Evolution of Software Product Development in Startup Companies

Mercy Njima
Department of Mathematics and Computer Science
University of Antwerp
Antwerp, Belgium
Email: mercy.njima@gmail.com

Serge Demeyer
Department of Mathematics and Computer Science
University of Antwerp
Antwerp, Belgium
Email: serge.demeyer@uantwerpen.be

***Abstract*—This position paper addresses the software engineering practices in startups. The focus of most software engineering research has been on established companies. However, startup technology companies have become important producers of innovative and software intensive products despite the fact that they are under severe time-to-market pressure. Given that software engineering is the core activity in said startups, inadequacies in such practices might be a substantial contributing factor to this pressure to keep up with the software industry competitive needs. Startups build non-traditional business architectures by taking the easy path to find a product-market fit and thus, accumulate large amounts of technical debt. We shed light on the major efforts in the domain and indicate the research directions we plan to explore further.**

I. Introduction

A. Software Engineering in Startups

The vast majority of software development projects today are small, with few people and a very short development cycle. These are often developed by young companies with no organizational history, no legacy code, and severely limited funding. These startup companies may still be in search of a business model and as a result, many applications are initially released in a matter of weeks or months (minimum viable product), rather than being developed and tested over years. The associated philosophy of "release early, release often" means that subsequent releases of those applications are sometimes made frequently [1], [2]. As a result of these constraints, startups tend to be very informal with their software engineering processes, focusing on key areas such as the user experience, product feature requirements, and the ability to use existing frameworks and libraries. These aspects are central to lean and agile development methods, and are not included among the key process areas of the traditional software engineering capability models.

Agile software development has been considered the most viable process for software startups as they allow teams to respond to the unpredictability of building software through incremental, iterative sprints [3], [4]. In this context, fast releases with an iterative and incremental approach shorten the lead time from idea conception to production with fast deployment. XP is the most used development process across startup companies, due to its reduced process costs and low documentation requirements [5]. The Lean Startup method [6], has been a common variant of Agile which advocates for the creation of value to customers and elimination of waste during the development phase. These methods are preferred as they enable a faster learning process especially since there's generally a lack of written architecture and design specifications. However, the absence of structure might hinder important activities, such as sharing knowledge, team coordination or implementing new features. Therefore, in preparation for growth, startups must plan for scalable processes that involve formulating initial architectural strategies from which they can benefit from reuse of components and shared architectures across projects to ensure long term evolution without significant schedule overhead.

This research process aims to answer the question "How can a startup continuously improve it's products and still meet customer requirements?" We hypothesize that the effective management of technical debt and the application of software product line engineering best practices would allow a startup to respond to the evolving needs of the market.

II. Related Work

Over the past couple of years, the software engineering community has shown interest in the software development practices of software startups. Unterkalmsteiner et al. provide a research agenda that focuses on software engineering in startups identifying, 70+ research questions in the areas of supporting startup engineering activities, startup evolution models and patterns, ecosystems and innovation hubs, human aspects in software startups, applying startup concepts in non-startup environments, and methods and theories for startup research [7]. Pantiuchina et al. [8], conducted a large survey of 1,526 software startups and examined the use of five agile practices, including quality related (regular refactoring and test first) and speed related (frequent release and agile planning) in an effort to understand how software startups can better use agile practices and eventually benefit from them. Klotins, applied a hybrid research method to design and launch a large scale study into software engineering aspects of startups [9]. Giardino et al. identified common software development startup practices in [4] while Paternoster et al. presented a detailed investigation and collection of all known empirical software engineering sources related to startups and their en-

gineering practices [3]. The following papers provide a general overview of the research done to investigate development practices in software startups [10], [11], [12], [13], [14], [15], [16].

The Lean startup method has also been applied in large software companies through experimentation in internal startups as they look for new ways to innovate. Edison et al. conducted a multiple case study at large software companies to shed light on this issue and examine how a new product was developed through the internal startup effort [17], [18].

We hope to build on the work done by Chicote and Yli-Huumo et al. in with regards to technical debt in software startups [19], [20]. Chicote introduces a technique for managing technical debt based on Visual Thinking. The technique addresses the problem of knowing how much debt is in place and how it is affecting the development cycle. On the other hand, Yli-Hummo et al. investigated one middle-size Finnish software company with two independent product lines to understand the causes and effects of technical debt.

III. Research Directions

A. Technical Debt in Startups

Startups search for product-market fit by building fast, testing and iterating until they find a solution. In the process of releasing a product, the development team knows that it has released software with flaws that must be fixed [21]. Balancing the choice of releasing poor-quality software early or high-quality software late is challenging. This leads startups to an awkward situation where they have to decide what quality is acceptable and what compromises in the development process they have to take [20], [22]. This is the basis of how startups accumulate technical debt. Technical debt is a metaphor reflecting technical compromises that can yield short-term benefit but may hurt the long-term maintainability and evolvability of a software system [23]. Accruing large amounts of technical debt may make new changes harder to implement especially when a pivot is triggered [24]. A pivot is a "structured course correction designed to test a new fundamental hypothesis about the product, strategy, and engine of growth" [6]. A startup that cannot implement new features quickly or recover swiftly from a failed project because their code base has technical debt is not likely to succeed [19]. In our opinion, managing technical debt for software startups deserves research attention. Meir Lehmann's second law of software evolution, *Increasing Complexity*, states that "As an evolving program is continually changed, its complexity, reflecting deteriorating structure, increases unless work is done to maintain or reduce it "[25]. This applies to the startup context as well.

