

## Association for Information Systems AIS Electronic Library (AISeL)

---

PACIS 2017 Proceedings

Pacific Asia Conference on Information Systems  
(PACIS)

---

Summer 7-19-2017

# Exploring Software Developers' Experiences in Startups: The Philippine

Ma. Beth S. Concepcion

*De La Salle University Manila*, [ma\\_beth\\_concepcion@dlsu.edu.ph](mailto:ma_beth_concepcion@dlsu.edu.ph)

Raymund C. Sison

*De La Salle University Manila*, [raymund.sison@delasalle.edu.ph](mailto:raymund.sison@delasalle.edu.ph)

Follow this and additional works at: <http://aisel.aisnet.org/pacis2017>

---

### Recommended Citation

Concepcion, Ma. Beth S. and Sison, Raymund C., "Exploring Software Developers' Experiences in Startups: The Philippine" (2017). *PACIS 2017 Proceedings*. 117.

<http://aisel.aisnet.org/pacis2017/117>

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2017 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# **Exploring Software Developers' Experiences in Startups: The Philippine Case**

*Completed Research Paper*

**Ma. Beth S. Concepcion**

De La Salle University

Manila, Philippines

ma\_beth\_concepcion@dlsu.edu.ph

**Raymund C. Sison**

De La Salle University

Manila, Philippines

raymund.sison@delasalle.edu.ph

## **Abstract**

*Though there may be a proliferation of technology startups, it is a sad fact that most of them fail. Because startups depend on people, there is a need to study not only the factors that lead to the success or failure of these startups, but also the experiences of the people on which these startups depend. This study explores the experiences of software developers in technology startups in the Philippines, a developing country that has consistently ranked highly in the annual Tholons Top Outsourcing Destinations Ranking and the Kearney Global Services Location Index. Thematic analysis of interview data revealed 7 themes: Startups are characterized by (1) Rapid Search, which refers to the need to look for or develop something innovative and useful under time pressure. Rapid Search in turn requires a lot of (2) Feedback, highly flexible (3) Development Strategies, a high degree of (4) Collaboration, and a lot of (5) Learning. To cope well with all the uncertainties that startups must face, startup software developers' (6) Motivations are more intrinsic than extrinsic, and are derived from a strong sense of (7) Community, from all the Learning (theme 5) that the software developer makes because he or she must, and from the internal gratification of having found or developed something innovative and useful, i.e., Rapid Search, which is theme 1.*

**Keywords:** startup, software developer experience, qualitative study

## **Introduction**

Startups have been defined as temporary organizations used to search for repeatable and scalable business models (Blank & Dorf 2012), designed to deliver new products or services under conditions of extreme uncertainty (Ries 2010). The proliferation of technology startups is mainly attributed to the omnipresence of the Internet, cloud services, rapid changes in computing power, and cheap and ubiquitous blocks for developing digital products (The Economist 2014). Despite notable success stories, however, startups face significant risks and challenges. In fact, a Harvard Business School study revealed that 75% of all startups fail (Blank 2013).

Though not all technology startups involve the development of innovative software or hardware, many do, whether as the startup's main product or service, or as a tool to support the development or delivery of the startup's main product or service. Therefore, quite a few studies have been conducted examining software development in startups. Paternoster et al. (2014) have already conducted a systematic mapping of 43 primary studies (chosen from an initial set of 1057 papers) on the subject, so no further reviews will be done here. Instead we point out, at the risk of oversimplification, what we believe to be their main result: that software development and management practices in startups tend to be lightweight and flexible. This is not inconsistent with the core practices of the lean startup method, as conceptualized by Steve Blank (2013) and popularized by Ries (2012): sketches of hypotheses instead of a detailed business plan, a customer-driven iterative approach to searching for the right business model, and the use of the agile development method. These are also not inconsistent with the research findings of Coleman and O'Connor (2008) in identifying how software development processes are formed in startups as well as the study of Giardino et al (2016) in understanding how software development strategies are employed in startups. In particular, their studies postulate that Agile and Lean are the most dynamic development approaches in startups in order to quickly deliver product to the market. In addition, Sutton (2000) acknowledged that the typical approach to software process may be incompatible in startups because it exists with dynamic technologies and markets in which they are often get caught in technological change sweeping the IT industry.

Andrews et al. (2005) and other researchers in Barrett (2005) have analyzed, mainly through qualitative methods, the management and labor practices in software startups, and their results are also not inconsistent with those above. For example, Andrews et al. conclude that the characteristics of software development and the competitive environment of software startups "militate against manufacturing type controls of software developers" (p.64). In the book's conclusion, Barrett (2005) writes that "it is not a question of management using one strategy or another to control the software development labor process, it is more about how different strategies are used separately and simultaneously" (p.178).

