Agile Practices as Solutions for Software Project Management Anti-patterns

Master Thesis

Gabriela Castro Flores

Madrid, June 2019
Agile Practices as Solutions for Software Project Management Anti-patterns
Master Thesis

This thesis is submitted to the ETSI Informáticos at Universidad Politécnica de Madrid in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering.

Master Thesis
Master Universitario en Ingeniería del Software – European Master in Software Engineering
Thesis Title: Agile Practices as Solutions for Software Project Management Anti-patterns
Thesis no: EMSE-2019-1
June, 2019

Author: Gabriela Castro Flores
Bachelor of Computational Systems Engineering
University of the presented title

Supervisor:

Ana M. Moreno
Ph. D. in Computer Science
Universidad Politécnica de Madrid

Department
School of Computing
Universidad Politécnica de Madrid

ETSI Informáticos
Universidad Politécnica de Madrid
Campus de Montegancedo, s/n
28660 Boadilla del Monte (Madrid)
Spain
# Table of contents

1. **Introduction** .............................................................................................................................. 8

2. **Agile Practices Research** ........................................................................................................... 12
   2.1 **Introduction** ......................................................................................................................... 12
   2.2 **Research methodology: Systematic mapping** ........................................................................ 12
      2.2.1 **Research questions** ........................................................................................................ 12
   2.3 **Search process** .................................................................................................................... 12
   2.4 **Study selection process** ....................................................................................................... 14
   2.5 **Data extraction process** ...................................................................................................... 15
   2.6 **Analysis process** ................................................................................................................. 15
   2.7 **Validity evaluation** .............................................................................................................. 16
   2.8 **Analysis of extracted data** .................................................................................................. 17
      2.8.1 **Data extracted** ................................................................................................................. 17
   2.9 **Consolidated list of agile practices** .................................................................................... 18
      2.9.1 **Consolidation process** .................................................................................................. 18
      2.9.2 **Final consolidated list** .................................................................................................. 24
   2.10 **Agile practices definitions** ............................................................................................... 26
      2.10.1 **Daily standup** ............................................................................................................... 27
      2.10.2 **Sprint/iteration planning** ............................................................................................. 28
      2.10.3 **Retrospectives** ............................................................................................................ 28
      2.10.4 **Sprint review** ............................................................................................................... 28
      2.10.5 **Short iterations** ........................................................................................................... 28
      2.10.6 **Release planning** ........................................................................................................ 29
      2.10.7 **Planning poker** ........................................................................................................... 29
      2.10.8 **Available Product owner** .......................................................................................... 29
      2.10.9 **Single team** ............................................................................................................... 29
      2.10.10 **Frequent releases** ..................................................................................................... 30
      2.10.11 **Common work area** ................................................................................................. 30
      2.10.12 **Product roadmapping** .............................................................................................. 30
      2.10.13 **Story mapping** ......................................................................................................... 30
      2.10.14 **Agile portfolio planning** ........................................................................................... 30
      2.10.15 **Agile UX** .................................................................................................................. 31

3. **Anti-patterns in Software Project Management Research** ....................................................... 33
   3.1 **Introduction** ....................................................................................................................... 33
3.2 Research questions ........................................................................................................................................ 33
3.3 Search process............................................................................................................................................... 33
3.4 Study selection process ................................................................................................................................. 35
3.5 Data extraction process................................................................................................................................ 35
3.6 Analysis process........................................................................................................................................... 35
3.7 Consolidated list of anti-patterns in software project management............................................................... 36
  3.7.1 Consolidated list ....................................................................................................................................... 36
  3.7.2 Anti-pattern software project management definitions ........................................................................... 37
4. Agile Practices as a Solution to Project Management Anti-patterns ............................................................ 42
  4.1 Introduction.................................................................................................................................................. 42
  4.2 Matching process ...................................................................................................................................... 42
    4.2.1 Matching criteria.................................................................................................................................. 43
    4.2.2 Agile Project Manager Profile ............................................................................................................... 44
  4.3 Lists of agile practices impacting Software Project Management Anti-patterns ....................................... 47
    4.3.1 Daily standup impact ............................................................................................................................ 47
    4.3.2 Sprint Planning impact ........................................................................................................................ 52
    4.3.3 Retrospectives impact ......................................................................................................................... 57
    4.3.4 Sprint review impact .......................................................................................................................... 62
    4.3.5 Short iterations impact ....................................................................................................................... 67
    4.3.6 Release planning impact ..................................................................................................................... 72
    4.3.7 Planning poker impact ....................................................................................................................... 77
    4.3.8 Available product owner impact ......................................................................................................... 82
    4.3.9 Single team impact ............................................................................................................................. 87
    4.3.10 Frequent releases impact .................................................................................................................. 91
    4.3.11 Common working area impact .......................................................................................................... 96
    4.3.12 Product roadmapping impact .......................................................................................................... 101
    4.3.13 Story mapping impact ..................................................................................................................... 106
    4.3.14 Agile portfolio planning impact ...................................................................................................... 111
    4.3.15 Agile UX impact .............................................................................................................................. 116
5 Discussion ....................................................................................................................................................... 123
  5.1 Number of anti-pattern related to agile practices ...................................................................................... 125
  5.2 Number agile practices related to anti-patterns ....................................................................................... 128
  5.3 Number of impacted anti-patterns classified by their activities ............................................................. 130
6 Conclusions .................................................................................................................................................... 138
7 References

7.1 Primary paper references for agile practices

7.2 Primary paper references for Software Project Management Anti-patterns

7.3 General paper references

Annex A.- Data extraction forms for agile practices lists

Annex B. Agile practices matching by name or concept

Annex C. Software project management anti-patterns extended definitions

Index of tables

Table 1: Summary of search results .............................................................. 14
Table 2: Rationale for excluded studies ....................................................... 14
Table 3: Data extraction form ................................................................. 15
Table 4: Daily standup appearances .......................................................... 20
Table 5: Sprint planning appearances ....................................................... 20
Table 6: Retrospectives appearances ......................................................... 20
Table 7: Sprint review appearances .......................................................... 20
Table 8: Short iterations appearances ....................................................... 20
Table 9: Release planning appearances .................................................... 20
Table 10: Planning poker appearances ..................................................... 21
Table 11: Available Product Owner appearances ...................................... 21
Table 12: Single team appearances ......................................................... 21
Table 13: Frequent releases appearances .................................................. 21
Table 14: Common work area appearances .............................................. 22
Table 15: Product roadmapping appearances .......................................... 22
Table 16: Story mapping appearances ...................................................... 22
Table 17: Agile portfolio planning appearances ........................................ 22
Table 18: Agile UX appearances ............................................................... 22
Table 19: Compiled agile practices shared by appearance ......................... 24
Table 20: Description of agile practices quick reference ......................... 26
Table 21: Summary of search results ...................................................... 35
Table 22: Final consolidated list of anti-patterns ..................................... 36
Table 23: Description of anti-patterns quick reference ........................... 37
Table 24: Anti-patterns categorized by impacted activity ....................... 38
Table 25: Daily standup – Project management anti-patterns ................. 47
Table 26: Sprint planning – Project management anti-patterns ............... 52
Table 27: Retrospectives – Project management anti-patterns ............... 57
Table 28: Sprint review – Project management anti-patterns ................. 63
Table 29: Short iterations – Project management anti-patterns ............................... 68
Table 30: Release planning – Project management anti-patterns ............................... 72
Table 31: Planning poker – Project management anti-patterns ................................. 77
Table 32: Available product owner – Project management anti-patterns ......................... 82
Table 33: Single team – Project management anti-patterns ........................................ 87
Table 34: Frequent releases – Project management anti-patterns ................................ 91
Table 35: Common working area – Project management anti-patterns .......................... 96
Table 36: Product roadmapping – Project management anti-patterns ........................... 101
Table 37: Story mapping – Project management anti-patterns ..................................... 107
Table 38: Agile portfolio planning – Project management anti-patterns .......................... 111
Table 39: Agile UX – Project management anti-patterns ............................................. 116
Table 40: Number of Agile Practices impacting the project management anti-patterns ........ 124
Table 41: Final compilation of the matching process of agile practices impacting .......... 125
project management anti-patterns by activity

Index of figures

Figure 1: Agile practices search process ................................................................. 13
Figure 2: Final consolidated agile practices list ......................................................... 25
Figure 3: Antipattern search process ................................................................. 34
Figure 4: Antipatterns categorized by impacted activity .......................................... 40
Figure 5: Number of anti-patterns related to the agile practices ............................. 127
Figure 6: Number of agile practices related to anti-patterns .................................. 129
Figure 7: Number of agile practices related to anti-patterns Controlling activity .......... 131
Figure 8: Number of agile practices related to anti-patterns Motivating activity .......... 132
Figure 9: Number of agile practices related to anti-patterns Planning activity ........... 133
Figure 10: Number of agile practices related to anti-patterns Scheduling activity ....... 134
Figure 11: Number of agile practices related to anti-patterns Staffing activity .......... 135
Figure 12: Anti-patterns activities impacted by agile practices ......................... 136
Chapter 1:

Introduction
1. Introduction

In our everyday life, there has been a quick acceleration of the adoption of new technologies, which require the development of specialized software to meet the needs that each person may have. The development of these systems requires to follow specific methodologies that allow keeping order and control over the activities that are executed.

For this reason, it is very important to have someone responsible for the management and lead of the activities that should be executed during a project development. But what happens when there is a lack of correct project management within the organization?

Different authors talk about the incorrect project administration and the consequences that this might bring into an enterprise. Many factors such as the establishment of realistic goals, competition, client satisfaction, a definite goal, profitability, implementation process, among others[1] can lead into a successful of failing project. Most of these activities are guided by the management, therefore, if the responsible for these activities is not right qualified the project would be in a high risk of failure.

The previously mentioned factors bring to us the concept of the software project management anti-patterns, which are the malpractices performed by the management software projects that leads to a negative consequence [2]. Hence, it is important to keep in mind that these wrong practices can be present at any moment within the organizations and it is essential to identify them to find a solution so that the previous outcomes will not be repeated.

On the other hand, in the late years has become trendy in the software field companies the use of agile practices. The enterprises have been changing from traditional schemes where the development of a product is performed on five stages (initiating, planning, executing, monitoring and controlling, and closing) to an agile development executed in short iterations.

This orientation to the agile schemes is due to they encourage the active involvement of the customer into the development process, which results into the constant improvement of the product and the constant observation of results [3]. Even when most of the organizations have leaned towards its implementation, it is important to mention that the use of these practices does not mean that all projects should be developed following them and they will solve of the problems derived from the traditional methodologies.

One advantage of the agile techniques is that most of the organizations can implement them customized based on their needs. However, they cannot be implemented randomly, it is necessary to analyze the unexpected results that bring unsuccessful projects and their possible causes. In many situations, the organization blames their processes or their teams about the failures that may appear during the development processes.

The enterprises sometimes do not consider that for having completely satisfactory processes there is more than one piece in the puzzle that might be failing. It is necessary to pay attention to every detail and observe that not necessarily the failures are due to the development processes or the
team in charge of it. The wrong administration of the projects or malpractices followed by the management can also be the cause of the arising problems within a project.

Moreover, the use of these practices implies a big change between the traditional and new tendencies which should be controlled in order to have a correct change management. Sometimes there are pieces on the traditional schemes that does not fit completely with the new agile schemes and must be treated carefully to have the expected results.

On agile practices some project management activities are treated as unnecessary, eventually, it is necessary to look a way to match the old processes with the new ones to bring quality to the products and cover the highest standards and client expectations.

This thesis will attempt to propose a solution for this traditional software project management anti-patterns. We look forward to proposing the implementation of agile practices to the software project management malpractices by answering the following three research questions:

- Which agile practices can help to prevent, reduce or avoid the impact of Project Management Anti-patterns?
- Which agile practices can help to identify the Project Management Anti-patterns?
- How these agile practices can help to reduce the impact of Project management anti-patterns?

In order to answer these questions, we will perform research on the most known and implemented agile practices within the software industry. Next, we will obtain a Project Management anti-patterns list with more incidences on the software industry. Based on these both lists, we plan to perform an analysis of each agile practice and the impact that they might have over each project management anti-pattern. Finally, we will explain the impact of these agile activities and how they help to reduce or solve the issues derived from the anti-patterns on which they have inference.

For the development of this project the thesis will be structured as follows:

- Chapter 1: *Introduction* which contains the explanation of the motivation of our work.
- Chapter 2: *Agile practices research* explains the research and systematic mapping process executed to obtain from the literature the most known and used agile practices on the industry and the final consolidated list of agile practices with the definitions of each of them.
- Chapter 3: *Anti-patterns in software project management* explains the research and systematic mapping process executed to obtain from the literature the most known and used anti-pattern project management on the industry and the final consolidated list of them with their correspondent definitions.
- Chapter 4: *Agile Practices as a solution to Project Management Anti-patterns* expose the matching process and explanation of the impact that each the agile practice has over the project management anti-pattern.
- Chapter 5: *Discussion and final remarks* the explanation of the important data obtained from the matching process using agile practices as a solution for the project management anti-patterns.
Agile Practices as Solutions for Software Project Management Anti-patterns
Master Thesis

- Chapter 6: Conclusions contains the conclusions with the final analysis of the results and some comments about the project.
- Chapter 7: References lists of references used for this work separated by primary papers, agile practices paper and, anti-pattern papers.
- Annexes: This part contains the data extraction forms for agile practices lists, the table with agile practices matching by name or concept and the software project management anti-patterns extended definitions.
Chapter 2:

Agile Practices Research
2. Agile Practices Research

2.1 Introduction
In this chapter we will discuss the research methodology that was selected to identify the existing agile practices, recognize which of them were the most known and implemented on the industry, to finally consolidate them on a list.

2.2 Research methodology: Systematic mapping
Systematic mapping is a useful methodology for documenting research works giving an overview of a topic, identifying and classifying contributions in relation to the subject. It involves searching, analyzing and synthesizing data or information extracted from different literature. For performing this research we have followed guidelines recommended by [4].

2.2.1 Research questions
The research question for the first part of the project was thought with the main goal of identifying the most known agile practices and which of them are implemented at industry.

RQ1. What are the most known/used agile practices in industry nowadays?

2.3 Search process
The search process represented on Fig. 1 took place in the first half of 2019 from January to June, the scope of this part of the research was to obtain existent agile practices lists to perform a selection of the most well-known and create a consolidated list. For that reason, we worked on papers that already contained the results of systematic literature reviews including systematic mappings. That is we performed a ternary study over secondary studies.

According to the research question (RQ1) the following key concepts were defined: Agile practices, Mapping study and Systematic review. The different combinations between these and concepts derived on the following string: (agile practices OR agile techniques OR agile activities) AND (systematic mapping OR mapping study OR systematic review OR systematic decomposition OR systematic literature review).

The research was executed in electronic databases that store publications from official journals and conferences:

- Science Direct: https://www.sciencedirect.com
- Research gate: https://www.researchgate.net/
- Institute of Electrical and Electronics Engineers (IEEEXplore): https://ieeexplore.ieee.org
- Association for Computing Machinery (ACM): https://www.acm.org/
- Google scholar: https://scholar.google.es/
Most of the research databases were selected following the recommendations found on articles [A2][5] and some others from [4]. Also, some of the papers used for the systematic mapping were taken from the references of previous papers. In Table 1 it is presented the reviewed databases and the total amount of papers selected from each of them, it also includes the result of the backward snowballing done after the first papers selection.
Table 1 Summary of search results

<table>
<thead>
<tr>
<th>Database</th>
<th>Search results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Science direct</td>
<td>6</td>
</tr>
<tr>
<td>Research gate</td>
<td>7</td>
</tr>
<tr>
<td>IEEEXplore</td>
<td>11</td>
</tr>
<tr>
<td>ACM</td>
<td>5</td>
</tr>
<tr>
<td>Google scholar</td>
<td>7</td>
</tr>
<tr>
<td>Backward Snowballing search</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>54</td>
</tr>
<tr>
<td>Total without duplicates</td>
<td>38</td>
</tr>
</tbody>
</table>

2.4 Study selection process

The selection process was executed in two parts, a first stage was based on where the studies took place (enterprise, university, etc.). The title and the abstracts were reviewed to decide if the article would be useful for the thesis.