B. Software Product Line Engineering for Startups

A software product line is a family of systems that share a common set of core technical assets, with preplanned extensions and variations to address the needs of specific customers or market segments [26]. Software product lines have achieved substantial adoption by the software industry. Consequently, a wide variety of companies have substantially decreased the cost of software development and maintenance and time to market and increased the quality of their software products [27].

Startup software companies usually start with a single idea and thus with a single product. Over time, if the company is successful, the product matures and the management sees that it can use the same idea (or slight variations of it) to develop a new set of products [28]. However, during this process most startups still have scarce financial resources, so they cannot invest heavily in development. Thus, they must make reuse a reality.

A few stages of startup development can be identified as startup phase, stabilization phase and growth phase [29]. In the startup phase the startup is just finding out ways to make business. In the stabilization phase, the organization starts gradually to find its way of working and the amount of uncertainty decreases. In the growth phase, the startup goes through a series of changes and the organization needs to restructure itself to support growth in all areas: business, personnel, the scope of the product, etc.

At this stage the company needs to work on how the systems can evolve later. When the products evolve, features need to be added, removed, turned into modules or separate products [30]. This requires attention to the architecture and implementation. Software product lines encourage an organization to reuse existing assets and capabilities rather than repeatedly developing them for new systems. Therefore, they use architectures, designs and implementation techniques that make features modular and independent so that they are easy to reuse, remove, turn into modules or separate products.

IV. Conclusion

Software engineering practice for startups is a rapidly evolving area of research. A mapping study is within our short term goals by which we aim to obtain an overview of the body of knowledge and identify inadequacies in used practices and proposed remedies.

References

[1] H. Erdogmus, A. Sillitti, and T. Wasserman, "Softstart 2017 workshop summary," in *2017 IEEE/ACM 1st International Workshop on Software Engineeting for Startups (SoftStart)*, 2017, pp. 1–1.

[2] E. Klotins, M. Unterkalmsteiner, and T. Gorschek, "Software engineering knowledge areas in startup companies: A mapping study," in *Software Business*, ser. Lecture Notes in Business Information Processing, J. M. Fernandes, R. J. Machado, and K. Wnuk, Eds. Springer International Publishing, 2015, vol. 210, pp. 245–257.

[3] N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson, "Software development in startup companies: A systematic mapping study," vol. 56, no. 10, pp. 1200–1218, 2014.

[4] C. Giardino, M. Unterkalmsteiner, N. Paternoster, T. Gorschek, and P. Abrahamsson, "What do we know about software development in startups?" *IEEE Software*, vol. 31, no. 5, pp. 28–32, Sept 2014.

[5] G. Coleman and R. O'Connor, "Using grounded theory to understand software process improvement: A study of irish software product companies," *Inf. Softw. Technol.*, vol. 49, no. 6, pp. 654–667, Jun. 2007.

[6] E. Ries, *The lean startup : how today's entrepreneurs use continuous innovation to create radically successful businesses*. New York: Crown Business, 2011.

[7] M. Unterkalmsteiner, P. Abrahamsson, X. Wang, A. Nguyen-Duc, S. M. A. Shah, S. S. Bajwa, G. H. Baltes, K. Conboy, E. Cullina, D. Dennehy, H. Edison, C. Fernández-Sánchez, J. Garbajosa, T. Gorschek, E. Klotins, L. Hokkanen, F. Kon, I. Lunesu, M. Marchesi, L. Morgan, M. Oivo, C. Selig, P. Seppänen, R. Sweetman, P. Tyrväinen, C. Ungerer, and A. Yagüe, "Software startups - A research agenda," *e-Informatica*, vol. 10, no. 1, pp. 89–124, 2016. [Online]. Available: https://doi.org/10.5277/e-Inf160105

[8] J. Pantiuchina, M. Mondini, D. Khanna, X. Wang, and P. Abrahamsson, *Are Software Startups Applying Agile Practices? The State of the Practice from a Large Survey*. Cham: Springer International Publishing, 2017, pp. 167–183.

[9] E. Klotins, "Using the case survey method to explore engineering practices in software start-ups," in *2017 IEEE/ACM 1st International Workshop on Software Engineeting for Startups (SoftStart)*, May 2017, pp. 24–26.

[10] C. Giardino, N. Paternoster, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson, "Software development in startup companies: The greenfield startup model," *IEEE Transactions on Software Engineering*, vol. 42, no. 6, pp. 585–604, June 2016.