While the results of the various studies above on software and management processes in startups are remarkably consistent and might therefore seem to be the final word, we believe that there still is a need to explore the experiences, to hear the voices, of software developers in technology startups in developing countries. Therefore, in this paper, we explore the experiences of software developers in technology startups in the Philippines, a developing country that has consistently ranked highly in the annual Tholons Top Outsourcing Destinations Ranking and the Kearney Global Services Location Index.

Despite its startup ecosystem being very young, it has had several technology startups that have exited, including Chikka, possibly the world's first commercially successful integration of Internet and mobile utilities, and Sulit, an online classified ads website that was eventually acquired by OLX. In 2015, the Department of Science and Technology launched a Philippine Roadmap for Digital Startups. In 2016, a bill called the Innovative Startup Act was filed at the Philippine Senate. Finally, early this year (2017), the Department of Trade and Industry launched its Startup Ecosystem Development Program.

In the rest of the paper, we first specify the research question and method, then present and discuss the results, and finally provide future directions.

## **Research Question**

In recent years, the Philippines has increased its focus on technological innovation and business ventures through the development of startups and the strengthening of the startup ecosystem. The increasing number of startups, co-working spaces, and incubation facilities in major and next wave cities in the country is a positive indicator of continuing interest in innovation and entrepreneurship.

Potential opportunities are clearly evident for technology startups to grow in the Philippines yet the lack of talent remains challenging (Hose 2013). Besides, the higher education institutions (HEIs) increasingly produce IT graduates every year but it is known that the demand for talented workforce overly exceeded the supply of qualified applicants. A job-skill mismatch still exists even though there is an existing development in improving the IT education in the country.

Furthermore, there is a vast area for software development sector in the Philippines waiting to be tapped. The Philippines is considered as a mature IT-BPO destination but the market share in the international market for software and IT outsourcing is small at 1 percent (del Prado 2015). Thus, the potential for software development sector to grow could also be explored in technology startups industries.

In order to realize these potential opportunities and address the impeding challenges, a deeper understanding of the startup phenomenon in the country is sought. Particularly, this study aims to explore the experiences of people working as software developers. It is important to understand software developers' work in the domain because startups possess a combination of unique characteristics that differ from the established companies and organizations (Unterkalmsteiner et al. 2016; Coleman & O'Connor 2008). The emergence of startups is argued to introduce a new kind of workplace, which faces a lot of demands and requires different kinds of mindset, skills and competencies apart from the traditional notion of a company.

This study is guided by the research question, "What are the experiences and concerns of software developers in Philippine startups?" This research question revolves around an understanding of the nature of their jobs, and the technological, organizational, and other factors that affect their ways of working.

By exploring software developers' experiences in startups, this study will provide insights on what necessary measures to be undertaken in order to nurture their total wellbeing and to keep these talented people from leaving. In addition, the awareness of their practices, culture, and social environment can provide guidance to those who wish to work in startups. This also aims to highlight the needed skills and attitudes software developers must acquire in order to be adaptable to startup environment knowing that a huge gap still exists between what has been offered by the academe and what has been practiced in this industry. Lastly, this study aims to understand the current state of startups in the country. This may serve as input in the development of governance policies and structures and in advancing support needed by industry to become sustainable.

## **Methodology**

The study used a qualitative, inductive approach to understand and describe detailed accounts of software developers' experiences in startups as their workplace setting. Sampling was purposive. The respondents must be working as software developers in a startup and must have had at least one year of work experience. Participants came from various startup companies located in two key cities: Metro Manila and Iloilo City. The choice of the two areas was based on the attempt to gather various perspectives from startups from two very different locations: Metro Manila, the capital and center of development and innovation; and Iloilo, dubbed as the "next wave city" in the Philippines. A total of 8 participants from 4 startup companies participated in the study. These companies focused on developing various software products (web, mobile apps) which fall under the following segments: health, legal, travel, education, games, finance, and lifestyle.

The primary mode of data collection was through in-depth interviews. Each interview lasted from 45 minutes to 1 hour. Interviews were semi-structured and participants were asked a series of open-ended questions. Before the actual conduct of the study, permission from startup companies was taken, and written or verbal informed consent from individual participants was obtained. The interviews were audio recorded while the researchers were taking down notes. In addition, the secondary mode of data collection was observation, where researchers examined the practices of software developers and the actual environment they worked working in. The audio recordings were then transcribed and the notes from the observation data were prepared for data analysis.

