

# Integrating Approaches in Software Development: A Case Analysis in a Small Software Company

Mary Sánchez-Gordón <sup>1</sup>[0000-0002-5102-1122], Ricardo Colomo-Palacios <sup>1</sup>[0000-0002-1555-9726], Alex Sánchez<sup>2</sup> and Sandra Sanchez-Gordon<sup>3</sup>

<sup>1</sup> Østfold University College, Halden, Norway  
{mary.sanchez-gordon, ricardo.colomo-palacios}@hiof.no

<sup>2</sup> LogicStudio Panamá, Panamá  
alex.sanchez@logicstudio.net

<sup>3</sup> Escuela Politécnica Nacional Quito, Ecuador  
sandra.sanchez@epn.edu.ec

**Abstract.** There are a myriad of software development methods, methodologies, frameworks, techniques and practices in both traditional and agile software development. Seeking synergy between these approaches has become necessary for the evolution of a software development process. Software companies deal with that challenge by combining well-structured comprehensive methods and flexible agile practices. In fact, some studies have revealed that mixed approaches in software industry are not uncommon. This paper analyzes a case study of the evolution of a software development process in a small company, which is based on Design Thinking, PMBOK and SCRUM. Results show the natural evolution of the software process over 15 years and lessons learned from a pragmatic process selection. The evolution in the company depends on its adaptability which captures the capacity of the company to learn, combine experience and knowledge, and adjust its software process to a changing context. Despite the results are promising further studies should be done.

**Keywords:** Agile Software Development, Case Study, Lessons Learned.

## 1 Introduction

Software practitioners, especially in small companies because of their nature [1], face a major challenge in shaping the many available methods, methodologies, frameworks and techniques [2, 3]. Even in the software process improvement field, there is a lot of diversity [1]. In fact, there is no silver-bullet or one-size-fits-all solution to all software development settings [3, 4]. Although, the literature has reported Waterfall as one of the most popular traditional (non-agile) approaches [4, 5], there are a large number of them, e.g. prototyping, spiral model and unified process. Additionally, Scrum is a well-known and popular agile approach [4, 5] but there are also a large number of them, e.g. Scrum, eXtreme Programming, Feature Driven Development and Crystal. Each approach, whether traditional or agile, is characterized by an individual specific structure that reflects the particular point of view and experiences of

who created it [2]. Given that each approach also has its own style and terminology to describe its selected practices, sometimes it is hard to distinguish common practices.

In this scenario, integrating agile and traditional approaches has become necessary for the evolution of a software development process [6]. It is revealed in the growing popularity of the mixed (hybrid) approaches in software industry [4, 7] while it is gaining increasing attention from the software engineering research community [2, 5, 8, 9]. Hybrid development is a fruitful research field [2, 8] in which there is a lack of evidence concerning combination patterns and contextual factors that drive the creation of hybrid approaches [9]. By conducting a case analysis of a software development process in a small software company, we this exploratory study aims to better understand the historical evolution. The findings confirm that many variations of software development approaches could occur, even within a small company.

