There's No Business Like Software Business: Trends in Software Intensive Business Research

Slinger Jansen(✉)

Utrecht University, Utrecht, The Netherlands
slinger.jansen@uu.nl

Abstract. Software intensive business research is rapidly evolving. Over the last decade we have witnessed a surge in research output but as the field matures, its future remains unsure. In this paper an overview is provided of the highlights and trends of software intensive business research. We briefly discuss the most cited papers in the domain and provide a hype cycle for software intensive business research. With this paper, we hope that researchers can forge more solid research strategies for themselves in the domain, to achieve longevity, academic depth, and impact.

Keywords: Software business hype cycle · Software intensive business · Software ecosystems

1 Introduction to This Keynote Paper

The role of software-intensive solutions is still, after decades, growing in our society. There is hardly a field or an industrial domain where software-intensive solutions have not revolutionized the business. Furthermore, due to the emergence of the platform economy, the role of software solutions and boundary resources, such as the quality of APIs, is rising. The biggest enabler for the growth of software-intensive solutions, is the fact that we are now living in a connected society, where devices, software, and business process can interact at relatively low cost.

Software Intensive Business (SiB) research is a domain that cross–sects software engineering, information systems, and economics. The scientific field of software-intensive business studies organizational arrangements, methods, and tools for value creation, capture, and delivery based on digital products and services. The International Conference on Software Business (ICSOB) was launched to provide a home for this type of research. To follow the trends in SiB research, work is highlighted from the International Conference on Software Business from 2010 to 2016, as these works have been able to amass a significant number of citations. In these works the following major topics are identified: *Software Product Management*, *Continuous X*, *Software Ecosystems*, and *Software Startups*.

© Springer Nature Switzerland AG 2019
S. Hyrynsalmi et al. (Eds.): ICSOB 2019, LNBIP 370, pp. 19–27, 2019.
https://doi.org/10.1007/978-3-030-33742-1_3

In the following Sections we discuss each of these topics. For each topic we highlight some of the most highly cited papers in the domain and identify, using dimensions.ai, the most welcoming forums for these topics.

2 Software Product Management

The domain of *Software Product Management (SPM)* gained its initial traction in the early 2000s, typically in information systems journals and software process conferences. The topic was continuously advanced upon and in 2006 the International Workshop on Software Product Management (IWSPM) was launched, a series that lasted until 2016. The workshop was a success from an academic point of view, as it launched the research area and in a relatively short time professionalized the field of SPM, even to the extent that an initiative was launched for the International Software Product Management Association, an open non-profit association of experts, companies, and research institutes with the goal to foster software product management excellence across industries. In a sense, this association is the culmination of a research domain, where the association actively supports researchers and professionals.

In 2010, when the ICSOB series was launched, it was a logical consequence that SPM would be a main topic of discussion in the conference and it is probably not a coincidence that the best cited paper from 2010 is the seminal work by van de Weerd, Bekkers, and Brinkkemper [18] on developing an SPM maturity model. SPM as a topic is still active, although there appears to be little growth in the domain. Also, the workshop ended in 2016, but that may be in part due to the fact that independent workshops are becoming less common. The main forums where SPM work is published are the Journal of Systems and Software (JSS), Lecture Notes on Computer Science (LNCS), Lecture Notes on Business Information Processing (LNBIP), and the Information and Software Technology Journal (IST).

3 Software Ecosystems

The domain of *Software Ecosystems* was launched with the first International Workshop on Software Ecosystems (IWSECO) in 2009, although earlier works had been using the term [15]. The workshop was launched as an initiative at the International Conference on Software Reuse, as that appeared to be a suitable place for work in this domain. The IWSECO started traveling from conference to conference and was eventually co-located with ICSOB several times. The domain has seen a resurgence in several workshops and is primarily addressed in the software engineering and repository mining communities. The domain is still rapidly growing and works are being written in several new domains, such as blockchain, cryptocurrency, safety and security, platforms, and mobile software.