Analysis of data was thematic based on (Braun & Clarke 2006). Transcripts and field notes were examined line by line and assign a label or code that best described the incident (see Table 1). Codes were then organized based on patterns and their commonalities to form categories. Third, categories were abstracted to form themes (higher-level categories). After identifying major themes and

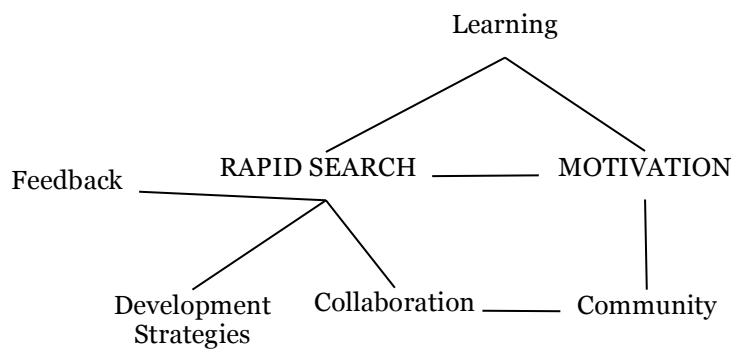
categories across data sets, corresponding transcripts and individual incidents were reviewed and extracted to support the detailed discussion on respondents' experiences.

Transcript	Codes
<p>Q: Tell me your work experiences in startup?</p> <p>A: In start up we <i>rely on technologies</i>. The <i>technologies are always changing</i>. So every time there are new technology updates and new programming tools, we have to <i>learn</i>. We need to <i>adapt and integrate updates in our products</i>. The problem usually happens when other <i>technologies are not compatible</i> with the recent version of OS.</p>	<ul style="list-style-type: none"> <li>• Technology-dependency</li> <li>• Constant Technology updates</li> <li>• Learning</li> <li>• Adapting</li> <li>• Compatibility problems</li> </ul>
<p>In solving problems usually I <i>learn it on my own way</i>. But, there are times that I don't know how to solve it, <i>I asked Sir John for help</i>. He made his initial version and showed me how he did it. My other <i>team member</i> also shared her ideas especially on the things that are new to me.</p>	<ul style="list-style-type: none"> <li>• Self-learning</li> <li>• Team work</li> <li>• Knowledge Sharing</li> </ul>

**Table 1. Example of Initial Coding of Data**

## Results and Discussion

Five major themes emerged, describing the experiences of software developers working in startups: Feedback, Development Strategies, Collaboration, Learning, and Community. As shown in Figure 1 we can connect and organize these five themes using two higher-level themes, which we shall call “theoretical themes” (from “theoretical codes” in grounded theory): Rapid Search and Motivation. We will explain the interconnections briefly now, and provide more detailed descriptions and sample instances after the next paragraph.



**Figure 1. Themes in startup software developers' experiences**

The first theoretical theme, Rapid Search, refers to the need to look for or develop something innovative and useful – whether a product, product feature, a marketing strategy, a technical solution, a venture capitalist – under time pressure. Rapid Search involves a lot of Feedback (from potential users and markets), highly flexible Development Strategies, a high degree of Collaboration, and a lot of Learning. These four themes pertain to the startup software developer’s work processes. The second theoretical theme, Motivation, refers to what keeps the startup software developers going and what

keeps him or her from leaving the startup in the face of all that Rapid Search entails. Our research shows that a strong sense of Community, all the Learning that the software developer makes because he or she must, and the internal gratification of having found or developed something innovative and useful (i.e., Rapid Search) all combine to powerfully motivate the software developer amid all the uncertainties associated with a technology startup.

We now proceed to detail each of the five major themes. Table 2 lists them together with their associated categories.

Themes	Categories indicating respondents' experience
Feedback	User metrics User feedbacks User retention Market
Development Strategies	Exploring Adapting and Tweaking Wait, Push or Quit
Collaboration	Knowledge sharing Communication Task assignment Specialization Swapping Tagging Task optimization
Learning	Self-learning Continuous Learning Skills Enrichment Internal Gratification
Community	Shared relationships Freedom Flexibility Discipline Trust Respect

**Table 2. Major themes and categories of software developers' experience**

### **Feedback**

To address Rapid Search, software developers engage with frequent feedbacks from users and markets. In particular, monitoring user metrics (i.e. user downloads, website traffic), listening to user feedbacks, and observing the markets seemed critical to the development of startup products, validate product functionalities and improve user experience.

Developers always pay attention to user response especially during the initial release of the product. In mobile applications in particular, the number of user downloads indicates the user acceptance and topping it in the online markets is an achievement for developers. A respondent shared, *"We were overwhelmed during the initial release of our mobile game because it became number 1 for a week based on the number of downloads in the online market."* This overwhelming experience may not be the same for some startups because the reality is that even when developers see their products as valuable and innovative, still user acceptance is not evident. One of our respondents shared that they were supposed to upgrade their existing travel site to a mobile app but they found out that there is no user traffic. Thus, the adverse reactions from users are equally valuable for developers in finding other mechanisms in order obtain wide user base.