## **2 Study Context**

### **2.1 Company Background**

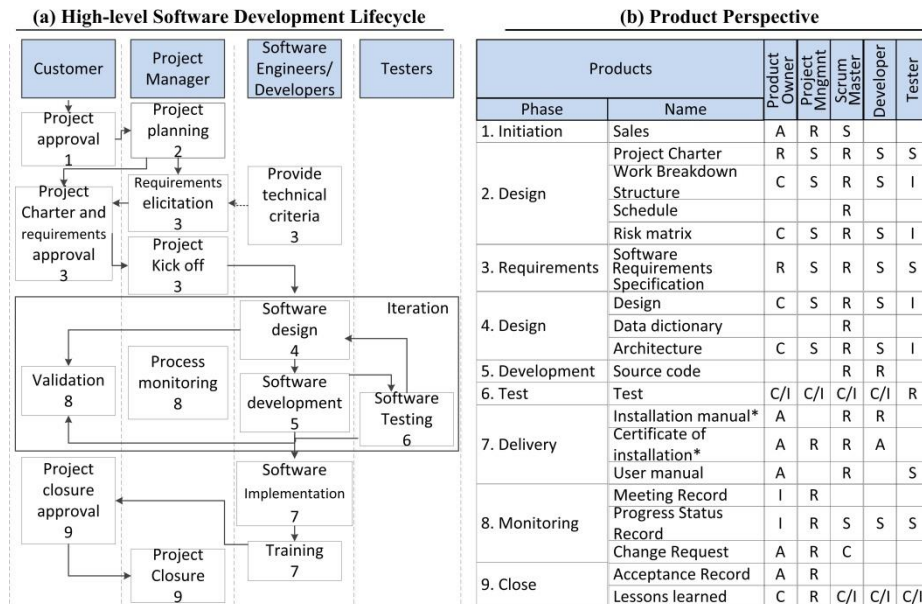
Logic Studio is a recognized Latin American outsourcing and software company that provides solutions to corporations, banks, and the public sector taking advantage of web and mobile technologies in order to create innovative services. Founded in 2003, Logic Studio has more than 100 successful projects in 7 countries. Logic Studio, with 90 employees, has branches in 2 countries, although its headquarters is located in Panama. The headquarters made around 1.8 million dollars in 2018 and 1.9 million dollars in 2019. Moreover, Logic Studio has commercial representatives in Lima-Peru and Florida-USA. Logic Studio is also a Microsoft Gold Certified Partner, whose experts have been awarded with the Microsoft Most Valuable Professional (MVP) and Microsoft Regional Director (RD) awards.

### **2.2 High-level Software Development Lifecycle and Product Perspective**

Logic Studio aims to empower their customers in order to significantly increase the success of their software development projects, while complying with the best practices on project management, software engineering, and agile development. To do so, Logic Studio manages the software development life cycle (SDLC) using an approach based on 1) the understanding of customer needs through Design Thinking (DT) [10]; 2) the best practices of the Project Management Body of Knowledge (PMBOK) [11]; and 3) SCRUM [12] for the agile development of software. The main goals are: 1) Delivery of products that meet company regulations and customer requirements on time and within budget; 2) Alignment of the vision of both customer and development team; 3) Motivated and involved team members to achieve better results; and 4) High productivity and quality.

Fig. 1(a) shows a high-level view of the SDLC in the company. The DT approach disclosed by d.school was chosen to address a better understanding of the customer need through a systematic exploration at the initiation phase before project approval.

The aim is to understand the right product to develop. This approach encompasses five phases: Empathize, Define, Ideate, Prototype and Test.



**Fig. 1.** High-level SDLC (a); product perspective on the SDLC (b).

The project management process is based on five main categories of PMBOK: Initiating, Planning, Executing, Monitoring and Controlling, and Closing. The processes are often iterated prior to completing the project and can have interactions within a Process Group and among Process Groups. Using the context of the software project, i.e. the specific project characteristics and environment, the project team seeks to understand the constraints in which the team should focus their efforts. The relationship among factors such as scope, quality, schedule, budget, resources and risks is such that if any factor changes, at least one of the other factors might be affected. Therefore, the project team needs to be able to assess every situation, balance demands and maintain proactive communication with stakeholders in order to deliver a successful project. Such project should be what stakeholders are expecting so they can find the project acceptable.

On the other hand, Scrum includes an iterative, development approach with early deliverables and well-defined responsibilities that promotes transparency, inspection and adaptation. A sprint is a 2-4 weeks period of development time. Moreover, the responsibilities of the traditional role of project manager are divided and complemented between three roles: Product Owner, Scrum Master and development team. In the last one, analysts, designers and testers are involved. The “look and feel” design is done by a graphic designer, if required, and validated by the customer in each delivery. The main focus is to deliver value continuously in a “time boxed” manner while the team members work together and not individually to build the product. The pro-

cesses are ISO-9001 certified and the engineering activities are operating at CMMi Level 3.

From the product perspective, there is a set of expected work products related to each SDLC phase. Fig. 1(b) shows the relationships between products and the actions of each role —Responsible (R), Approve (A), Read (C), Informed (I), and Support (S). The specific project management approach is selected according to the nature and duration of the project as well as the amount of the project budget.

