Agile and Software Project Management Antipatterns: Clarifying the Partnership

Gabriela Castro Flores, netLogistik

Ana M. Moreno, Universidad Politécnica de Madrid

Lawrence Peters, Universidad Politécnica de Madrid and Iowa State University

// We address the relationship between Agile and software project management (SPM), discussing the extent to which agile practices help to reduce or avoid different SPM antipatterns. Our results may be particularly useful to organizations that are becoming agile. //



Digital Object Identifier 10.1109/MS.2020.3001030 Date of current version: 20 August 2021

PROJECT MANAGEMENT IS recognized as the primary cause of software development failure. A study by IBM¹ found that 53% of project breakdowns were attributable to poor management, while 3% were due to technical challenges. Companies such as Google have recognized the relevance of software management and ensured its success through careful vetting, training people for administrative roles, and having leaders evaluated by the people they supervise.²

Dysfunctional approaches to software project management (SPM) have been discussed for a long time in software engineering literature.³ They have also been synthesized and reported on in the form of SPM antipatterns.^{4,5} Antipatterns formally describe counterproductive approaches to problem solving.⁵ They can be a bad practice, a wrong reaction to a combination of events, a failure to control a project factor, and so on. In any case, they represent problematic scenarios that, if unresolved, will eventually lead to additional breakdowns. The notion of antipatterns is closely related to that of project risks, in the sense that antipatterns are potential hazards to initiatives. Table 1 shows an example of the classic Detailitis Plan SPM antipattern.

The project manager's lack of knowledge has been identified by Silva et al. as the most common cause of SPM malpractices, followed by sloth, pride, and haste.⁶ The authors have also highlighted the impact of such malpractices on customers, team members, and, paradoxically, project managers. Additionally, they discuss the direct impact of antipatterns on SPM activities, including controlling, motivating, and staffing.

The application of agile methods has been recognized as providing

relevant benefits that are directly related to SPM. Murphy et al. state that project managers have the highest understanding of agile benefits compared to other team roles.⁷ Among the traditionally mentioned benefits of Agile are time control, continuous software process improvement and refinement, team satisfaction, cooperation and adaptability skills, and increased productivity.8

In this article, we address the relationship between Agile and SPM, discussing the extent to which the use of agile practices helps to reduce or avoid the occurrence of different SPM antipatterns. We worked with a consolidated list of existing SPM

agile practices. Similarly, we used an amalgamated list of existing SPM antipatterns. We then conducted an in-depth analysis of the impact of such agile practices on the antipatterns. To confirm the applicability of those results, we validated the outcomes with a representative set of Agile researchers and practitioners. We concluded that several agile practices can effectively help to prevent the effects of specific SPM antipatterns. Other agile practices, even if they cannot help to prevent the consequences of the antipattern, can help the project manager identify a misleading situation so he/she can take the corresponding corrective

	ino Botantio Flan antipattorin
Antipattern name	Detailitis Plan
Also known as	Death by planning
Refactored solution name	Rational planning
Refactored solution type	Process
Anecdotal evidence	"We can't get started until we have a complete program plan." "The plan is the only thing that will ensure our success." "As long as we follow the plan and don't diverge from it, we will be successful."
Background	An issue occurs when detailed plans for software projects are taken too seriously.
Short description	Excessive planning for software projects leads to complex schedules that cause downstream problems.
Symptoms and consequences	Symptoms: inability to plan at a pragmatic level; focus on costs rather than delivery; spending more time planning and replanning than on delivering software. Consequences: endless planning and replanning causes further planning and replanning; the objective shifts from delivery of software to delivery of a set of plans; continual delays to software delivery and eventual project failure.
Typical causes	Overzealous continual planning to attempt to enforce absolute control of development. Forced customer compliance. Forced executive management compliance.
Solution	A project plan should primarily show deliverables at two levels: products and components. The deliverable plans should be updated weekly. Tracking is done on the estimated level of completeness.

action. Unexpectedly, we found a few exceptions where a particular agile practice contributed to the negative effects of certain antipatterns.

