

# Exploring the Prevalence of Anti-patterns in the Application of Scrum in Software Development Organizations

Michał R. Wróbel\*, Dorota Przała\* and Paweł Weichbroth\*

\*Faculty of Electronics, Telecommunications and Informatics,  
 Gdansk University of Technology, Gabriela Narutowicza 11/12, Gdańsk 80-233, Poland  
 Email: [michal.wrobel@pg.edu.pl](mailto:michal.wrobel@pg.edu.pl), [przala.dorota@gmail.com](mailto:przala.dorota@gmail.com), [pawel.weichbroth@pg.edu.pl](mailto:pawel.weichbroth@pg.edu.pl)

**Abstract**—The paper presents a survey-based study that aimed to determine the prevalence of anti-patterns in the Scrum software development methodology. A total of 35 anti-patterns were selected from the literature review, and 42 respondents working in software development organizations located in Poland indicated whether they had encountered each anti-pattern in their organizations. The study found that “Unfinished Tasks” was the most prevalent anti-pattern, highlighting the importance of proper planning and task management within sprints. Additionally, several other common anti-patterns were identified, including daily scrums being extended beyond the recommended time, user stories not being fully refined, and the sprint goal not being defined at the sprint planning meeting. The findings of this study provide valuable insights into the current state of Scrum methodology in software development organizations and highlight areas where there is room for improvement.

## I. INTRODUCTION

**W**E are agile! – many organizations proudly proclaim. However, as practice shows, it is always easy to say, but much more harder to accomplish [1]. Since the publication of the Agile Manifesto in 2001 [2], the number of development projects conducted according to the agile approach has been steadily increasing. Nowadays, this approach is also increasingly used outside the domain of software development [3]. However, implementing an agile approach requires more than putting the right processes in place. It requires a change in mindset and attitude toward managing projects.

Scrum has gained significant traction over the years due to its flexibility, iterative approach, and focus on cross-functional team collaboration. It is the most popular software development methodology of all available agile approaches [4], with 9 out of 10 respondents in the State of Agile Report claiming to use it [5]. However, as with any methodology, there are always risks of anti-patterns – common mistakes or misapplications of the methodology that can hinder its effectiveness. A Scrum anti-pattern is defined as a harmful practice within the Scrum framework that may appear convenient at first but proves detrimental in the long run [6].

The aim of this study was to investigate the current state of Scrum implementation in software development organizations located in Poland. For this purpose, a survey was conducted among practitioners to determine the prevalence of Scrum anti-patterns, which are common mistakes or pitfalls in the appli-

cation of a methodology. They can lead to poor performance, low productivity, and other negative outcomes.

## II. RELATED WORK

Based on the interview conducted among 18 respondents, representing 11 IT organizations located in Finland, Eloranta et al. [6] identified ways of potentially harmful mishandling of Scrum. The findings show that a) sprints were too long since the teams could not respond to customer requests and the feedback loop became too long, b) the system testing was performed in the next sprint by a separate team, as well as testing took place at the end of the sprint; in addition, in some cases there was the lack of automation tools employed in the testing, and c) the progress of the work was not made visible with burn-down charts to the teams, thus a phenomenon, termed as “invisible progress” occurred.

Matthies et al. [7] discussed experiences from a classroom project in which a group of 38 students was responsible for developing a single system by using a scaled version of Scrum. In conclusion, the authors argue that the combination of tutor observations, surveys, along with initial testing of automated process analysis leads to better understanding the Scrum adoption, as well as detecting agile practices violations for every team in every sprint, including prolonged and varied duration of sprint, moving testing to the next sprint, and invisible progress.

On a basis of a grounded theory approach, Carew and Glynn [8] investigated how productivity, effectiveness and workflow priorities were influenced by adopting the Scrum. In general, a number of agile anti-patterns were recognized with a negative impact on these three facets, including decision-making incapacity, incomplete deliverables (working code), ad-hoc work requests, and an inability to actually define when work items were completed to the required “Definition of Done”, just to name a few.

Through the observation of the two teams, including 14 members who in majority were software developers, as well as acting as a product owner, agile coach, or scrum master, Mortada et al. [9] investigated the cases of four Scrum activities separately practiced among these teams. In total, 13 deviations from Scrum guidelines were identified, including:

- four related to the daily scrum event, namely: (1) not all key questions are addressed, (2) not all team members contribute, (3) daily scrum events take longer than 15 minutes, and (4) no fixed time for the daily scrum event;
- seven related to the sprint planning, namely: (5) stories are not refined in the product backlog, (6) no calculation of resources available for the upcoming sprint, (7) sprint goal is defined at the end of the planning meeting, (8) no break down of large stories, (9) no agenda used for the planning meeting, (10) stories are not estimated, and (11) stories not formulated completely;
- two related to the sprint demonstration, namely: (12) sprint does not end with a demonstration, and (13) demonstration to the wrong audience.

Moreover, these agile anti-patterns were also reported to have negative impact on the product quality and team morale.

Based on the interviews conducted with software professionals working in Scrum teams, Çetin and Durdu [10] aimed to explore how the Scrum was practically implemented, and in particular uncover what were the differences between these implementations and the original Scrum model. Their findings show that in some organizations daily Scrum meetings, sprint retrospectives, as well as the burndown charts, were not always implemented.

Having collected data from approximately 40 different software development teams of a large Scrum organization, Heikkila et al. [11] investigated how the requirements were planned and managed, how well the requirements planning and management practices matched Scrum guidelines, and whether the changes were perceived harmful. The findings show that only 30% of user stories were set in progress, while 32% of user stories were closed during the sprint planning day and the sprint review day, respectively. The respondents indicated two reasons why user stories last for multiple sprints, namely the inter-team dependencies and the dependencies to third parties, and the difficulty of splitting user stories into pieces that can be implemented in a single sprint. In conclusion, the authors argue that these two deviations from the Scrum user story management and planning process cannot be categorized as harmful.