The second stage consisted on reading and reviewing the complete articles to be selected based on the inclusion or exclusion criteria. The definition of these criteria were based on our interest about agile practices that were used or implemented in the late years.

Inclusion criteria

1. Studies that presented a concrete list of agile practices, techniques or activities.
2. Studies that were part of a conference or a journal.

Exclusion criteria

1. Studies dealing just with a particular agile methodology or an approximation without discussing the specific practices.
2. Studies whose full-text was not accessible.
3. Studies not written in English.

For the first part of the selection process the search revealed a total of 38 studies extracted from the previously mentioned databases. After the first review of them 17 were marked as relevant for the study (Table 2).

Table 2 Rationale for excluded studies

<table>
<thead>
<tr>
<th>Reason</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicates</td>
<td>18</td>
</tr>
<tr>
<td>Studies dealing just with an agile methodology, agile development or an approximation without discussing the specific practices</td>
<td>16</td>
</tr>
<tr>
<td>Studies whose full-text was not accessible</td>
<td>1</td>
</tr>
<tr>
<td>Studies not written in English</td>
<td>0</td>
</tr>
<tr>
<td>Studies dealing with agile practices without listing them</td>
<td>3</td>
</tr>
<tr>
<td>Backward snowballing</td>
<td>2</td>
</tr>
</tbody>
</table>
After a rigorous reviewing of the left 15 papers we excluded 2 of them because they did not accomplish with the inclusion criteria. The backward snowballing was also applied obtaining 2 new papers relevant for the study.

### 2.5 Data extraction process

The data extraction process was performed following the recommendations and the process designed in [5] and the guidelines from [4]. It was divided into three steps: design of data extraction form, data extraction and validation of extracted data.

Data extraction form is based on [4]. However, it was necessary to analyze and adapt information from [6][7] in order to complement the form.

**Table 3. Data extraction form**

<table>
<thead>
<tr>
<th>Data item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study ID</td>
<td>Own database number + Author’s name + year of publication</td>
</tr>
<tr>
<td>Article Title</td>
<td>Name of the article</td>
</tr>
<tr>
<td>Author Name</td>
<td>Name of the authors</td>
</tr>
<tr>
<td>Year of Publication</td>
<td>Calendar year</td>
</tr>
<tr>
<td>Venue</td>
<td>Name of publication venue</td>
</tr>
<tr>
<td>List of agile</td>
<td>What are the agile practices implemented or investigated in this paper?</td>
</tr>
<tr>
<td>practices</td>
<td></td>
</tr>
<tr>
<td>Notes</td>
<td>Any note used for the record</td>
</tr>
</tbody>
</table>

**List of agile practices** refers to the specific list showed on the paper that was applied or studied for the creation of the article.

### 2.6 Analysis process

In this section will be explained a general overview about how was performed the analysis process for this first systematic mapping. It was necessary to read and review on detail each of the 19 papers that were selected as primary papers and considered relevant for the research following a document analysis process. As it is explained in [8], document analysis is a systematic procedure for reviewing or evaluating documents. The interpretation of documents is performed in order to elicit meaning, gain understanding, and develop empirical knowledge.

We reviewed different literature for understanding and incorporate relevant information to our work. The analyzed documents where part of scientific and technical studies, but also, non-technical documents such as reports that are official documents developed by companies or professional organizations which share empirical information extracted from the daily work within the organizations.
Our document analysis was performed to provide a confluence of evidence to introduce credibility into our study. For this study it was necessary to examine different sources to corroborate that the lists of practices identified were representative of industrial practice.

Those lists of practices were recorded in a data extraction form, that was used for the matching process. For that aim, it was selected a reference list that would be the guidance for having a comparison point of the agile practices and their definitions. The criteria for selecting that list was the latest publication available, which on the other hand, had a clear professional base.

Subsequently, we evaluated the obtained data to decide if it was necessary to dismiss any concept that was not relevant for the study or to put them together in case their meaning were the same but with different names, finally, we obtained the consolidated list and formulated a definition for each agile practice, summarizing the found definitions on each paper.

2.7 Validity evaluation

To accomplish the quality criterion that should be covered as it is mentioned in [4], is important to identify and discuss about the validity threats. For this study it was identified the following.

Search coverage: This threat refers to the importance of ensuring that any relevant studies were ignored or not found. As a first action it was selected the most used and known scientific databases, where each search was going to be done.

After this selection, it was established a list of keywords used to obtain in the most accurate way studies that were related to our main scope. Limiting the big extension of agile topics to the main agile practices, studies about agile practices implemented with systematic mapping methodology and studies about agile techniques, practices or activities.

Then it was applied the string created with the different keywords in the selection of the scientific databases. It is important to mention that to ensure the obtention of the right topics, it was selected only papers which study cases were about agile practices and not specifically an agile methodology. This way it was open the review scope, considering that nowadays it is extremely difficult to find a project which was developed with a pure specific agile methodology instead of the practices that fit best in a project.

Finally, most of the selected systematic mappings about agile practices that were found during the search process were free of this threat due that it was easy to specify and limit the research to agile activities.

Study selection: This threat refers to the possibility of have a bias deciding if any paper should be part of the study or if it will be useful. To reduce its impact, it was followed the subsequent strategies to deal with this problem. Firstly, it was reviewed and done a selection process for the inclusion and exclusion criteria, this way it is easier to identify the specific data included in the studies that was going to be helpful for the research.

Secondly, the search was complemented with snowball sampling. Thus, after having a set of articles that were studied it was possible to increase the information that was going to be analyzed for the study.
Data extraction and classification: This threat focus on the misinterpretation of the information and data obtained from the selected articles. Besides researcher bias would still be a threat, to mitigate them was created the data extraction form.

This form was prepared in a specific way to obtain the information as concisely as possible. After obtaining the data from each paper each form was reviewed and analyzed a second time to ensure that the selected data was meaningful for the project. 2.8 Consolidated list of agile practices

2.8 Analysis of extracted data
To obtain the consolidated list, we followed the process explained from section 2.1 to 2.7 Agile Practices Research (Chapter 2). After performing the paper selection, we proceed with the matching process. During this process it was observed that many of the practices listed on each paper are either repeated or have a different name for a similar practice.

In order to manage the concepts in a practical way, avoiding redundant information or techniques, we perform this consolidation considering two different criteria. First, some authors use different names for related or same concepts, we reviewed these names and compared the concepts to obtain a standard name and include it on the list.

Second, some of the practices are a variation or an extension of other. Due to this, we decided to put them together and consider them as one. Having these variations separated might not give the same relevance using them as a solution for the anti-patterns.

The detail of this selection process and the consolidated agile practices list is presented in the following sections of this chapter.

2.8.1 Data extracted
As it was explained on the data extraction process section, a special form was designed to facilitate the compilation of the data from each analyzed paper. This was implemented as a support tool, for later obtain the information and perform the matching process.

In the Annex A are available all the extraction data forms that were fulfilled. Below is an example of one of these extraction data forms fulfilled.

Adapting Agile Practices in University Contexts [A6]

Study ID: Masood2018

Article title: Adapting Agile Practices in University Contexts

Author Name: Zainab Masood, Rashina Hoda, Kelly Blincoe

Year of publication: 2018

Venue: Journal of Systems and Software

Type of research: Validation research
List of agile practices:

<table>
<thead>
<tr>
<th>Scrum board</th>
<th>Release planning</th>
<th>Sprint review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily stand-up</td>
<td>Retrospective</td>
<td>Product owner</td>
</tr>
<tr>
<td>Sprint planning</td>
<td>Cross functionality</td>
<td>Pair programming</td>
</tr>
<tr>
<td>Review meeting</td>
<td>Sprint length and weight</td>
<td></td>
</tr>
</tbody>
</table>

2.9 Consolidated list of agile practices

2.9.1 Consolidation process

As previously discussed, the consolidated list was obtained thorough a document analysis process. To help with the document analysis we work with specific data forms to collect the results of each source, that is the specific list of agile practices suggested. The 17 selected primary papers had a concrete list of agile practices.

The next step was to select a reference list. This allows to have a base to follow and match with it the other agile practices lists. Based on the nearest publication dates, the method it was obtained and the source, we selected the list of the 12th Annual State of Agile report [A3] from the year 2018. This report is published every year by CollabNet VersionOne.

The enterprise works with many different companies focused on the software area, they obtain the information by means of surveys; evaluates the insight into agile trends, best practices and lessons learned; for finally create this report and publish it as a support for research area and improvement for those companies interested on agile practices. The lists that were taken from this report, for this research project interests, are classified into two different main concepts: agile techniques and engineering practices.

The agile techniques are those practices that are implemented in agile schemes which are more related to deal with project management activities and issues. On the other hand, the engineering practices are software engineering techniques that can be implemented in both traditional or agile schemes; these practices are principally used in order to introduce better quality to the software or improve the development processes. Below is presented both lists of concepts.

Agile techniques:
- Daily standup
- Sprint/Iteration planning
- Retrospectives
- Sprint/Iteration review
- Short iterations
- Release planning
- Planning poker/ team estimation
- Kanban
- Dedicated customer/ Product owner
- Single team (integrated dev and test)
- Frequent releases

Engineering practices:
- Unit testing
- Coding standards
- Continuous integration
- Refactoring
- Continuous deployment
- Pair programming
- TDD
- Automated acceptance testing
- Collective code ownership
- Sustainable pace
- BDD
Subsequent to the reference list selection, it was performed the matching of the concepts of each of the other 16 remaining lists with the principal one. All the practices mentioned on each list were analyzed, afterward, each of the practices were matched with the reference list. It is necessary to emphasize that most of these lists contain both agile practices and engineering practices, but they are not classified separately.

Each author includes those engineering practices on their agile lists because they are useful and fit well on the agile schemes, but they do not have a direct impact on project management activities. Due to this project is focused on software project management activities and it is not necessary to relate them with activities proper of the software development, we decided to not include the engineering concepts on our agile practices list.

The matching process was divided into two stages, on the first, were matched the practices with the same name and concept. For the second stage, the practices that had a different name or similar one were matched based on their definition. In only one case, one of the concepts was a variant of other; Kanban is part of the agile portfolio planning, so we decided to combine them and keep the second one that has the general idea of this kind of agile tools.

After the second stage some concepts were still unmatched. Based on the definition that each paper had about them and due to most of them were not suitable to incorporate with other practice, we took the decision to not include them on the reference list in order to delimit a concise list.

Those excluded concepts were found in the literature in different papers only on one occasion. The lack of repetitions and the low rate of implementation of these subjects, motivate us to take them out from the reference list, those concepts are listed as follows: defect reports, defect trend metrics, data naming conventions, self-assignment, instant messages, pair planning, informal communication, shared conceptualization, team documentation decisions rather than planning.

Summarizing, after analyzing the found lists of practices, we grouped those ones that were dealing with the same or related process. The consolidated list is presented in the next section. The group of tables (Tables from 4 to 17) show the practices list matched per names, and they are separated by each practice to facilitate their understandability.

The tables were organized by columns. The first column presents the reference from each paper where the lists where taken, the second column contains the agile practices as they were named by each author. The blank spaces or where the is possible to see “-“ mean that the practice that we were matching did not appear on the referenced article. The practices were split by agile practice name in different tables to allow the reader to have a better overview of the showed information, however, on Annex B are listed all the agile practices in a unique table.
<table>
<thead>
<tr>
<th>Reference</th>
<th>Agile practice name</th>
<th>Reference</th>
<th>Agile practice name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A1]</td>
<td>Daily stand up meeting</td>
<td>[A1]</td>
<td>Iteration task list</td>
</tr>
<tr>
<td>[A8]</td>
<td>Standup meetings</td>
<td>[A8]</td>
<td>Planning meeting</td>
</tr>
<tr>
<td>[A9]</td>
<td>Scrum meeting / daily stand up</td>
<td>[A9]</td>
<td>-</td>
</tr>
<tr>
<td>[A10]</td>
<td>-</td>
<td>[A10]</td>
<td>-</td>
</tr>
<tr>
<td>[A12]</td>
<td>Stand up / scrum meeting</td>
<td>[A12]</td>
<td>-</td>
</tr>
<tr>
<td>[A14]</td>
<td>-</td>
<td>[A14]</td>
<td>Continuous planning</td>
</tr>
<tr>
<td>[A15]</td>
<td>Daily meetings/ Stand up Meetings</td>
<td>[A15]</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6. Retrospectives appearances

<table>
<thead>
<tr>
<th>Reference</th>
<th>Agile practice name</th>
<th>Reference</th>
<th>Agile practice name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A2]</td>
<td>-</td>
<td>[A2]</td>
<td>Outcome review</td>
</tr>
<tr>
<td>[A4]</td>
<td>Retrospectives</td>
<td>[A4]</td>
<td>-</td>
</tr>
<tr>
<td>[A5]</td>
<td>Retrospectives</td>
<td>[A5]</td>
<td>Iteration/Sprint reviews</td>
</tr>
<tr>
<td>[A8]</td>
<td>Retrospectives</td>
<td>[A8]</td>
<td>Sprint review/ Sprint demo</td>
</tr>
<tr>
<td>[A9]</td>
<td>Retrospectives</td>
<td>[A9]</td>
<td>-</td>
</tr>
<tr>
<td>[A10]</td>
<td>-</td>
<td>[A10]</td>
<td>Iterative controls</td>
</tr>
<tr>
<td>[A12]</td>
<td>Retrospectives</td>
<td>[A12]</td>
<td>Iteration reviews/ demos</td>
</tr>
<tr>
<td>[A13]</td>
<td>Retrospectives</td>
<td>[A13]</td>
<td>-</td>
</tr>
<tr>
<td>[A15]</td>
<td>Retrospectives</td>
<td>[A15]</td>
<td>Sprint review meeting</td>
</tr>
</tbody>
</table>

Table 8. Short iterations appearances

<table>
<thead>
<tr>
<th>Reference</th>
<th>Agile practice name</th>
<th>Reference</th>
<th>Agile Practice Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A2]</td>
<td>Time boxing/ Delivering frequent releases</td>
<td>[A2]</td>
<td>-</td>
</tr>
<tr>
<td>[A7]</td>
<td>Sprint/ iterations</td>
<td>[A7]</td>
<td>-</td>
</tr>
<tr>
<td>[A8]</td>
<td>Short iteration proxy customer</td>
<td>[A8]</td>
<td>-</td>
</tr>
</tbody>
</table>
### Agile Practices as Solutions for Software Project Management Anti-patterns

#### Master Thesis

<table>
<thead>
<tr>
<th>Reference</th>
<th>Agile Practice Name</th>
<th>Reference</th>
<th>Agile Practice Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A10]</td>
<td>-</td>
<td>[A10]</td>
<td>-</td>
</tr>
<tr>
<td>[A11]</td>
<td>-</td>
<td>[A11]</td>
<td>-</td>
</tr>
<tr>
<td>[A14]</td>
<td>-</td>
<td>[A14]</td>
<td>-</td>
</tr>
<tr>
<td>[A15]</td>
<td>-</td>
<td>[A15]</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 10. Planning poker appearances**

