

Software Processes Used in University, Government, and Industry: A Systematic Review

Caroline Guterres Silva, Evandro Luis Fensterseifer Filho and Lisandra Manzoni Fontoura
*Computer Science Post Graduate Program, Federal University of Santa Maria (UFSM),
Avenida Roraima, n°1000, Santa Maria, RS, Brazil*

Keywords: Software Process Development, Agile Methods, Plan-driven Methodology, Hybrid Methodology.

Abstract: Software processes are essential for software development organizations to deliver quality software. There are currently several software processes to meet different needs. However, it is difficult to find in the literature software processes focused on university projects involving other institutions, such as government and industry. This article aims to conduct a systematic literature review to identify the characteristics and limitations of agile and plan-oriented methodologies, which processes were used in software development projects and to establish a relationship between organizational characteristics and best methodologies successful. As a research method, we conducted a systematic study of the literature associated with a snowball strategy, identified and structured the literature on the use of agile and plan-oriented methodologies. We selected 12 studies using the systematic review and added 5 more using the snowball method, totaling 17 selected articles. We note that there is no specific methodology to be used in software development, each organization has its characteristics. The lack of specific processes for university projects is evident, and the differences between this environment and industry require processes tailored. Beside, a large number of projects use practices of more than one method, called hybrid methodologies, to exploit the best of agile and plan-oriented methodologies.

1 INTRODUCTION

In the late 1970s, the software crisis occurred, leading to a change in the way software was developed, requiring companies to conduct their projects in a standardized way (Pressman, 2011). Since then, various software development methods have emerged, and since the mid-1990s, various processes have been proposed.

In general, software processes can be grouped into two different approaches according to their characteristics, which are: plan-driven (or traditional) and agile methods (Silva, Oliveira, Canedo & Martins 2016). Agile methods focus on people rather than processes, communication, and interaction between people. Less documentation is created, and code is seen as part of the documentation. On the other hand, plan-driven methodologies focus on predictability and detailed planning. To define the process to be used in each project, it is necessary to consider the characteristics of each methodology and the application context.

University projects are different from government and industry projects (Abbas, Avdic, Xiaobao, Hasan, & Ming, 2018). Identify the characteristics of each project, such as developer profile, work time, criticality, team size; It is relevant to define the most appropriate processes for each development (Silva & Melo, 2016).

Besides, today projects involving more than one of these organizations are common, following the Triple Helix model, especially in projects involving science, research, and innovation. This model focuses on university-industry-government relations as a strategy to encourage the dynamics of innovation (Mineiro, Souza, Vieira, Castro & Brito 2019).

Therefore, rethinking and redesigning the software development process is an essential task to address different application contexts, executing it more effectively, thereby reducing software development costs and increasing end-product quality and customer satisfaction (Canedo, Martins, Oliveira & Silva, 2016).

In this sense, it is essential to analyze the ideal characteristics of the use of each of the process approaches, the characteristics of projects developed in universities, government, and industry to establish a relationship between the characteristics of process approaches versus the characteristics of project/organization.

Therefore, the purpose of this paper is to investigate the characteristics and limitations of using agile and planned methods and which processes are used in software projects involving at least one of these organizations (university-business-government).

To this end, a systematic literature review and snowballing technique were applied as a research methodology to identify studies related to the area that meet the research objectives.

The paper was organized as follows: Section 2 describes a background with concepts relevant to the understanding of the work. Section 3 describes the research methodology, and section 4 describes the final considerations and future work.

2 BACKGROUND

The term Triple Helix was coined by Henry Etzkowitz in the 1990s to describe the model of innovation based on the Government-University-Enterprise relationship (Etzkowitz, 1994). The authors Etzkowitz and Zhou (2017) define that Triple Helix as a model of innovation in which the university/academy, industry, and government interact to promote development through innovation and entrepreneurship.