McKenzie et al. [12] conducted interviews with eight New Zealand video game development studios with the aim to empirically determine how and why agile frameworks were applied. In this extent, the authors recognized several limitations due to the common misunderstanding in key areas around project management and collaboration, demonstrated by missing retrospectives, and ambiguity related to the status of tasks.

Last but not least, Perry [13] discussed the issue of misunderstanding of the end user role with the ultimate customer. The author concludes that “by focusing on the intermediary rather than on the end user that we do the people who have to work with the system a disservice”.

TABLE I  
ANTI-PATTERNS SELECTED AFTER LITERATURE REVIEW

Anit-pattern	Source
Sprints that are too long	[6], [7]
Variable lengths of sprints	[7]
Unfinished tasks	[8]
Testing in next sprint	[6], [7]
Invisible progress	[6], [7]
Not everyone actively participate in the meeting	[9]
Daily Scrum lasts longer than 15 minutes	[9]
The Daily Scrum does not take place at a fixed time of day	[9]
Not all key issues are addressed	[9]
Daily scrums are not held every day	[10]
Disordered product backlog	[7], [6]
Extensive requirements documentation instead of user stories	[7], [6], [9]
User stories that are too extensive	[11]
PO without authority, has negligible influence on the selection of tasks to be implemented	[7], [6]
Stakeholder indecisiveness – requirements come from multiple stakeholders and compete for prioritization	[8]
Lack of unanimity whether the task has been completed	[8], [12]
An indirect customer who is service provider of its own customer	[13]
PO delegated by a client and does not understand the role	[7], [6]
Exact estimation instead of relative estimation of items	[7]
Item estimation imposed to the team	[7], [6]
No estimation of resources available for the upcoming sprint	[9]
Sprint planning meeting has no agenda	[9]
Sprint goal set at the end of the meeting or not at all	[9]
Not breaking large user stories into smaller ones during a meeting	[9], [11]
Disruptions in the development process – other areas of the project are developed in turn without one being completed	[8]
Adding items during sprint	[8], [11], [7], [6]
User stories are not fully refined	[9]
A burndown chart is not used	[10]
Semi functional teams	[7], [6]
Insufficient technical knowledge[8]	
Lack of business knowledge	[8]
No or too long waiting for feedback (lack of Sprint review or stakeholders do not show interest)	[7], [12], [9]
The sprint does not end ends with a demonstration	[9]
Demonstration for the wrong targets	[9]
Missing retrospectives	[7], [12]
Combining the two meetings into one	[10]

### III. SURVEY DESIGN

To prepare the survey, a literature review was conducted to identify common Scrum anti-patterns. For this purpose, articles in the Web of Science, IEEE Explore, and Scopus databases were searched for the keywords *Scrumbut*, *Scrumfall*, or *Scrum anti-patterns*. A total of 81 articles were found, which were then filtered by title, abstract and content. Finally, 8 articles were identified that described common Scrum anti-patterns [6], [7], [8], [9], [10], [11], [12], [13].

Based on this literature review, 35 Scrum anti-patterns were selected for the survey, which are shown in Table I. To

systematize, they were then assigned to one of 7 sections, i.e. Sprint, Daily Scrum, Product Backlog, Product Owner, Sprint Planning, Development Team, and Completing the Sprint. For each of the anti-patterns, survey respondents who indicated that they use or have used Scrum in their development process responded whether and how often they had encountered such a problem in their organizations. A five-point Likert scale was used, along with an additional option of “I don’t know”. The survey was prepared in Polish.

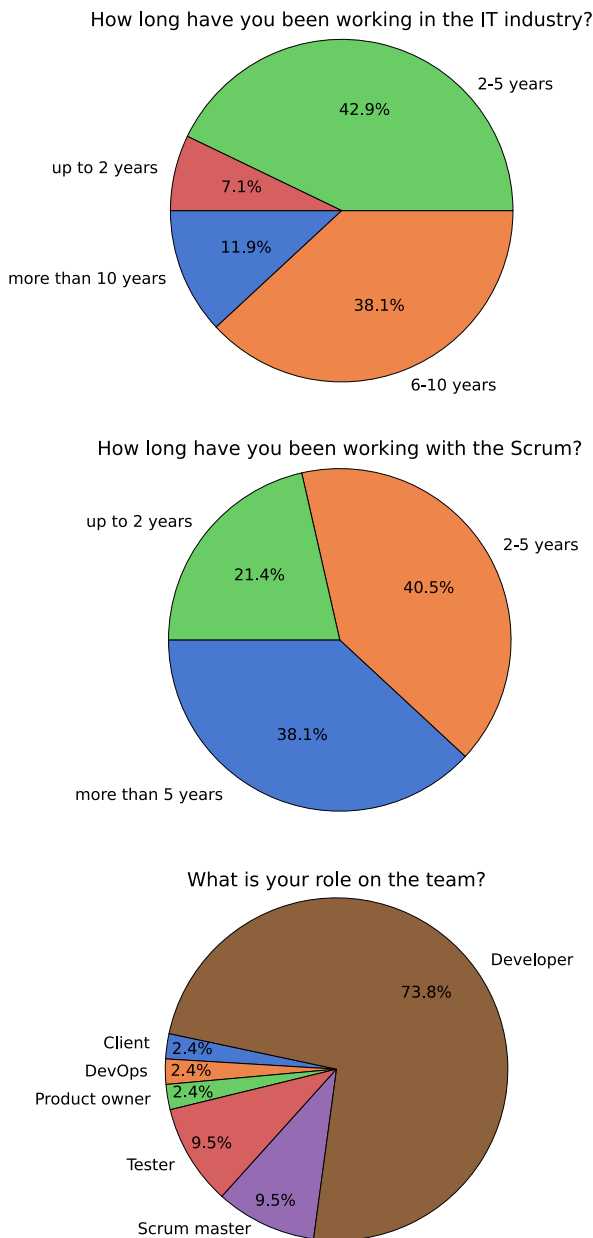


Fig. 1. Experience of survey participants

A. *Sprint*

The Scrum Guide suggests that sprints should last one month or less and that their length should remain constant throughout the project. Testing and all tasks related to both adding new functionality and testing new code should be completed at the end of the sprint. In addition, progress should be tracked using a burndown chart [14]. The identified anti-patterns that belong to this section are presented in Table II.