### 3 Research Method

An single-case study was carried out to explore the natural evolution of software process in a small software company. Hence, we used qualitative data regarding the (general) process use and experiences in order to gain an understanding of the underlying motivation, opinions, and practice of software engineers when tailoring the software process. . In what follows a brief description of the procedure is described.

*Documentation analysis.* Two authors were granted access to the documentation, related to both the software development processes and/or work products (December 2018). The preliminary documentation analysis was used as a basis to define the study context. Moreover, the documentation analysis was important when validating research and analyzing the content or reasoning contained within a document.

*Survey.* Two authors conducted the survey (January 2019) with project team members identified by the Chief Executive Officer (CEO). The questionnaire is based on the instrument developed by HELENA study [5]. The survey was anonymous and participation was voluntary. However one reminder was sent to each potential participant. The data was gathered from 17 projects in order to better understand the software development process in the company. Only one of the responses was incomplete —as respondents were given the option to skip questions. Specifically, three questions were not answered. The HELENA is an international exploratory multistage survey-based study on the use of “**H**ybrid **dE**ve**L**opm**EN**t **A**pproaches in software systems development” [13] that was launched in 2016. Two of three stages have been conducted globally in more than 25 countries and involved about 75 researchers [5]. The questionnaire comprises 38 questions aimed at collecting data on general process use, process use in the context of norms and standards, process improvement, and experiences. Results revealed that hybrid development approaches in software system development are a reality that affects companies regardless of size and industry sector.

*Interview.* After a first analysis of the survey results, two authors conducted the interview with the CEO/Funder (beginning February 2019) in order to understand better the results of the survey and the evolution of the software process in the company. The interview allowed more in-depth discussions and targeted questions about the previous results. The interview (semi-structured and in-depth) lasted approximately

45 minutes and it was voice recorded and transcribed verbatim. Finally, the transcript was sent to the interviewee in order to correct misunderstandings.

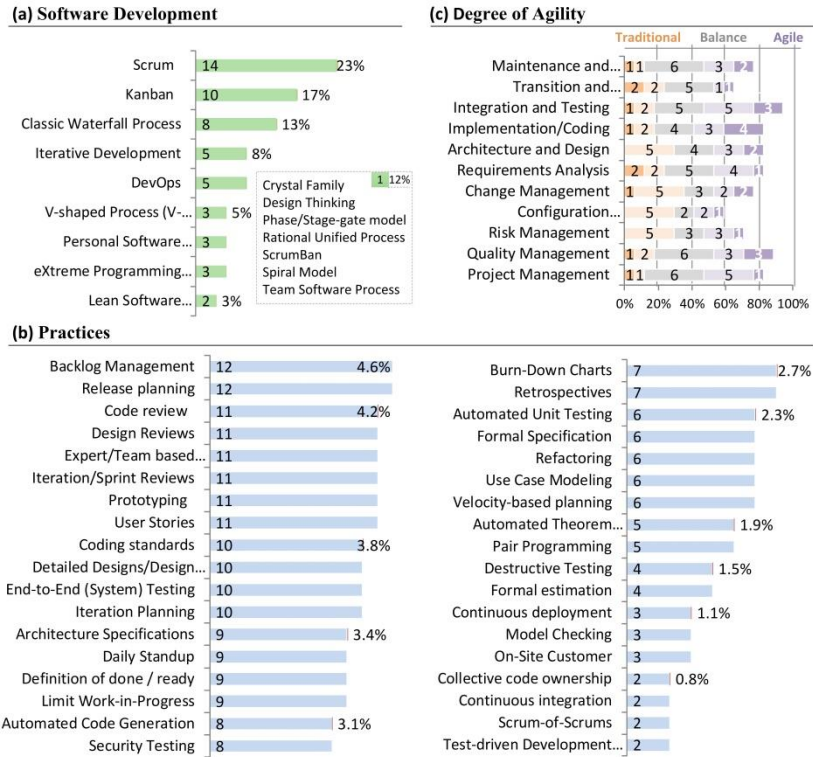
*Data analysis.* Two authors constantly analyzed, coded and reviewed the transcript in the light of data collected from the previous survey and documentation analysis. To do so, the constant comparative method used within the Grounded Theory (GT) method was employed to evaluate new data. Thus, the iterative analysis of the data from the interview was augmented by the documentation analysis. It was achieved by constantly referring to the information provided in the intranet for checks and validation. The findings were also consolidated by using constant comparative method. The author not involved in the data analysis was tasked to provide the quality assurance.