This representation seeks the production of new knowledge, technological innovation, and economic development through dynamic processes of experiences in the relationships between science, technology, research, and development, in a spiral of endless transitions (Mikosz, 2017). Based on this assumption, it is analyzed that university-industry-government interactions, which form a “triple helix” of innovation, are critical points for knowledge-based economic growth and social development (Etzkowitz & Zhou, 2017).

Given the scenario described, using methodologies in the implementation of projects that meet the diversity of this market with precision and without fail is paramount (Silva & Melo, 2016). Cereci and Karakaya (2018), define that software development methodologies are used to organize and monitor the software process.

Currently, it is considered that there are two main approaches related to software processes, which are: plan-driven methodologies and agile methods (Fowler, 2005).

The basic idea of the plan-driven methodology is that projects are relatively simple, predictable, and linear with clearly defined boundaries, which facilitates detailed planning and follow-up without significant changes (Boehm & Turner, 2003). Plan-driven methodologies focus on planning, and work begins by collecting and documenting a complete set of requirements, followed by the development and inspection of high-level architectural designs (Mushashu & Mtebe, 2019).

Agile methods, according to Koskella (2003), are related to rapid software development focusing on less time in analysis and design, thus being an iterative and incremental approach. According to Pressman (2011), the agile methodology employs simplicity in development, focused on customer satisfaction, with small and motivated teams.

Mamoghli and Cassivi (2019) describe that the implementation of agile practices allows gaining experience with software throughout the project. In this development, communication between developers and clients is prioritized, giving preference to delivery over project analysis.

However, according to Comfort and Amaral (2016), due to an innovation environment, the form of software development has changed, implying new strategies and techniques to combine simplicity, speed, and flexibility.

According to Kruchten (2011), due to routine and constant changes in software design requirements, the team should use a dynamic and adaptive software process. Software projects are affected by various factors such as cost, scope, quality, and success criteria vary from project to project. Adaptive processes can facilitate these changes.

Hybrid methodologies, that is, combining practices from agile and traditional methods, have also been used to deal with the dynamism of specific projects, increase productivity, and improve the quality of the final product. Hybrid methodologies can be characterized by the combined adoption of principles, practices, techniques, and tools from different methodologies to adapt management to the context of software design and provide the balance between flexibility and predictability, reducing risk and increasing innovation (Conforto, Amaral, Silva, & Rebentisch, 2015).

In general, to define the most appropriate processes for developing a project, it is necessary to

consider the project objectives, the development team, and the software business domain. Silva et al. (2015) state that traditional and agile methods have their advantages and disadvantages so the choice of one depends on the context of the application.

Given this Silva and Melo (2016) highlight that it is necessary to have a clear understanding of the characteristics of the project, of the structure and strategy of the company when defining the process. These understandings will support the definition of the best and most appropriate practices of each methodology to be associated with the achievement of the project objective (Silva & Melo, 2016).

3 RESEARCH METHODOLOGY

A systematic literature review aims to provide a fair assessment of a research topic using a reliable, rigorous, and auditable review methodology (Kitchenham, 2004). This review evaluates and interprets the relevant literature available to understand and analyze the development methodologies used in projects in universities, companies, and governments. This study followed the guidelines proposed by (Kitchenham and Charters, 2007), which involve three main phases: planning, conducting, and reporting the review results.

After SLR, we performed a manual search using the snowballing guidelines proposed by Wohlin (2014), to perform an analysis of primary studies and obtain secondary studies in the face of the research proposal.

3.1 Planning the Review

This step aims to define the search plan, determining the issues of interest, data sources, strategy and search terms, inclusion and exclusion criteria, detailed below.

The quality of the systematic review is related to the proper definition of the research questions. The questions guide the review, so they are elaborated to meet the proposed objectives (Kitchenham, 2004). For this research, we defined the following questions:

RQ1: What are the characteristics and limitations of plan-driven and agile methodology?

RQ2: What are the characteristics of each organization (university, government, and industry)?