Our results are useful for organizations that are taking the daunting journey of becoming agile.⁹ For example, such results include the following:

- Encourage organizations to bring to the surface their SPM malpractices and identify the agile approaches that can address mismanagement.
- Use in tradeoff analysis to define action plans containing the agile practices to be adopted according to their potential effects on the most critical antipatterns, among other variables.
- Provide decision-making support for senior management and teams to evaluate the potential benefits of transitioning to Agile.

Agile Practices

A quick Internet search provides several lists of agile practices. One of the most well-known catalogs is reported in "The 12th Annual State of Agile Report."10 It was gathered from organizations in the global software development community. The agile practices described in this report are classified into two groups: agile techniques and engineering practices. In this article, we focus on the agile techniques, as the reported engineering practices generally refer to technical activities (e.g., unit testing and refactoring) that are not directly linked to management issues.

To confirm that the agile techniques in the report¹⁰ accord with the scientific community, we performed a tertiary study of literature published during the past 10 years that collected agile practices. The details are

Table 1. The Detailitis Plan antipattern ⁴

presented by Castro Flores.¹¹ We confirmed that all practices provided by the agile report¹⁰ were included in the scientific sources (although different names were sometimes used) and that there were no extra agile practices included in the scientific literature that were not considered in the report.¹⁰ Table 2 contains all such agile practices with one exception. We included Kanban under the general practice of agile portfolio planning, as, according to the report,¹⁰ Kanban is a specific way to address this task, and both generated similar results in relation to SPM antipatterns.

The validity threats for this study relate to the risks suggested by Petersen et al.¹² Regarding search coverage, we worked with the most well-known scientific databases and defined the search string through several keywords to obtain the most accurate secondary papers within our scope. We might have used the list of agile practices suggested by the report,¹⁰ but we wanted to confirm such a list by using the existing literature. To avoid bias in the study selection, we defined detailed inclusion and exclusion criteria, and the first two authors independently agreed on the final choice of the secondary papers according to these stipulations. Then, the search was complemented with snowball sampling. Finally, to deal with the data extraction and classification threat, a data extraction form was completed by the first author, using the information about the secondary papers, and analyzed and evaluated by the second author to ensure that the selected data were meaningful for our study.

SPM Antipatterns

Our goal is to discuss the effect of agile practices on SPM antipatterns so that this information can be useful

Table 2. The consolidated list of agile practices, ordered by their frequency of use in industry.¹⁰

Agile practice	Description
Daily stand-up	Daily stand-ups are one of the most common practices, followed by agile teams. They are meetings that take place every day, preferably face-to-face, to report updates on the daily work.
Sprint planning	Sprint-planning sessions held at the beginning of the sprint (iteration cycle) are intended to review and analyze possible changes and define the sprint backlog, with the tasks to be completed in the sprint.
Retrospectives	Retrospectives are meetings held after the completion of an iteration to suggest process improvements for the following sprint.
Sprint review	The sprint review is a meeting where the team usually presents the sprint work to the product owner for feedback.
Short iterations	Short iterations typically seek to maintain a consistent sprint length. Their implementation behaves as a support for agile team accuracy while calculating the amount of work that the team would be able to perform.
Release planning	Release planning is performed as an independent session for scheduling every release. Its purpose is to estimate which features will be delivered at the established release deadlines.
Planning poker	Planning poker is shaped by numbered cards in a Fibonacci series. It is used by the agile team as a tool to estimate the value of each task that should be performed.
Available product owner	A product owner is responsible for managing the customer requirements, transforming the prerequisites on backlog items, and prioritizing the stipulations for communication to the agile team.
Single team	Single team is a practice in which, theoretically, every team member is open to take up any work, irrespective of skill sets.
Frequent releases	Frequent releases enable the team to see the state of the system and track whether deadlines are being meet.
Common work area	A common work area supports frequent meetings that lead to informal communication among stakeholders, which aids the constant evolution of the requirements.
Product road mapping	Planning the road map of a product is one of the practices that help to provide a general overview of what is expected from the product and how the system is being constructed based on the features.
Story mapping	Story mapping is a technique used to organize the product backlog by release and functionalities.
Agile portfolio planning	Agile portfolio planning supports the tracking of activities that are in process. It shows the status of each of the tasks that was planned for the sprint.
Agile UX	Agile UX seeks to integrate user experience practices into the agile development cycle. It includes performing corporate design standards and getting feedback from clients to refine the requirements and prototypes.