In addition, developers always figure out what needs to be improved in their products based on user comments. Typically, user feedback provides insight on what features need to be added, removed, or improved in the next iteration of product development. One respondent who did the mobile application for child development shared that when they deployed the first version of their product someone made a comment that there is something lacking in the app. He reiterated the feedbacks as, *"We saw your app and it is good but how can I add additional frog because I have three kids."* With that, their team decided to integrate an option for users to add a frog as child's pet based on user suggestions.

Direct feedback from clients is as important as feedback generated online. As software developers, they throw a lot of features, get users to test it, and identify what is best for the customers and for the business. For applications that involve clients (e.g. teachers, restaurants owners, physicians, lawyers), developers usually engage with them and inquire for additional features since they are very familiar of the processes and needs in their fields. Some specified, *"We consulted teachers and they suggested to add assignment and parent notification. So, we thought of integrating those suggested features in our next version"*. As one of the strategies, startups also partnered with other entities in the development of their products. In gathering what clients and partners want, they usually convene either face to face or via teleconference to talk about the intended product features, do user testing, and make changes as needed. One described, *"We partnered with a fitness gym. They lay down the features and we confirm it if it is feasible."*

Acquiring users is not a guarantee for market-hit; users need to be retained and kept active. Software developers underscored the importance of user metrics because it enabled them to observe user growth, analyze user activities, and determine if there is user inactivity or discontinuance. One respondent narrated, *"We are happy that there were many user downloads during the initial release of our app. However, as the time went by, we noticed in the metrics that users were active only just for two weeks or one month. We recognized that our app had transformed children's behavior that is why people stopped using it. However, we realized we were the ones who killed our own product. This triggered us to strategize ways in order to keep users active by introducing additional features. So, we decided to add a reward system feature so that the app could be used continuously."* Thus, this instance shows that the concern of software developers is not only on instantly hitting the market and acquiring new customers but also on the long-term survival by keeping and retaining their current customers.

In addition, markets greatly influence the Rapid Search of software developers. They regularly explore different market segments to determine the trend and think of products that are unique and valuable. They have to move first considering a very competitive ecosystem where other startups or established companies plan to do similar software products. As an example, one startup company we studied originally focused on developing mobile applications and games, but when there was a tremendous buzz that smart watch will hit the market they instantly delved into developing products that can be deployed in that platform. Accordingly, this strategy enabled them to acquire first set of customers before other entrants do and to obtain acknowledgement from tech enthusiasts.

On the other hand, being the first mover is not always an assurance that the product will gain competitive advantage. One breaking change happens when big players introduced product features similar to what startups originally offer. In some cases, startup companies could not scale their products because of Facebook, their major competitor. Facebook is not only limited to a social media platform but also extends its services to restaurant reservation, food delivery, and ticket booking. Thus, these new features triggered the startup companies we interviewed to change its directions and strategies. One company shared that they did not pursue the launching of their restaurant application because of such threat. In addition, it is very hard to compete with Facebook in terms of user experience. This holds true for some startups we studied when they developed a travel site. They found out that there was no traffic because they realized that it is difficult to break the habit of people who are used to do similar things in Facebook. One developer described, *"People are now using Facebook to join trips, organized events, and even made payments. We realized that our app adds another steps but has the same goal that is already been doing by users in Facebook."*

Startup companies also need a positive environment where it fosters them to develop, scale, and grow maturely. The presence of government, accelerators, investors, venture capitalist, and receptive market is crucial to their development. As one of their strategies, some startup companies started to establish linkages with local chambers and government agencies for endorsement and support. However, some experienced difficulties in engaging with public institutions because of bureaucracy. As one respondent shared, *"We have a private school partner, their CEO already approved our proposal. It is very easy to penetrate private schools than public institutions because it is less bureaucratic."* Others had problems in looking for backup and funding because of the conservative environment and the close mindset of people. This is particularly true for companies located in an emerging city in the Philippines. One shared, *"Even if our product has a unique value, the challenge is that investors and people here in Iloilo are very conservative."*

## **Development Strategies**

There are three mechanisms that being practice by software developers in product development considering the breaking changes brought by rapid changes in technologies and demands from users and markets: (1) Exploring, (2) Adapting and Tweaking, and (3) Wait, Push or Quit.

Since startups are in a search for a scalable business model, software developers explore different techniques in developing their products. Developers always thrive on exploring what product to develop that is different from what has been done before. In creating unique products, most respondents think of an existing social problem that needs technological solutions. One shared his experience in developing an application, which is rooted in his observations on children's behavior. A respondent narrated, *"Our vision is to change the next generation because we have seen the problem in today's behavior of children. We all know that child development always starts at home. So we built an application that may guide parents in disciplining their children."* In addition, startup companies develop products in order to innovate existing processes or even disrupt other industries. One shared that they are developing a legal application because they observed that there is a surplus of lawyers every year yet there are also lots of legal problems in the shelf that left unanswered. Thus, this huge disconnect enables them to explore and develop an online market place for clients who look for lawyers and for lawyers to find prospect clients.