## **4 Results and Lessons Learned**

### **4.1 Status Report**

Respondents informed that, they work in all size projects, but the majority of them are medium (2-6 person months). The projects are running in some industry sectors: “Financial Services”, “Cloud Applications and Service”, “Web Applications and Services”, “Mobile Applications” and “Public Sector/Public Contracting”. Almost all respondents recognized that a software failure conceivably can impact the company's reputation and company's business. About half of the respondents also believe that such a failure can lead to system (service) degradation. Similarly, legal consequences (civil law) and financial loss were identified in 3 and 2 projects, respectively. It is not surprising that respondents informed all projects of the company are operated according to the same (potentially customized) standard process. Although the project-specific development approach was mostly defined by a project manager who tailors the process in the beginning of a project, respondents recognized others ways as well. Sometimes, the process follows defined rules or specific practices and methods that are selected in the project according to customer demands. However, two less experienced (1-2 years) respondents pointed out that specific practices and methods are selected in the project on demand and the process is not tailored at all.

An overview of the use of software development approaches is shown in Fig. 2 (a). It is based on the respondents' rating in four categories: we rarely use it, we sometimes use it, we often use it, and we always use it. Other categories such as “do not know a specific framework or method” and “do not use it” are not depicted. As it was expected Scrum is on the top, followed by Kanban, Classic Waterfall, Iterative Development and DevOps while HELENA study [5] for the combined process use in the entire (non-filtered) dataset shows Scrum followed by Iterative Development, Kanban, Classic Waterfall and DevOps. Therefore, there is no surprise in top-5. A likely explanation for the use of Waterfall in LogicStudio is in the outsourcing requirements. Regarding PMP, it was not mentioned at all. Although PMP is part of the homegrown process, the respondents would probably be not aware of it. Finally, DT was incorporated as part of the homegrown process and its use is limited at the time.



**Fig. 2.** a) Overview of the use of software development (a); practices (b); degree of agility (c).

Despite that, previous studies [5, 8, 14] have revealed the diversity of combinations of individual practices and methods from both traditional and agile approaches, it was unexpected the variety of them within a small company (see Fig. 2(a) and (b)). The results of the analyses of the entire (non-filtered) dataset in HELENA study [5, 15] identified three practices —Code Review, Coding Standards, Release Planning— as the most commonly used practices. In LogicStudio, Backlog Management and Release planning were two of the most used practices along with other six practices —Code Review, Design Reviews, Expert/Team based estimation, Iteration/Sprint Reviews, Prototyping and User Stories.

In spite of the fact that 71% of the respondents pointed out that implementing agile is easy in their context project, the remaining 29% perceive challenges. In HELENA study [5], half of the participants state that implementing agile is easy in their context (59%). The literature, e.g. [8, 14], identifies as a major challenge how to combine mixed approaches. However, in this study the identified challenges were documentation requirements to compliance with external standards (50%), resistance to change (25%) and lack of knowledge about the agile methods and practices (25%). By analyzing the data gathered through the inspection to the project repositories we did not find the description of all the used methods and practices. Therefore, it seems that some respondents rather use their experience to define their software processes. A

likely explanation is that an explicit construction process is missing. In the literature, we found two relevant proposals that highlight its importance and aim to address this issue at both organizational and project levels [2, 8]. With regard to the degree of agility, respondents state that the approach tends to be agile as Fig. 2(c) shown. However, it is based on self-perception of the terms “agility” and “traditional”, and classified according to SWEBOK categories. When compared to the results of the HELENA study [5], there were less “don’t know” and “not answered” answers than the results in Fig. 2(c). However, 7 of the 11 SWEBOK categories seem to be aligned in both studies. In particular, “Change Management” is perceived more “agile” in the HELENA study [5] while “Maintenance and Evolution”, “Quality Management” and “Project Management” are perceived more “traditional” than the results in Fig. 2(c).