U.X.: user experience.

Table 3. The consolidated list of SPM antipatterns.⁶

Antipattern name	Description
Absentee Manager	Manager who engages in avoidance behavior or is unavailable for long periods of time
All You Have Is a Hammer	1D management, where the same techniques are used on all subordinates and in all situations
Appointed Team	False assumption that a group of people selected by management will immediately gel and become a team
Detailitis Plan	Excessive planning leading to complex schedules with high level of detail, giving the false perception that the project is fully under control
Dry Water Hole	Specifying stringent skill and knowledge requirements for a job when they are not strictly necessary, resulting in a very limited pool of available talent
Fire Drill	Months of boredom, followed by demands for immediate delivery
Glass-Case Plan	Lack of tracking and updating of initial plans, assuming the plans are adequate
Inflexible Planning	Lack of flexible plans and processes
Irrational Management	Irrational management can be viewed as a skewed set of actions, where the manager's personal priorities guide the software project in illogical directions.
Leader, not Manager	Manager with a vision (leader) but no plan or management methodology
Micromanagement	Excessive involvement in tasks beyond a manager's responsibility
Mushroom Management	Isolating developers from end users, under the mistaken assumption that requirements are stable and well understood by both end users and the software team at project inception
Myopic Delivery	Management insisting on original delivery date even when reducing staff or funding
Process Disintegration	Failing processes due to an underlying decline in overall cooperation and morale
Project Mismanagement	Lack of proper software project monitoring and control
Proletariat Hero	False assumption that coercion is an efficient way to increase productivity
Rising Upstart	Superstars who cannot wait for their time and want to skip learning phases
Road to Nowhere	Lack of planning
Size Isn't Everything	Assuming developers are interchangeable and that the number of people working on a problem is inversely proportional to development time
The Brawl	Project manager with no leadership or management experience
The Domino Effect	Moving critical resources between projects, blurring boundaries
Ultimate Weapon	Relying heavily on a superstar in the team

for organizations that are exploring the possibility of transitioning to agile or that are already in the midst of the process. Therefore, we focus on general SPM antipatterns and not specific agile SPM antipatterns, as the latter refer to malpractices by organizations that already use agile techniques. The study of these agile antipatterns would also be highly relevant, but it is not related to our scope.

In 2015, Silva et al. published an integrated list of SPM antipatterns that were gathered through a tertiary study of the literature.⁶ This work analyzed different lists of SPM antipatterns, resulting in a consolidated inventory of 22 SPM malpractices. To check if there exists a more recent list of SPM antipatterns, we performed another ternary study of scientific literature dealing with SPM antipatterns that was published from 2015 to June 2019. The details can be found in Castro Flores.¹¹ We did not find any secondary study with SPM antipatterns that was published later than 2015. However, we located information about SPM antipatterns in gray literature (sources other than scientific documents, such as books, journals, and conference papers), including "Death by Planning"13 and Basaraner.¹⁴ We confirmed that the set of antipatterns in these sources is a subset of the ones in Silva et al.⁶ However, these publications provide evidence of the community's interest in SPM antipatterns. In sum, the 22 SPM antipatterns identified in Silva et al.⁶ represent the consolidated list of SPM malpractices that we worked with. They are described in Table 3.