B. *Daily Scrum*

Daily Scrum should take place every day at a fixed time. Each team member should actively participate in the meeting and answer key questions (What did I do yesterday? What will I do today? What obstacles did I encounter?) The meeting should last no longer than 15 minutes [14]. The anti-patterns that have been identified and fall under this section are presented in Table III.

C. *Product Backlog*

The product backlog should contain items rather than traditional large documentation. The items should be sorted by risk factor and value, and can take the form of user stories that are small enough to be completed in a single sprint [14]. The identified anti-patterns are presented in Table IV.

D. *Product Owner*

The product owner (PO) represents the interests of the stakeholders and manages the product backlog. Any changes to the backlog can only be introduced by the product owner, and the decisions made should be respected by the entire team [14]. The identified anti-patterns that are part of this section are listed in Table V.

E. *Sprint Planning*

Sprint planning is the process of deciding what will be worked on in the sprint. Developers should be involved in the selection of tasks, but the product owner has the final decision. Tasks are broken down into smaller ones as needed [14]. The identified anti-patterns are presented in Table VI.

F. *Development Team*

Programmers in Scrum should form a self-sufficient team with good knowledge of the technical and business knowledge. They should have the competence to both develop and test new versions of software [14]. In this section, 3 anti-patterns were identified and are presented in Table VII.

G. *Completing the Sprint*

Two meetings should be held at the end of the Sprint. A Sprint Review for the customer, where feedback is received and a demonstration is given, and a Retrospective for the Scrum Team, where they discuss what went well and what went wrong during the Sprint [14]. Table VIII presents the anti-patterns associated with this section.

#### IV. RESULTS

The survey was conducted in May and June 2022. The invitation to participate was published on LinkedIn and on social media. A total of 42 people took part in the survey.

At the beginning of the survey, information about the participants' experience in IT and Scrum projects was collected. The results are shown in Figure 1. Most of the respondents have commercial IT experience of more than 5 years, and only three have worked in the industry for less than 2 years. Regarding the work with the Scrum methodology, 16 had more than 5 years of experience, 17 between 2 and 5 years, and only 9 less than 2 years. Developers dominated the survey with 31 respondents. In addition, 4 scrum masters and 4 testers participated in the survey, as well as one product owner, one DevOps, and one customer representative.

##### A. Sprints

The results of the survey regarding sprints are shown in Table II and Figure 2. Of the 5 anti-patterns identified, two were mostly indicated as occurring rarely or never. All respondents indicated that the sprint length is always or mostly in line with the Scrum Guide. In addition, 66% indicated that sprints are always the same length, and only one participant indicated that the opposite is often the case.

In the case of the invisible progress anti-pattern, responses were fairly evenly distributed. Twenty respondents reported

that this never or rarely occurs, and 21 reported that it sometimes, often, or always happens.

The results of the survey, on the other hand, confirmed the presence, the last two anti-patterns. None of the respondents stated that there is never a situation where all the tasks that are supposed to be completed in a sprint are completed. In addition, 66% indicated that always or often not all tasks are completed in a given sprint, and 26% that this occurs sometimes. This led to the identification of another anti-pattern – testing in the next sprint. Only 11 respondents reported that this doesn't or rarely happen, while as many as 19 reported that it often or always is the case.

##### B. Daily Scrum

Table III and Figure 3 shows the results of the daily scrum related anti-patterns. Based on the responses, it can be concluded that daily Scrum in most cases actually take place every day. Only 2 respondents reported that it sometimes happens that a meeting is not held every day, and another two that it is the norm.

For the next two anti-patterns "Not everyone actively participates in the meeting" and "Not all key issues are addressed," the responses indicate that such situations do occur in some organizations. For the latter, only 8 respondents indicated that this never happens.

TABLE II  
SPRINT RELATED ANTI-PATTERNS

Question	I don't now	Never	Rarely	Sometimes	Often	Always
Sprints that are too long	1	33	5	3	0	0
Variable lengths of sprints	1	28	10	2	1	0
Unfinished tasks	0	0	3	11	19	9
Testing in next sprint	3	5	6	9	15	4
Invisible progress	1	5	15	13	6	2