Similarly, according to the respondents’ perception, the current development approach is helpful. On one hand, about half of the respondents state that they are generally satisfied with the level of agility and they believe it is sufficient but 80% of them would change or improve it if they could. Those results seem to be aligned with the results of HELENA study [5]. On the other hand, 59% of the respondents reported that they (intentionally) combine different development approaches while 78% of participants positively answered this question in the HELENA study [5]. When comparing the reasons for external standards reported by this study with the reasons reported by HELENA study [5], most respondents agreed that, external models and standards such as CMMI and ISO 9001 are due to mainly the internal policies (65%, 60%), follows by external triggers (35%, 58%) and requirements of the company or the project business (12%, 52%). Therefore, despite that the reasons were the same their distribution was different. Regarding the assessment of compliance of both studies, in most of the projects, the compliance is firstly assessed by internal assessments (82%, 74%). However, respondents also point out that they have applied constructive measures (35%, 43%) and analytical measures (29%, 40%). Additionally, HELENA study [5] reported external project assessments (63%).

## 4.2 Lessons Learned

According to Jacobson and Stimson [2], it is evident that “we need better ways of working that put us on the road to real software engineering”. In fact, literature has underlined that the mixed approaches are a reality in the software industry and the situation reported in this case study provides empirical evidence to better understand the historical evolution of the software process in a small software company. Since the beginning the CEO/Funder realized that having a good software method provides a competitive advantage, but he knew by experience that one size does not fit all projects. Obviously, programming was the major concern in the early years so that the other important things were ad hoc. Over the period from 2003 to 2007, Logic Studio grew in terms of sales, customers and employees. However, the individual perspectives, prejudices and experiences of project managers were reflected, and not to what the company as a whole had collectively learned during those years. In other words, every “method” was controlled by a “warden” as explained in [2]. By 2007, the corporate customers such as HSBC, Banesco, ASSA, Copa Airlines, Adidas and Sage

USA expected that the projects were managed according PMBOK then Logic Studio adopted it. It was agreed that traditional processes provide predictability, stability and high quality assurance [6]. In fact, the CEO got a Project Management Professional (PMP) Certification that year. Then, the first approach reused what the company considered the best practices for their particular challenges and purposes. This finding is supported by Kuhrmann et al. [14], who concluded that experience, learning and pragmatism driven a natural process evolution.

Eventually, it was necessary that such an approach was explained and explicit. Hence, a digital repository was implemented. However, the schedules were inflexible and predetermined so that it was frequently necessary to adjust the estimated time as the company is mainly focused on innovation. In that context, agile offered the flexibility to more easily adjust to changes in project requirements [6]. Therefore, a small team was immediately given Scrum training. As a result, the CEO was the first person in becoming a Certified Scrum Master (CSM, 2008), Certified Scrum Product Owner (CSPO, 2010) and Agile Certified Professional (PMI-ACP, 2014). Nevertheless, adopting an agile approach was challenging not only for the practices, but also because, customers approved a fixed budget. Indeed, contract negotiation is an important aspect that remains a fundamental business instrument in many engagements [1]. Despite that fact, the company decided to take that risk in 2013 by developing a mixed approach that combined the unique strengths, and lessened the weaknesses of both approaches Scrum and PMBOK. Thus, project-specific software processes were built by choosing practices from PMBOK to give a “safe” environment for managers and practices from Scrum to achieve freedom for developers. In this sense, the systematic literature review carried out by Theocharis et al. [9] in 2015 reported that there is a clear trend toward adopting Scrum and some indication that Scrum is often used in combination with other software development approaches. Once the mixed approach was defined, the company developed a training plan, including conferences and workshops about Scrum, PMP and the mixed approach. Project managers were the first to attend the workshop followed by all development team members. Moreover, new employee orientation and induction processes were defined.