Relationship Between Agile Practices and SPM Antipatterns

The effect of each agile practice on the antipatterns (in total, 330

	Daily stand-up	Sprint planning	Retrospectives	Sprint review	Short iterations	Release planning	Planning poker	Available product owner	Single team	Frequent releases	Common working area	Product roadmapping	Story mapping	Agile portfolio planning	Agile UX
Absentee Manager	\checkmark	~	~	~	~	~	~	×	×	~	~	×	~	\checkmark	~
All You Have Is a Hammer	€ €	×	~	×	×	×	~	×	~	B	B	×	×	@	~
Appointed Team	\checkmark	~	~	×	×	~	~	×	~	æ 、	~	×	~	×	~
Detailitis Plan	×	~	€	×	~	~	~	×	~	×	~	~	×	~	~
Dry Water Hole	×	×	×	×	×	×	×	×	A	×	⊕	×	×	×	×
Fire Drill	~	~	ଙ୍କ୍	~	~	~	×	×	~	~	~	×	~	~	~
Glass-Case Plan	~	~	ଙ୍କ	~	~	~	×	~	×	~	~	~	~	~	~
Inflexible Plan	~	~	@	~	~	~	~	~	~	~	~	~	~	×	~
Irrational Management	æ,	~		×	×		~	×	×	B	~		B	×	Q
Leader, not Manager	€ €	~	ଙ୍କ୍	ଙ୍	×	ଙ୍	~	×	×	B	×	ଙ୍କ	B	×	~
Micromanagement	×	ଙ୍କ	ଙ୍କ	×	×	×	~	×	~	×	~	×	×	~	×
Mushroom Management	×	~	ଙ୍କ	~	×	~	~	~	×	~	@	~	~	×	~
Myopic Delivery	~	~	ଞ୍	~	~	~	~	×	A	~	×	~	×	~	~
Process Disintegration	€	~	~	×	ଞ୍	~	~	~	~	~	~	×	~	æ,	~
Project Mismanagement	\checkmark	×	Θ 、	Θ 、	~	~	×	~	×	~	~	~	~	~	\checkmark
Proletariat Hero	×	~	€	×	×	×	~	×	~	×	~	×	×	×	×
Rising Upstart	Θ 、	ଞ୍	ଞ୍	ଞ୍	×	ଞ୍	⊕	×	ଞ୍	×	⊕	×	×	æ,	æ,
Road to Nowhere	B	~	ଙ୍କ୍	×	~	~	~	~	×	~	B	~	~	ଞ୍	~
Size Isn't Everything	×	~	ଙ୍କ	×	ଙ୍କ	×	~	×	×	B	×	~	×	ଞ୍	×
The Brawl	B	ଙ୍କ	ଙ୍କ	ଙ୍କ	×	ଙ୍କ	B	×	×	B	B	×	×	×	ଞ୍
The Domino Effect	×	×	ଙ୍କ	×	ଙ୍କ	ଙ୍କ	×	×		B	×	×	×	⊕	×
Ultimate Weapon	æ,	~	æ,	×	×	æ,	~	×	~	×	æ 、	×	×	A	×

Table 4. The relationship between agile practices and SPM antipatterns.

combinations, with 15 agile practices across 22 antipatterns) is shown in Table 4. The most relevant issues are discussed in the following (the detailed rationale for each of the 330 cases is provided in Castro Flores¹¹). Agile practices might have different effects on antipatterns, and those consequences have been represented with different icons in Table 4.

