



On the Characteristics of Internal Software Startups

Anastasiia Tkalich¹(✉)  and Henry Edison² 

¹ SINTEF, Trondheim, Norway
Anastasiia.Tkalich@sintef.no

² Blekinge Institute of Technology, Karlskrona, Sweden

Abstract. In recent years, more attention has been given to *internal software startups* in practice and in research alike, yet the concept is not fully understood. Nor is it clear whether or not it significantly differs from stand-alone software startup, and if yes, then how. In this position paper, we propose to conceptualize internal software startups as a hybrid of two related concepts: stand-alone software startup and internal corporate venture (ICV). We derive characteristics of the both concepts from the earlier literature and use our previous research on internal software startups to uncover the differences and the similarities across the three concepts.

Keywords: Internal software startup · Internal corporate venture · Software Startup · Lean Startup · Innovation

1 Introduction

Innovation is becoming very important in an ever-paced world. Corporations must look beyond their trusted walls for opportunities to lead in the business competition. Over the years, corporations have been looking externally to tap into the right skills, which usually includes acquisition (e.g., the acquisition of Skype by Microsoft, Whatsapp by Facebook or Waze by Google) and corporate venture capital (e.g., Google Venture, Intel Capital) to improve their innovation capabilities. In fact, the practice of corporate venturing has grown in its popularity from 2% to 44% in six years [17]. According to the latest report [8], total venture capital hits a new record high in 2021 - reaching \$171.1 billion worldwide.

In recent years, an increasing amount of attention is being given to “copycat” startups by corporations, which are called internal startups [1]. The maturity of the startup approach with the constant growth of investment and proven business models have attracted corporations who look for new ways to expand their business development. Today, internal startup is a new normal as we have seen more and more corporations launching new software products or services as a startup e.g. Niantic (Google) or MESH (Sony). This phenomenon has caught the attention of software engineering (SE) researchers. For example, Marijarvi

et al. [11] reported the experience of large Finnish companies in developing new software through internal startups. The SE literature refers to the phenomenon differently (e.g., internal startups [18], Lean internal startups [5], or internal software startups [20,23]) and there is no consistency in its definition. A clearly stated definition and/or defining characteristics are necessary for scientific understanding, explanation, and prediction [12]. Moreover, it helps researchers to build on each other's work and for practitioners to decide whether research findings are relevant and applicable to their particular situation. As the field of software startup is still in its infancy, the time is ripe to work on the clarification of the existing terminology.

The objective of this position paper is to differentiate an important type of software innovation initiatives, which we call *internal software startup*, from other similar types of initiatives, such as stand-alone software startups and internal corporate ventures (ICVs). To illustrate our point, we look back at our own research on internal software startups and summarize their characteristics. Based on what is already known about software startups and ICVs with their key characteristics, we show how the three concepts differ and where they overlap.

2 Software Startups and ICVs

In this section, we present relevant literature on the both concepts to show which characteristics have been put forward as defining.

2.1 Software Startup

There is a lack of consensus regarding the definition and central characteristics of software startups. In one of the earliest and most-cited papers that used “software startup” term [22], Sutton identified four major characteristics of software startups: youth and immaturity, limited resources, multiple influences and dynamic technologies and markets. Paternoster et al. [16] extended the list further and enumerated 15 characteristics of software startups where the most frequent were: 1) lack of resources and dependency on external sponsors, 2) time pressure, 3) innovation, 4) rapidly evolving, 5) lack of experience, and 6) highly risky. After analysing these six characteristics, Klotins [7] argued that while some of these characteristics have but anecdotal evidence (e.g., lack of experience), others are not unique to software startups (e.g. innovation, lack of resources, time pressure). Nevertheless, many of the aforementioned characteristics are typically used to specify what one means by software startups. For example, Melegati [14] highlights the following characteristics: 1) innovation, 2) software-intensive, 3) searching for repeatable and scalable business model, and 4) high uncertainty.