For the ICSOB community it was logical to also address the topic of software ecosystems, as software ecosystems take a somewhat broader view on the SiB domain. In 2011 the paper with the most citations in the ICSOB community was the work by Jansen and Kabbedijk [12], which presents one of the first in-depth case studies of an open source software ecosystem. In 2013 Bloemendal and Jansen [9] formulate a definition of the concept of an app store at ICSOB and achieve the most citations in that year.

The most welcoming forums for software ecosystems work are JSS, IST, LNBIP, LNCS, the Empirical Software Engineering journal (EMSE), and IEEE Software. The domain is still growing steadily. In this Section we highlight some of the early findings from the creation of an updated research agenda on software ecosystems. The source of the research challenges consists of the notes of a Dagstuhl meeting in 2018 [1] and a series of surveys that has been released to the SiB community.

Engineering Ecosystems. Ecosystems cannot be created: they must be cultivated and grown to enable keystones to gain power from their ecosystems. We find that enabling technologies, such as plug-in architectures, app store architectures, and API architectures create the infrastructure that enables partners to co-create and innovate within an ecosystem. Our first set of challenges centers around enabling partners to engage into the ecosystem through technical infrastructure. There are many enabling technologies for ecosystems, such as plug-in architectures, application stores, and block chains. To engage developers there is a need for code repositories, IDE integrations of ecosystem resources, sand boxes, license protection mechanisms, and even fully integrated development stacks. To monitor and enable partners, there exists a need for incentive systems [16], partner quality monitoring systems, and API performance monitoring. These systems need to be studied more extensively to establish how they contribute to the ecosystem, developer satisfaction, and overall partner performance.

Secondly, there exists a challenge in identifying the barriers to entry for new partners in an ecosystem. We need to study manners to keep thresholds low and employ network effects for the growth of the partner ecosystem.

Analysis of Ecosystem Data. Analyzing ecosystem data is essentially (big) data analytics and techniques from this domain are presently insufficiently applied. Studying a repository such as Github is often compared to drinking water from a fire hose, especially when a research project is focused on the needle in the (Github) haystack. We identify the following data analytics challenges.

First, there is the challenge that most researchers typically limit themselves to a snapshot of one or several ecosystems. However, to answer some of the deeper questions on ecosystem health and attracting new developers and partners to an ecosystems, concrete recipes need to be evaluated in terms of metrics that can be gathered from repositories over time. A large challenge here, is that in software ecosystems data is generally hard to collect. Data is of different types, is hidden behind organizational barriers, and sometimes simply overwritten and unavailable. Monitoring ecosystems over time has become a challenge that requires extensive and long-lasting efforts. Fortunately, with initiatives such as GHTorrent (http://ghtorrent.org/), we are actively curating the data that is needed for durable ecosystem analysis.

Secondly, the concept of ecosystem health, i.e., the propensity for growth of an ecosystem, has been extensively studied. These frameworks have become increasingly elaborate and comprehensive, whereby making them challenging to apply to a research project. Some even call for a customized set of health metrics for every (type of) ecosystem. More research is required into the main performance indicators of ecosystems, the recipes that aim to influence them, and their measurable influence on these performance indicators.

Modeling of Ecosystem Structure and Behavior. Ecosystems are increasingly used as tools for reasoning about an organization's business model [11], market position, opportunities and threats. However, we have not been able to reason at the highest levels of fidelity. There is a need for the development of modeling languages that provide insight and enable analysis at different levels of scale. There are several modeling languages used in the field, such as social network models, goal models, and supply chain models. These models appear to have significant overlap, as they all aim to model actors, software structures, and relationships, and yet each serves a different purpose.

A second challenge is that the current languages do not scale upwards easily. Ecosystems with up to 5 actors can still be modeled in goal modelling languages and power models, but beyond those numbers these models become too complex. Finally, even with such models it becomes complex to monitor and model ecosystems over time.

Management of Developer Ecosystems and Platforms. Ecosystem join decisions are made both on a strategic level but also on an operational level by senior software engineers. Some have coined these software engineers "Kingmakers", as these decisions may lead to long lasting relationships with the technical platform and the keystone organization that supports it.