- Applying an agile practice can contribute to reducing the impact of a particular SPM anti*pattern* (\checkmark): This means that the negative effect of the antipattern can be diminished when applying a particular agile practice or that the application of the agile practice avoids the antipattern. This is the most common situation, as it occurs in 42% of the combinations. One example is the daily meeting/absenteemanager pair. Daily meetings require the presence of all team members, including the project manager. Problems may arise during these daily meetings, requiring actions from the project manager and forcing him or her to make decisions. Other cases where this practice has a positive effect include frequent releases and the fire drill. Frequent releases promote the constant delivery of working, valuable software to the client, avoiding idle times and the postponement of work until late in a project.
- Applying an agile practice may not help to negate an antipattern, but it can help to identify the symptoms of the malpractice: Agile practitioners can be aware of a potential risk and take corresponding corrective actions (Q). Table 3 identifies this situation in 35% of the combinations. An

example of this is sprint planning/ rising upstarts. Rising upstarts are people who imprudently try to promote their professional skills. They attempt to stand out because they are potentially able to perform more activities than others. During sprint-planning sessions, the team estimates and distributes the tasks to be performed to facilitate the ease of identifying people who may try to overload themselves with activities to perform.

- The agile practice does not directly help to reduce the impact of, or help to identify the risk associated with, the antipat*tern* (\mathbf{X}): Roughly 22% of the combinations fall into this category. One example is the sprint review, with antipatterns such as All You Have Is a Hammer, Appointed Team, and Detailitis Plan. The objective of the sprint review is to present the working software developed during the sprint to the product owner or customer; thus, it does not have any direct relationship with the monotonous behavior of the project manager, lack of practices for team cohesion, and possible detailed plans, respectively.
- Implementing an agile practice might aggravate or contribute to the existence of a particular antipattern (A): This situation has been identified with only the single-team agile practice, representing 1% of the combinations. At least in theory, the single team enables each member be able to take any role, irrespective of his or her main skill set. But having a multidisciplinary team does not mean that its members can be moved to other initiatives and activities outside the project,

with the assumption that other members will be able to perform the vacated tasks (The Domino Effect antipattern). Other harmful combinations of this agile practice have been identified with the Myopic Delivery and Dry Water Hole antipatterns, which will be discussed later.

Table 4 was created through detailed analysis and discussion among the authors according to the definition of agile practices and antipatterns provided in Tables 2 and 3, respectively, and from the evidence gathered from our 40-plus years of experience with the software industry, SPM, and consulting with firms ranging from high-technology start-ups and business management companies to multinational corporations. Additionally, we surveyed a total of 30 Agile practitioners through face-to-face interviews to discuss the matching results. The interviewed practitioners came from a wide range of domains (banking, the stock market, transportation, ehealth, and education). They were geographically distributed across North America, South America, and Europe and had a mean number of six years of experience in Agile. More than 80% of them were certified as Scrum masters. We surveyed 30 professionals, as this number is used in empirical studies in software engineering to produce a representative sample.¹⁵

Unstructured interviews were used to discuss the matching, which was distributed to the participants in advance. Later, the face-to-face interviews were set for approximately one hour. During the first round, we interviewed 10 participants, who contributed to consolidate the initial version of the matching, mainly by providing a better rationale for specific combinations. Afterward, this matching was used for the interviews with the rest of participants. The interviews concerned the details of the Agile/SMP antipattern mapping and the potential benefits of the results for the industry.

Regarding the Agile/SPM relationship, the personal experience of each interviewee had a relevant impact of the interpretation of each combination. Even if a consensus was reached, it was clear that, although each agile practice retains its essence, it can be implemented and adapted in different ways by various organizations and even by dissimilar projects within the same organization. Consequently, the matching process might provide slightly different results for various organizations. Regarding specific combinations, six participants explicitly recognized the negative effects of the single team and the domino effect inside their own companies.

In regards to the potential benefits, the interviewees highlighted the idea that the matching could indicate potential SPM issues and explicitly facilitate discussions about how Agile can contribute to address the problems. More than half the participants emphasized the need for commitment and support from management during the transition process and how these results could contribute in that direction.