[11] X. Wang, H. Edison, S. S. Bajwa, C. Giardino, and P. Abrahamsson, *Key Challenges in Software Startups Across Life Cycle Stages*. Cham: Springer International Publishing, 2016, pp. 169–182.

[12] S. S. Bajwa, X. Wang, A. N. Duc, R. M. Chanin, R. Prikladnicki, L. B. Pompermaier, and P. Abrahamsson, "Start-ups must be ready to pivot," *IEEE Softw.*, vol. 34, no. 3, pp. 18–22, May 2017. [Online]. Available: https://doi.org/10.1109/MS.2017.84

[13] E. Klotins, M. Unterkalmsteiner, and T. Gorschek, "Software-intensive product engineering in start-ups: a taxonomy," *IEEE Software*, vol. 0, no. 0, p. 0, 2017, in Print.

[14] R. Chanin, L. Pompermaier, K. Fraga, A. Sales, and R. Prikladnicki, "Applying customer development for software requirements in a startup development program," in *Proceedings of the 1st International Workshop on Software Engineering for Startups*, ser. SoftStart '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 2–5. [Online]. Available: https://doi.org/10.1109/SoftStart.2017...3

[15] S. S. Bajwa, X. Wang, A. N. Duc, and P. Abrahamsson, *How Do Software Startups Pivot? Empirical Results from a Multiple Case Study*. Cham: Springer International Publishing, 2016, pp. 169–176.

[16] A. Nguyen-Duc, X. Wang, and P. Abrahamsson, *What Influences the Speed of Prototyping? An Empirical Investigation of Twenty Software Startups*. Cham: Springer International Publishing, 2017, pp. 20–36.

[17] H. Edison, *Software Product Innovation Through Startup Experimentation in Large Companies*. Cham: Springer International Publishing, 2016, pp. 751–756.

[18] H. Edison, X. Wang, and P. Abrahamsson, "Product innovation through internal startup in large software companies: A case study," in *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Aug 2016, pp. 128–135.

[19] M. Chicote, "Startups and technical debt: Managing technical debt with visual thinking," in *Proceedings of the 1st International Workshop on Software Engineering for Startups*, ser. SoftStart '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 10–11. [Online]. Available: https://doi.org/10.1109/SoftStart.2017...6

[20] J. Yli-Huumo, A. Maglyas, and K. Smolander, *The Sources and Approaches to Management of Technical Debt: A Case Study of Two Product Lines in a Middle-Size Finnish Software Company*. Cham: Springer International Publishing, 2014, pp. 93–107.

[21] B. Curtis, J. Sappidi, and A. Szynkarski, "Estimating the size, cost, and types of technical debt," in *Proceedings of the Third International Workshop on Managing Technical Debt*, ser. MTD '12. Piscataway, NJ, USA: IEEE Press, 2012, pp. 49–53. [Online]. Available: http://dl.acm.org/citation.cfm?id=2666036.2666045

[22] H. Terho, S. Suonsyrjä, and K. Systä, *The Developers Dilemma: Perfect Product Development or Fast Business Validation?* Cham: Springer International Publishing, 2016, pp. 571–579.

[23] Z. Li, P. Avgeriou, and P. Liang, "A systematic mapping study on technical debt and its management," *J. Syst. Softw.*, vol. 101, no. C, pp. 193–220, Mar. 2015. [Online]. Available: http://dx.doi.org/10.1016/j.jss.2014.12.027

[24] S. S. Bajwa, X. Wang, A. Nguyen Duc, and P. Abrahamsson, ""failures" to be celebrated: an analysis of major pivots of software startups," *Empirical Software Engineering*, vol. 22, no. 5, pp. 2373–2408, Oct 2017.

[25] M. M. Lehman, "Programs, life cycles, and laws of software evolution," *Proceedings of the IEEE*, vol. 68, no. 9, pp. 1060–1076, Sept 1980.

[26] *Software Product Lines: Practices and Patterns*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001.

[27] J. Bosch, "Maturity and evolution in software product lines: Approaches, artefacts and organization," in *Software Product Lines, Second International Conference, SPLC 2, San Diego, CA, USA, August 19-22, 2002, Proceedings*, 2002, pp. 257–271.

[28] P. Knauber, D. Muthig, K. Schmid, and T. Widen, "Applying product line concepts in small and medium-sized companies," *IEEE Software*, vol. 17, no. 5, pp. 88–95, 2000.

[29] V.-P. Eloranta, "Towards a pattern language for software start-ups," in *Proceedings of the 19th European Conference on Pattern Languages of Programs*, ser. EuroPLoP '14. New York, NY, USA: ACM, 2014, pp. 24:1–24:11.

[30] A. Dande, V.-P. Eloranta, A.-J. Kovalainen, T. Lehtonen, M. Leppänen, T. Salmimaa, M. Sayeed, M. Vuori, C. Rubattel, W. Weck *et al.*, "Software startup patterns-an empirical study," *Tampereen teknillinen yliopisto. Tietotekniikan laitos. Raportti-Tampere University of Technology. Department of Pervasive Computing. Report; 4*, 2014.