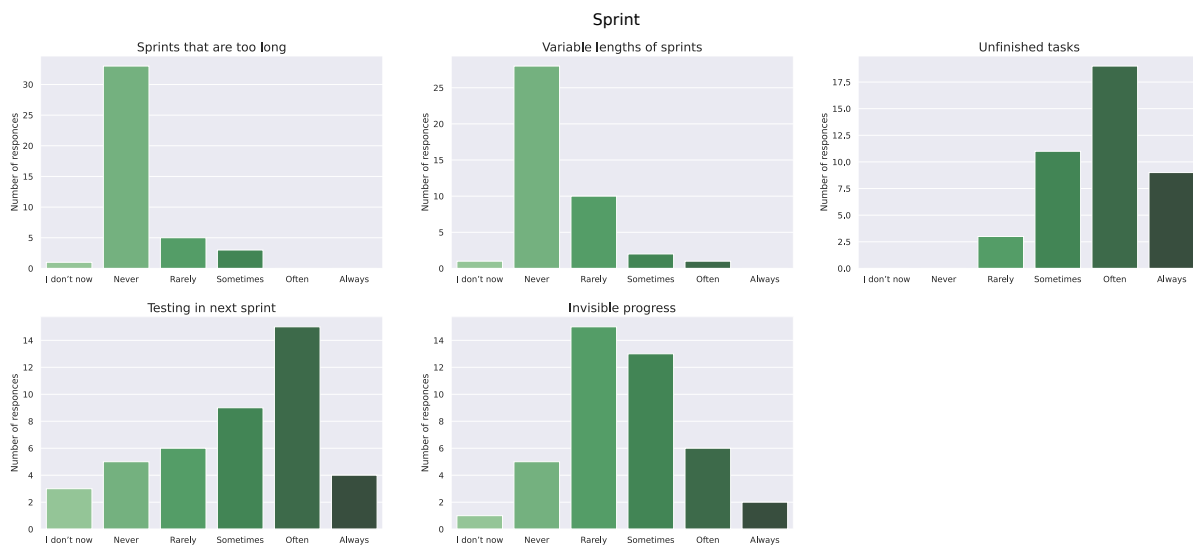


Fig. 2. Sprint related anti-patterns

TABLE III  
DAILY SCRUM RELATED ANTI-PATTERNS

Question	I don't now	Never	Rarely	Sometimes	Often	Always
Not everyone actively participate in the meeting	0	15	9	12	3	3
Daily Scrum lasts longer than 15 minutes	0	1	9	9	16	7
The Daily Scrum does not take place at a fixed time of day	0	33	6	3	0	0
Not all key issues are addressed	1	8	16	13	2	2
Daily scrums are not held every day	1	28	9	2	0	2

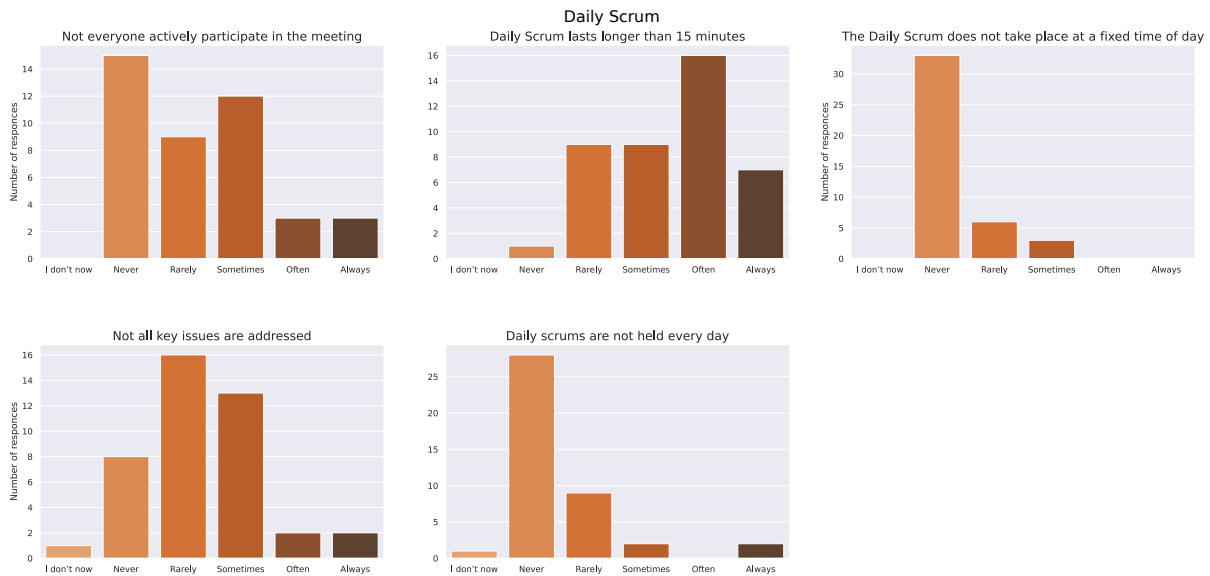


Fig. 3. Daily scrum related anti-patterns

The analysis of the survey results shows that the Daily Scrum goes beyond the recommended 15 minutes in a significant number of cases. Only one respondent reported that it never happens, and as many as 23 that always or often.

C. Product Backlog

The analysis of the responses related to the product backlog showed that the anti-patterns “Extensive requirements documentation instead of user stories” and “User stories that are too extensive” occur rarely in projects. None of the respondents indicated that this situation always occurs, and only 2 and 7, respectively, indicated that it often happens.

On the other hand, in the case of the anti-pattern concerning disordered backlog, more than half of the responses indicated that this situation happens sometimes, often or always. Detailed data with results are shown in Table IV and Figure 4.

D. Product Owner

The analysis of the responses related to the product owner showed that only the anti-pattern “Stakeholder indecisiveness” is the common issue among the respondents’ organizations. It was reported by 32.5% as often or always occurring. Of the remaining anti-patterns, all received more than 63% of responses that they never or rarely occur. The detailed results of the survey are presented in Table V and Figure 5.

It should be noted that this section of questions was characterized by the highest number of “I don’t know” responses. This may indicate that respondents do not fully understand the role of the product owner.

E. Sprint planning

Among the 10 anti-patterns for sprint planning, as many as 2 received 40% or more “Often” or “Always” responses. These are “Sprint goal set at the end of the meeting or not at all”, “User stories are not fully refined” and “A burndown chart is not used”.

Of the remainder, only the “No estimation of resources available for the upcoming sprint” anti-pattern can be reported as relatively rare, with 64% reported for the “Never” and “Rarely” responses. All others are relatively often identified as occurring in respondents’ projects. As can be seen in Table VI and Figure 6, all the others are relatively common in respondents’ projects.

F. Development team

The results of the survey with respect to the development team are shown in Table VII and Figure 7. The anti-pattern “Insufficient technical knowledge” was not indicated by any respondent as occurring always and only by four as sometimes.