Figure 1 quantifies the effect of each agile practice on the antipatterns. Sprint planning followed by planning

poker and the Agile user experience (UX) are the agile practices that contribute to address the highest number of antipatterns (15, 13, and 13, respectively). Among them, the Fire Drill, Mushroom Management, and Myopic Delivery have special negative implications for the customer and therefore should be carefully considered.⁶ Other antipatterns that these practices address are the Detailitis Plan, Process Disintegration, and Project Mismanagement, which, according to Silva et al.,⁶ have an important effect on developers and project managers. Therefore, previous agile practices might be good candidates for early incorporation into an Agile transition process. However, the specific impact, including the possible collateral harm, of



FIGURE 1. The number of antipatterns addressed by Agile practice.

each SPM malpractice in an organization, and even a project, should be carefully analyzed to evaluate the criticality of each mismanagement. This information, along with the implementation cost, should be part of the decision-making process about adopting an agile practice. In this sense, it is worth mentioning the Agile UX, which addresses multiple antipatterns An unexpected result was the possible negative contribution of a specific agile practice, the single team. We have already mentioned the combination of the single team and the domino effect. This practice can also have other negative effects, for example, wrongly assuming that having a team where members can take different roles enables setting a fixed

Sprint planning followed by planning poker and the Agile user experience are the agile practices that contribute to address the highest number of antipatterns.

but, according to the agile report,¹⁰ is the least-used agile practice. Its implementation is costly and has important implications for the agile process itself as well as for the team.

Retrospectives constitute the agile approach that most contribute to identifying SPM malpractices. They are aimed at reviewing the whole development process to search for problems and potential improvements. Consequently, in the case of SPM, retrospectives are the place where problems should be identified. Retrospectives help to reduce the impact of a few antipatterns, Absentee Manager, All You Have Is a Hammer, Appointed Team, and Process Disintegration. So, in general, they have an overall positive effect on SPM malpractices. They can be implemented at a low cost and are good candidates for inclusion in the first phases of an agile transition. In fact, they are one of the most-used agile practices.¹⁰

delivery date (Myopic Delivery) according to the incorrect belief that multidisciplinary units will be able to recover the work. Additionally, the idea that team members need to be able to perform any kind of technical activity might raise the expectations of an organization, leading a company to define very strict requirements and exacerbating the effects of the dry water hole antipattern, resulting in a very limited pool of available talent. Even when the application of this practice has recognized positive effects and is an essential element of the agile philosophy, the case of these combinations represents possible risky situations that should be carefully considered.

In this study we focus on agile practices and their potential effects on SPM malpractices, but the fact that one agile practice can have a smaller impact on solving SPM risk scenarios does not mean that it will contribute less to the success of the project. For example, the available product owner practice has a minimal direct impact on SPM malpractices. However, it is essential from the requirements perspective, and therefore it is critical to a project's success.

Practical Implications and Final Remarks

Project management malpractices need to be brought to the surface, discussed, and addressed. Table 4 can be used to identify agile practices that can be useful to address most relevant SPM problems. But each organization is responsible for determining the negative effects that particular SPM malpractices have on its business value and consequently the associated risks. The decision of whether to incorporate an agile practice in a software development process is a tradeoff between different variables, among them, an approach's ability to address the most harmful and frequent SPM malpractices and its implementation cost. We do not mean that one practice will solve a particular antipattern. Most generally, the implementation of several practices working altogether is needed for this purpose. Again, each organization must monitor the process and take the corresponding corrective actions, if needed.

inally, this article discusses the link between Agile and SPM malpractices, providing organizations with a preliminary mapping between the two, which might need to be adapted according to each company's specifics. This information can provide decision support in an agile transition process. However, the adoption of agile practices will not free teams of SPM problems. Other problematic scenarios that are related to the



GABRIELA CASTRO FLORES is a project administrator at netLogistik. Her research includes testing and software usability for agile schemes. Castro Flores received her M.Sc. in software engineering from Universidad Politécnica de Madrid. Contact her at gabriela.castro@netlogistik.com.