RQ3: What methodologies are used in projects developed at universities, government, and industry?

RQ4: What works describes the use of hybrid development methodologies?

For the analysis and selection of primary studies, we searched the following sources: IEEE, ACM, and Google Scholar. To perform the search, we define the search strings, in Portuguese and English, using the logical operators “AND” and “OR”; so the search string used was:

(“software development” AND “process” AND (university OR company OR industry OR govern) AND (“agile method” OR “planned methodology” OR hybrid))

We have defined a set of precise selection criteria (inclusion and exclusion) to select the most relevant articles for systematic review, ie, those that help answer the survey questions. Studies were selected according to the following criteria:

1. Articles describing software processes adopted in software development projects involving universities/academia, government and /or industry;
2. Case studies involving the adoption of software processes by organizations;
3. Articles published between 2013 and 2019;
4. Works wrote in Portuguese or English;

The exclusion criteria defined were:

1. Articles outside the scope of the research;
2. Articles other than Portuguese or English;
3. Published outside the defined date range;
4. Availability of title and abstract only;
5. Full document access unavailable;
6. Duplicate works.

3.2 Conducting the Review

According to the research protocol explained earlier, we searched the databases and retrieved the relevant studies. Using search strings, we analyze the title and summary of each search and apply the exclusion criteria. This first round resulted in 24 candidate articles.

In the second iteration, we read the articles in their whole to make sure the papers cover the scope of the research. In this analysis, 3 articles were excluded because they were duplicate, and 6 articles did not cover the scope of the search. Also, 2 articles were not found full text, which was deleted, resulting in 12 articles selected.

The snowballing technique was used (Wohlin, 2014), which refers to the use of reference lists of selected articles or citations of these articles, that is,

it allows the search of works from references of selected articles through a systematic review. It is possible to refine the searches with the articles being cited, thus allowing us to find new articles. The application of Snowballing allowed the selection of 5 new works, thus totaling 16 works in all.

After selecting the articles, data were extracted, considering the questions defined in the research. Besides, we extracted information such as year of publication, authors, proposal of each study and source of publication, and the research method used. Table 1 lists the resulting works.

After data extraction, a synthesis of such data was performed to facilitate data analysis. The synthesis lists the characteristics of each methodology, as well as its limitations and characteristics. Based on the similarity of the extracted data, a categorization was performed for each research question.

3.3 Reporting the Results

This section summarizes the search results. The analysis of the information is presented, considering each research question.

RQ1: What are the characteristics and limitations of plan-driven and agile methodology?

The authors Jusoh, Gorment, Nor and Muhamad (2017) [E08] describe that agile methodology is widely used, especially in the education sector, because it contributes to reducing waste, increasing speed, as well as improving productivity, decision making and confidence.

Already the authors Vijayasathy and Butler (2016), in addition to an online survey, state that the approaches are used in statistics with a small number of employees in low budget and medium or high critical projects, in a single and small team. Meanwhile, traditional approaches are used by organizations with large numbers of employees, high-budget and risky projects, and in medium and multiple teams.

The authors Boehm and Turner (2003) define several characteristics that can be used to choose between these two approaches, which are: as application characteristics (size and environment); management resources, including customer relations, planning and control; technical characteristics, including requirements, development and testing approaches; and personal resources, including customer, developer, and organizational culture particularities. In Table 2, the characteristics and limitations of agile and plan-driven methodology are summarized.

RQ2: What are the characteristics of each organization (university, government, and industry)?

In the university, authors Cereci and Karakaya (2018) [E01] report that meetings occur less frequently because people work part-time on projects, reconciling them with academic activities. Also, there are not always exist end users to perform software testing.

Brondani, Mello, Fontoura (2019) [E08] state that university teams are composed of workers with different skill levels, including undergraduate, graduate, and highly experienced researchers. These teams, in addition to focusing on development activities, need to investigate highly complex research solutions. Another characteristic pointed out by the authors is the high turnover of the team.. As they develop projects in partnership with different organizations, the team is unfamiliar with the business domain, making it challenging to define software requirements.