Software producing organizations address the groups of software engineers in their software ecosystems as "Developer Ecosystems". Managing developer ecosystems is a challenge for software producing organizations in four different ways. First, the platform that the developer ecosystem focuses on needs to be extensible, flexible, robust, evolvable, and provide facilities for rapid development of new solutions. Secondly, the developer community must be managed, by organizing events, coordinating feedback, helping developers help each other, etc. Thirdly, the software producing organization must be ready to accommodate developers, by readily providing them with easy access to the platform as well as to support, knowledge, and advice. Finally, the organization needs to keep track of other ecosystems, the role of open source in the platform, and invest in supporting platforms and ecosystems.

The position of organizations in software ecosystems as a keystone is largely dependent on how they conduct their developer ecosystem, i.e., the ecosystem of collaborating developers that add value to the platform. The field of software ecosystem governance is maturing, but many organizations are still reinventing tools and methods for becoming stronger in a software ecosystem. There is a need for research into the mechanisms that entice, attract, keep, and lock-in developers. These mechanisms range from tools for knowledge sharing, such as joint repositories and API documentation, to release coordination, where release candidates are released to partners early, to enable them to do compatibility checking of their extensions. Another concept that is insufficiently studied is the role of dominant design and standardization in industries. Finally, ecosystem life cycles have not yet been researched comprehensively, and it is as of yet unclear how ecosystems start, grow, develop, die out, and renew again.

4 Continuous X, Agile, and Technical Debt

One topic that has been harder to assign a title to, is the topic of Continuous X, where X can be replaced with testing, improvement, and development. The theme within this domain is the fact that software products are seen decreasingly as finalized artifacts that improve a business process, but increasingly as living artifacts that can be used to evolve with the business process. During the 2018 Dagstuhl meeting on the topic was decided that our field should let go of the 'Software Business' title and move towards "Software intensive Business", a change that can perhaps be attributed to this topical change.

We notice the following patterns in the Continuous X theme and recognize that these are direct lessons from the agile methodology. First, customers should be involved often and early in the development process, to reduce the significant waste that is part of any development process. Secondly, delivery should be frequent, to enable an organization to fail fast and fail often. In industry there are two trends that are at the basis of this research domain: software businesses are maturing into large inflexible organizations and traditional businesses are increasingly finding that they too, are software intensive businesses.

The best cited work of ICSOB 2012 is the work on innovation experiment systems for software products by Jan Bosch [5]. In it, Bosch lays out the first steps towards becoming a more responsive software development organization. Later Fabijan, Ollson, and Bosch [7] again achieve the most cited paper of ICSOB 2015, by elaborating on the methods for data gathering in continuous X processes.

We find that the outlets for work on these topics is best received by IEEE Software, JSS, IST, LNBIP, and LNCS. Also, there appears to be significant growth in this domain, as the number of published papers is steadily increasing.

A topic that deserves more attention from both an academic and a practitioner perspective, is the problem of scaling up from a small to medium software business to a larger one. Even though many companies grow to a healthy five to twenty employees, the hurdle to become a 10 million euro revenue company seems insurmountable for most. We hypothesize that the practices proposed in the domain of Continuous X, together with topics such as agile and aligned autonomy, could support smaller organizations in growing larger; but more research is needed.

5 Software Startups

The topic that has seen the fastest growth over a short time is software startups. With the surge of new startups and startup incubators, there is an audience for deliberate and planned growth of software startups. Several workshops are being started in this domain, but there are few serial academic events. That said, several relatively good articles are published in journals, such as the work by Bajwa, Wang, and Abrahamsson [3].

The community has brought forth several excellent ICSOB papers, such as the best paper of 2014 by Giardino, Wang, and Abrahamsson [8], which proposes a startup behavioral framework that shows reasons for and ways to avoid startup failure. In 2016 the community also has a most cited paper in the ICSOB with work on pivots in startups [2], work that would later be reworked into the journal article in the Empirical Software Engineering Journal (EMSE). Overall, the startup community publishes in forums such as LNBIP, LNCS, Communications of the ACM, JSS, and IEEE Software. A quick analysis of the domain shows a rapid increase in articles year over year and the growth is steadily increasing as well. This makes the topic of startups a relatively safe bet for future work.