Combining the insights from earlier literature, in this paper we use the following six characteristics of stand-alone software startups: 1) **innovation** - given the highly competitive ecosystem, startups need to focus on highly innovative segments of the market [14,16]; 2) **high uncertainty** - due to the innovative

nature of the products, startups face high level of uncertainty which they handle through experimentation (e.g., lean startup) [14]; 3) **intention to create a repeatable and scalable business model** [14] - startups are not only about new functionality but intend to create new revenue; 4) **rapidly evolving** - unlike established companies that are constrained by their existing processes and infrastructure, startups can quickly react to changes in the underlying market and technologies, thus rapidly changing in their several aspects all at once [7, 16]; 5) **dependency on external sponsors** - due to a lack of resources software startups heavily rely on external solutions (e.g., APIs, Open Source Software, outsourcing, COTS,) and investment [7, 16]; **software as outcome or ubiquity** - the “software” aspect is present in startups where software is a ubiquity in entrepreneurial operations [21] or outcome of such operations [13].

2.2 Internal Corporate Venture (ICV)

Internal software startups have a lot in common with the concept of internal corporate ventures (ICVs), which has been scrutinized by corporate entrepreneurship (CE) literature for decades. ICV is one of four routes (i.e. research and development - R & D, joint ventures, and acquisitions) to CE [10]. CE research focuses, in its turn, on ways that companies can use to create new businesses that generate new revenue streams and create value for shareholders, thus achieving innovation [15]. There are several important distinctions between ICV and other ways of CE. First, R & D is responsible for leveraging resources in the company to support the main core of the business [2]. The goal of ICV is to **create a new business model** empowered by a new product, extension of a current product to the existing market or extension of the current product to a new market [9]. Second, participants in ICVs are responsible for driving the end-to-end innovation process while at the same time they may continue to fulfill their other responsibilities in the company, whereas the R&D is fully dedicated to the innovation process [2].

This approach is also known as **employee-driven innovation** [6]. The employees can be gradually reassigned to the ICVs full-time as the products mature [10]. Third, ICVs are different from joint ventures (JVs) because they primarily **rely on the resources (skills, finances, marketing) of the parent company** [15], while JVs rely on the pool of resources from several firms. For example, the ideas of ICVs frequently emerge and are carried out in-house [15] (at least in the initial stages), which is different from acquisition where the ideas are a result of purchase or merger.

In recent SE literature, ICVs are sometimes studied under the term *Lean internal startups* [3]. There are three characteristics of lean internal startups [4]: institution, innovation, and extreme uncertainty. *Institution* is about the startup being **integrated within the parent company and relying on its resources functions** (e.g. sales, marketing, testing, security). **Innovation** encompasses the need for different ways of working compared to the routines to allow teams to move fast but at the same time do not threaten the product in the parent company. **Extreme uncertainty** is the main reason for the lean startup

approach, which is basically about reducing the uncertainty. Internal startups are often assigned to an untapped market segment. Thus with all resources from the parent company, there is no guarantee that the internal startup will succeed [4].

To summarise, the defining characteristics of ICV are: 1) innovation [4], 2) high uncertainty [3,4,19], 3) intention to create a new business model [2,9], 4) reliance on the resources of the parent company [4,15], and 5) being driven and owned by individuals or teams of employees [2,6,9,10].

3 Internal Software Startups: Defining Characteristics and Case Illustrations

Examining the characteristics of software startups and ICVs, as discussed in the previous section, one can see that some of them overlap, whereas others are distinct. As shown in Fig. 1 we mapped both lists of the characteristics and used them to illustrate how they relate to each other and to internal software startups that we studied earlier.

Internal software startup: characteristics		Lokki [3,4]	Fast- Café [3,5]	See-Say [3,5]	Small- Ads [3]	Car Ads [3]	Startup 1 [19]	Startup 2 [23]
◆ +	Innovation [14,16,4]	✓	✓	✓	✓		✓	✓
◆ +	High uncertainty [14,4,19,3]	✓	✓	✓	✓	✓	✓	✓
◆+	Intention to create a scalable business model [14,9,2]	✓	✓	✓	✓	✓	✓	✓
◆	Rapidly evolving [7,16]	✓	✓	✓	✓		✓	
◆	Dependency on external sponsors [16,7]							✓
◆	Software as outcome or ubiquity [21,13]	✓	✓	✓	✓	✓	✓	✓
	Reliance on the resources of the parent company [15,4]	✓	✓	✓	✓	✓	✓	✓
	+ Driven and owned by internal employees [10,6,9,2]	✓	✓	✓	✓	✓	✓	✓
Software startup	Internal corporate venture (ICV)	Cases						