According to Dias, Kodikara, and Jayawardena (2013) [E02], academic institutions focus on learning, research, and innovation, as the result of research projects intended to works publications.

Already in organizational contexts, Cereci and Karakaya (2018) [E01], who report that work is done full time, software development is for-profit and limited budget product delivery (Dias, Kodikara & Jayawardena, 2014) [E02]. Kuhrmann et al. (2019) [E05] declare that company policy must be strictly followed, and this often limits the choice of the most appropriate process.

When referring to the government, it is possible to verify aspects related to government laws, and the need for transparency in data and control of internal organs related to the institution (Dos Santos and Canedo, 2014) [E06].

In the case studies by the authors, Benedicenti et al. (2016) [E07], describe how army software development should be of high quality and have specific and complex requirements.

Cotugno and Messina (2014) [E17] describe that, in military environments, system reliability is essential. The criticality of systems is usually high. Planning costs must be within budget, and there is a need to follow software development standards. Table 3 shows the characteristics of each institution.

RQ3: What methodologies are used in projects developed at universities, government, and industry?

Table 3 describes the methods proposed by the works selected by the organization in which they were applied.

Table 1: All the papers included in the SLR.

Id	Authors	Year	Title
E01	Cereci and Karakaya	2018	Need for Software Development Methodology for Research Based Software Projects
E02	Dias et al.	2013	The Need for Novel Development for Software Projects in Universities: A Sri Lankan Case Study
E03	Dias, Kodikara, and Ekanayaka	2014	Differences between universities and industry in software development
E04	Turke, France, and Rumpe	2014	Limitations of Agile Software Processes
E05	Kuhrmann et al.	2019	Hybrid Software Development Approaches in Practice: A European Perspective
E06	Dos Santos, and Canedo	2014	Development Methodology Case Study: Brazilian Electoral Justice
E07	Benedicenti et al.	2016	Applying Scrum to the Army - A Case Study
E08	Brondani et al.	2017	A Case Study of a Software Development Process Model for SIS-ASTROS
E09	Vijayasarathy and Butler	2016	Choice of Software Development Methodologies Do Organizational, Project and Team Characteristics Matter?
E10	Mushashu and Mtebe	2019	Investigating Software Development Methodologies and Practices in Software Industry in Tanzania
E11	Awad	2005	A Comparison between Agile and Traditional Software Development Methodologies
E12	Fitriani, Rahayue and Sensuse	2016	Challenges in Agile Software Development: A Systematic Literature Review
E13	Spundak	2013	Mixed agile/traditional project management methodology reality or illusion?
E14	Riesener, Dölle, and Ays	2018	Hybridization of Development Projects Through Process-related Combination of Agile and Plan-Driven Approaches
E15	Klunder, Hohl, Fazal-Baqaie, Krusche, Küpper, Linssen & Prause	2017	HELENA study: Reasons for combining agile and traditional software development approaches in german companies
E16	Marinho, Noll Richardson and Beecham (2019)	2019	Plan-Driven Approaches Are Alive and Kicking in Agile Global Software Development
E17	Cotugno and Messina	2014	Adapting SCRUM to the Italian Army: Methods and (Open) Tools

Table 2: Characteristics and limitations of plan-driven and agile methodologies.