<table>
<thead>
<tr>
<th>Reference</th>
<th>Agile Practice Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A3]</td>
<td>Planning poker/ team estimation</td>
</tr>
<tr>
<td>[A1]</td>
<td>Planning game</td>
</tr>
<tr>
<td>[A2]</td>
<td>-</td>
</tr>
<tr>
<td>[A4]</td>
<td>Agile games/ Team based estimations</td>
</tr>
<tr>
<td>[A5]</td>
<td>Expert estimation</td>
</tr>
<tr>
<td>[A6]</td>
<td>-</td>
</tr>
<tr>
<td>[A7]</td>
<td>Planning game/ Estimation meeting</td>
</tr>
<tr>
<td>[A8]</td>
<td>Planning game</td>
</tr>
<tr>
<td>[A9]</td>
<td>Trust in teams</td>
</tr>
<tr>
<td>[A10]</td>
<td>-</td>
</tr>
<tr>
<td>[A11]</td>
<td>-</td>
</tr>
<tr>
<td>[A12]</td>
<td>Planning poker</td>
</tr>
<tr>
<td>[A13]</td>
<td>-</td>
</tr>
<tr>
<td>[A14]</td>
<td>-</td>
</tr>
<tr>
<td>[A15]</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 11. Available Product Owner appearances**

<table>
<thead>
<tr>
<th>Reference</th>
<th>Agile Practice Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A3]</td>
<td>Dedicated customer/ Product owner</td>
</tr>
<tr>
<td>[A1]</td>
<td>Active stakeholder participation</td>
</tr>
<tr>
<td>[A2]</td>
<td>Product vision/ Customer involvement</td>
</tr>
<tr>
<td>[A4]</td>
<td>Dedicated product owner</td>
</tr>
<tr>
<td>[A5]</td>
<td>-</td>
</tr>
<tr>
<td>[A6]</td>
<td>-</td>
</tr>
<tr>
<td>[A7]</td>
<td>-</td>
</tr>
<tr>
<td>[A8]</td>
<td>-</td>
</tr>
<tr>
<td>[A9]</td>
<td>-</td>
</tr>
<tr>
<td>[A10]</td>
<td>Customer participation</td>
</tr>
<tr>
<td>[A11]</td>
<td>Client interaction</td>
</tr>
<tr>
<td>[A12]</td>
<td>Features in iteration customer valued</td>
</tr>
<tr>
<td>[A13]</td>
<td>-</td>
</tr>
<tr>
<td>[A14]</td>
<td>Customer involvement</td>
</tr>
<tr>
<td>[A15]</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 12. Single team appearances**

<table>
<thead>
<tr>
<th>Reference</th>
<th>Agile practice name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A3]</td>
<td>Single team</td>
</tr>
<tr>
<td>[A1]</td>
<td>-</td>
</tr>
<tr>
<td>[A2]</td>
<td>Small cross-functional teams/ Common knowledge</td>
</tr>
<tr>
<td>[A4]</td>
<td>Integrated dev/QA</td>
</tr>
<tr>
<td>[A5]</td>
<td>-</td>
</tr>
<tr>
<td>[A6]</td>
<td>Cross functionality</td>
</tr>
<tr>
<td>[A7]</td>
<td>-</td>
</tr>
<tr>
<td>[A8]</td>
<td>One team</td>
</tr>
<tr>
<td>[A9]</td>
<td>Collaborative software dev process/ Small team</td>
</tr>
<tr>
<td>[A10]</td>
<td>Small team</td>
</tr>
<tr>
<td>[A11]</td>
<td>-</td>
</tr>
<tr>
<td>[A12]</td>
<td>Whole multidisciplinary team</td>
</tr>
<tr>
<td>[A13]</td>
<td>-</td>
</tr>
<tr>
<td>[A14]</td>
<td>Cross-functional teams</td>
</tr>
</tbody>
</table>

**Table 13. Frequent releases appearances**

<table>
<thead>
<tr>
<th>Reference</th>
<th>Agile practice name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A3]</td>
<td>Frequent releases</td>
</tr>
<tr>
<td>[A1]</td>
<td>-</td>
</tr>
<tr>
<td>[A2]</td>
<td>Delivering frequent releases</td>
</tr>
<tr>
<td>[A4]</td>
<td>-</td>
</tr>
<tr>
<td>[A5]</td>
<td>-</td>
</tr>
<tr>
<td>[A6]</td>
<td>-</td>
</tr>
<tr>
<td>[A7]</td>
<td>-</td>
</tr>
<tr>
<td>[A8]</td>
<td>Continuous/automated builds</td>
</tr>
<tr>
<td>[A9]</td>
<td>-</td>
</tr>
<tr>
<td>[A10]</td>
<td>Frequent builds</td>
</tr>
<tr>
<td>[A11]</td>
<td>-</td>
</tr>
<tr>
<td>[A12]</td>
<td>Frequent release of working software</td>
</tr>
<tr>
<td>[A13]</td>
<td>-</td>
</tr>
<tr>
<td>[A14]</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 14. Common work area appearances

<table>
<thead>
<tr>
<th>Reference</th>
<th>Agile practice name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A3]</td>
<td>Common work area</td>
</tr>
<tr>
<td>[A1]</td>
<td>Co-located team</td>
</tr>
<tr>
<td>[A2]</td>
<td>-</td>
</tr>
<tr>
<td>[A4]</td>
<td>Open work area</td>
</tr>
<tr>
<td>[A5]</td>
<td>-</td>
</tr>
<tr>
<td>[A6]</td>
<td>-</td>
</tr>
<tr>
<td>[A7]</td>
<td>-</td>
</tr>
<tr>
<td>[A8]</td>
<td>Sit together</td>
</tr>
<tr>
<td>[A9]</td>
<td>Effective communication/ Face-to-Face communication</td>
</tr>
<tr>
<td>[A10]</td>
<td>-</td>
</tr>
<tr>
<td>[A11]</td>
<td>Direct communication</td>
</tr>
<tr>
<td>[A12]</td>
<td>Synchronous communication</td>
</tr>
<tr>
<td>[A13]</td>
<td>Open work area</td>
</tr>
<tr>
<td>[A14]</td>
<td>Face-to-face communication</td>
</tr>
<tr>
<td>[A15]</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 15. Product roadmapping appearances

<table>
<thead>
<tr>
<th>Reference</th>
<th>Agile Practice Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A3]</td>
<td>Product roadmapping</td>
</tr>
<tr>
<td>[A1]</td>
<td>-</td>
</tr>
<tr>
<td>[A2]</td>
<td>Progress monitoring</td>
</tr>
<tr>
<td>[A4]</td>
<td>-</td>
</tr>
<tr>
<td>[A5]</td>
<td>Model checking</td>
</tr>
<tr>
<td>[A6]</td>
<td>-</td>
</tr>
<tr>
<td>[A7]</td>
<td>Requirements workshop</td>
</tr>
<tr>
<td>[A8]</td>
<td>-</td>
</tr>
<tr>
<td>[A9]</td>
<td>Roadmap</td>
</tr>
<tr>
<td>[A10]</td>
<td>Feature growth not size growth</td>
</tr>
<tr>
<td>[A11]</td>
<td>-</td>
</tr>
<tr>
<td>[A12]</td>
<td>Task board</td>
</tr>
<tr>
<td>[A13]</td>
<td>-</td>
</tr>
<tr>
<td>[A14]</td>
<td>-</td>
</tr>
<tr>
<td>[A15]</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 16. Story mapping appearances

<table>
<thead>
<tr>
<th>Reference</th>
<th>Agile Practice Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A3]</td>
<td>Story mapping</td>
</tr>
<tr>
<td>[A1]</td>
<td>-</td>
</tr>
<tr>
<td>[A2]</td>
<td>-</td>
</tr>
<tr>
<td>[A4]</td>
<td>Story mapping</td>
</tr>
<tr>
<td>[A5]</td>
<td>User stories</td>
</tr>
<tr>
<td>[A6]</td>
<td>-</td>
</tr>
<tr>
<td>[A7]</td>
<td>User stories</td>
</tr>
<tr>
<td>[A8]</td>
<td>User stories</td>
</tr>
<tr>
<td>[A9]</td>
<td>User story</td>
</tr>
<tr>
<td>[A10]</td>
<td>-</td>
</tr>
<tr>
<td>[A11]</td>
<td>Provision of user stories/ Priorization of user stories</td>
</tr>
<tr>
<td>[A12]</td>
<td>Requirements written as informal stories</td>
</tr>
<tr>
<td>[A13]</td>
<td>User stories</td>
</tr>
<tr>
<td>[A14]</td>
<td>User stories</td>
</tr>
<tr>
<td>[A15]</td>
<td>User stories</td>
</tr>
</tbody>
</table>

Table 17. Agile portfolio planning appearances

<table>
<thead>
<tr>
<th>Reference</th>
<th>Agile Practice Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A3]</td>
<td>Agile portfolio planning</td>
</tr>
<tr>
<td>[A1]</td>
<td>-</td>
</tr>
<tr>
<td>[A2]</td>
<td>-</td>
</tr>
<tr>
<td>[A4]</td>
<td>Digital taskboard/ Analog taskboard</td>
</tr>
<tr>
<td>[A5]</td>
<td>Limit work in progress/ Definition done/ready</td>
</tr>
<tr>
<td>[A6]</td>
<td>Scrum board</td>
</tr>
<tr>
<td>[A7]</td>
<td>-</td>
</tr>
<tr>
<td>[A8]</td>
<td>Virtual scrum wall</td>
</tr>
<tr>
<td>[A9]</td>
<td>-</td>
</tr>
<tr>
<td>[A10]</td>
<td>-</td>
</tr>
<tr>
<td>[A11]</td>
<td>-</td>
</tr>
<tr>
<td>[A12]</td>
<td>-</td>
</tr>
<tr>
<td>[A13]</td>
<td>-</td>
</tr>
<tr>
<td>[A14]</td>
<td>-</td>
</tr>
<tr>
<td>[A15]</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 18. Agile UX appearances

<table>
<thead>
<tr>
<th>Reference</th>
<th>Agile Practice Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A3]</td>
<td>Agile UX</td>
</tr>
<tr>
<td>[A1]</td>
<td>Whiteboard sketches/sketching modelling/ Simple design/ evolutionary</td>
</tr>
<tr>
<td></td>
<td>design/ paper-based modelling/ Paper models</td>
</tr>
<tr>
<td>---</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>[A2]</td>
<td>-</td>
</tr>
<tr>
<td>[A4]</td>
<td>-</td>
</tr>
<tr>
<td>[A5]</td>
<td>Design reviews</td>
</tr>
<tr>
<td>[A6]</td>
<td>-</td>
</tr>
<tr>
<td>[A7]</td>
<td>Simple/ Incremental design</td>
</tr>
<tr>
<td>[A8]</td>
<td>Simple/ Incremental design</td>
</tr>
<tr>
<td>[A9]</td>
<td>Periodic design Review / Prototyping/ End-user-focused and value-centric system design</td>
</tr>
<tr>
<td>[A10]</td>
<td>-</td>
</tr>
<tr>
<td>[A11]</td>
<td>Prototyping</td>
</tr>
<tr>
<td>[A12]</td>
<td>Informal design/ design inspections/ emergent design</td>
</tr>
<tr>
<td>[A13]</td>
<td>-</td>
</tr>
<tr>
<td>[A14]</td>
<td>Prototyping</td>
</tr>
<tr>
<td>[A15]</td>
<td>-</td>
</tr>
</tbody>
</table>

In the following section is possible to observe the complete list of agile practices. This table (Table 19) was classified with the following color code:

- No concept was found referencing to the agile practice
- The practice is mentioned using the same name and concept
- The practice appears with a different name, but the definition is the same as the one of the reference list
## Table 19. Compiled agile practices shared by appearance

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>[A1]</th>
<th>[A2]</th>
<th>[A3]</th>
<th>[A4]</th>
<th>[A5]</th>
<th>[A6]</th>
<th>[A7]</th>
<th>[A8]</th>
<th>[A9]</th>
<th>[A10]</th>
<th>[A11]</th>
<th>[A12]</th>
<th>[A13]</th>
<th>[A14]</th>
<th>[A15]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily standup</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sprint planning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retrospectives</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sprint review</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Short iterations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release planning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planning poker</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Available product</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>owner</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single team</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frequent releases</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Common work area</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product roadmapping</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Story mapping</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agile portfolio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>planning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agile UX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 2.9.2 Final consolidated list

After this consolidation, it was obtained the following list of 15 agile practices. Figure 2 presents the reference list with the names as are presented on [A3].
Figure 2. Final consolidated agile practices list
2.10 Agile practices definitions

In this section we provide a quick reference about the description of the listed practices (Table 20). On the first column is situated the practice name, the second column contains a brief description of the practice, and the third column contains the references where the agile practices appeared (in bold the references from which the description was obtained).

Table 20. Description of agile practices quick reference

<table>
<thead>
<tr>
<th>Agile practice</th>
<th>Description</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily standup</td>
<td>Daily Stand-ups is one of the most common practices followed by agile teams. Is a meeting that takes place every day, preferably face-to-face, to report updates on work done.</td>
<td>[A1], [A2], [A3], [A4]–[A9], [A12], [A13], [A15]</td>
</tr>
<tr>
<td>Sprint planning</td>
<td>Sprint planning sessions held at the beginning of the sprint, are intended to review and analyze possible changes and to define the sprint backlog, with the tasks to be completed in the sprint.</td>
<td>[A1]–[A4], [A5], [A6]–[A8], [A11], [A13], [A14], [A12]</td>
</tr>
<tr>
<td>Retrospectives</td>
<td>Retrospectives are the meetings held after the completion of an iteration to suggest process improvements for the following sprint.</td>
<td>[A3], [A4], [A5], [A14], [A6]–[A9], [A12]–[A15]</td>
</tr>
<tr>
<td>Sprint review</td>
<td>Sprint review is a meeting where the team usually presents the sprint work to the product owner for their feedback.</td>
<td>[A1]–[A3], [A5]–[A10], [A11], [A12], [A14], [A15]</td>
</tr>
<tr>
<td>Short iterations</td>
<td>Short iterations typically seek to maintain a consistent sprint length. Its implementation behaves as a support for agile team accuracy while calculating the amount of work that they would be able to perform.</td>
<td>[A1], [A2], [A3], [A4], [A6]–[A9], [A12], [A13]</td>
</tr>
<tr>
<td>Release planning</td>
<td>Release planning is performed as an independent session for planning every release. It purpose is to estimate which features will be delivered in the established release deadlines.</td>
<td>[A1], [A3], [A4], [A5], [A6], [A9], [A12], [A13]</td>
</tr>
<tr>
<td>Planning poker</td>
<td>Planning poker is shaped by numbered cards with Fibonacci series. It is used by the agile team as a</td>
<td>[A1], [A3]–[A5], [A7], [A9], [A12]</td>
</tr>
</tbody>
</table>
## Agile Practices as Solutions for Software Project Management Anti-patterns

**Master Thesis**

<table>
<thead>
<tr>
<th>Practice</th>
<th>Description</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Available product owner</strong></td>
<td>A product owner is responsible for managing the customer requirements, transforming them on backlog items and prioritizing it for then communicate them to the agile team.</td>
<td>[A1]-[A4], [A10]-[A12], [A14], [10]</td>
</tr>
<tr>
<td><strong>Single team</strong></td>
<td>Single team is a practice in which theoretically, every team member is open to take up any work irrespective of their skill set.</td>
<td>[A2]-[A4], [A6], [A8]-[A10], [A12], [A14]</td>
</tr>
<tr>
<td><strong>Frequent releases</strong></td>
<td>Frequent releases allow the team to see the state of the system and track if the deadlines are being meet.</td>
<td>[A2], [A4], [A8], [A10], [A12]</td>
</tr>
<tr>
<td><strong>Common work area</strong></td>
<td>Common working area supports frequent meetings, that lead to informal communication among stakeholders, which aids to the constant evolution of the requirements</td>
<td>[A1], [A3], [A4], [A8], [A9], [A11]-[A14], [A14]</td>
</tr>
<tr>
<td><strong>Product roadmapping</strong></td>
<td>Planning the roadmap of a product is part of the practices that help to have a general overview of what is expected from the product and how the system is being constructed based on the features.</td>
<td>[A2], [A4], [A5], [A7], [A9], [A10], [A12]</td>
</tr>
<tr>
<td><strong>Story mapping</strong></td>
<td>Story mapping is a technique to organize the product backlog by release and functionalities.</td>
<td>[A3]-[A5], [A7]-[A9], [A11]-[A14], [A15]</td>
</tr>
<tr>
<td><strong>Agile portfolio planning</strong></td>
<td>An agile portfolio planning is a support to keep track of the activities that are in process. It shows the status of each of the tasks that were planned for the sprint.</td>
<td>[A3]-[A6], [A8]</td>
</tr>
<tr>
<td><strong>Agile UX</strong></td>
<td>Agile UX seeks to integrate the user experience practices into the agile development cycle. Performing corporate design standards and getting feedback from clients to refine the requirements and prototypes.</td>
<td>[A1], [A3], [A5], [A7]-[A9], [A11], [A12], [A14], [9]</td>
</tr>
</tbody>
</table>

Continuing with the definitions, we will present each agile practice name, a deep definition and other possible names on which could be found on bibliography.