The mixed approach was defined as follows: starting with the most critical requirements for the specific project in order to deliver maximum value to the customer in the first iterations while the remaining requirements are left for next versions and additional planning cycles. In this way, according to the company, customers could see something tangible and feel pleased. These results are consistent with the benefit reported in some studies, e.g. [16, 17]. The company also acknowledged that the specific momentum was crucial in the adoption. It coincided with the banking innovation in Panama so that such a sector really appreciated the workshops and training courses about Scrum and PMP offered by the company. In 2014, the results provided some indication not only of the benefits but also of the need to strengthen the process. In this context, an increasing interest in software process improvement (SPI) enriched by continuous learning practices [18] emerged. Thus, a training course focused on “CMMI for Development v1.3” was undertaken and SPI became a high priority. In 2015, the software process were ISO 9001:2008 certified and after three annual audits the company has achieved an ISO 9001:2015 certification. In 2018, after 2 years of



efforts, the company accomplished the goal of receiving a CMMI Level 3 certification. As a result, the defects have decreased although previous studies, e.g. [16, 17], have reported such an effect in CMMI Level 5 companies. Scrum and CMMI together bring a combination of adaptability and predictability [16]. The company relies on the current processes and the development team's freedom to pick between a full agile or full predictive project management and mix those practices.

Emerging challenges and new business opportunities, especially in USA, required more flexibility. In this context, DevOps captured the attention of the company due to its strong focus on rapid and continuous delivery. Then, a subset of DevOps practices was chosen in 2017 as a feasible approach to minimize risks. Apart from that, the SPI initiatives helped the company to realize that innovation projects should be a majority in the future. Therefore, after attending training on DT, it has been adopted since 2018. The initial results are in line with a previous study [19] that point out the extra effort spent on it helps the development teams to have a deeper understanding of the problem to be solved.

Regarding the effects of the whole mixed approach, two kinds of perceptions were identified. They are well illustrated by the following two quotes: “for our customers, more innovative projects which generate/add value from the beginning that are easier to justify and win approval of the [customer] management” and “for us, a more efficient planning and the satisfaction of keeping our customers satisfied”. However, it is worth noting that the approach has a homegrown presentation that makes hard to compare it with others [2]. So far, the company cannot see value on use a formal specification to define it. Finally, it is worth to note that Logic Studio plans to adopt the indicators proposed in the CMMI Level 4 in the near future.

### 4.3 Threats to validity

There are some threats to the validity of this study. The major threat is related to the sample size since this study is focused on only one small software company. The case company is not representative for all software small companies therefore the results must be interpreted with some caution when moving away from the features of the studied company. Moreover, surveys and interviews may be subject to post hoc rationalization and recall biases. However, the research approach allowed us to perform an in-depth qualitative analysis that used multiple sources of evidence. Such a triangulation process allowed us to validate insights by accessing different perspectives.

Another potential threat is related to conducting interviews due to the fact that it always includes some possible bias from the interviewer. To mitigate that threat two authors were involved and an interview guideline, which was previously reviewed by the other authors, was developed. The transcript was also sent to the interviewee to correct misunderstandings. The number of participants —convenience sampling strategy— and the self-reporting structure of the survey is another limitation of this study. To handle the risk related to convenience sampling strategy, this study followed the approach used in the HELENA study [15]. It means before analyzing the data, data pre-processing including a consistency check of the data was implemented. Regarding the HELENA survey instrument, it was developed and refined in several iterations by

a team of researchers who built the questionnaire, tested and revised it. The threats to validity reported in [15] regarding the use of this instrument are briefly discussed below.

The online questionnaire might lead to incomplete or wrong answers since multiple-choice questions might have been incomplete and respondents may have misunderstood questions or answer options. To mitigate these threats, multiple-choice questions were complemented with a free-text option and a qualitative analysis of the free-text answers were carried out by the authors. The questionnaire was administered in Spanish to mitigate the risk of misunderstandings due to language issues. Finally, the results and conclusions were validated by the company.