	Characteristics	Limitations
Agile methodology	<p>Turke, France and Rumpe (2014) [E04]; Fitriani, Rahayue and Sensuse (2016) [E12]:</p> <ul style="list-style-type: none"> • Customer communication; • Software requirements evolve as software is developed; • Change-adapted software development process; • Project visibility can be achieved mainly through the delivery of increments and some metrics; • Software can be developed in increments. <p>Awad (2005) [E11]:</p> <ul style="list-style-type: none"> • Customer collaboration. <p>Dos Santos and Canedo (2014) [EO6]:</p> <ul style="list-style-type: none"> • Well defined roles; • Anticipated deliveries of key system features; • Simplicity; • Deliver frequently running software. 	<p>Turke, France and Rumpe (2014) [E04]:</p> <ul style="list-style-type: none"> • Distributed development environment; • Development involving several teams; • Limited to software with high criticality; • Complex software development. <p>Awad (2005) [E11]:</p> <ul style="list-style-type: none"> • Large scale software. <p>Mushashu and Mtebe(2019) [E10]:</p> <ul style="list-style-type: none"> • Lack of documentation; • Insufficient software design. <p>Dos Santos and Canedo (2014) [EO6]:</p> <ul style="list-style-type: none"> • Requires team members greater skill and responsibility.
Plan-driven methodology	<p>Spundak (2013) [E13]:</p> <ul style="list-style-type: none"> • Formal documentation; • Used in critical projects; • Work control; • Projects with high criticality; • Focus on verification and validation; • Used in larger projects. 	<p>Spundak (2013) [E13]:</p> <ul style="list-style-type: none"> • Robustness; • Project isolated from its environment; • Time constraints; • Uncertainty in setting goals.

Table 3: Characteristics of each institution and methods used.

	Characteristics	Methods Used
University	<p>Cereci e Karakaya (2018) [E01]:</p> <ul style="list-style-type: none"> • Meetings less frequently; • Lack of end users; • Turnover of people in the project; • Limited time to work on a project (undergraduate and graduate students). <p>Dias, Kodikara e Jayawardena (2013) [E02]:</p> <ul style="list-style-type: none"> • Learning and research; • Innovation; • Publications of works; • Flexibility. 	<p>Cereci e Karakaya (2018) [E01]:</p> <ul style="list-style-type: none"> • Methodologies similar to prototyping; • Experience-based software development process.
Company	<p>Cereci e Karakaya (2018) [E01]:</p> <ul style="list-style-type: none"> • Work full time; • To profit; • Limited budget; • Product delivery <p>Dias, Kodikara, e Ekanayaka, (2014)[E03]; Kuhrmann et al. (2019)[E05]:</p> <ul style="list-style-type: none"> • Company Politics. 	<p>Dias, Kodikara, e Ekanayaka, (2014) [E04]</p> <ul style="list-style-type: none"> • Scrum. <p>Kuhrmann et al. (2019) [E05]:</p> <ul style="list-style-type: none"> • Scrum and waterfall model; • Kanban , Scrum and Unified Agile Process. <p>Vijayarathy e Butler (2016) [E09]; Klunder et al. (2017)[E15]; Marinho et al. (2019) [E16]:</p> <ul style="list-style-type: none"> • Hybrid Methodology. <p>Mushashu e Mtebe (2019) [E10]:</p> <ul style="list-style-type: none"> • Traditional methodology; • Cascade model and prototyping
Government	<p>Dias, Kodikara e Jayawardena (2013) [E02]:</p> <ul style="list-style-type: none"> • Government laws; • Data transparency. <p>Dos Santos e Canedo (2014) [E06]:</p> <ul style="list-style-type: none"> • Internal control organs. 	<p>Dos Santos e Canedo (2014) [E06]:</p> <ul style="list-style-type: none"> • Scrum.

Given this information, it was found that in the university environment, according to the study by authors Cereci and Karakaya (2018) [E01], the methodologies are similar to the prototyping and software development process is based on experiences from other projects. According to Dias et al. [03] describes that projects carried out at a university use whiteboard to draw diagrams and concept maps to illustrate their ideas.

According to Dias, Kodikara, and Ekanayaka (2013) [E02], there is no standard method to be used in the university context, which focuses on innovation, research, and learning.