### 2.10.1 Daily standup

**Agile practice name:** Daily standup
Optional name: Daily standup meeting, Scrum meeting, standup meeting

Definition: One of the most common agile practices followed by agile teams. Is a meeting that takes place every day, preferably face-to-face and should not take more than fifteen minutes. Highly recommended to conduct it when workday starts. Its main purpose is to report updates on work done, the activities that will be performed during the day and if there exist any impediment executing them [A5].

2.10.2 Sprint/ iteration planning
Agile practice name: Sprint/ iteration planning

Optional name: Iteration task list, Planning meeting, Continuous planning,

Definition: Agile development implies continuous planning, where the team should be prepared for any upcoming change from the customer. Sprint planning sessions are intended to review and analyze these possible changes and define sprint backlog, selecting tasks that the team consider they will be able to complete during the following sprint. This meeting is held at the beginning of the sprint [A5],[A14], [12].

2.10.3 Retrospectives
Agile practice name: Retrospectives

Optional name:

Definition: Generally, retrospectives are meetings held after each sprint completion. Often review of how the sprint went regarding the process, people and tools. During them it is identified what was well employed, executed and potential improvements in order to upgrade the activities within the team and the process. Finally, it is created a plan to implement those improvements [A5],[A14].

2.10.4 Sprint review
Agile practice name: Sprint/ Iteration review

Optional name: Outcome review, review meeting, demo, working demoable software

Definition: As the name implies, sprint review is a meeting oriented to present the sprint work done to the product owner to get his feedback. It also works to review if the planned work for that sprint were completed, how good is the backlog being accomplished and for planning team decisions [A5], [A10], [A14].

2.10.5 Short iterations
Agile practice name: Short iterations

Optional name: Incremental delivery, Time boxing, Iterations, small increments, sprint length and weight

Definition: Short iterations has been transformed into one of the most important agile practices. Handling it, provides advantages on agile schemes. Its implementation behaves as a support for the agile team accuracy, while calculating the amount of work that they would be able to perform in a small amount of time.
Besides, the system should be easier to test and correct any failure that might appear since the product of each iteration is smaller and easier to manipulate. In addition, possible changes would affect gradually the development process instead of flipping the complete path as it might be in a traditional methodology project [A2], [A6].

2.10.6 Release planning  
**Agile practice name:** Release planning  

**Optional name:**

**Definition:** This activity is performed by the agile team as an independent session. It seeks to estimate which features will be delivered in the established release deadlines (in case they should be fixed). These estimations are based on prioritized features and depend on teams velocity. Concretely, the plan represents the scope of what the team expect to deliver by a given deadline [A6], [A3].

2.10.7 Planning poker  
**Agile practice name:** Planning poker/ Estimation team  

**Optional name:** Agile games, Planning game, Team based estimation, estimation meeting  

**Definition:** We can see that many practices are close related between them, one of those are the techniques and tools recommended by agile to estimate the necessary effort by the team to execute any of the tasks of each sprint.

The most found on literature is the planning poker. Basically, is shaped by numbered cards with Fibonacci series, each team member should have them (could be physically or any electronic application).

Previous starting with the estimation, the team members will select the activity that they consider the most difficult and the easiest one to perform, assigning the highest and lowest value to each. Subsequently, they rate sprint backlog items based on their experience [A1].

2.10.8 Available Product owner  
**Agile practice name:** Available Product owner  

**Optional name:** Dedicated customer, Product vision  

**Definition:** A product owner is a role that forms part of the activities performed in agile environments. The responsible for this action should manage customer requirements, transforming them on backlog items and prioritizing it for then communicate them to the agile team. Besides, the product owner should have at least a general overview and technical knowledge about each system under development on their projects [10].

2.10.9 Single team  
**Agile practice name:** Single team  

**Optional name:** Cross-functional teams; Integrated Dev/QA, Cross functionality, Close collaboration, Small team, whole multidisciplinary team
Definition: In theory, agile supports the concept of a working team open to take up any activity disregarding their skillset. Also known as cross-functionality the team members should be able to perform any task, with the main purpose of successfully accomplish the estimated work planned for the sprint [A6].

2.10.10 Frequent releases
Agile practice name: Frequent releases

Optional name: Frequent builds, frequent release of working software, continuous builds

Definition: Agile promotes the constant communication and status reviewing of the project in order to know how it is proceeding. Frequent releases allow the team to see the state of the system and track if the deadlines are being meet. On the other side, development team can communicate any possible concern about following deliveries. Finally, enable stakeholders to keep monitoring if the progress of the project is correct [A10].

2.10.11 Common work area
Agile practice name: Common working area

Optional name: Open work area, sit together

Definition: To maintain the agility of a project, one crucial practice is the constant communication. Keep all the team together and having an open work area is one typical solution. Thereby, the team can solve easily any doubt or issue on a development process, avoiding unnecessary waiting time [A14].

2.10.12 Product roadmapping
Agile practice name: Product roadmapping

Optional name: Progress monitoring, Requirements workshop, Feature growth

Definition: Planning the roadmap of a product is a practice that helps to have a general overview of what is expected and how the system is being constructed. This activity should be based on the features and not in size, the creation of the roadmap must consider the most important functionalities that will shape the product [A10].

2.10.13 Story mapping
Agile practice name: Story mapping

Optional name: User stories, Priorization of user stories

Definition: Story mapping is a technique to organize the product backlog by release and functionalities. Therefore, it is necessary to write user stories that are the agile format for the specification of the customer requirements. This facilitates stakeholders communication and reduces the need for documentation as in traditional project development since the user stories are short and simple [A9], [A14].

2.10.14 Agile portfolio planning
Agile practice name: Agile portfolio planning
Optional name: Task board, Scrum board, Scrum wall

Definition: Nowadays it exists many different options to implement this kind of tool, it can be created as a conventional physical scrum board or is possible to find online tools. They work as a support to keep track on the activities that are in process. Basically, it shows the status of each of the tasks that were planned for the sprint [A6].

2.10.15 Agile UX
Agile practice name: Agile UX

Optional name: Evolutionary design, design reviews, End-user-focused design, Prototyping, Emergent design

Definition: This practice seeks to integrate the user experience practices into the agile development cycle. What it intends, is to have constant feedback from the customer through reviewing the requirements and interfaces, evaluate if it works as expected to finally, perform the necessary changes to improve the final design [A11], [9].
Chapter 3:

Anti-patterns in Software Project Management Research
3. Anti-patterns in Software Project Management Research

3.1 Introduction
This chapter contains the explanation of the process executed for the second systematic mapping related to the identification of software project management anti-patterns.

During this search we seek for papers dealing with project management anti-patterns that present a concrete lists of them. In the following sections we will explain how this process was performed and the obtained results.

3.2 Research questions
For performing the search of the related papers to this systematic mapping it was necessary to establish a research question. It was thought with the main goal of identifying which are the most relevant software project management anti-patterns in industry. We focused the research on the technological and development industry, where might be present this malpractices.

RQ2. What are the existing software project management anti-patterns?

3.3 Search process
Figure 4 represents the search process executed. Due to the process that we were following for this project, the research was focused on the obtention of papers were they already performed a systematic review presenting consolidated lists of anti-patterns.

According on the research question for this part of the work (RQ2) we defined for the search the key concepts: Anti-pattern, project management, systematic review and systematic mapping. With the different combinations between them it was created the following search string: (anti-pattern OR antipattern OR mal practice OR bad practice) AND (project management OR software project management OR Management) AND (systematic mapping OR mapping study OR systematic review OR systematic decomposition OR systematic literature review).

The same electronic databases that store publications from official journals and conferences were employed:

- Science Direct: [https://www.sciencedirect.com/](https://www.sciencedirect.com/)
- Institute of Electrical and Electronics Engineers (IEEEXplore): [https://ieeexplore.ieee.org](https://ieeexplore.ieee.org)
- Association for Computing Machinery (ACM): [https://www.acm.org/](https://www.acm.org/)
- Google scholar: [https://scholar.google.es/](https://scholar.google.es/)
Table 39 presents the reviewed databases and the total amount of selected papers. Unfortunately, after a deep search it was not possible to obtain more papers that were relevant for our study, two papers were chosen for this systematic mapping.
Table 21. Summary of search results

<table>
<thead>
<tr>
<th>Database</th>
<th>Search results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Science direct</td>
<td>0</td>
</tr>
<tr>
<td>Research gate</td>
<td>1</td>
</tr>
<tr>
<td>IEEEXplore</td>
<td>1</td>
</tr>
<tr>
<td>ACM</td>
<td>0</td>
</tr>
<tr>
<td>Google scholar</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>2</strong></td>
</tr>
<tr>
<td><strong>Total without duplicates</strong></td>
<td><strong>2</strong></td>
</tr>
</tbody>
</table>

### 3.4 Study selection process

We decided to follow the same selection process as the previously executed in the previous chapter Agile Practices Research (Chapter 2). It was considered for the selection: where the study was performed (enterprise, university, etc.), the major topic explained on the abstract and if it contained a concrete list of project management anti-patterns.

After executing the search on all the selected digital databases, unfortunately, we were not able to find more than two documents that were helpful for the project. One of those documents was [B1] and the second paper found was [B2].

After conducting a deep revision of both articles, it was identified that the second paper was already referenced in the first one. In order to avoid duplicated information and due to the first publication was the most up to date it was decided to base the research just on it.

The selected publication is a 2014 journal paper that explains the work done on the thesis “Categorization of software project management anti-patterns” [2]. On this publication is explained how was performed a review of other articles and books dealing with software anti-patterns and how they perform the analysis to create a consolidated list of them.

After evaluating the overview of the situation and the lack of new studies dealing with this topic, we took the final decision of using the consolidated list that was developed on the mentioned thesis and is referenced on the publication.

### 3.5 Data extraction process

Given that the selected work is a systematic mapping procedure for anti-patterns, it already has an official consolidated list. The data extraction process was performed in a different way as the previous list. Therefore, was necessary to identify which sections of the document were relevant and useful for the project to perform the data extraction.

For this case we did not create a data extraction form, we focused the extraction work on identifying what was going to be helpful for our study. Finally, it was selected to include on the thesis the anti-pattern list categorized per name and per activity.

### 3.6 Analysis process

Even having a consolidated list on [2], it was necessary to analyze the document and identify which information was going to be taken and displayed on this study.
We identified as relevant information the consolidated list of anti-patterns and the anti-pattern software project management categorized per activity. The extracted data are: a simplified list of anti-patterns with short definitions, the list with extended definitions included on the annex C, the list categorized per activity and the activities definitions.

We considered that all this information together would be helpful to understand and identify how anti-patterns work, to finally, propose a possible solution to avoid or reduce their impact on the industry.

3.7 **Consolidated list of anti-patterns in software project management**

This section will present the most up to date list related to anti-patterns in software project management. In order to have an autonomous work, we will include the definition of each of them in a summarized format and it will be possible to find on the annex C an extended version of them.

It is important to mention that some of the information presented on this part of the project was taken directly from our main source. We decided to maintain the definitions and present the data the most possible attached to [2] to exhibit accurately the information.

3.7.1 **Consolidated list**

On this section it is possible to observe the consolidated list (Table 40) of project management anti-patterns. For the purposes of this work it was decided to only show the final consolidated list, any other deeper information can be found on the references.

<table>
<thead>
<tr>
<th><strong>Anti-pattern Name</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Absentee manager</td>
</tr>
<tr>
<td>All you have is hammer</td>
</tr>
<tr>
<td>Appointed team</td>
</tr>
<tr>
<td>Detailitis Plan</td>
</tr>
<tr>
<td>Dry waterhole</td>
</tr>
<tr>
<td>Fire drill</td>
</tr>
<tr>
<td>Glass case plan</td>
</tr>
<tr>
<td>Inflexible plan</td>
</tr>
<tr>
<td>Irrational management</td>
</tr>
<tr>
<td>Leader not manager</td>
</tr>
<tr>
<td>Micro-management</td>
</tr>
<tr>
<td>Mushroom management</td>
</tr>
<tr>
<td>Myopic Delivery</td>
</tr>
<tr>
<td>Process disintegration</td>
</tr>
<tr>
<td>Project mismanagement</td>
</tr>
<tr>
<td>Proletariat hero</td>
</tr>
<tr>
<td>Rising upstart</td>
</tr>
<tr>
<td>Road to nowhere</td>
</tr>
<tr>
<td>Size isn’t everything</td>
</tr>
<tr>
<td>The brawl</td>
</tr>
<tr>
<td>The domino effect</td>
</tr>
</tbody>
</table>
3.7.2 Anti-pattern software project management definitions

The following table (Table 41) was taken directly from [2]. On the first column are listed the anti-pattern names and in the second is possible to find a short description for each of them, the extended definitions are available on the Annex C.

Table 23. Description of anti-patterns quick reference [2]

<table>
<thead>
<tr>
<th>Anti-pattern Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absentee manager</td>
<td>Manager who engages in avoidance behavior or is invisible for long periods of time.</td>
</tr>
<tr>
<td>All you have is hammer</td>
<td>One-dimensional management, where the same techniques are used on all subordinates and in all situations.</td>
</tr>
<tr>
<td>Appointed team</td>
<td>False assumption that a group of people selected by management will immediately gel and become a team.</td>
</tr>
<tr>
<td>Detailitis Plan</td>
<td>Excessive planning leading to complex schedules with high level of detail, giving the false perception that the project is fully under control.</td>
</tr>
<tr>
<td>Dry waterhole</td>
<td>Specifying stringent requirements for a job when it is not strictly necessary, resulting in a very limited pool of available talent.</td>
</tr>
<tr>
<td>Fire drill</td>
<td>Months of boredom followed by demands for immediate delivery.</td>
</tr>
<tr>
<td>Glass case plan</td>
<td>Lack of tracking and updating of initial plans, assuming the plan is enough.</td>
</tr>
<tr>
<td>Inflexible plan</td>
<td>Lack of flexible plans and processes.</td>
</tr>
<tr>
<td>Irrational management</td>
<td>Irrational management decisions, habitual indecisiveness and other negative management practices.</td>
</tr>
<tr>
<td>Leader not manager</td>
<td>Manager with a vision (leader) but no plan or management methodology.</td>
</tr>
<tr>
<td>Micro-management</td>
<td>Excessive management involvement in tasks beyond their responsibility.</td>
</tr>
<tr>
<td>Mushroom management</td>
<td>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.</td>
</tr>
<tr>
<td>Myopic Delivery</td>
<td>Management insisting on original delivery date even when reducing staff or funding.</td>
</tr>
<tr>
<td>Process disintegration</td>
<td>Failing processes due to an underlying decline in overall cooperation and morale.</td>
</tr>
<tr>
<td>Project mismanagement</td>
<td>Lack of proper software project monitoring and controlling.</td>
</tr>
<tr>
<td>Proletariat hero</td>
<td>False assumption that coercion is an efficient way to increase productivity.</td>
</tr>
<tr>
<td>Rising upstart</td>
<td>Superstars who cannot wait their time and want to skip learning phases.</td>
</tr>
<tr>
<td>Road to nowhere</td>
<td>Lack of planning.</td>
</tr>
<tr>
<td>Size isn’t everything</td>
<td>Assuming developers are interchangeable and that the number of people working on a problem is inversely proportional to development time.</td>
</tr>
<tr>
<td>The brawl</td>
<td>Project manager with no leadership or management experience.</td>
</tr>
<tr>
<td>The domino effect</td>
<td>Moving critical resources between projects, blurring project boundaries.</td>
</tr>
<tr>
<td>Ultimate weapon</td>
<td>Relying heavily on a superstar in the team.</td>
</tr>
</tbody>
</table>

3.7.3 Anti-pattern software project management categorization per activity

As a matters of this study we identified some information presented on [2] that would be helpful for the final systematic mapping. The agile practices include the performance of some activities that can be related to some of the software project management activities.
Part of the activities performed on the agile processes can be matched using the following classification enabling us to verify if the impacted activities by the anti-patterns listed on table 42, are being correctly related to agile practices.