TABLE IV  
PRODUCT BACKLOG RELATED ANTI-PATTERNS

Question	I don't now	Never	Rarely	Sometimes	Often	Always
Disordered product backlog	1	6	12	12	7	4
Extensive requirements documentation instead of user stories	2	14	13	10	3	0
User stories that are too extensive	1	11	12	11	7	0

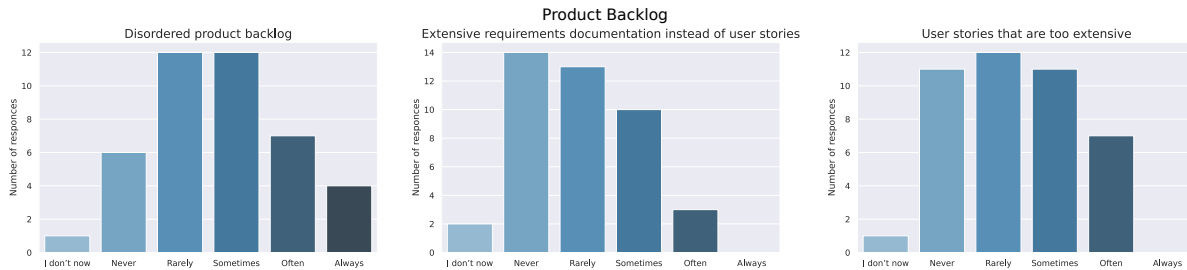


Fig. 4. Product backlog related anti-patterns

TABLE V  
PRODUCT OWNER RELATED ANTI-PATTERNS

Question	I don't now	Never	Rarely	Sometimes	Often	Always
PO without authority, has negligible influence on the selection of tasks to be implemented	1	19	10	6	5	1
Stakeholder indecisiveness – requirements come from multiple stakeholders and compete for prioritization	1	8	11	7	8	5
Lack of unanimity whether the task has been completed	1	12	16	10	2	1
An indirect customer who is service provider of its own customer	9	14	7	7	4	1
PO delegated by a client and does not understand the role	7	17	8	5	4	0

TABLE VI  
SPRINT PLANNING RELATED ANTI-PATTERNS

Question	I don't now	Never	Rarely	Sometimes	Often	Always
Exact estimation instead of relative estimation of items	2	11	8	10	5	6
Item estimation imposed to the team	0	17	6	6	10	3
No estimation of resources available for the upcoming sprint	0	16	11	7	5	3
Sprint planning meeting has no agenda	3	15	9	9	4	3
Sprint goal set at the end of the meeting or not at all	2	7	8	6	10	10
Not breaking large user stories into smaller ones during a meeting	3	8	12	10	8	2
Disruptions in the development process – other areas of the project are developed in turn without one being completed	2	7	8	17	8	1
Adding items during sprint	0	3	18	11	8	2
User stories are not fully refined	0	2	5	18	10	7
A burndown chart is not used	2	8	5	11	5	11

More than half of respondents indicated that they had never or rarely encountered “Semi functional teams” anti-pattern.

Lack of business knowledge, on the other hand, received the most responses about its occurrence, with more than 65% of the responses indicating that it happens sometimes, often, or always.

### G. Completing the sprint

None of the anti-patterns classified in the sprint completion section are present in a significant proportion of respondents' organizations. Table VIII and Figure 8 shows that only “The sprint does not end ends with a demonstration” was reported as often or always present in 36.8% of the responses.

Among the rest, however, “Missing retrospectives” and “Combining the two meetings into one” can still be distinguished as the ones with a significant proportion of “Sometimes”, “Often” and “Always” answers.

### H. Summary of the results

To identify the most common anti-patterns, an prevalence rate was calculated based on the survey results. The weighted sum of responses was calculated using the following weights: “I don't know” – 0, “Never” – 0, “Rarely” – 1, “Sometimes” – 2, “Often” – 3, and “Always” – 4.

The final score was then calculated for each sum as a fraction of the maximum score possible, i.e. 168. Table IX lists all the anti-patterns ordered by calculated score.

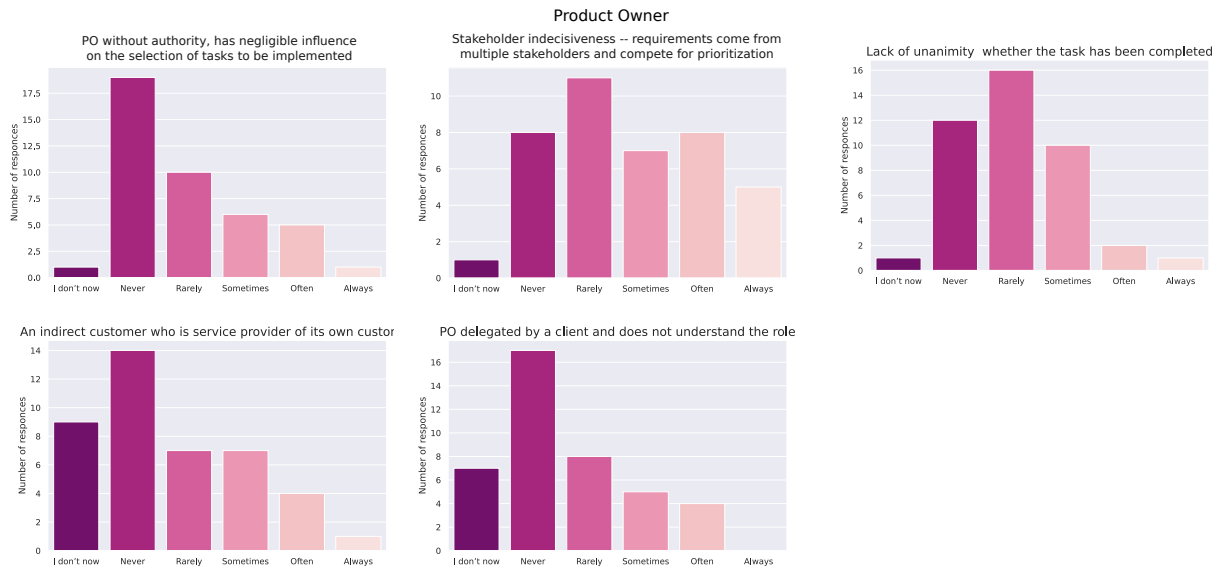


Fig. 5. Product owner related anti-patterns

TABLE VII  
DEVELOPMENT TEAM RELATED ANTI-PATTERNS

Question	I don't now	Never	Rarely	Sometimes	Often	Always
Semi functional teams	2	12	10	7	7	4
Insufficient technical knowledge	0	13	14	10	5	0
Lack of business knowledge	1	8	6	10	15	2