6 Software Intensive Business Hype Cycle

In Fig. 1 we illustrate a hype cycle, as inspired by Gartner [14]. First, we address the four themes that can be identified in the best cited papers of ICSOB, being SPM, Software Ecosystems, Continuous X, and Software Startups. Secondly, we place several topics on the hype cycle, to indicate which topics we foresee as potential topics that will yield success in the future.

We identify that robots are an upcoming topic in the field of information systems and could potentially extend the field towards hardware. Secondly, a theme that is increasingly gathering popularity in software engineering and information systems is the application of AI techniques, for instance to support developers in task assignment in large projects. We expect that AI techniques may be able to for instance support task assignment in SPM or be able to support requirements identification in requirements engineering.

We also see a rise in publications on software business and blockchain [4,6] and expect that this is a relatively solid bet for future research in terms of impact and citations. There are two directions that SiB research could take in combination with blockchain. First, there are the cryptocurrencies, which can support new business models in the SiB domain. With cryptocurrencies, software businesses can invent currencies that are only used on their platform, thereby creating complete platform economies, new investment models, and distributed governance mechanisms. Secondly, the blockchain itself can be used to create new software intensive products. Examples are products that enable gateways between cryptocurrencies, products that use smart contracts, or products that can benefit from the fact that the shared distributed ledger is guaranteed to be correct.

One topic that is not addressed in this paper, is the topic of business modeling [13]. This topic is not experiencing the prolific growth of the others and appears not to be gaining much more traction after the seminal work on the business model canvas of Osterwalder and peers [17]. With the word "business"

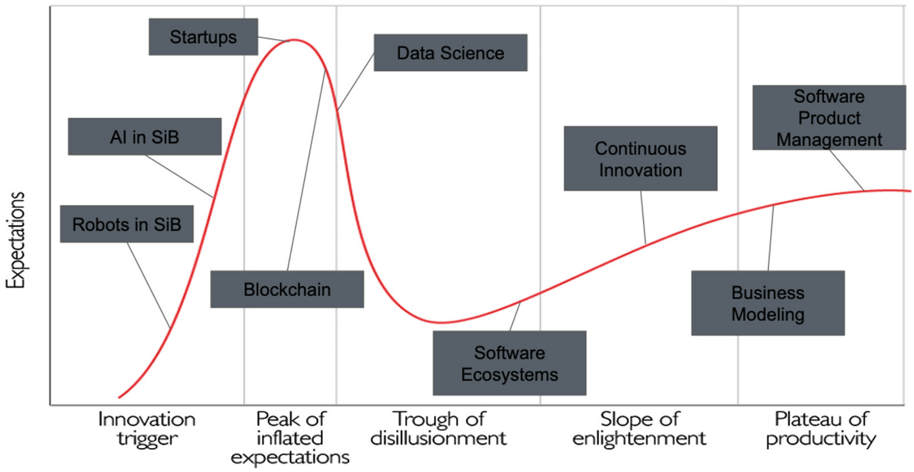

Fig. 1. Software intensive business hype cycle

in the name of the conference, one could have expected more from this domain. However, it is unlikely that the topic of business modeling will experience another surge such as around 2010.

7 Discussion and Conclusions

The future of the domain looks bright. Pockets of researchers will continue their lines on software ecosystems, continuous X, software product management, and software startups. We recognize new trends and research tooling in the domains of data science, artificial intelligence, and blockchain to further facilitate excellent research.

Simultaneously the field suffers from a lack of impact or perhaps a way to signify this impact to the outside world. Few initiatives are undertaken for grant proposal writing in, for instance, a European context. Also, support from companies should be extensive, but only few research groups are supported by industry with data or other resources.

If there is one thing this paper does, it is recognize that there are several sources in which we publish our work. Primarily, SiB work is published in LNCS, LNBIP, IEEE Software, the journal of Information Management, IEEE Computer, JSS, and IST. By recognizing these outlets as our primary outlets, it is perhaps also beneficial to invest heavily in these journals through guest editorships.

Acknowledgement. Please note that parts of the Section on software ecosystems have been discussed by Michael Cusumano, Karl Popp, and myself in an IEEE Software Special Issue on Software Ecosystems [10]. Also, I thank Krzysztof Wnuk for his excellent comments on this paper.