Table 24. Anti-patterns categorized by impacted activity [2]

<table>
<thead>
<tr>
<th>Anti-pattern Name</th>
<th>Impact Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fire Drill</td>
<td>Controlling</td>
</tr>
<tr>
<td>Glass Case Plan</td>
<td>Controlling</td>
</tr>
<tr>
<td>Mushroom Management</td>
<td>Controlling</td>
</tr>
<tr>
<td>Myopic Delivery</td>
<td>Controlling</td>
</tr>
<tr>
<td>Project Mismanagement</td>
<td>Controlling</td>
</tr>
<tr>
<td>The Domino Effect</td>
<td>Controlling</td>
</tr>
<tr>
<td>Absentee Manager</td>
<td>Controlling, Motivating</td>
</tr>
<tr>
<td>Irrational Management</td>
<td>Controlling, Motivating</td>
</tr>
<tr>
<td>The Brawl</td>
<td>Controlling, Motivating</td>
</tr>
<tr>
<td>All You Have Is a Hammer</td>
<td>Motivating</td>
</tr>
<tr>
<td>Micro-Management</td>
<td>Motivating</td>
</tr>
<tr>
<td>Process Disintegration</td>
<td>Motivating</td>
</tr>
<tr>
<td>Proletariat Hero</td>
<td>Motivating</td>
</tr>
<tr>
<td>Rising Upstart</td>
<td>Motivating</td>
</tr>
<tr>
<td>Ultimate Weapon</td>
<td>Motivating</td>
</tr>
<tr>
<td>Inflexible Plan</td>
<td>Planning</td>
</tr>
<tr>
<td>Leader Not Manager</td>
<td>Planning</td>
</tr>
<tr>
<td>Road to Nowhere</td>
<td>Planning</td>
</tr>
<tr>
<td>Detailitis Plan</td>
<td>Scheduling</td>
</tr>
<tr>
<td>Size Isn't Everything</td>
<td>Scheduling</td>
</tr>
<tr>
<td>Appointed Team</td>
<td>Staffing</td>
</tr>
<tr>
<td>Dry Waterhole</td>
<td>Staffing</td>
</tr>
</tbody>
</table>

We will add the definition of each of those activities to any necessary lookup and better understanding of the activities.

- **Controlling**: An activity which is both passive, monitoring using Earned Value Management, and active, re-planning to respond to some unexpected event, such as tasks running behind schedule, over budget, or changing requirements.

- **Motivating**: Creating and maintaining a collegial environment where teams participate in the creation of their schedules and goals and are efficiently led and managed by a "servant leader", who relies on his/her strengths, selects a complimentary team, and negotiates with teams, other leaders, and managers to achieve challenges.

- **Planning**: A proactive activity, which involves creating a project plan that includes, as a minimum, a business case, a cost/benefit analysis, a risk analysis, and task delineation, including estimated resources and schedule.

- **Scheduling**: The creation of the project schedule, which can be described as a list of events where each event has a calendar date for starting and finishing. The schedule includes resource and time allocation for each task.
• **Staffing:** Consists of recruiting the correct people while keeping teams small, reviewing and evaluating team effectiveness and productivity and removing roadblocks when needed, as well as being aware of team conduct and disruption
Figure 4. Anti-patterns categorized by impacted activity
Chapter 4:

Agile Practices as a Solution to Software Project Management Anti-patterns
4. Agile Practices as a Solution to Project Management Anti-patterns

4.1 Introduction
In this section, we present the matching process that was executed in order to study how the agile practices can contribute to address the different software project management anti-patterns. After performing the systematic mapping of the agile practices and the project management anti-patterns, we matched both lists by reviewing each anti-pattern and identifying the possible impact of the agile practice on it.

It was necessary to establish a matching criteria to classify the impact that each agile practice would have on each malpractice, and determine the profile based on the activities that the Project Manager should execute in agile. In the subsequent points, the matching process that was followed will be explained.

4.2 Matching process
Previous to perform the matching process, we thought about the endless different situations within an organization and how each agile practice would impact the anti-patterns based on those possibilities. Therefore, it was necessary to set up some strategic comparison points and delimit to the most simple and probable scenarios.

Additionally, it was necessary to define the concept of each impact that might have an agile practice over the management malpractices. The fact of having a proposal for a solution, in this case the agile practice, does not mean that each practice will fit and solve the problems derived from the anti-patterns. Some practices might contribute to solve the problems generated by an anti-pattern, others might only help to identify them or even have no direct impact on them.

It is also important to remark that they can exist cases where one agile practice might contribute solving derived issues from malpractices. This does not mean that the selected agile technique will solve all the problems within a project, sometimes will be required the implementation of other agile techniques, that all together will support the daily activities and, consequently contribute to reduce or solve the impact of the problems.

Furthermore, it was necessary to determine the profile that the representative of the Project Manager should accomplish in agile. On agile schemes there is no official PM role, therefore, it was necessary to analyze which management activities are performed by the representatives of the team inside an agile scheme and define a profile which matches as much as possible with a traditional PM role.

The detail of the matching criteria and the establishment of the Project Manager profile are explained in the following subsections 6.2.1 and 6.2.2, respectively.

Once we had established these important parameters, we proceed with the matching process where we evaluate the impact that each agile practice would have over the anti-patterns. Based on the established impacts we select which was more compatible with each malpractice.
The matching process was executed over a total of 23 anti-patterns matched with 15 agile practices. Based on the established impacts for each case, we explain in detail the selected result for better understandability of the selected matching option.

These explanations of each agile practice impacting the anti-patterns are based on the definitions found in the literature. During the document analysis, we extracted formal explanations from research documents, and we based our description of each case from them. Even when we used official surveys and reports from organizations, we preferred to keep the formality of this project using the scientific project definitions.

On the following section, it is found the matching lists of agile practices vs. anti-patterns and the explanation for each of them.

4.2.1 Matching criteria
Nowadays most of the organizations are interested in implementing agile schemes for their development processes, however, there exist an endless world of possibilities where these practices can work as they wished or not function as expected.

Each software enterprise is different, the working environment can have many different scenarios and issues within the organization. The problems can arise depending on their employee's diversity, the projects, clients, among other possible situations.

Due to this endless world of possibilities, it was necessary to establish matching criteria for evaluating each anti-pattern and give them a solution, if possible, using agile practices.

First, it is important to mention that sometimes when enterprises are applying a specific agile methodology, most of the times they are implementing the agile practices that fit better in their organization.

Each practice can be implemented in different ways and they are adaptable depending on the organization needs. For this research, the matching process was executed taking the definitions on the literature and explained from the point of view of each author (Chapter 2), but each organization is free to implement and adapt this agile practices, based on the situations they are facing.

Second, we also consider that, as one practice can directly contribute to solve an issue, it might exist the possibility where only one practice will not help to avoid the consequences of an anti-pattern. Sometimes, the implementation of more than one agile practice will be needed, to fix the problem working altogether.

Third, during the evaluation of each agile practice over the anti-patterns, we noticed that each agile practice might have a different impact on them. We classify those possibilities as follows:

- An agile practice that contributes to solving the anti-pattern
- An agile practice that does not contributes directly to solve the anti-pattern
- An agile practice that contributes to identify the anti-pattern or problems derived from it, but not directly to solve it
Agile practices that are special cases, these are cases where the implementation of the agile practice can collide with other activities within the processes aggravating the anti-pattern.

Fourth, on agile schemes does not exist a specific Project Manager profile. For this reason, we perform a research (Section 6.2.2) to establish a profile that should be fulfilled by the person representative of the PM on agile schemes.

Every time that we mention the Project Manager on this matching process, we are referring to this representative, taking into account that this profile should have the responsibility of performing the management activities on the agile scheme.

4.2.2 Agile Project Manager Profile

For the needs of this project, it was necessary to research how would be a Project Manager profile on the agile scheme due to this position does not explicitly exist as in traditional schemes. The research was executed in the Project Management Institute website: https://www.pmi.org

On the website information it is explained the equivalence of the traditional PM and the roles that a PM can take on the agile world [12]. They split the profiles in three different: strategic, leading and lagging PM.

These profiles are separated at the same time in different roles, that are assigned to the PM depending on their skills or expertise. Based on it a PM can be working as a financial advisor, training engineer, release manager, product owner or scrum master.

After analyzing this information, we extracted the main characteristics and the management activities performed during a development process, to establish an agile PM profile. Below, we include the characteristics and activities that each profile should have, further information can be found on the literature [12].

Traditional PM

As it is defined by the PMI, a traditional PM leads projects during a five-phase delivery life cycle that includes: initiating, planning, executing, monitoring and controlling, and closing.

During these stages they need to perform activities such as identify stakeholders, create a project chart, establish the scope of the project, refine objectives, define corrective actions, maintain objectives, verify if the work defined in the plan is being completed, track and review the activities, regulate the project progress and performance and formally close the project or phase.

A PM should be prepared and have key characteristics that includes leadership, team building, motivation, communication, influencing, decision making, political and cultural awareness, negotiation, trust building, conflict management and coaching.

Agile PM profiles

On agile projects the PM shift from leading projects to the role of a servant leader. Therefore, focuses on serving the team members to remove impediments that block the product release to market.

On agile schemes the role of the PM is split into three different profiles:
- Strategic PM: Supports the organization’s strategic objectives.
- Leading PM: Identifies ways to minimize impediments.
- Lagging PM: Provides answers to past team behaviors.

Each of these profiles performs different activities and can take different roles in agile based on their expertise and skills.

**Strategic PM [12]:**

This project manager provides information that enables the business leaders to make decisions to select the appropriate business to fund. This profile is more focused on budget management; however, it performs some activities that might be of interest to our research.

The strategic PM performs activities such as:

- Provides agile coaching
- Enable future innovations
- Facilitates organization alignment
- Collaborates with portfolio management and business leaders
- Evaluates strategic insights implementation
- Measures the “voice of the customer” deliverables
- Tracks implementation portfolio performance metrics

**Leading PM [12]:**

Is responsible for the risk management, anticipates and prevent issues that are potential impediments for the team. Is key leading the organizational change management that enables cross-functional teams to adapt to the changes in the working environment.

The leading PM can be in charge of different roles within the agile projects, we list here those roles but for the interest of this project, the ones that we took to be evaluated in our agile PM profile are the Product Owner and Scrum Master activities.

**Release train engineer:**

- Facilitates processes and programs for the agile release train by escalating impediments
- Managing risks
- Coaching continual improvement for maximum outcome

**Release manager:**

- Accountable for product release market

**Product owner:**

- Accountable for prioritized work for the team

**Scrum master:**
Agile Practices as Solutions for Software Project Management Anti-patterns

Master Thesis

- Accountable for team agile practices
- Enables the transition from non-agile practices to agile thinking

Lagging PM [12]:

The lagging PM monitors and communicates learning from the execution team’s behaviors. He/She implements corrective actions from the gathered information and promotes continual improvement within the team.

This type of PM can perform activities from two different roles within an organization with an agile scheme:

Product owner:
- Validates the completed and prioritized work
- Works with the product management team to plan releases
- Ensures that the team pursues a common vision
- Drives business value through a prioritized backlog
- Defines and accepts user stories
- Represents the customer to answer questions by the product development team

Scrum master:
- Measures velocity, quality and continual improvement
- Tracks dependency, risks and integration issues
- Removes impediments
- Facilitates team meetings
- Coaches and facilitates agile behavior and practice
- Facilitates integration meetings that manage dependencies and risks
- Enables the transition from non-agile to agile thinking

After obtaining and reviewing the previous information, we consider that those activities performed by the Scrum Master role on leading and lagging profiles were the most suitable for the agile PM profile.

Even when we selected the activities performed by the Scrum Master, we decided to keep the term Project Manager “PM” for the definitions and explanations within the document. The previous decision was taken in order to focus on the concept of the agile practices and avoid as much as possible to terms used on any specific agile methodologies.

Thus, the Project Manager profile that we establish and that will be the one used for the explanations along the matching process should perform the following activities:

- Accountable for team agile practices
- Enables the transition from non-agile practices to agile thinking
- Measures velocity, quality and continual improvement
- Tracks dependency, risks and integration issues
• Removes impediments
• Facilitates team meetings
• Coaches and facilitates agile behavior and practice
• Facilitates integration meetings that manage dependencies and risks
• Enables the transition from non-agile to agile thinking.

Finally, we can conclude that on agile, the activities are split in different resources, the team supports the PM and takes more responsibility for its execution and improvements. Besides, the PM does not need to perform some activities that are part of the traditional project development because on agile are treated as unnecessary. Based on this, the established agile PM profile can work as the responsible of performing the management activities equivalents to the PM on traditional projects, but not more beyond on agile schemes.

4.3 Lists of agile practices impacting Software Project Management Anti-patterns

On this section is going to be presented the consolidated list for each agile practice as a solution for the software project management anti-patterns. Each agile practice will have a table where it will be possible to have a general overview of what is its relation to the anti-patterns. The detailed explanation of the impact that the agile practice has with the anti-patterns will be included after each table.

A specific code has been used to identify the relationship of each agile practice with the anti-patterns. The different options have been previously explained on section 6.2.1 Matching criteria.

☑ The agile practice contributes to solve the anti-pattern.
☒ The agile practice does not contributes directly to solve the anti-pattern.
☒ The agile practice contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.
⚠ Especial cases.

4.3.1 Daily standup impact

In this section it will be presented the matching results between the agile practice Daily standup and the anti-patterns. On table 25, can be observed impact of the agile practice over the malpractices and subsequently the explanation of each selected impact.

Table 25. Daily standup – Project management anti-patterns

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>Project Management Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily standup</td>
<td>Absentee manager</td>
<td>☑</td>
</tr>
<tr>
<td></td>
<td>All you have is a hammer</td>
<td>☒</td>
</tr>
<tr>
<td></td>
<td>Appointed team</td>
<td>☑</td>
</tr>
<tr>
<td></td>
<td>Detailitis plan</td>
<td>☒</td>
</tr>
<tr>
<td></td>
<td>Dry waterhole</td>
<td>☒</td>
</tr>
<tr>
<td></td>
<td>Fire drill</td>
<td>☑</td>
</tr>
<tr>
<td></td>
<td>Glass case plan</td>
<td>☑</td>
</tr>
<tr>
<td></td>
<td>Inflexible plan</td>
<td>☑</td>
</tr>
</tbody>
</table>
4.3.1.1 Absentee manager
Agile Practice: Daily standup

Impact: Contributes to solve the anti-pattern

Detail: Stand-up meetings would be a useful technique to avoid an absentee manager. When subordinates do not have a primary decision maker, they are forced to make crucial decisions or delay them. On these meetings the development team members and the PM should be present, they talk about which activities are performed and if there exist any impediment executing it.