Brondani et al. (2019) [E08] point out that in the development of a project for the government, carried out at a university, a hybrid methodology was used. Practices related to plan-driven methodologies were used for the activities of analysis, design, and verification activities, supporting the development of documents related to each phase of the project, as the condition of the project contract required formal deliverables. Agile methodologies were used to manage implementation and testing activities, emphasizing short development cycles, team, and end-user communication and collaboration principles, enabling incremental deliveries, and change requests without affecting the project.

Regarding industries, Kuhrmann et al., (2019) [E05] state that traditional and agile methodologies are used, with a more significant predominance of the cascade and scrum method of structure, as well as Kanban.

In a research conducted by the authors Dias et al. (2014) [E03], they observed that the industries that participated in the interview, use the Scrum software development standard, emphasizing on-going user

engagement to understand requirements and improve as feedback

In contrast, in a study by the authors Klunder et al. (2017) [E15] and Marinho et al. (2019) [E16], they report that most projects use a combination of agile and traditional approaches called the hybrid methodology.

Authors Mushashu and Mtebe (2019) [E10], describe that the methods chosen by companies are: plan-driven methodologies, such as cascade and prototyping method.

In the context related to government, Dos Santos and Canedo (2014) [E06] report that agile methods are widely used because they are objective, roles are well defined, easy to learn and provide project visibility to the team. The authors describe some aspects of implementing agile government-related project methodologies such as response to changes, delivery of key system functionality, increased customer collaboration in the development process, and the possibility of effort estimation.

RQ4: What works describes the use of hybrid development methodologies?

Theocharis et al. (2015) argue that instead of using only traditional or agile processes individually, various methods should be combined to adapt their development process to a specific context. Thus, it is considered that the combination of agile and plan-driven methods should be used to leverage the strengths of both approaches. The authors describe that agile methods emphasize communication, knowledge sharing, and project visibility. The plan-driven methodology proposes detailed planning, control, and estimation of tasks.

Riesener et al. (2018) [E14] describe that plan-driven development processes are described by attributes such as predictability and stability. On the other hand, agile processes apply to projects with many requirements changes and where the customer is heavily involved in the development process.

Given this perspective, it is suggested to adopt hybrid development approaches, which will allow the benefits of both methodologies, providing secure management and a flexible environment (Kuhrmann et al., 2019) [E05]. Thus, it is necessary to balance the need for planning and control with time and flexibility when adopting different methodologies (Marinho et al. (2019) [E16].

The authors Comfort and Amaral (2016) argue that the adoption of hybrid methodologies proves improvements in process performance, especially issues related to added value, flexibility, and accuracy of project information.

The authors Vijayarathy and Butler (2016) [E09] describe that the choice of a methodology for a project depends on the type of organization, project characteristics, and team. According to the research described by the authors, 45.3% of the projects use hybrid approaches, which are relevant to organizations of different sizes, with medium and high criticality budget and in small teams.

Brondani et al. (2019) [E08] used a hybrid approach in which contract assets (deliverables) were managed using planned approaches, while short development cycles were defined to manage internal deliveries, bringing agility.

4 CONCLUSION

This research discussed the importance of adequate methods for the specifications of each context and project in software development, as well as the appropriate combination of methods proposed by different approaches, leveraging their potentials and consequently improving the quality of the software process. There is no single methodology that can cover the needs of all software projects, as well as different contexts.

Plan-driven and agile methodologies have their advantages and disadvantages, so it is not possible to standardize and claim that one approach is better than another. Some factors influence the choice of software methodologies, such as project type, project size, project duration, level of understanding of user requirements, project complexity, customer involvement, and team size.

It is observed that, by answering the questions raised, there is a lack of studies related to projects developed in universities. This issue is understood to be necessary, especially at a time when the university is no longer having a secondary, though important, social role in providing higher education and research, and is assuming a primary role equivalent to that of industry and government, as a generator of new industries and companies.

As future work, it is proposed to expand the research to deepen the requirements for a process involving university and government and/or industry.