TABLE VIII  
COMPLETING THE SPRINT RELATED ANTI-PATTERNS

Question	I don't now	Never	Rarely	Sometimes	Often	Always
No or too long waiting for feedback (lack of Sprint review or stakeholders do not show interest)	2	13	13	4	9	1
The sprint does not end with a demonstration	4	12	8	4	8	6
Demonstration for the wrong targets	10	13	6	6	5	2
Missing retrospectives	0	23	5	5	5	4
Combining the two meetings into one	4	19	6	5	3	5

V. DISCUSSION

In terms of the prevalence of anti-patterns studied, “Unfinished Tasks” received the highest score. A situation in which some tasks are not completed within a sprint is in clear contradiction to Scrum’s guidelines, which states that “Sprint may be considered a short project” [14]. Of the respondents, only 3 people reported that such a situation rarely happens, and none that it never does. In our opinion, this is a serious violation of the guidelines and indicates not so much poor work by the development team, but rather poor planning of the tasks to be completed within the sprint.

The second most common anti-pattern is extending daily scrums beyond the recommended 15 minutes. Only one respondent reported never having such a situation and 9 rarely. This situation may stem from an incomplete understanding of the Scrum Guide, which openly states that “Daily Scrum is not the only time developers are allowed to adjust their

plan” [14]. Daily Scrum should only focus on progress and planning for the next day’s work. In practice, in addition to progress reports and planning for the next day’s work, this meeting is often used to discuss other issues, such as resolving technical problems that have arisen. We believe this is not a serious problem, but it should be encouraged to limit these meetings to recommended purposes only, and other issues should be resolved in additional meetings, perhaps in a limited group.

Another anti-pattern with the comparable prevalence score is “User stories are not fully refined”. This problem stems directly from the problem of requirements engineering. Unlike traditional software development methods, the Scrum methodology does not formally define how requirements are elicited. This can lead to less commitment and dedication to creating comprehensive user stories. Therefore, it is necessary to promote a balanced approach to requirements definition in

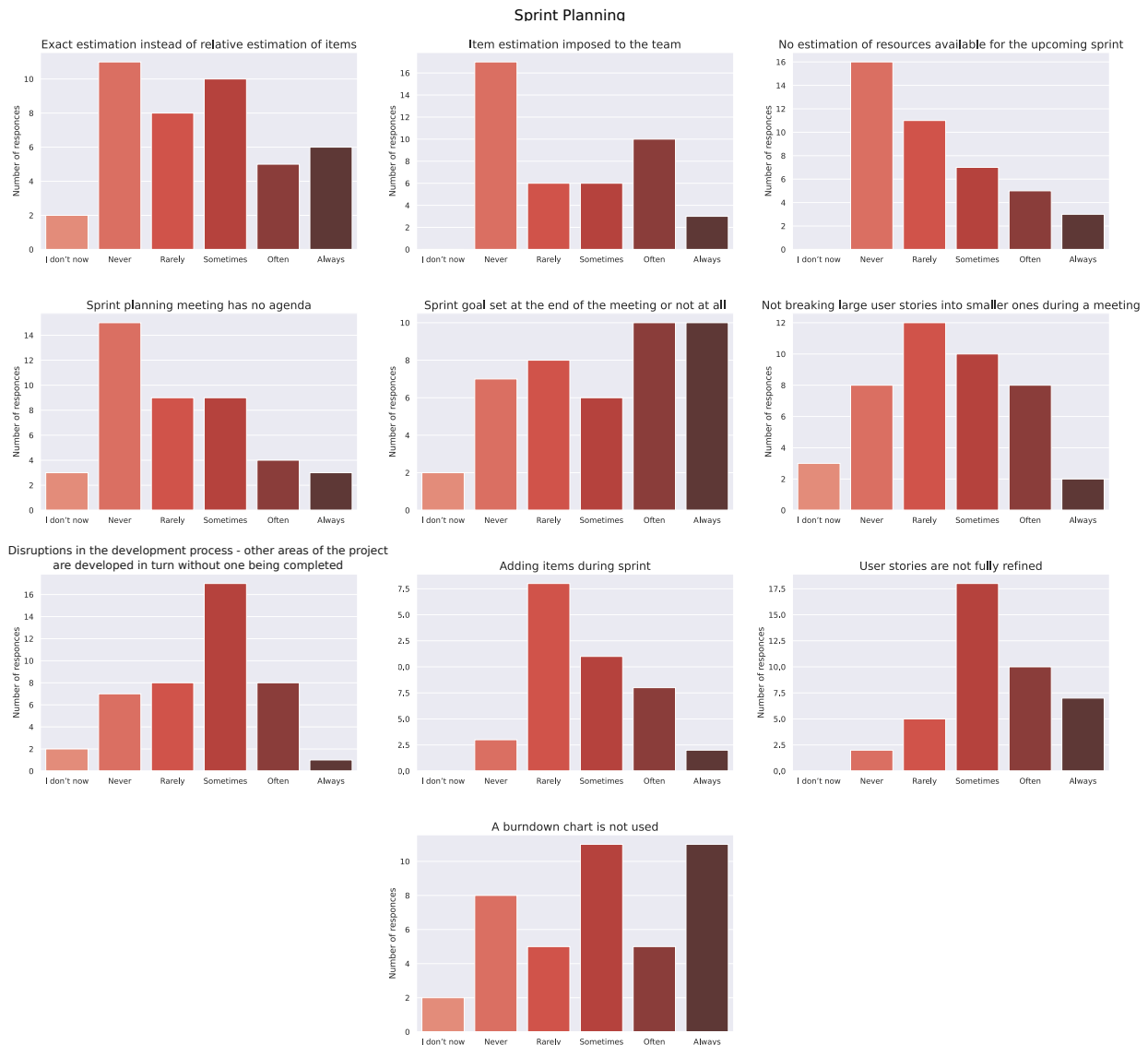


Fig. 6. Sprint planning related anti-patterns

agile software development organizations.