4.3.1.2 All you have is a hammer
Agile Practice: Daily standup

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Daily stand-ups practice help to identify if this anti-pattern is happening. If the project manager is trying to impose the same techniques to solve the issues within a project, it might not work for all of them and the problems would be persisting. The PM might be able to identify the source problem that is the anti-pattern and not the project per-se, for then try to find a solution.

4.3.1.3 Appointed team
Agile Practice: Daily standup

Impact: Contributes to solve the anti-pattern

Detail: Not always the people on a team can immediately start working as is expected, there are no perfect criteria for screening team members. Daily meetings can help to the team members to identify the way of each person works, how fast and easy is for them to execute their tasks and identify how they can be helped to finally, gel and improve to start feeling the sensation that they are in a team.
4.1.1.1 Detailitis plan
Agile Practice: Daily standup
Impact: Does not contributes directly to solve the anti-pattern
Detail: Daily standup does not have a direct relation to this anti-pattern, due to on this meetings are not performed any planification activity.

4.1.1.2 Dry waterhole
Agile Practice: Daily standup
Impact: Does not contributes directly to solve the anti-pattern
Detail: Establishing stringent requirements for team members selections might keep aside very talented people who can be prepared and developed for the future. Daily standup is not direct solution for this anti-pattern, due to on it there are any team selection activity.

4.1.1.3 Fire drill
Agile Practice: Daily standup
Impact: Contributes to solve the anti-pattern
Detail: The advantage of having daily meetings is that the PM can observe the progress of the project every day. Having a daily status help to avoid idle times and enormous amounts of accumulated work to deliver at the last minute.

4.1.1.4 Glass case plan
Agile Practice: Daily standup
Impact: Contributes to solve the anti-pattern
Detail: The main purpose of daily meetings is to have an overview of how the project is being executed and to keep track on the activities. This give the advantage to identify if it is necessary to do any modification on the initial sprint plan before any possible issue arise.

4.1.1.5 Inflexible plan
Agile Practice: Daily standup
Impact: Contributes to solve the anti-pattern
Detail: Agile is oriented to be open to flexibility and changes during the development process. Daily meetings are meant to permit the team to know the status of the project and if something is not moving in the correct way in the sprint, react before relevant consequences arise.

4.3.1.4 Irrational management
Agile Practice: Daily standup
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Daily meetings are meant to know the daily status of the product by the team and the PM. Due to it is mandatory for all the involved people to express their own status about the tasks that they are performing, it could help to identify if the PM is being indecisive or his/her decisions are negative for the path of the project.

4.3.1.5 Leader not manager
Agile Practice: Daily standup
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Implementing daily meetings can help to identify if there exist a lack of a management methodology, it would be more evident if not exist a control or a right planification to be followed on the day by day.

4.3.1.6 Micro-management
Agile Practice: Daily standup
Impact: Does not contributes directly to solve the anti-pattern

Detail: Daily standup does not have a direct impact on PM over-managing in any other aspects of the project. This meeting is exclusive for updating the team about the status of daily activities.

4.3.1.7 Mushroom management
Agile Practice: Daily standup
Impact: Does not contributes directly to solve the anti-pattern

Detail: Daily standup meetings are performed every day with the development team and the PM, the user does not participate in them. This agile practice does not have direct inference solving the lack of communication between the team and the user.

4.3.1.8 Myopic delivery
Agile Practice: Daily standup
Impact: Contributes to solve the anti-pattern

Detail: Having a daily status of the project is a support for the development team and the manager to let know the stakeholders how the activities are being carry out. If there exist any slip on the schedule, the management can take decisions on time and notify them to all the interested people on the project.

4.3.1.9 Process disintegration
Agile Practice: Daily standup
Impact: Does not contributes directly to solve the anti-pattern
Detail: Daily standup meetings can help to identify if this antipattern is present on the project. The overall demotivation can be reflected in this kind of meetings where all the team is present and it would be easier for the PM notice that might something is failing within the process and the team.

4.3.1.10 Project mismanagement
Agile Practice: Daily standup
Impact: Contributes to solve the anti-pattern

Detail: This kind of meetings are oriented to carry out in a simpler way the monitoring and control of the activities executed by the team. The project manager can have an accurate overview of what is happening during the sprint.

4.3.1.11 Proletariat hero
Agile Practice: Daily standup
Impact: Does not contribute directly to solve the anti-pattern

Detail: These meetings are meant to give a daily status about the performed activities, it does not have direct impact avoiding bad practices from management side to increase the productivity.

4.3.1.12 Rising upstart
Agile Practice: Daily standup
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Rising upstarts want to forego the requisite time to learn and show up how supposedly they are the best. Daily stand-ups allow the project manager to identify if there is any person with this profile during the meeting, due to all the team members should have active participation.

4.3.1.13 Road to nowhere
Agile Practice: Daily standup
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Daily standups are a useful tool to identify if there exist a lack of planning for each sprint, if that is the situation the team will not know what they need to execute during the day and what is already done.

4.3.1.14 Size isn’t everything
Agile Practice: Daily standup
Impact: Does not contribute directly to solve the anti-pattern

Detail: During standup sessions, each team member explains their progress on their assigned activities for the sprint. These sessions are not meant to take decisions, discuss arising problems or assigning activities. Standup meetings do not have direct impact solving or identifying the
problem of interchanging team members, having a false idea that development activities will increase this way.

4.3.1.15 The brawl
Agile Practice: Daily standup

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: The teamwork together every day and follow the status of each team member on these meetings; everyone present expects guidance, but if there is a lack of leadership it will be evident and the daily standup would help to identify the lack of leadership or management skills.

4.3.1.16 The domino effect
Agile Practice: Daily standup

Impact: Does not contribute directly to solve the anti-pattern

Detail: During standup sessions, each team member explains their progress on their assigned activities for the sprint. These sessions are not meant to take decisions such as exchanging critical resources between projects. Standup meetings do not have direct impact solving or identifying the problem of interchanging team members, blurring project results.

4.3.1.17 Ultimate weapon
Agile Practice: Daily standup

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Ultimate weapons can do great things within a project; however, this kind of people can fall into the arrogance affecting themselves and the project. Also, team members could feel carefree about the possible problems waiting for this person to solve it. Daily meetings can help to identify persons with this profile or if these situation is happening.

4.3.2 Sprint Planning impact

In this section it will be presented the matching results between the agile practice Sprint planning and the anti-patterns. On table 26, can be observed impact of the agile practice over the malpractices and subsequently the explanation of each selected impact.

Table 26. Sprint planning – Project management anti-patterns

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>Project Management Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint planning</td>
<td>Absentee manager</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>All you have is a hammer</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Appointed team</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Detailitis plan</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Dry waterhole</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Fire drill</td>
<td>✔</td>
</tr>
</tbody>
</table>
### 4.3.2.1 Absentee manager

Agile Practice: Sprint planning

Impact: Contributes to solve the anti-pattern

Detail: This meeting requires the participation of the development team and the PM and maybe other involved people on the project. The active participation of the PM is indispensable due to during this meeting relevant decisions should be taken; even when decisions should not be taken unilaterally by the PM, but all team members should participate.

### 4.3.2.2 All you have is a hammer

Agile Practice: Sprint planning

Impact: Does not contributes directly to solve the anti-pattern

Detail: Sprint planning does not have a direct relation solving this anti-pattern, due to this meeting is meant to plan the sprint activities and not solve problems related with the team management or processes.

### 4.3.2.3 Appointed team

Agile Practice: Sprint planning

Impact: Contributes to solve the anti-pattern

Detail: Sprint planning is an opportunity to permit the team members to have open participation, give their opinion and share their experience. This technique allows to identify the abilities and growth opportunities for each of them and promotes team coordination. Consequently may be easier to integrate themselves, empowering them to meet project goals and become a team.
4.3.2.4 Detailitis plan
Agile Practice: Sprint planning
Impact: Contributes to solve the anti-pattern

Detail: Based on the progress of the previous sprint, the sprint planning allows the team to prepare the tasks that should be executed on the following sprint generating the sprint backlog. This agile practice helps to avoid the generation of a complex schedule due to is only prepared a plan for each sprint instead of a complete project.

4.3.2.5 Dry waterhole
Agile Practice: Sprint planning
Impact: Does not contributes directly to solve the anti-pattern

Detail: During sprint planning meetings it is decided what the team should perform during the sprint. This agile practice does not have direct inference when establishing the selection requirements for the team members that will be working on a project.

4.3.2.6 Fire drill
Agile Practice: Sprint planning
Impact: Contributes to solve the anti-pattern

Detail: Constant planning help to avoid immediate demands pressuring the team executing the activities till the last minute. Sprint planning meetings are the perfect option, to have a periodical planning, selecting the tasks that should be performed during the sprint and avoiding last minute demands.

4.3.2.7 Glass case plan
Agile Practice: Sprint planning
Impact: Contributes to solve the anti-pattern

Detail: The sprint planning helps to include any possible change facing the first previsions about the sprint, backlog or release planning. This practice avoids the situation where the project is not being tracked and the initial release plan is not updated.

4.3.2.8 Inflexible plan
Agile Practice: Sprint planning
Impact: Contributes to solve the anti-pattern

Detail: Sprint planning meetings allow flexibility in terms of the work to be developed in each sprint according to possible changes in the product backlog, as well as changes in the development process, that will be reflected into the sprint backlog.

4.3.2.9 Irrational management
Agile Practice: Sprint planning
Impact: Contributes to solve the anti-pattern
Detail: Sprint plannings are meant to discuss and evaluate what should be done during the next sprint. The decisions should be discussed by the entire development team and the project manager. Sprint planning meetings avoid the possibility of having an irrational decision derived from the manager’s personal priorities, as decisions should be taken collaboratively by the agile team.

4.3.2.10 Leader not manager
Agile Practice: Sprint planning
Impact: Contributes to solve the anti-pattern

Detail: One of the outcomes from the sprint planning meetings is the sprint backlog with the list of tasks and schedule for the user stories to be performed. This enforces the manager and the team to create a plan for each sprint and follow a management methodology.

4.3.2.11 Micro-management
Agile Practice: Sprint planning
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: In the sprint planning is prepared the different tasks to be performed in the iteration as well as the team member’s that will be executing those activities. So, if the project manager takes tasks out of his/her responsibility, this will be reflected in the sprint backlog. This agile practice can help to identify if this is the case and how it might affect the project, for later try to find a solution.

4.3.2.12 Mushroom management
Agile Practice: Sprint planning
Impact: Contributes to solve the anti-pattern

Detail: Sprint planning meetings may require the participation of users as changes in the product backlog can arise. So the interaction between the agile team and the user is promoted by this agile practice.

4.3.2.13 Myopic delivery
Agile Practice: Sprint planning
Impact: Contributes to solve the anti-pattern

Detail: During the sprint planning, the team should decide about the number of tasks that will be performed during the sprint. Original plans are open to changes according to different factors like changing requirements, priorities, previous sprints velocity, etc. This agile practice promotes the open discussion about rescheduling deliveries in favor of the project progress and quality.

4.3.2.14 Process disintegration
Agile Practice: Sprint planning
Impact: Contributes to solve the anti-pattern

Detail: During sprint planning meetings all the team has to cooperate and participate to be on the lookout for the project. The team feels integrated and comfortable giving their opinions about how
to drive the sprint in the best way. This can work as a solution to avoid defeated teams without
motivation, implicating failures in the process.

4.3.2.15 Project mismanagement
Agile Practice: Sprint planning
Impact: Does not contributes directly to solve the anti-pattern

Detail: Monitoring and controlling activity are not the primary activities on this sprint planning
meetings. In consequence, this agile practice does not have a direct impact on solving the lack of
control over the project.

4.3.2.16 Proletariat hero
Agile Practice: Sprint planning
Impact: Contributes to solve the anti-pattern

Detail: During these meetings, the team members create their own working plan. They select and
estimate the tasks that they are compromised to complete during the sprint. This way is avoided
the false perception that some PM may have about that people will work better if they are punished
or coerced, becoming more efficient. The efficiency comes from their own compromise.

4.3.2.17 Rising upstart
Agile Practice: Sprint planning
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not
contribute to solve it.

Detail: Rising upstarts are people who try to show up their best professional skills, trying to stand
out because of the abilities that will allow them to perform more activities than others. During
sprint planning sessions, the estimation of activities to be performed is executed and can be easier
to identify people with this profile, trying to overload themselves with activities to perform during
the sprint.

4.3.2.18 Road to nowhere
Agile Practice: Sprint planning
Impact: Contributes to solve the anti-pattern

Detail: The sprint plannings are meant to have a plan for the work to be done in the sprint. This
allows having a established road to follow during the sprint achieving short goals and planning
constant progress.

4.3.2.19 Size isn’t everything
Agile Practice: Sprint planning
Impact: Contributes to solve the anti-pattern

Detail: Every sprint planning the members are in charge to estimate the amount of work that they
are able to perform, the plans are adapted to each team member, so team members are not
interchangeable. With this practice, the PM knows what each person is capable to execute, making changes in staff provokes changes in the plan.

4.3.2.20  The brawl
Agile Practice: Sprint planning
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Sprint planning meetings are a good tool to identify if leadership or management is missing. The team attempts to position themselves into a leadership role, skipping the point of view or authority of the responsible of the project; or no leadership is present, dificulting the process.

4.3.2.21  The domino effect
Agile Practice: Sprint planning
Impact: Does not contributes directly to solve the anti-pattern

Detail: During sprint planning sessions, decisions are not made about other projects. This anti-pattern talks about moving critical resources between the projects, this misstep can occur at any moment, not necessarily during these sessions. The sprint planning do not have direct inference, avoiding the PM moving resources provoking a domino effect within the projects.

4.3.2.22  Ultimate weapon
Agile Practice: Sprint planning
Impact: Contributes to solve the anti-pattern

Detail: Having knowledge about how the work will be done and who is going to perform it would help to avoid relying on everything on only one team member, forcing each person on the team to cooperate. In this case, the sprint backlog will reflect this situation helping to solve it.

4.3.3  Retrospectives impact

In this section it will be presented the matching results between the agile practice Retrospectives and the anti-patterns. On table 27, can be observed impact of the agile practice over the malpractices and subsequently the explanation of each selected impact.

Table 27. Retrospectives – Project management anti-patterns

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>Project Management Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrospectives</td>
<td>Absentee manager</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>All you have is a hammer</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Appointed team</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Detaillitis plan</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Dry waterhole</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Fire drill</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Glass case plan</td>
<td>✔</td>
</tr>
</tbody>
</table>
Inflexible plan
Irrational management
Leader not manager
Micro-management
Mushroom management
Myopic Delivery
Process disintegration
Project mismanagement
Proletariat hero
Rising upstart
Road to nowhere
Size isn’t everything
The brawl
The domino effect
Ultimate weapon

4.3.3.1 Absentee manager
Agile Practice: Retrospective
Impact: Contributes to solve the anti-pattern
Detail: Retrospectives are performed at the end of each sprint and require the participation of the development team and the PM. So, this practice enforces actions from PM at least once per sprint.

4.3.3.2 All you have is a hammer
Agile Practice: Retrospective
Impact: Contributes to solve the anti-pattern
Detail: The projects are not always the same and should not be treated in the same way either. If the same techniques are imposed and they are not giving the expected results, retrospectives are a good option to identify and propose alternative solutions to solve the problem.

4.3.3.3 Appointed team
Agile Practice: Retrospective
Impact: Contributes to solve the anti-pattern
Detail: Retrospectives are a good option to solve the problem of teams that are not well integrated. During these meetings the cooperation of all team members is necessary, everyone works together to find weaknesses, propose solutions and improve the processes.