ACKNOWLEDGEMENTS

We gratefully acknowledge financial support from CAPES-Brazil, and the Brazilian Army for the financial support through the SIS-ASTROS Project (813782/2014), developed in the context of the PEE-ASTROS 2020.

REVIEW REFERENCES

- Awad, M. A., 2005. A Comparison between Agile and Traditional Software Development Methodologies. In *2nd National Conference on Applied Research. Computer Science and Information Technology*.
- Benedicenti, L., Cotugno, F., Ciancarini, P., Messina, A., Pedrycz, W., Sillitti, A. & Succi, G., 2016. Applying scrum to the army: a case study. In *IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*, Austin, TX. 725-727. 10.1145/2889160.2892652.
- Brondani, C., Mello, O., & Fontoura, L., 2019. A Case Study of a Software Development Process Model for SIS-ASTROS. 600-605. 10.18293/SEKE2019-098.
- Cereci, I., & Karakaya, Z., 2018. Need for a Software Development Methodology for Research-Based Software Projects. In *3rd International Conference on Computer Science and Engineering (UBMK)*, Sarajevo: pp. 648-651.
- Cotugno, F., & Messina, A., 2014. Adapting SCRUM to the Italian Army: Methods and (Open) Tools. In *IFIP Advances in Information and Communication Technology*. 61-69. 10.1007/978-3-642-55128-4_7.
- Dias, D. M. P., Kodikara, N. D., & Jayawardena, M., 2013. The Need for Novel Development Methodologies for Software Projects in Universities: A Sri Lanka Case Study. In *International Journal of Future Computer and Communication* (January): 494-98.
- Dias, M., Kodikara, N., & Ekanayaka, Y., 2014. Differences between Universities and Industry in Software Development. In *32nd National Information*