References

1. Abrahamsson, P., Bosch, J., Brinkkemper, S., Mädche, A.: Software business, platforms, and ecosystems: fundamentals of software production research (Dagstuhl seminar 18182). Dagstuhl Rep. **8**(4), 164–198 (2018)
2. Bajwa, S.S., Wang, X., Duc, A.N., Abrahamsson, P.: How do software startups pivot? Empirical results from a multiple case study. In: Maglyas, A., Lamprecht, A.-L. (eds.) Software Business. LNBIP, vol. 240, pp. 169–176. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40515-5_14
3. Bajwa, S.S., Wang, X., Nguyen Duc, A., Abrahamsson, P.: "failures" to be celebrated: an analysis of major pivots of software startups. Empir. Softw. Eng. **22**(5), 2373–2408 (2017)
4. Berkhout, M., van den Brink, F., van Zwienen, M., van Vulpen, P., Jansen, S.: Software ecosystem health of cryptocurrencies. In: Wnuk, K., Brinkkemper, S. (eds.) ICSOB 2018. LNBIP, vol. 336, pp. 27–42. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04840-2_3
5. Bosch, J.: Building products as innovation experiment systems. In: Cusumano, M.A., Iyer, B., Venkatraman, N. (eds.) ICSOB 2012. LNBIP, vol. 114, pp. 27–39. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30746-1_3
6. Boshuis, S., Braam, T.B., Marchena, A.P., Jansen, S.: The effect of generic strategies on software ecosystem health: the case of cryptocurrency ecosystems. In: Proceedings of the 1st International Workshop on Software Health, pp. 10–17. ACM (2018)
7. Fabijan, A., Olsson, H.H., Bosch, J.: Customer feedback and data collection techniques in software R&D: a literature review. In: Fernandes, J., Machado, R., Wnuk, K. (eds.) ICSOB 2015. LNBIP, vol. 210, pp. 139–153. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19593-3_12
8. Giardino, C., Wang, X., Abrahamsson, P.: Why early-stage software startups fail: a behavioral framework. In: Lassenius, C., Smolander, K. (eds.) ICSOB 2014. LNBIP, vol. 182, pp. 27–41. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08738-2_3
9. Jansen, S., Bloemendal, E.: Defining app stores: the role of curated marketplaces in software ecosystems. In: Herzwurm, G., Margaria, T. (eds.) ICSOB 2013. LNBIP, vol. 150, pp. 195–206. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39336-5_19
10. Jansen, S., Cusumano, M., Popp, K.M.: Managing software platforms and ecosystems. IEEE Softw. **36**(3), 17–21 (2019)
11. Jansen, S., Cusumano, M.A., Brinkkemper, S.: Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry. Edward Elgar Publishing, Cheltenham (2013)
12. Kabbedijk, J., Jansen, S.: Steering insight: an exploration of the ruby software ecosystem. In: Regnell, B., van de Weerd, I., De Troyer, O. (eds.) ICSOB 2011. LNBIP, vol. 80, pp. 44–55. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21544-5_5
13. Khurum, M., Gorschek, T., Wilson, M.: The software value map - an exhaustivecollection of value aspects for the development of software intensiveproducts. J. Softw. Evol. Process. **25**(7), 711–741 (2013)
14. Linden, A., Fenn, J.: Understanding gartner's hype cycles. Strategic Analysis Report No R-20-1971. Gartner, Inc. (2003)

15. Messerschmitt, D.G., Szyperski, C.: Software Ecosystem: Understanding An Indispensable Technology and Industry. The MIT Press, Cambridge (2005)
16. Nakasai, K., Hata, H., Matsumoto, K.: Are donation badges appealing?: a case study of developer responses to eclipse bug reports. IEEE Softw. **36**(3), 22–27 (2019)
17. Osterwalder, A., Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. Wiley, Hoboken (2010)
18. van de Weerd, I., Bekkers, W., Brinkkemper, S.: Developing a maturity matrix for software product management. In: Tyrväinen, P., Jansen, S., Cusumano, M.A. (eds.) ICSOB 2010. LNBIP, vol. 51, pp. 76–89. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13633-7_7