According to the Scrum Guide, the sprint goal should be defined at the sprint planning meeting. However, the survey results show that this is often not the case and that the goal is not defined at all or only at the end of the sprint. In our view, this is a serious violation of Scrum principles. Clear visibility of the purpose of the sprint leads to better coordination of work and greater involvement of team members.

Another anti-pattern in order of the prevalence score relates to not using the burndown chart. However, this is not the only way to visualize the project's progress. Therefore, we cannot draw too far-fetched conclusions here.

The last anti-pattern with prevalence score above 50% is "Testing in next sprint". This can be linked to the previously described prevalence of not completing tasks defined for the

sprint. In our opinion, this is also a very important issue. Lack of testing leads to delivery and demonstration of potentially non-working software at the end of Sprint. Scrum guide defines increments that are produced within a single sprint must be usable. Without thorough testing, this cannot be guaranteed. Therefore, it is important to emphasize to the stakeholders of the Scrum project the importance of continuous testing of the delivered functionality.

The least common anti-patterns are related to the length of sprints and Daily Scrums. The respondents confirmed that in their organizations, the length of the sprint is usually within the guidelines and is constant throughout the project. Furthermore, Daily Scrums are held daily and at fixed times.



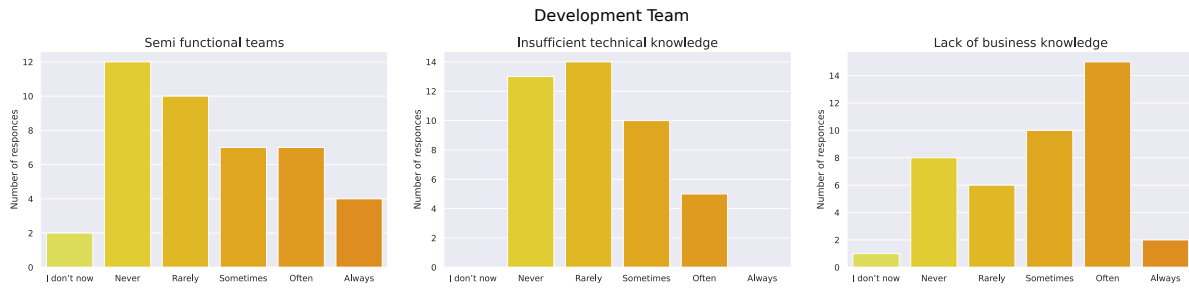


Fig. 7. Development team related anti-patterns

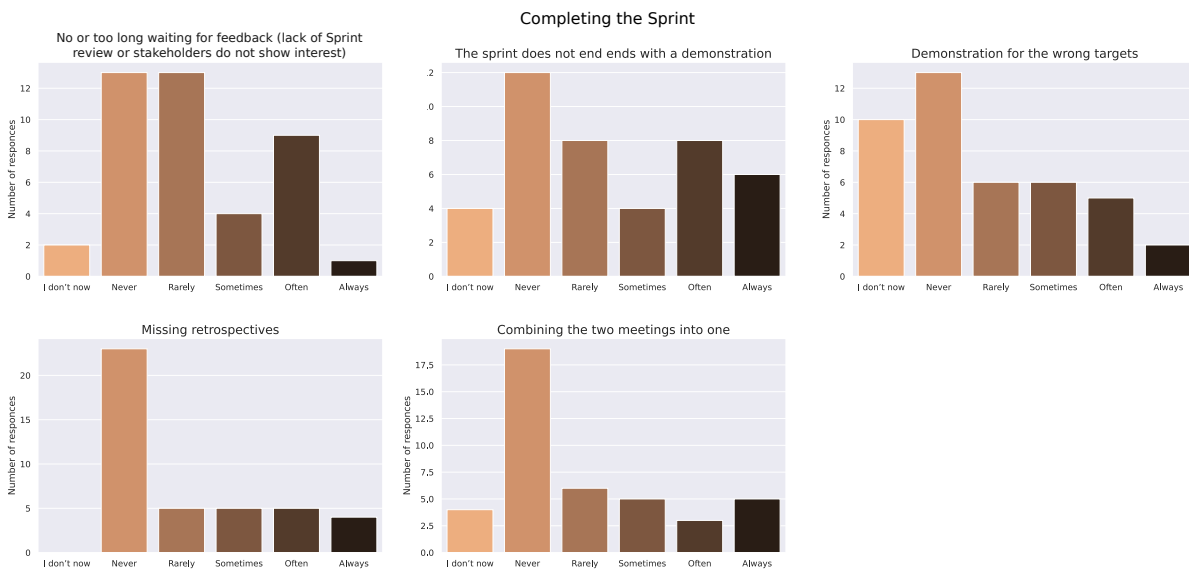


Fig. 8. Completing the sprint related anti-patterns

VI. CONCLUSION

The aim of the study was to determine the prevalence of 35 anti-patterns in Scrum software development methodology in Polish companies. Through a survey of industry professionals, we identified 6 most common issues in the organizations of the respondents. Among them, we believe that “Unfinished Tasks”, “User stories are not fully refined”, “Sprint goal set at the end of the meeting or not at all” and “Testing in next sprint” are the most critical. Although the presence of the remaining anti-patterns may not be critical to project success, they do indicate areas where teams can improve their adherence to Scrum principles. Overall, the findings of this study provide valuable insights into the current state of Scrum methodology in software development organizations.

While this study has provided valuable insights into the current state of Scrum methodology in software development organizations, further research is needed to understand the underlying causes of these issues. In future research, we aim to focus on investigating why these anti-patterns occur and what factors contribute to their persistence. This could be achieved through in-depth interviews with experienced project

managers and Scrum Masters who have encountered these anti-patterns in their work. Such interviews could shed light on the organizational and cultural factors that lead to the breaking of Scrum rules and provide guidance on how to address these issues effectively. By gaining a deeper understanding of the root causes of these anti-patterns, organizations can take more targeted and effective action to improve their Scrum practices and increase the success of their projects.