4.3.3.4 Detailitis plan
Agile Practice: Retrospective
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: There exist other agile practices that avoid the creation of an excessively detailed plan, but if this situation is present, it can be also identified by retrospectives. These meetings are essentially performed to identify possible opportunity areas within the processes so it is a good tool to identify those problems.

4.3.3.5 Dry waterhole
Agile Practice: Retrospective

Impact: Does not contributes directly to solve the anti-pattern

Detail: Retrospectives are meetings performed along the project development and it is meant for process improvement. This agile practice does not have a direct impact on the establishment of requirements for selecting team members for a project.

4.3.3.6 Fire drill
Agile Practice: Retrospective

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Idle times can be detected in these retrospective meetings, this will allow the team members to try to find a solution for future sprints.

4.3.3.7 Glass case plan
Agile Practice: Retrospective

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: There exist other agile practices that might help to solve the lack of tracking and updating of the plans. In case this anti-pattern was not detected or solved because those practices were not correctly implemented, the objective of retrospectives is helping to identify the problems and opportunity areas within the projects to propose a possible solution.

4.3.3.8 Inflexible plan
Agile Practice: Retrospective

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Retrospectives are meetings specifically create to identify the different opportunity areas within a project. If there exists the situation where the process is rigid and the team needs to change something on them, this agile practice can help to identify the problem for search a solution.

4.3.3.9 Irrational management
Agile Practice: Retrospective
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: If the PM show indecisiveness or any other negative management habits it will be noticed by the team during the execution of the sprint, the retrospective can be a tool to identify the problem and propose it as an opportunity area to be improved.

4.3.3.10  
**Leader not manager**
Agile Practice: Retrospective

Impact: Contributes to identify the anti-pattern or the problems derived from it, but does not contribute to solve it.

Detail: The retrospectives are sessions created to identify the problems and opportunity areas within the development process. If the PM show a lack of management skills it can be noticed by the team during the execution of the sprint. This problem can be mentioned during the retrospective session for its improvement.

4.3.3.11  
**Micro-management**
Agile Practice: Retrospective

Impact: Contributes to identify a problem, but it might be the anti-pattern or any other issue.

Detail: Excessive management involvement could be a problem that arises during the retrospective. This practice could help in that way to identify the problem and propose it as an improvement action.

4.3.3.12  
**Mushroom management**
Agile Practice: Retrospective

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Retrospectives can help to identify if there exists a lack of communication between the development team and the users. The team can suggest that they might have more doubts than expected about the requirements and identify that they need to discuss them with the user.

4.3.3.13  
**Myopic delivery**
Agile Practice: Retrospective

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: During retrospectives can be identified if there exists an insistent demand by the management on delivering in the original estimated dates, even when the progress of the project is correct, but the initial estimations were not as accurate as expected. The situation can be identified and discussed during these meetings.

4.3.3.14  
**Process disintegration**
Agile Practice: Retrospective
Impact: Contributes to solve the anti-pattern

Detail: During retrospectives, the PM can be able to identify if the team is demotivated or if their attitude is not the best facing the problems. Also can be identified if the processes are not working as expected, provoking the feeling of defeat by the team. This antipattern can be also addressed by this practice, as proposals for improvement should require all team involvement and therefore all team cooperation.

4.3.3.15 Project mismanagement
Agile Practice: Retrospective

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Retrospectives can help as support for other agile practices to identify different issues derived from bad practices. The lack of proper project software monitoring and controlling might be identified during these sessions by the team or the PM.

4.3.3.16 Proletariat hero
Agile Practice: Retrospective

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Some PM has the false idea that coercion would increase the productivity of the team. During retrospectives, the team is able to openly treat what they think is wrong executed within the project, such as this miss practice and the problem can be identified.

4.3.3.17 Rising upstart
Agile Practice: Retrospective

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: These sessions can help to identify team members with this profile. Rising upstarts are people who take any opportunity to show up and stand out, these sessions where all the team should participate is a good option to find those problems.

4.3.3.18 Road to nowhere
Agile Practice: Retrospective

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: During retrospectives can be identified the lack of planning, the team can communicate the need of a guide to follow or if the sprint planning has any failure when the planification it is being performed.

4.3.3.19 Size isn’t everything
Agile Practice: Retrospective
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Every sprint has a plan that is adapted to each team member, so team members are not interchangeable. The PM knows what each person is capable to execute, making changes in staff provokes changes in the plan. In case this anti-pattern occurs the retrospective can help to identify the problem for later try to find a solution.

4.3.3.20 The brawl
Agile Practice: Retrospective

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: If the PM shows a lack of leadership or management experience it will be noticed by the team during the execution of the sprint or during meetings as retrospectives. Also, can be presented the situation where the team attempts to position themselves into a leadership role, skipping the point of view or authority of the responsible of the project. The retrospective can be a tool to identify the problem and propose it as an opportunity area to be improved.

4.3.3.21 The domino effect
Agile Practice: Retrospective

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Every sprint has a plan that is adapted to each team member, so team members should not be moved between the projects. The PM knows what each person is capable to execute, making changes in staff provokes changes in the plan. In case this anti-pattern occurs the retrospective can help to identify the problem either because of the slips on the schedule or the lack of an important resource for that project.

4.3.3.22 Ultimate weapon
Agile Practice: Retrospective

Impact: Contributes to identify the anti-pattern or the problems derived from it, but does not contribute to solve it

Detail: Ultimate weapon is the superstar of the team which is able to have an outstanding performance, this can provoke that other team members rely on all the activities on people with this profile. Retrospectives can work as a support to discuss this findings and identify if this anti-pattern represents the current situation of the team to propose a solution.

4.3.4 Sprint review impact

In this section it will be presented the matching results between the agile practice Sprint review and the anti-patterns. On table 28, can be observed impact of the agile practice over the malpractices and subsequently the explanation of each selected impact.
Table 28. Sprint review – Project management anti-patterns

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>Project Management Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Absentee manager</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>All you have is a hammer</td>
<td></td>
<td>✗</td>
</tr>
<tr>
<td>Appointed team</td>
<td></td>
<td>✗</td>
</tr>
<tr>
<td>Detailitis plan</td>
<td></td>
<td>✗</td>
</tr>
<tr>
<td>Dry waterhole</td>
<td></td>
<td>✗</td>
</tr>
<tr>
<td>Fire drill</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Glass case plan</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Inflexible plan</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Irrational management</td>
<td></td>
<td>✗</td>
</tr>
<tr>
<td>Leader not manager</td>
<td></td>
<td>☺</td>
</tr>
<tr>
<td>Micro-management</td>
<td></td>
<td>✗</td>
</tr>
<tr>
<td>Mushroom management</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Myopic Delivery</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Process disintegration</td>
<td></td>
<td>✗</td>
</tr>
<tr>
<td>Project mismanagement</td>
<td></td>
<td>☺</td>
</tr>
<tr>
<td>Proletariat hero</td>
<td></td>
<td>✗</td>
</tr>
<tr>
<td>Rising upstart</td>
<td></td>
<td>☻</td>
</tr>
<tr>
<td>Road to nowhere</td>
<td></td>
<td>✗</td>
</tr>
<tr>
<td>Size isn’t everything</td>
<td></td>
<td>✗</td>
</tr>
<tr>
<td>The brawl</td>
<td></td>
<td>☻</td>
</tr>
<tr>
<td>The domino effect</td>
<td></td>
<td>✗</td>
</tr>
<tr>
<td>Ultimate weapon</td>
<td></td>
<td>✗</td>
</tr>
</tbody>
</table>

4.3.4.1 Absentee manager
Agile Practice: Sprint review

Impact: Contributes to solve the anti-pattern

Detail: During the sprint review, it is necessary the presence of the PM, the product owner, and the development team. The PM is expected to have an active role to present the product and manage the communication with the PO.

4.3.4.2 All you have is a hammer
Agile Practice: Sprint review

Impact: Does not contribute directly to solve the anti-pattern

Detail: Sprint review is a specific meeting to review what was developed during the sprint, on those sessions are not evaluated any issues that might arise about processes or team members. This agile practice does not have a direct impact solving the problem of using always the same techniques to solve the arising problems.
4.3.4.3 Appointed team
Agile Practice: Sprint review

Impact: Does not contribute directly to solve the anti-pattern

Detail: Sprint review is a meeting meant for the review the progress of the project during each sprint, not related to the team management or their integration. This agile practice does not have a direct relation in solving the issues derived from this anti-pattern.

4.3.4.4 Detailitis plan
Agile Practice: Sprint review

Impact: Does not contribute directly to solve the anti-pattern

Detail: During sprint reviews, it is not performed any planning related to the project. This meetings only serve as a review session to know if what was developed during the sprint accomplishes the expected requirements, for this reason, this agile practice does not help directly solving the problem of having very complex schedules.

4.3.4.5 Dry waterhole
Agile Practice: Sprint review

Impact: Does not contribute directly to solve the anti-pattern

Detail: These sprint review meetings are performed during the life of the project. Normally, the selection process and the requirements set for each profile are activities performed before the project starts. Due to this, sprint reviews are not an option that helps directly to identify or solve the problem of establishing stringent requirements for team selection.

4.3.4.6 Fire drill
Agile Practice: Sprint review

Impact: Contributes to solve the anti-pattern

Detail: Due to each sprint should have a sprint review meeting, the progress of the product is reviewed and some improvements can arise generating new activities that should be executed on the following sprints. This does not allow boredom times avoiding big delays and hard demands.

4.3.4.7 Glass case plan
Agile Practice: Sprint review

Impact: Contributes to solve the anti-pattern

Detail: Essentially the sprint review is to review the progress of the product during each sprint, and consequently keep a track of what was successfully accomplished or if there exist any task that was not finished. This promotes the constant update to the initial plans that also will work as a comparison point for future projects planning.

4.3.4.8 Inflexible plan
Agile Practice: Sprint review
Impact: Contributes to solve the anti-pattern

Detail: During the sprint review, it is identified if the product needs any improvement or change. Consequently, changes in the backlog will arise that should be updated to keep track of them for their implementation, avoiding the idea of having a definitive plan that should not be changed.

4.3.4.9 Irrational management
Agile Practice: Sprint review

Impact: Does not contribute directly to solve the anti-pattern

Detail: Sprint review does not have a direct relation with this anti-pattern, due to during these meetings the management does not evaluate or take decisions.

4.3.4.10 Leader not manager
Agile Practice: Sprint review

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: During sprint reviews the is expected that the PM leads the meeting, having control of what was performed during the sprint. If the PM does not follow a management methodology some difficulties might arise due to the lack of control. This agile practice can help to identify if it is necessary that the PM improve their management skills to improve the quality of the processes.

4.3.4.11 Micro-management
Agile Practice: Sprint review

Impact: Does not contribute directly to solve the anti-pattern

Detail: In sprint reviews are not performed any activity related to the assignation of activities. Even when the PM might be involved in activities beyond their responsibility, this agile practice does not have a direct impact on solving this problem.

4.3.4.12 Mushroom management
Agile Practice: Sprint review

Impact: Contributes to solve the anti-pattern

Detail: During sprint reviews it is necessary that the product owner is present, who is the representative of the final user. This interaction between the development team and the customer is a perfect option to exchange doubts, ideas, and solutions for the creation of the final product, promoting communication between the PO and the team.

4.3.4.13 Myopic delivery
Agile Practice: Sprint review

Impact: Contributes to solve the anti-pattern

Detail: Sprint reviews are one of the complements for review what is happening within the creation of the product, the advance and the remaining tasks. This help to make the stakeholders
consciousness about the possible retards on the first established delivery date as a consequence of the demand for any modification.

4.3.4.14  Process disintegration
Agile Practice: Sprint review

Impact: Does not contributes directly to solve the anti-pattern

Detail: Even when the team might be demotivated due to constant failures, sprint reviews are only a review session with the client. This agile practice does not have a direct impact solving the problem of a within the team and their performance.

4.3.4.15  Project mismanagement
Agile Practice: Sprint review

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: If the management does not have the proper control and monitoring about what is being performed, this leads to a difficult situation the presentation of the progress of the project. Sprint reviews can help to identify if there exists a lack of control about the developed software in case the sprint review does not flows as it is expected.

4.3.4.16  Proletariat hero
Agile Practice: Sprint review

Impact: Does not contributes directly to solve the anti-pattern

Detail: Sprint reviews are led by the PM, even when the team is present on the meetings is not a suitable situation where they participate a lot. This agile practice does not help directly solving or identifying the wrong actions that the PM can execute to increase team productivity.

4.3.4.17  Rising upstart
Agile Practice: Sprint review

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Rising upstarts want to forego the requisite time to learn and show up how supposedly they have better skills than other teammates in any opportunity that they find. Can occur the unwanted situation where a person with this profile tries to stand out with the user, therefore, the PM can identify the situation to later try to find a solution.

4.3.4.18  Road to nowhere
Agile Practice: Sprint review

Impact: Does not contributes directly to solve the anti-pattern

Detail: Any planning activity is performed during sprint review sessions, this agile practice might not be supportive of finding a solution for a lack of planning within the project.
4.3.4.19  
**Size isn’t everything**  
Agile Practice: Sprint review  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: During sprint reviews, the team presents what was performed during the sprint to verify if the developed features work as expected. During these sessions is not reviewed any activity respect to the team assignments, due to this, the agile practice does not have direct inference solving this anti-pattern.

4.3.4.20  
**The brawl**  
Agile Practice: Sprint review  
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.  
Detail: Sprint reviews are meetings where the development team present to the PO the work executed during the sprint. The PM is responsible to present the product and manage the communication during these meetings, but if there exists a lack of management experience the problem will be identified and later try to find a solution.

4.3.4.21  
**The domino effect**  
Agile Practice: Sprint review  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: If the PM takes the decision of moving critical resources between the projects provoking slips or problems within them, does not mean that they will not have something to present during sprint review sessions. This agile practice does not help as expected to solve this situation, and even implementing it the problem might persist.

4.3.4.22  
**Ultimate weapon**  
Agile Practice: Sprint review  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: Even when the team is present during sprint reviews, the interaction and the activities performed during this meeting does not lead to a suitable situation where it can be identified teammates with this superstar profile, then this agile practice does not have direct impact solving the problem.

4.3.5  
**Short iterations impact**  

In this section it will be presented the matching results between the agile practice Short iterations and the anti-patterns. On table 29, can be observed impact of the agile practice over the malpractices and subsequently the explanation of each selected impact.
Table 29. Short iterations – Project management anti-patterns

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>Project Management Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absentee manager</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>All you have is a hammer</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Appointed team</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Detailitis plan</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Dry waterhole</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Fire drill</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Glass case plan</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Inflexible plan</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Irrational management</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Leader not manager</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Micro-management</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Mushroom management</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Myopic Delivery</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Process disintegration</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Project mismanagement</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Proletariat hero</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Rising upstart</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Road to nowhere</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Size isn’t everything</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>The brawl</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>The domino effect</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Ultimate weapon</td>
<td></td>
<td>×</td>
</tr>
</tbody>
</table>

4.3.5.1 Absentee manager
Agile Practice: Short iterations
Impact: Contributes to solve the anti-pattern

Detail: Short iterations should promote the PM actions in short periods of times, in opposition to long iterations where a possible absence of the PM could go unnoticed.

4.3.5.2 All you have is a hammer
Agile Practice: Short iterations
Impact: Does not contributes directly to solve the anti-pattern

Detail: Short iterations do not have a direct relation with one-dimensional management trying to use the same techniques to solve the problems. This practice promotes the agility, improving the team's calculations estimating amounts of work per sprint.

4.3.5.3 Appointed team
Agile Practice: Short iterations
Impact: Does not contributes directly to solve the anti-pattern
Detail: There exist many agile practices that can help to solve the problem of lack of team integration. Short iterations is a practice that promotes agility and helps the team improving their estimations, but does not have a direct impact on promoting their integration.