ANA M. MORENO is a full professor of software engineering at Universidad Politécnica de Madrid, Madrid, Spain. Her research interests include agile methods and process improvement. Moreno received her Ph.D. in computer science from Universidad Politécnica de Madrid in 1997. Contact her at ammoreno@fi.upm.es.



LAWRENCE PETERS received his Ph.D. in engineering management in 2002 from California Coast University. His research interests include the incorporation of proven management methods into the practice of software project management and raising awareness regarding the critical role of the software project manager in the success of software projects. He has published books and papers on the practice of software project management. Contact him at ljpeters42@gmail.com.

appearance of specific agile SPM antipatterns might arise and need to be considered. \mathfrak{P}

Acknowledgments

The authors extend special thanks to the practitioners that participated in the interviews.

References

- J. Gulla, "Seven reasons IT projects fail," IBM Systems Media, Feb. 1, 2012. [Online]. Available: https:// ibmsystemsmag.com/Power-Systems/ 2/2012/7-reasons-it-projects-fail
- D. Garvin, "How Google sold its engineers on management," *Harv. Bus. Rev.*, Dec. 2013. [Online]. Available: https://hbr.org/2013/12/how-google -sold-its-engineers-on-management

- F. Brooks, The Mythical Man-Month: Essays on Software Engineering. Reading, MA: Addison-Wesley, 1979.
- W. J. Brown, H. W. McCormick, and S. W. Thomas, *AntiPatterns in Project Management*. New York: Wiley, 2000.
- I. Stamelos, "Software project management antipatterns," *J. Syst. Softw.*, vol. 83, no. 1, pp. 52–59, Jan. 2010. doi: 10.1016/j.jss.2009.09.016.
- P. Silva, A. M. Moreno, and L. Peters, "Software project management: Learning from our mistakes," *IEEE Softw.*, vol. 32, no. 3, pp. 40–43, May–June 2015. doi: 10.1109/MS.2015.71.
- B. Murphy, C. Bird, T. Zimmermann, and L. Williams, "Have agile techniques been the silver bullet for software development at Microsoft?" in

Proc. 7th Int. Symp. Empirical Software Engineering and Measurement, 2013. doi: 10.1109/ESEM.2013.21.

- K. Petersen and C. Wohlin, "A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case," *J. Syst. Softw.*, vol. 82, no. 9, pp. 1479–1490, Sept. 2009. doi: 10.1016/j.jss.2009.03.036.
- R. Hoda and J. Noble, "Becoming agile: A grounded theory of agile," in Proc. 2017 IEEE/ACM Int. Conf. Software Engineering (ICSE). doi: 10.1109/ICSE.2017.21.
- VersionOne, "The 12th annual state of agile report," CollabNet. Alpharetta, GA, 2018. [Online]. Available: https://stateofagile.com/#ufh-i -613553652-12th-annual-state-of -agile-report/7027494
- G. Castro Flores, "Agile practices as solutions for software project management anti-patterns," M.S. thesis, Dept. Comput. Languages, Systems, Softw. Eng., Universidad Politécnica de Madrid, Madrid, Spain, 2019.
- 12. K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Inf. Softw. Technol.*, vol. 64, pp. 1–18, Aug. 2015. doi: 10.1016/ j.infsof.2015.03.007.
- SourceMaking, "Death by planning." Accessed on: June 10, 2020. [Online]. Available: https://sourcemaking.com/ antipatterns/death-by-planning
- 14. C. Basaraner, "5 common antipatterns in software project management," DZone, June 14, 2012. [Online]. Available: https://dzone.com/articles/ 5-common-antipatterns-software
- N. Juristo and A. M. Moreno, Basics of Software Engineering Experimentation, 2nd ed. New York: Springer-Verlag, 2011.