The recent buzz in technologies in the market for example is a potential opportunity to build applications. One startup company we interviewed tried exploring and building different applications for smart watches. They were successful in their utility applications (e.g. tip calculator, fitness app). However, they failed when they tried developing games in the same device. One respondent recalled, *"We developed games for smart watches but then we realized its impracticality because smart watch is not a gaming platform."*

Software developers are exposed to frequent iterations in product development. Specific approaches used are adapting and tweaking. In their perspective, rapidly adapting to changes in technologies, market and user preferences is challenging. Hence, they need to find innovative features to suit the demands. One shared his experience, *"In startups, we find what is best. By finding what's best we need to adjust product features. Sometimes we have to tweak it very drastic so that we can find the product fit."* In addition, technological dependency caught practitioners' attention specifically in addressing compatibility problems. According to them, the problems usually happen when their products are not compatible with the recent versions of platforms as well as the programming languages used. One respondent described, *"I started using version 19 of a particular language and now it is in version 38. The package I used in the previous version is not compatible with the current one."* These scenarios triggered them to move fast and adjust for compatibility.

In software development, some developers ascribed to Agile method because it makes the development process more flexible to accommodate rapid changes brought by customers and markets. Others may find it very interesting to explore different paradigms to address the demands from the customers as well to suit the needs of the team. Some specified, *"Our practice before was Agile development using Kanban. Now, we used the SCRUM. As a SCRUM master, I let my team members decide what specific task they would like to work."* This strategy is believed to facilitate creativity, empowerment, and decision-making capabilities of team members. On the other hand, others set no particular method in software development as long they quickly develop the product, ensure the software passes several tests, and launch the product to the market.

Wait, Push or Quit is another strategy used by our respondents in developing their products. To wait is to pause the development activity until product dependencies have been normalized. One respondent shared an instance, *"If the new version of a platform is not yet stable, we wait until the final version is out"*. While this happens, others find ways to do other task or projects just to optimize time. He said, *"We find other work-around task and wait for the platform to fix its bugs."* Push or quit is another mechanism for software developers to deal with the project opportunities and breaking changes. One startup company decided to push the development when they found the opportunity for their application to progress. They shared, *"We are developing a scholarship crowd funding platform to support scholars. We will continue the development because we received support from the local chambers and networked schools."* One startup company shared a different experience when they were developing restaurant application wherein customers can reserve seats and place orders in advance. They already received positive feedbacks from partner restaurants. But their momentum was impeded when one of them read an article that Facebook will release the same feature in their page. As a developer, she was hesitant to continue the development. She explained, *"We think our products will*



*not work because we are fighting a losing battle if ever. The food applications in the US are already affected and we anticipate that too will happen in the Philippines soon."*

## **Collaboration**

Collaboration shows how work is being facilitated within individuals and teams in startups. Sub-concepts were abstracted from the respondents' interviews, thus include knowledge sharing, communication, task assignment, specialization, swapping, task overlapping, and tagging.

The core to collaboration is knowledge sharing wherein individuals in a startup team interact with each other to exchange ideas. The spontaneous flow of tacit knowledge and direct physical interactions are very apparent because of the informal structure of the organization. The open communication lines between individuals in a team thus underpin positive outcomes specifically in the resolution of problems. One respondent attested, *"I don't know about the design since I am a developer, so I approach people like Ben, John and Eric. They are willing to share their experience and very open to share solutions."* Communication is also fluid because it comprised of smaller teams with narrow hierarchical boundaries. One respondent who is a developer and co-founder at the same time shared, *"We can set a meeting anytime even just the two of us. We can discuss in the pantry or in their workstations"*. One respondent from other startup company also supports this practice, *"It is better to talk with my peers face to face that talk via chat. I can ask help and questions anytime."*

Moreover, software developers use online collaboration and source code management tools to facilitate coding, task management and communication needs. One respondent shared, *"Using Slack, I can create channel where I can communicate with my-co developers and other members in the design team."* In addition, they also used tools to create repositories and to manage their codes, *"My usual practice in managing my task is to pull repository every first hour in the morning and push changes at later part of the day."* Others specified, *"I used GitHub repository. When I want to test new feature, I create a branch for us to test."*