- Technology Conference*, Colombo, Sri Lanka (August): 207–11.
- Dos Santos, V., Alves, S., & Canedo, E. D., 2014. Metodologia Ágil No Desenvolvimento de Software: Estudo de Caso: Justiça Eleitoral Do Brasil. In *Iberian Conference on Information Systems and Technologies, CISTI*: 1–6.
- Fitriani, W R, Rahayu, Puji & Sensuse, D. I., 2017. Challenges in Agile Software Development: A Systematic Literature Review. In *International Conference on Advanced Computer Science and Information Systems, ICACSIS 2016*: 155–64.
- Klunder, J., Hohl, P., Fazal-Baqaie, M., Krusche, S., Küpper, S., Linssen, O. & Prause, C., 2017. HELENA Study: Reasons for Combining Agile and Traditional Software Development Approaches. German Companies. 10.1007/978-3-319-69926-4_32.
- Kuhrmann, M. (Ed.). n/a et al., 2019. Hybrid Software Development Approaches in Practice: A European Perspective. In *IEEE Software* 36(4): 20–31.
- Marinho, M., Noll, J., Richardson, Ita, & Beecham, S., 2019. Plan-Driven Approaches Are Alive and Kicking in Agile Global Software Development. In *International Symposium on Empirical Software Engineering and Measurement* Septe: 1–11.
- Mushashu, E. T. & Mtebe, J. S., 2019. Investigating Software Development Methodologies and Practices in Software Industry in Tanzania. In *IST-Africa Week Conference, IST-Africa 2019*: 1–11.
- Riesener, M., C. Dölle, Ays, J., & Ays, J. L., 2019. Hybridization of Development Projects Through Process-Related Combination of Agile and Plan-Driven Approaches. In *IEEE International Conference on Industrial Engineering and Engineering Management*.
- Turk, D., France, R. & Rumpe, B., 2014. Limitations of Agile Software Processes. In *Third International Conference on Extreme Programming and Flexible Processes in Software Engineering, XP2002*, Italy.
- Vijayarathy, L. R., & Butler, C. W., 2016. Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter?. In *IEEE Software* 33(5): 86–94.
- E., 2015. Modelos Híbridos: Unindo Complexidade, Agilidade e Inovação. *Revista Mundo PM*.
- Etzkowitz, H. & Zhou, C., 2017. Hélice Tríplice: inovação e empreendedorismo universidade-indústria-governo. *Estudos Avançados*, 31(90), 23-48. <https://dx.doi.org/10.1590/s0103-40142017.3190003>.
- Etzkowitz, H., 1994. *Academic-industry relations: a sociological paradigm for economic development*. Evolutionary Economics and Chaos Theory: New Directions in Technology Studies, p. 139 – 151.
- Fowler, M., 2005. The New Methodology. Viewed 21 Feb 2020. <https://www.martinfowler.com/articles/newMethodology.html>
- Jusoh, Y. Y., Gorment, R. N. H., Nor, S. A. M. & Muhamad, S., 2017. A study on the current practices of software development in Malaysia. In *3rd International Conference on Science in Information Technology: Theory and Application of IT for Education, Industry and Society in Big Data Era, ICSITech* -Janua: 716–21.
- Kitchenham, B., 2004. Procedures for Performing Systematic Reviews. In: Relatório Técnico, TR/SE-0401 ISSN: 1353-7776.
- Kitchenham, B. & Charters, S., 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.
- Kruchten, P., 2013, Contextualizing agile software development. In *Journal of Software: Evolution and Process* 25 (4): 351-361.
- Mamoghli, S., & Cassivi, L., 2019. Agile ERP Implementation: The Case of a SME. *ICEIS*.
- Mikosz, V. M., 2017. A Relação Universidade-Empresa-Governo no contexto no Sistema Nacional de Ciência, Tecnologia e Inovação Brasileiro: um estudo de caso dos mecanismos de cooperação e seus intervenientes em uma universidade pública. Dissertação (Mestrado em Planejamento e Governança Pública) – Programa de Pós-Graduação em Planejamento e Governança Pública – Universidade Tecnológica Federal do Paraná. Curitiba.
- Mineiro, A., Souza, D. L, Vieira, K. C., Castro, C & Brito, M. J., 2019. Da hélice tríplice a quintupla: uma revisão sistemática. In: *Revista Economia & Gestão*. 18. 77-93. 10.5752/P.1984-6606.2018v18n51p77-93.
- Pressman, R. S., 2011. *Engenharia de Software: uma abordagem profissional*. 7º Ed. Porto Alegre: AMGH.
- Silva, D., Oliveira, E. C., Canedo, E. D., & Martins, H. F., 2016. Application of a hybrid process software requirements management. In *11th Iberian Conference on Information Systems and Technologies (CISTI)*, Las Palmas, doi: 10.1109/CISTI.2016.7521442
- Silva, R. F; Melo, F. C., 2016. Modelos Híbridos de Gestão de Projetos Como Estratégia na Condução de Soluções em Cenários Dinâmicos e Competitivos. *Revista Brasileira de Gestão e Desenvolvimento Regional*, p. 443-457.
- Wohlin, C., 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *ACM International Conference Proceeding Series*. 10.1145.2601268.

REFERENCES

- Abbas, A., Avdic, A., Xiaobao, P., Hasan, M. M., & Ming, W., 2018. University-government collaboration for the generation and commercialization of new knowledge for use in industry. *Journal of Innovation & Knowledge*. doi:10.1016/j.jik.2018.03.002
- Boehm, B., & Turner, R., 2003. *Balancing agility and discipline: A guide for the perplexed*. Boston, MA: Addison Wesley.
- Canedo, E.D., Martins, H, Oliveira, E. C. & Silva, D., 2016. Adapted Software Requirements Management for Public Organization. 1234-1246. 10.5748/9788599693124-13CONTECSI/PS-3832.
- Conforto, E. C., Amaral, D. C., Silva, F. B., & Rebentisch,