◆ Characteristics related to software startups
 + Characteristics related to ICVs

Fig. 1. Internal software startups: characteristics and case examples

The figure shows that software startups and ICVs share three characteristics: innovation, high uncertainty and intention to create a scalable business model. There are some characteristics that are inherent to software startups, but not to ICVs, i.e. rapidly evolving, dependency on external sponsors, and software as outcome or ubiquity of entrepreneurial operations. ICVs are in their turn unique because they rely of the resources of the parent company and are driven by people that are already employed there. When it comes to our previously published cases

of what we believe were internal software startups, the figure shows that most of them contained some characteristics that are inherent to software startups but not to ICVs (e.g. rapidly evolving, software as outcome or ubiquity), some that are only typical to ICVs but not to software startups (e.g. reliance on the resources of the parent company, driven by internal employees), and those that are common for the both concepts. Our first conclusion is, therefore, that internal software startups are entities on their own, which cannot be described by the related concepts, but at the same time partly overlap with them.

Looking at which characteristics were prominent in the cases of internal software startups (the green checks), we see that many of them were surprisingly consistent across all the cases, while others were less consistent. For example, dependency on external resources was only identified in one case, which can suggest that this trait may occur in certain internal software startups but is not as central as the others. In the same way, there was not consistency in terms of whether all of the internal software startups were rapidly evolving or not. This may be due to the lack of clarity of what exactly “rapidly evolving” means and how it can be observed. Klotins [7] argues that this trait is inherent to software startups, because unlike the established organizations, they are not constrained by the existing structures and routines, and can thus change several aspects at once. However, it is hard to specify which exactly aspects one should look for and how rapid should the change be for this characteristic to be present. Another explanation is that while some internal software startups evolve faster, others are constrained by their parent companies. To conclude, we encourage further research on whether this characteristic is significant for internal software startups, how it can manifest, and how it is influenced by the parent companies.

Further, the figure demonstrates that *innovation, high uncertainty, software as ubiquity, reliance on resources of the parent company* and *being driven by internal employees* were present in all our cases. This suggests that these are candidates for defining characteristics of internal software startups. Indeed, the most internal software startups we had studied, provided new offerings that could be scaled and that were expected to bring stable income after a while. Since the offerings were new, the initiatives inevitably faced uncertainty - for example with regard to the market segment, the functionality or the user needs. The internal software startups also relied on the resources of the parent companies who function not only as investors, but also as infrastructure, often providing work stations, competency, and salary to the startups teams. In return, the teams were often accountable for the result of their initiatives and functioned as main drivers behind the new product while still fulfilling other responsibilities at the parent companies (which is different from conventional R&D innovation initiatives that do not have to result in new business models and are fully dedicated to the innovation process [2]). While ICVs concern all sorts of innovations, the internal software startups we studied, had software as the essence of their business or their outcome (see Fig. 1). This implies that the lessons learned from software engineering literature can and should be applied to understand and

succeed with internal software startups. This implication is important because we have so far seen that internal software startups in established companies are sometimes being developed using elements from conventional approaches (e.g. top down decision-making, centralized budgeting, silo-formed context [20]).

4 Conclusions and Further Work

To summarize, we have used earlier literature on software startups and ICVs to derive possible key characteristics of what we argue is *internal software startup*. Using our previous research on this phenomenon, we showed that internal software startups cannot be described by the related concepts, but at the same time overlap with them. Internal software startups are a subset of ICVs, as they have software as their essence, which is not specifically addressed in the ICV literature. However, it is still unclear how internal software startups evolve equally rapidly as stand-alone startups. Future research should address this inconsistency. Other avenues of research should concern validation of the characteristics suggested in this position paper and whether they can be used to further data collection and analyses on internal software startups. Overall, the characteristics we have suggested are a first step toward conceptual understanding of internal software startups that can help researchers better specify their objects of study.

Acknowledgements. This study was supported by the 10xTeams and Digital Class research projects; and by the Research Council of Norway through grants 309344 and 309631.