## 5 Conclusions

In this paper, the historical evolution of a software development process in a small company was analyzed. In this case, such an approach was motivated by improving the quality of the services and products, not only for providing compliance with the international standards, but also for improving the flexibility and speed of response to customer needs and innovation challenges. The company was founded in 2003 but an incremental harmonization of different software development approaches started in 2007 and took around a decade. The first approach adopted by the company was PMBOK (2007). After five years (2013), the natural evolution resulted in a mixed approach based on PMBOK and SCRUM. That triggered a gradual improvement of company's capabilities for compliance with ISO 9001 (2015) and CMMI Level 3 (2018). However, emerging challenges and new business opportunities, especially in USA, required more flexibility, and rapid and continuous delivery then DevOps was integrated (2017). Lately, global competition has forced the company to adopt an innovation process so that DT (2018) has been incorporated. In this way, the company aims to make business successful while faces the need for change which is in line with *Business* and *Change* values of SPI Manifesto [20]. Agile approaches offer an appealing combination of economy and simplicity that allows small software company to increase the degree of success when they undertake software projects. In this sense, the CEO pointed out "we tried several other approaches to improve our software development life cycle, most of them with the precept of '*no pain, no gain*'. Once we found agility our customer easily come on board with the focus on early delivery of results and continuous value generation, now our motto is '*no gain, no way*'".

The findings from this study about the pragmatic process selection and its evolution over time are aligned with the results of HELENA study [5], but this study obtains further in-depth information. The sense of urgency and vision, called "common sense" by the CEO, along with continuous learning practices have addressed the evolution of the software development process. But the actual practice is what makes the company agile and enables it to adapt to changing situations without sacrificing formality. In fact, the actual practice reflects the mixed approach developed. One interesting point is the unexpected diversity of software development approaches and prac-

tices founded within a small company. Therefore, further research is needed to understand this fact.

Despite the effort made, this mixed approach also requires a construction process in order to enable a systematic design of the project-specific development approach as suggested by [2, 8]. The evolution of software process in the company depends on its adaptability which captures its capacity to learn, combine experience and knowledge, and tailor its software process to changing environment. The main limitation of this study is the sample, which is one small company as well as the collected and analyzed data is from (17) respondents mostly located in Panama. Therefore, further research is also needed to increase coverage. Although the results of the mixed approach are promising, additional studies are necessary to know how agile and traditional approaches are combined and how they relate to the particular company context [9]. Moreover, it is worth noting that the DT approach integrated with SCRUM has been little studied in the scientific literature [19].

Finally, this study aims to encourage empirical research and documentation of the lessons learned from companies that seek synergy between traditional and agile software development approaches.

## References

1. Sánchez-Gordón, M., Colomo-Palacios, R., de Amescua Seco, A., O'Connor, R.V.: The Route to Software Process Improvement in Small- and Medium-Sized Enterprises. In: *Managing Software Process Evolution*. pp. 109–136. Springer International Publishing (2016). [https://doi.org/10.1007/978-3-319-31545-4\\_7](https://doi.org/10.1007/978-3-319-31545-4_7).
2. Jacobson, I., Stimson, R.: Escaping Method Prison – On the Road to Real Software Engineering. In: *The Essence of Software Engineering*. pp. 37–58. Springer International Publishing (2018). [https://doi.org/10.1007/978-3-319-73897-0\\_3](https://doi.org/10.1007/978-3-319-73897-0_3).
3. Clarke, P., O'Connor, R.V., Yilmaz, M.: In search of the origins and enduring impact of Agile software development. In: *Proceedings of the 2018 International Conference on Software and System Process*. pp. 142–146. ACM Press, Gothenburg, Sweden (2018). <https://doi.org/10.1145/3202710.3203162>.
4. Vijayarathy, L.R., Butler, C.W.: Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter? *IEEE Softw.* 33, 86–94 (2016).
5. Kuhrmann, M., Tell, P., Klünder, J., Hebig, R., Licorish, S., MacDonell, S.: Complementing Materials for the HELENA Study (Stage 2), DOI: 10.13140/RG.2.2.11032.65288.
6. Boehm, B., Turner, R.: *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley, Boston (2003).
7. Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Trektene, K., Caffery, F.M., Vahid, G., Felderer, M., Linssen, O., Hanser, E., Prause, C.: Hybrid Software Development Approaches in Practice: A European Perspective. *IEEE Softw.* 36, 20–31 (2018).
8. Küpper, S., Rausch, A., Andelfinger, U.: Towards the Systematic Development of Hybrid Software Development Processes. In: *Proceedings of the 2018 International Conference on Software and System Process*. pp. 157–161. ACM, New York, NY, USA (2018). <https://doi.org/10.1145/3202710.3203158>.