REFERENCES

- [1] T. Raharjo and B. Purwandari, “Agile project management challenges and mapping solutions: A systematic literature review,” in *Proceedings of the 3rd International Conference on Software Engineering and Information Management*, 2020. doi: 10.1145/3378936.3378949 pp. 123–129.
- [2] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries *et al.*, “Manifesto for agile software development,” 2001.
- [3] P. Weichbroth, “A case study on implementing agile techniques and practices: Rationale, benefits, barriers and business implications for hardware development,” *Applied Sciences*, vol. 12, no. 17, p. 8457, 2022. doi: 10.3390/app12178457
- [4] S. Hassani-Alaoui, A. F. Cameron, and T. Giannelia, ““we use scrum, but ...”: Agile modifications and project success,” in *Hawaii International Conference on System Sciences*, 2020. doi: 10.24251/HICSS.2020.765

TABLE IX  
PREVALENCE SCORE OF ANTI-PATTERNS

Anti-pattern	Weighted sum	Score
1. Unfinished tasks	118	70.24%
2. Daily Scrum lasts longer than 15 minutes	103	61.31%
3. User stories are not fully refined	99	58.93%
4. Sprint goal set at the end of the meeting or not at all	90	53.57%
5. A burndown chart is not used	86	51.19%
6. Testing in next sprint	85	50.60%
7. Lack of business knowledge	79	47.02%
8. Disordered product backlog	73	43.45%
9. Adding items during sprint	72	42.86%
10. Disruptions in the development process – other areas of the project are developed in turn without one being completed	70	41.67%
11. Stakeholder indecisiveness – requirements come from multiple stakeholders and compete for prioritization	69	41.07%
12. Invisible progress	67	39.88%
13. Exact estimation instead of relative estimation of items	67	39.88%
14. Not breaking large user stories into smaller ones during a meeting	64	38.10%
15. The sprint does not end ends with a demonstration	64	38.10%
16. Semi functional teams	61	36.31%
17. Item estimation imposed to the team	60	35.71%
18. Not all key issues are addressed	56	33.33%
19. User stories that are too extensive	55	32.74%
20. Not everyone actively participate in the meeting	54	32.14%
21. No estimation of resources available for the upcoming sprint	52	30.95%
22. No or too long waiting for feedback (lack of Sprint review or stakeholders do not show interest)	52	30.95%
23. Sprint planning meeting has no agenda	51	30.36%
24. Insufficient technical knowledge	49	29.17%
25. Lack of unanimity whether the task has been completed	46	27.38%
26. Missing retrospectives	46	27.38%
27. Combining the two meetings into one	45	26.79%
28. Extensive requirements documentation instead of user stories	42	25.00%
29. PO without authority, has negligible influence on the selection of tasks to be implemented	41	24.40%
30. Demonstration for the wrong targets	41	24.40%
31. An indirect customer who is service provider of its own customer	37	22.02%
32. PO delegated by a client and does not understand the role	30	17.86%
33. Daily scrums are not held every day	21	12.50%
34. Variable lengths of sprints	17	10.12%
35. The Daily Scrum does not take place at a fixed time of day	12	7.14%
36. Sprints that are too long	11	6.55%

- [5] "State of agile report," <https://digital.ai/resource-center/analyst-reports/state-of-agile-report/>, 2022, accessed: 2023-07-31.
- [6] V.-P. Eloranta, K. Koskimies, T. Mikkonen, and J. Vuorinen, "Scrum anti-patterns—an empirical study," in *2013 20th Asia-Pacific Software Engineering Conference (APSEC)*, vol. 1. IEEE, 2013. doi: 10.1109/APSEC.2013.72 pp. 503–510.
- [7] C. Matthies, T. Kowark, K. Richly, M. Uflacker, and H. Plattner, "How surveys, tutors, and software help to assess scrum adoption in a classroom software engineering project," in *Proceedings of the 38th International Conference on Software Engineering Companion*, 2016. doi: 10.1145/2889160.2889182 pp. 313–322.
- [8] P. J. Carew and D. Glynn, "Anti-patterns in agile adoption: A grounded theory case study of one irish it organisation," *Global Journal of Flexible Systems Management*, vol. 18, pp. 275–289, 2017. doi: 10.1007/s40171-017-0162-8
- [9] M. Mortada, H. M. Ayas, and R. Hebig, "Why do software teams deviate from scrum? reasons and implications," in *Proceedings of the International Conference on Software and System Processes*, 2020. doi: 10.1145/3379177.3388899 pp. 71–80.
- [10] E. Çetin and P. Onay Durdu, "Blended scrum model for software development organizations," *Journal of Software: Evolution and Process*, vol. 31, no. 2, p. e2147, 2019. doi: 10.1002/smr.2147
- [11] V. T. Heikkilä, M. Paasivaara, and C. Lassenius, "Scrumbut, but does it matter? a mixed-method study of the planning process of a multi-team scrum organization," in *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. IEEE, 2013. doi: 10.1109/ESEM.2013.27 pp. 85–94.
- [12] T. McKenzie, M. Morales-Trujillo, S. Lukosch, and S. Hoermann, "Is agile not agile enough? a study on how agile is applied and misapplied in the video game development industry," in *2021 IEEE/ACM Joint 15th International Conference on Software and System Processes (ICSSP) and 16th ACM/IEEE International Conference on Global Software Engineering (ICGSE)*. IEEE, 2021. doi: 10.1109/ICSSP-ICGSE52873.2021.00019 pp. 94–105.
- [13] T. Perry, "The intermediate customer anti-pattern," in *Agile 2008 Conference*. IEEE, 2008. doi: 10.1109/Agile.2008.74 pp. 280–283.
- [14] K. Schwaber and J. Sutherland, "The scrum guide," <https://www.scrum.org/resources/scrum-guide>, 2020, accessed: 2023-04-10.