In dealing with task, teams used approaches such as assigning, swapping, tagging and optimizing. Typically, task assignment is normally done based on specialization. One respondent explained, *"In our case, we have our own specialization or forte. For example, John specializes in web development but he does not have a sufficient experience in native app and game development. If there is an mobile app and game dev project, I am the one who does it."* Some also tried the task swapping strategy. This approach is a way for them to learn skills that have already been doing by their colleagues. One respondent shared an instance of this experience; *"There are times that we worked for a certain projects, utility apps both for Android and iOS. Christy worked with one app developed using iOS. I developed the second app developed for Android. After we were done, we swap. I developed her project using Android and she developed mine using iOS."*

Startup companies used another strategy in assigning task when task overlapping happen. One of our respondents shared that they tried optimizing task by empowering other members to choose what task they want to own and what they want to explore. One developer narrated, *"Before, I assigned task based on their specialization. However, if their specialization overlaps, I tried to optimized task. If there are task that are not in their specialized fields, I need to assign to them so that they will learn. Now our practice is different. I put the tasks and let them decide what specific task they would like to work."* In most cases, developers in startups acquire variety of skills as they move from one task to another. Hence, this mechanism enabled them to become generalist instead of being specialist.

Startups usually comprised of young developers yet it has an environment where everyone is encouraged to ask for help. One strategy used is tagging experienced developers for the resolution of difficult tasks when self-learning is not sufficient. One shared, *"Normally for certain problems that I don't know how to solve on my own, I ask another person who encountered the same problem. In case the team can't solve it, we ask the boss. Once we tagged Ian and Mark, it is really a serious problem."* Thus, all these practices show synergy and symbiotic relationships among team members in achieving a common purpose - to create a valuable and innovative product.

## **Learning**

Learning relates to the intellectual undertakings of software developers as they deal with the Rapid Search. Working in startup provides a best learning experience for software developers. As observed, common to them started as neophyte but being inexperienced was compensated by their enthusiasm and eagerness to learn. One respondent specified, *"Almost 75 % of the stuff I am doing is self-learn because the things I do were not taught in school."* Others said, *"The programming languages I used*

*were introduced 2 or 3 years ago.*" This situation acknowledges that self-learning is a way for them to address huge skills gap between what had been taught in school and what have been doing in startups.

For some, they are motivated to learn on their own because no senior or experienced developers who can train them in the process. While certifications and trainings are significant for IT professionals working in established companies and organizations, people in startups see it as less important. One developer specified, *"I don't see any point of having certifications. Trainings can be learned on our own. Probably, what I only need someone who can share and explain how to do it."*

Accordingly, in order to adapt to the technological change and skills demand, software developers need to expand their abilities through continuous learning. They developed learning goals by applying lessons they have learned in previous experiences and continuous discovery of things through research. One shared his insights, *"The learning process here is continuous. That is why, I do not stop researching."* Furthermore, they regularly update their knowledge by exposing themselves to opportunities such as developers' conference. One respondent also shared, *"We grab the opportunity to attend the developers' convention. I kept updated with the recent development in mobile, VR (virtual reality), UX, Internet of things, among others."* Various startup communities also conduct startup meetups and events to gather founders, developers and creatives. This is an avenue for members of the startup ecosystem and interested parties to connect, gain support, share insights, and learn from each other's experiences that will help them enliven their technopreneurial spirits.

In addition, learning is also one motivating factor that keeps software developers from leaving their work despite of the challenges brought by Rapid Search. For them, working in startups is way of improving their technical abilities and expanding their skills sets. When they asked what keep them working, responses received were, *"The main contribution of startup is to enhance my skills", "For the short term plan, I guess is improving my skills sets", and "I am here to find my niche."* They also emphasized that the skills they gained is an important element to be added in their portfolio. For some, this is beyond improving technical and problem-solving skills but more on uncovering their potentials in business, management, communication and interpersonal relations. The reason behind this is they played multiple roles at the early stage of startup since they were constrained by human and financial resources. One detailed how his work in startup contributes to his personal and professional growth, *"I have a good problem-solving skills but I didn't have an experience in business. So working here forced me to learn skills in connection with business like processing business permits, meeting and connecting with people, and leading the team."* By learning other things, some argued that their overall experience improved their personality and boosted their confidence. One narrated, *"I am very introvert person. I am a techie, which is innate in our fields. But there is a shift in my attitude when I am in startups because it is a need. Nobody do these business related things except me."*

Another motivating factor that keeps the enthusiasm of software developers in their work is the feeling of gratification and achievement. For them the work in startups can be draining but it can be rewarding if they done something challenging and worthwhile. As recalled by one participant, *"We are exposed to constantly moving targets, which is very stressful. Surprisingly, this is what excites me."* In addition, the element of gratification bolstered them to continue what they are doing. They usually feel anxious for the problems they cannot solve. However, they always take pride and esteemed if their solutions worked. One detailed his experience, *"Problem solving is really addicting. It is a feeling that even in my sleeping time I am still thinking of the codes and sometimes get stuck of the problem. I didn't stop thinking until I was able to solve it. But the feeling is different when I was the one who provide solutions for hard problems."*

## **Community**

A sense of community is an interesting culture among startups that motivates people to enjoy their work. In startups, the lines between work, personal and social life are becoming blurred. The office becomes their place of work, their home, their playground, and a place where they can freely express themselves.