9. Theocharis, G., Kuhrmann, M., Münch, J., Diebold, P.: Is Water-Scrum-Fall Reality? On the Use of Agile and Traditional Development Practices. In: Abrahamsson, P., Corral, L., Oivo, M., and Russo, B. (eds.) *Product-Focused Software Process Improvement*. pp. 149–166. Springer International Publishing (2015).
10. The Hasso Plattner Institute of Design: Get Started with Design Thinking, <https://dschool.stanford.edu/resources/getting-started-with-design-thinking>.
11. Project Management Institute: A Guide to the Project Management Body of Knowledge (PMBOK Guide). Project Management Institute (2013).
12. Schwaber, K., Sutherland, J.: Home | Scrum Guides, <https://www.scrumguides.org/>.
13. Tell, P., MacDonell, S., Licorish, S.A.: 3rd Workshop on Hybrid Development Approaches in Software System Development. In: Kuhrmann, M., Schneider, K., Pfahl, D., Amasaki, S., Ciolkowski, M., Hebig, R., Tell, P., Klünder, J., and Küpper, S. (eds.) *Product-Focused Software Process Improvement*. pp. 433–440. Springer International Publishing (2018).
14. Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., Trektene, K., McCaffery, F., Linssen, O., Hanser, E., Prause, C.R.: Hybrid Software and System Development in Practice: Waterfall, Scrum, and Beyond. In: *Proceedings of the 2017 International Conference on Software and System Process*. pp. 30–39. ACM, New York, NY, USA (2017). <https://doi.org/10.1145/3084100.3084104>.
15. Tell, P., Klünder, J., Küpper, S., Raffo, D., MacDonell, S., Münch, J., Pfahl, D., Linssen, O., Kuhrmann, M.: What are Hybrid Development Methods Made Of? An Evidence-based Characterization. Presented at the International Conference on Software and Systems Process ICSSP, Montreal, QC, Canada, Canada (2019). <https://doi.org/10.1109/ICSSP.2019.00022>.
16. Sutherland, J., Jakobsen, C.R., Johnson, K.: Scrum and CMMI Level 5: The Magic Potion for Code Warriors. In: *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*. pp. 466–466 (2008). <https://doi.org/10.1109/HICSS.2008.384>.
17. Ortiz, G.A., Trujillo, M.E.M., Oktaba, H., Hernandez, E.R.: Integrating Agile Methods into a Level 5 CMMI-DEV Organization: a Case Study. *IEEE Lat. Am. Trans.* 14, 1440–1446 (2016). <https://doi.org/10.1109/TLA.2016.7459632>.
18. Kuhrmann, M., Muench, J.: SPI is Dead, isn't it? Clear the Stage for Continuous Learning! In: *2019 IEEE/ACM International Conference on Software and System Processes (ICSSP)*. pp. 9–13 (2019). <https://doi.org/10.1109/ICSSP.2019.00012>.
19. Pereira, J.C., Russo, R. de F.S.M.: Design Thinking Integrated in Agile Software Development: A Systematic Literature Review. *Procedia Comput. Sci.* 138, 775–782 (2018). <https://doi.org/10.1016/j.procs.2018.10.101>.
20. Pries-Heje, J., Johansen, J.: SPI Manifesto, [http://www.iscn.com/Images/SPI\\_Manifesto\\_A.1.2.2010.pdf](http://www.iscn.com/Images/SPI_Manifesto_A.1.2.2010.pdf).