References

1. Babych, M.: Internal startups: the new normal for business? (2021)
2. Bart, C.K.: New venture units: Use them wisely to manage innovation. *MIT Sloan Manag. Rev.* **29**(4), 35 (1988)
3. Edison, H.: Lean internal startups: challenges and lessons learned. In: Nguyen-Duc, A., Münch, J., Prikladnicki, R., Wang, X., Abrahamsson, P. (eds.) *Fundamentals of Software Startups*, pp. 251–268. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-35983-6_15
4. Edison, H., Wang, X., Abrahamsson, P.: Lean startup: why large software companies should care. In: *Scientific Workshop Proceedings of the XP*, pp. 1–7 (2015)
5. Edison, H., Wang, X., Jabangwe, R., Abrahamsson, P.: Innovation initiatives in large software companies: a systematic mapping study. *Inf. Softw. Technol.* **95**, 1–14 (2018)
6. Høyrup, S.: Employee-driven innovation and workplace learning: basic concepts, approaches and themes. *Transf. Eur. Rev. Labour Res.* **16**(2), 143–154 (2010)
7. Klotins, E.: Software start-ups through an empirical lens: are start-ups snowflakes? In: *CEUR Workshop Proceedings, Volume 2305, CEUR-WS* (2018)
8. KPMG: Q3 2021 venture pulse report - global trends (2021)
9. Kuratko, D.F., Covin, J.G., Garrett, R.P.: Corporate venturing: insights from actual performance. *Bus. Horiz.* **52**(5), 459–467 (2009)

10. Lengnick-Hall, C.A.: Innovation and competitive advantage: what we know and what we need to learn. *J. Manag.* **18**(2), 399–429 (1992)
11. Märijärvi, J., et al.: The cookbook for successful internal startups. In: *Digile N4S* (2016)
12. McKelvey, B.: *Organizational Systematics: Taxonomy, Evolution, Classification*. University of California Press, Berkeley (1982)
13. Melegati, J., Edison, H., Wang, X.: XPro: a model to explain the limited adoption and implementation of experimentation in software startups. In: *IEEE Transactions on Software Engineering 1–1 Conference Name: IEEE Transactions on Software Engineering* (2020)
14. Melegati, J., Guerra, E., Wang, X.: Understanding hypotheses engineering in software startups through a gray literature review. *Inf. Softw. Technol.* **133**, 106465 (2021)
15. Narayanan, V.K., Yang, Y., Zahra, S.A.: Corporate venturing and value creation: a review and proposed framework. *Res. Policy* **38**(1), 58–76 (2009)
16. Paternoster, N., et al.: Software development in startup companies: a systematic mapping study. *Inf. Softw. Technol.* **56**(10), 1200–1218 (2014)
17. Prats, M., Siota, J., Canonici, C., Contijoch, X.: *Corporate venturing: challenges and best practices of large firms innovating with start-ups* (2018)
18. Raatikainen, M., Komssi, M., Kiljander, H., Hokkanen, L., Märijärvi, J., Mohout, O.: Eight paths of innovations in a lean startup manner: a case study. In: Abrahamsson, P., Jedlitschka, A., Nguyen Duc, A., Felderer, M., Amasaki, S., Mikkonen, T. (eds.) *PROFES 2016. LNCS*, vol. 10027, pp. 15–30. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49094-6_2
19. Ries, E.: *The lean startup: how today's entrepreneurs use continuous innovation to create radically successful businesses*. In: *Crown Books* (2011)
20. Sporse, T., Tkalich, A., Moe, N.B., Mikalsen, M.: Understanding barriers to internal startups in large organizations: evidence from a globally distributed company. In: *Proceedings of 16th ICGSE* (2021)
21. Steininger, D.M.: Linking information systems and entrepreneurship: a review and agenda for IT-associated and digital entrepreneurship research. *Inf. Syst. J.* **29**(2), 363–407 (2019)
22. Sutton, S.M., Jr.: Role of process in a software start-up. *IEEE Softw.* **17**(4), 33–39 (2000)
23. Tkalich, A., Moe, N.B., Ulfsnes, R.: Making internal software startups work: how to innovate like a venture builder? In: Wang, X., Martini, A., Nguyen-Duc, A., Stray, V. (eds.) *ICSOB 2021. LNBIP*, vol. 434, pp. 152–167. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91983-2_12

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