We found out that people create interpersonal connections and social bonds apart from being colleagues. They also feel that they are secured, empowered and accepted despite of differences because of the mutual trust and respect. As described by our participants, startup is a community of people with shared relationship, passion and spirit. They know how to relate and how to work with one another because they treat peers as friends and even families. These observations are prevalent among our respondents when they replied, *"I see them more as friends rather than colleague", "I am at home here", and "In our office, we don't compete with other people because they treat you as a friend."*

There are lots of attractive work opportunities outside that are waiting for these skilled developers however they intend not to leave because of the startup culture. One co-founder shared his observations that people stay in his company not because of salaries but because their friends are there. Also, respondents appreciate the mentorship way of leadership, which is contrary to the command-and-control that is being practiced in large companies. When asked why he favored working in startups, our respondent narrated, *"I really don't like the culture in big companies. I don't like it the way they were too controlling. I guess I have more freedom here."* In startups, the participants even feel that their needs and opinions are being acknowledged. One specified, *"If you did something wrong, the boss will talk to you as a friend not as a boss."* Others responded, *"They never get upset if you think you can't solve but they are just going to measure you."*

Freedom and flexibility are startup characteristics that fit the personality of participants. Some startup companies offer an environment with no rigid protocols and people can choose their working preferences. It is not uncommon to see people working in their t-shirts, shorts and flip-flops contrary to the casual or corporate attire required for employees in most established companies and organizations. On an interview, one specified, *"The major difference is culture. I can't do the office wearing shorts without being reprimanded."*

In addition, startups provide an environment where workers are allowed to play games and do anything just to lessen their anxiety. Accordingly, this is a way for them to have a preventive maintenance before they totally get stressed and becomes unproductive. One co-founder said, *"I let them do anything they want. It is their way to recharge."*

Some startups companies provide flexible work options where employees can work at their own time and their own pace as long as they accomplish the required task and comply the specified number of hours. One developer said that he usually arrived in the office at 3 in the afternoon and went home at 2 or 3 in the morning. When asked why he preferred to work at that time, he responded, *"Because it fits my sleep cycle. They are not strict as long as I deliver what is required. Other share similar experience, "Working time is varied but at the end of the day, we have to deliver the product update."*

In case target demands their time, they are extending their work time in the office. Others also brought their work at home when need arises, *"In some cases during deployment when the Internet connection in the office is slow, sometimes I bring my work at home when there is faster Internet speed."* This observation is true to locations, which is challenge by poor Internet infrastructure.

Amidst freedom and flexibility, some prefer to work on a regular working hours, which usually starts at 9 o'clock in the morning and ends at 6 in the afternoon. One respondent justified, *"because I like talking and working with my co developers. I can ask help and questions anytime."* Other participant emphasized that he wanted to work at regular hours where everybody is present. That lax of schedule according to him causes problems in development, *"Because of the flexible time, when time comes you need to develop solutions you will wait for them to arrive. It is hassle sometimes because we wait for them just to finish our work and before we go home."*

Promotion is a problem in startups because of its flat structure yet founders ensure that the salary of their developers is reasonable. Some even raised the concern of having the medical benefits similar to what BPO companies provided to their employees. Notwithstanding, they understand that the company cannot provide them their request because it is not yet sustainable. To compensate this, some even allowed their employees to work on their freelancing jobs especially if the company is not yet earning money from their startup projects. This is a way for their employees to find additional sources of income. One respondent cited an instance, *"When we didn't receive any money from the Alpha project, our founder Fred even gave us freelance works."*

It is apparent that startups are inherently chaotic because they often face a lot of uncertainties. However, chaos in the workplace actually fosters creativity and sparks productivity when it is properly managed. To establish order amidst chaos, individuals must value self-discipline, and establish coordination, trust and respect with peers. As emphasized by one respondent, *"Having chaos actually works for us but we have to finish the work because the project is also dependent on other people. It is more like developing own self-discipline rather than discipline being imposed to you."*

## Conclusion and Recommendations

This study conceptualized from interview data seven themes that describe the experiences of software developers working in startups. Startups are characterized by Rapid Search, which refers to the need to look for or develop something innovative and useful –a product, product feature, a marketing strategy,

a technical solution, a venture capitalist – under time pressure. Rapid Search requires a lot of Feedback (from potential users and markets), highly flexible Development Strategies, a high degree of Collaboration, and a lot of Learning. To cope well with all the uncertainties that startups must face (which Rapid Search addresses), startup software developers' Motivations are more intrinsic than extrinsic, and are derived from a strong sense of Community, all the Learning that the software developer makes because he or she must, and the internal gratification of having found or developed something innovative and useful (i.e., Rapid Search).

The above results, particularly the use of highly flexible Development Strategies, confirm those of Paternoster et al. (2014) and Giardino et al. (2016) that the software development and management practices in startups tend to be agile. The Rapid Search theoretical theme also shares several features with the lean startup method of Blank (2013) – both involve a high degree of Collaboration with, and Feedback from, the customer, and both involve the use of agile Development Strategies. However, Rapid Search is also tightly coupled with another theoretical theme, Motivation, which in turn involves Community and Learning, themes which are not present, at least not explicitly, in the lean startup method.

Organizations in charge of educational programs and specific courses (including, e.g., the entrepreneurship course that the Commission on Higher Education has recently mandated for all engineering degree programs, and the Department of Education, for Grade 12) might consider the seven-theme conceptualization and other findings described above in curriculum and instructional (re)design. Similarly, government agencies, firms, and other groups that provide startup training and support might also consider the above findings in the (re)design of their programs.

Though the above conceptualization may already be used as basis for recommending the adoption by startups of certain specific practices, or even the design of new practices, we believe that the story could be detailed further. We have therefore begun to plan a grounded theory study of software developers in technology startups. The classic grounded theory method (GTM) of Glaser and Strauss (1967), as well as the other GTMs that grew out of it, differ from the thematic analysis approach we used here in two major respects. First, GTM's sampling (called theoretical sampling) is more careful, with a clearer termination condition. Second, GTM's analysis (using constant comparison throughout open, selective, and theoretical coding) is more rigorous. The greater rigor entailed by theoretical sampling and constant comparison should enable the discovery of a fuller and denser theory of the substantive area.

## Acknowledgements

We would like to extend our gratitude to the respondents of the study for willingly sharing their experiences and for the startup companies for their approval.

## References

- Andrews, C. et al. 2005. "The Labor Process in Software Startups: Production on a Virtual Assembly Line?" in *Management, Labour Process and Software Development: Reality Bites*, R. Barret (ed.) Routledge, pp. 40-67
- Barret, R. 2005. *Management, Labour Process and Software Development: Reality Bites*. Routledge.
- Blank, S. 2013. "Why the lean start-up changes everything," *Harvard Business Review* (91:5), pp. 63-72
- Blank, S. G., & Dorf, B. 2012. *The Start-up Owner's Manual: The Step-by-step Guide for Building a Great Company*. Pescadero, Calif.: K&S Ranch.
- Braun, V., & Clarke, V. 2006. "Using Thematic Analysis in Psychology," *Qualitative Research in Psychology*, (3:2), pp. 77-101.
- Coleman, G., & O'Connor, R. V. 2008. "An Investigation into Software Development Process Formation in Software Start-ups," *Journal of Enterprise Information Management*, (21:6), pp.633-648.
- Del Prado, F. 2015. "The BPO Challenge: Leveraging Capabilities, Creating Opportunities," PIDS Discussion Paper Series, (No. 2015-36)
- Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., & Abrahamsson, P. 2016. "Software Development in Startup Companies: The Greenfield Startup Model," *IEEE Transactions on Software Engineering*, (42:6), pp. 585-604.
- Glaser, B. and Strauss, A. 1967. *The Discovery of Grounded Theory Strategies for Qualitative Research*.
- Hose, N. 2013. *Philippine Startup Report*.

- ICTO. 2015. The Philippine Roadmap for Digital Startups: 2015 and Beyond.
- Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., & Abrahamsson, P. 2014. "Software Development in Startup Companies: A systematic Mapping Study," *Information and Software Technology*, (56:10), pp. 1200-1218.
- Ries, E. 2010. *Startup Lessons Learned*. Retrieved January 12, 2017, from <http://www.startuplessonslearned.com/2010/06/what-is-startup.html>
- Ries, E. 2012. *The Lean Startup Methodology*. Retrieved March 4, 2017, from <http://theleanstartup.com/principles>
- Sutton, S. M. 2000. "The role of process in software start-up," *IEEE Software*, (17: 4), pp. 33-39.
- The Economist. 2014. A Cambrian Moment: Special Report Tech Startups. Retrieved January 12, 2017 from <http://www.economist.com/>
- Unterkalmsteiner, M., Abrahamsson, P., Wang, X., Nguyen-Duc, A., Shah, S., Bajwa, S. S., ... & Edison, H. 2016. "Software Startups—A Research Agenda," *e-Informatica Software Engineering Journal*, (10:1), pp 89-123.