4.3.5.4 Detailitis plan
Agile Practice: Short iterations
Impact: Contributes to solve the anti-pattern
Detail: Due to each iteration should be short, the plan is limited by the tasks that will be performed during the sprint avoiding complex schedules.

4.3.5.5 Dry waterhole
Agile Practice: Short iterations
Impact: Does not contributes directly to solve the anti-pattern
Detail: The team selection process is an activity performed before starting the project. Short iterations is a practice implemented during the project development, this means that this agile practice cannot help directly solving the problem of establishing stringent requirements for team profiles.

4.3.5.6 Fire drill
Agile Practice: Short iterations
Impact: Contributes to solve the anti-pattern
Detail: Short iterations promote having constant progress on a project in small amounts of time with accurate estimations, this avoids to leave all the work till the last minute and getting enormous amounts of work till the end.

4.3.5.7 Glass case plan
Agile Practice: Short iterations
Impact: Contributes to solve the anti-pattern
Detail: Short iterations allow keeping the initial release plan updated according to the work done in each iteration, this way is avoided the lack of tracking and updating the initial plans.

4.3.5.8 Inflexible plan
Agile Practice: Short iterations
Impact: Contributes to solve the anti-pattern
Detail: From a project point of view, short iterations allow the release plan to be updated and flexible. At a sprint point of view, due to short iterations are shaped by small amounts of time, the process should be flexible and adaptable to each iteration.

4.3.5.9 Irrational management
Agile Practice: Short iterations
Impact: Does not contributes directly to solve the anti-pattern

Detail: Even if the PM is making wrong decisions or his/her indecisiveness is provoking problems within the project, short iterations do not have direct impact avoiding this situation. The performance of the PM will keep incorrect until the problem is identified.

4.3.5.10  Leader not manager
Agile Practice: Short iterations

Impact: Does not contributes directly to solve the anti-pattern

Detail: Having short iterations will not help to avoid if there exists a lack of management methodology to follow. The problems might persist and the agile practice not necessarily will help to solve or identify which is the problem.

4.3.5.11  Micro-management
Agile Practice: Short iterations

Impact: Does not contributes directly to solve the anti-pattern

Detail: The situation where the PM feel insecure and take responsibility for tasks beyond their scope can be present in any situation no matters if the scheme is traditional or agile. Short iterations is not a direct solution to solve this problem.

4.3.5.12  Mushroom management
Agile Practice: Short iterations

Impact: Does not contributes directly to solve the anti-pattern

Detail: Short iterations is a practice that helps to improve the agile process, it does not have a direct inference solving the lack of communication and interaction between the team and the user.

4.3.5.13  Myopic delivery
Agile Practice: Short iterations

Impact: Contributes to solve the anti-pattern

Detail: Short iterations help to estimate accurately the amount of work to be executed each sprint, this also helps the management to know the progress of the product and establish accurate delivery dates.

4.3.5.14  Process disintegration
Agile Practice: Short iterations

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: The constant delivery during short iterations allows the PM to review the results constantly, making more evident if there exist any failure deriving in a demotivated team. If the results are not as expected due to failing processes and the team looks unsteady, they can be related, and the anti-pattern would be identified.
4.3.5.15  Project mismanagement  
Agile Practice: Short iterations  
Impact: Contributes to solve the anti-pattern  
Detail: Short iterations help to keep a constant tracking on the activities performed during each sprint, this way the project will be properly monitored and controlled. At least, their revision will be constant avoiding lack of control on the project.

4.3.5.16  Proletariat hero  
Agile Practice: Short iterations  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: The wrong action of coercion within the team members from the management side cannot be avoided even if the project has short iterations or a traditional scheme. This problem is related to the PM performance and the agile practice is related to improving agile processes.

4.3.5.17  Rising upstart  
Agile Practice: Short iterations  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: Short iterations do not have a direct relation with the identification or the solution of a team member with a profile that want to stand out all the time, affecting the results of the project.

4.3.5.18  Road to nowhere  
Agile Practice: Short iterations  
Impact: Contributes to solve the anti-pattern  
Detail: The planning of the amount of work that will be performed during each short iteration is one of the principal activities that this agile practice promotes, this way it works as a solution of the lack of planning anti-pattern.

4.3.5.19  Size isn’t everything  
Agile Practice: Short iterations  
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.  
Detail: If the PM is interchanging team members and the objectives per sprint are not being achieved due to the constant changes. Short iterations can help to identify sooner this anti-pattern.

4.3.5.20  The brawl  
Agile Practice: Short iterations  
Impact: Does not contributes directly to solve the anti-pattern
Detail: The lack of management experience or leadership is an issue that can be solved with the support of some agile practices. Short iterations do not have direct influence solving this anti-pattern because is a practice-oriented to the processes improvement.

4.3.5.21 The domino effect
Agile Practice: Short iterations

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: If the PM is moving critical resources between projects and the objectives per sprint are not being achieved, causing a domino effect. Short iterations can help to identify sooner this anti-pattern.

4.3.5.22 Ultimate weapon
Agile Practice: Short iterations

Impact: Does not contributes directly to solve the anti-pattern

Detail: Short iterations are meant to improve team estimations. Even implementing this agile practice, the team can still rely heavily on the superstar of the team, letting that all estimations are done by him/her level of knowledge and the problem would persist.

4.3.6 Release planning impact

In this section it will be presented the matching results between the agile practice Release planning and the anti-patterns. On table 30, can be observed impact of the agile practice over the malpractices and subsequently the explanation of each selected impact.

Table 30. Release planning – Project management anti-patterns

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>Project Management Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release planning</td>
<td>Absentee manager</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>All you have is a hammer</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Appointed team</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Detailitis plan</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Dry waterhole</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Fire drill</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Glass case plan</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Inflexible plan</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Irrational management</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Leader not manager</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Micro-management</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Mushroom management</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Myopic Delivery</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Process disintegration</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Project mismanagement</td>
<td>✓</td>
</tr>
</tbody>
</table>
4.3.6.1 Absentee manager
Agile Practice: Release planning
Impact: Contributes to solve the anti-pattern
Detail: The active participation of the PM and development team is necessary to create this release plan. Due to this, the release planning sessions can work as an agile practice that contributes to avoiding the absence of the PM.

4.3.6.2 All you have is a hammer
Agile Practice: Release planning
Impact: Does not contribute directly to solve the anti-pattern
Detail: Release planning are meetings to create a release schedule based on the priorities of the user and the team possibilities. This agile practice does not have direct impact solving or finding a possible solution the management using always the same techniques to resolve the problems present in the project.

4.3.6.3 Appointed team
Agile Practice: Release planning
Impact: Contributes to solve the anti-pattern
Detail: Release planning sessions are performed with the whole people involved in the project, communication and cooperation are promoted to achieve common agreements. This practice also encourages people to work together, helping to integrate team members.

4.3.6.4 Detailitis plan
Agile Practice: Release planning
Impact: Contributes to solve the anti-pattern
Detail: Release planning is an overview of the general features that should be delivered for the product. These features will be decomposed into deeper tasks planned for each sprint. Therefore, this agile practice helps to avoid the generation of unnecessary complex schedules.

4.3.6.5 Dry waterhole
Agile Practice: Release planning
Impact: Does not contributes directly to solve the anti-pattern

Detail: During the release planning is planned the general features that should be delivered for the product. This sessions are part of the activities executed during the project development. This agile practice does not have direct inference avoiding the problem of setting stringent requirements for the team selection.

4.3.6.6 Fire drill
Agile Practice: Release planning

Impact: Contributes to solve the anti-pattern

Detail: The release planning requires the active participation of the development team, this way the team has the knowledge of the tasks that should perform and when to develop them. This agile practice contributes to avoiding idle times and deliver on the established due dates.

4.3.6.7 Glass case plan
Agile Practice: Release planning

Impact: Contributes to solve the anti-pattern

Detail: The release planning is meant to plan release dates for each feature. This plan is updated according to the results of the different sprints, promoting the updates on the original plan.

4.3.6.8 Inflexible plan
Agile Practice: Release planning

Impact: Contributes to solve the anti-pattern

Detail: When this planning is being performed, the team and the PM have in mind that the delivery dates are not fixed and can be changed as the project is going on. This agile practice work as a complement to solve the lack of a flexible plan.

4.3.6.9 Irrational management
Agile Practice: Release planning

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: All team members and PM participate in the release planning meeting. They cooperate and establish due dates that they are sure they can accomplish. This session contributes to identify if the PM is indecisive or take wrong decisions during estimations, due to the PM is not the only one planning and try to find a solution.

4.3.6.10 Leader not manager
Agile Practice: Release planning

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.
Detail: It is necessary to follow a management methodology for the creation of any kind of plan, in this case, the release planning. If there exist a lack of knowledge in management from the PM it would be more evident during this kind of meetings and it can be identified the problem.

4.3.6.11 Micro-management
Agile Practice: Release planning
Impact: Does not contributes directly to solve the anti-pattern

Detail: The release planning is meant to plan release dates for the features of the product, this practice is not directly related to solving the wrong performed management activities, such as the PM taking responsibility of tasks beyond their scope.

4.3.6.12 Mushroom management
Agile Practice: Release planning
Impact: Contributes to solve the anti-pattern

Detail: During release planning sessions, the customer needs to present his/her prioritize features and the team needs to estimate how many time does it take to develop them. This way these meetings are a useful tool to promote the communication between developers and users.

4.3.6.13 Myopic delivery
Agile Practice: Release planning
Impact: Contributes to solve the anti-pattern

Detail: When release planning sessions take place, the user and the team generates the plan and create the estimations according to their knowledge on the area to have a general idea about the delivery dates. However, if there exist any slip on the calendar the stakeholders are conscient that the project is being developed on an agile scheme and changes on it can arise.

4.3.6.14 Process disintegration
Agile Practice: Release planning
Impact: Contributes to solve the anti-pattern

Detail: The release planning is meant to plan release dates for each feature. This process is done in a cooperative way; also, the planning of the activities that should be performed help to increase security on the team because they have a path to follow and do not feel lost trying to understand what they need to perform.

4.3.6.15 Project mismanagement
Agile Practice: Release planning
Impact: Contributes to solve the anti-pattern

Detail: The release planning is meant to plan release dates for each feature. This plan should be controlled and updated according to the results of the different sprints, otherwise, it would be probably that the releases do not coincide with what it was established at the beginning.
4.3.6.16 Proletariat hero
Agile Practice: Release planning
Impact: Does not contributes directly to solve the anti-pattern

Detail: Release planning are meetings related to planning the features that should be developed. The malpractices executed by the PM, such as coercion to increase team productivity cannot be solved directly with the agile practice.

4.3.6.17 Rising upstart
Agile Practice: Release planning
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: It is possible to find a situation where a person with this profile tries to stand out and show up how supposedly he/ she is the best within the team, therefore, the PM can identify this problem to later try to find a solution.

4.3.6.18 Road to nowhere
Agile Practice: Release planning
Impact: Contributes to solve the anti-pattern

Detail: Release planning is not the only practice that helps to generate a plan but is useful and one of the pieces that complement the planning activities within an agile schedule.

4.3.6.19 Size isn’t everything
Agile Practice: Release planning
Impact: Does not contributes directly to solve the anti-pattern

Detail: Release planning is a session where the team plans release dates for each feature, it does not have a direct relation with the development process that is executed or who is assigned to perform each activity within the project.

4.3.6.20 The brawl
Agile Practice: Release planning
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: During release planning meetings it can be identified if leadership or management is missing. The team attempts to position themselves into a leadership role, skipping the point of view or authority of the responsible of the project; or no leadership is present dificulting the process.

4.3.6.21 The domino effect
Agile Practice: Release planning
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: The planification of the release plan is performed by all team members, the estimations are based on their knowledge. If the PM is moving constantly the resources the plan will also constantly modified and he/she can identify that there exist a problem due to the changes within the team.

4.3.6.22 Ultimate weapon
Agile Practice: Release planning

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: On release planning can be evident if the team is relying heavily on the superstar of the, letting that all estimations are done by his/her level of knowledge. These sessions can be a good tool to identify if there exist a person with this profile on the team and try to find a solution to the problem.

4.3.7 Planning poker impact

In this section it will be presented the matching results between the agile practice Planning poker and the anti-patterns. On table 31, can be observed impact of the agile practice over the malpractices and subsequently the explanation of each selected impact.

Table 31. Planning poker – Project management anti-patterns

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>Project Management Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning poker</td>
<td>Absentee manager</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>All you have is a hammer</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Appointed team</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Detailitis plan</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Dry waterhole</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Fire drill</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Glass case plan</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Inflexible plan</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Irrational management</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Leader not manager</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Micro-management</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Mushroom management</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Myopic Delivery</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Process disintegration</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Project mismanagement</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Proletariat hero</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Rising upstart</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Road to nowhere</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Size isn’t everything</td>
<td>✔</td>
</tr>
</tbody>
</table>
4.3.7.1 Absentee manager
Agile Practice: Planning poker
Impact: Contributes to solve the anti-pattern
Detail: Planning poker is a tool to estimate the value of each task that should be performed at the beginning of each sprint with the participation of all the team. Therefore, this tool encourages PM active participation.

4.3.7.2 All you have is a hammer
Agile Practice: Planning poker
Impact: Contributes to solve the anti-pattern
Detail: Planning poker requires the discussion among the team member about the estimation of each user story. Those discussions can help to identify and propose different solutions based on the expertise of each team member. This agile practice promotes solving the issues using different techniques depending on the proposals obtained during these sessions.

4.3.7.3 Appointed team
Agile Practice: Planning poker
Impact: Contributes to solve the anti-pattern
Detail: Even if the planning poker is a tool, motivates the cooperation and open discussion about the work estimations. This activity is performed with all the development team, enforcing them to contribute and give their opinion, helping them for constant integration and cooperation.

4.3.7.4 Detailitis plan
Agile Practice: Planning poker
Impact: Contributes to solve the anti-pattern
Detail: Planning poker generates an estimation in story points which are an abstract metric far away from detailed measures and estimations. The complexity of schedules is avoided due to is not an official metric that gets into deeper detail.

4.3.7.5 Dry waterhole
Agile Practice: Planning poker
Impact: Does not contributes directly to solve the anti-pattern
Detail: Planning poker is a tool for tasks value estimation, this agile practice is implemented during the project execution. Team selection process and the requirements for each profile are usually
established at the very beginning of the project. Planning poker does not have a direct relation in solving the problems during the team selection process.

4.3.7.6 Fire drill
Agile Practice: Planning poker
Impact: Does not contributes directly to solve the anti-pattern

Detail: Planning poker is an activity performed at the beginning of each sprint and is an estimation activity. The use of this agile practice does not represent an activity that would have a direct relationship with the team performance and if they have idle times that affect the project.

4.3.7.7 Glass case plan
Agile Practice: Planning poker
Impact: Does not contributes directly to solve the anti-pattern

Detail: The estimation process about the activities that will be performed at the beginning of each sprint should be performed at the beginning and is a part of the planning poker. Even having this part of the plan not necessarily the management will keep tracking and updating the initial plans. This agile practice does not have direct inference solving the anti-pattern.

4.3.7.8 Inflexible plan
Agile Practice: Planning poker
Impact: Contributes to solve the anti-pattern

Detail: Planning poker is by definition a flexible process, based on discussion and relative metrics. Each person cooperates based on their experience and abilities promoting processes open to take different ways to find the solutions instead of only one path.

4.3.7.9 Irrational management
Agile Practice: Planning poker
Impact: Contributes to solve the anti-pattern

Detail: Planning poker is based on discussions among all team members. So indecisiveness or irrational decisions of the PM can be detected and avoided. Otherwise, as a cooperative process, those irrational decisions could be minimized.

4.3.7.10 Leader not manager
Agile Practice: Planning poker
Impact: Contributes to solve the anti-pattern

Detail: Planning poker by itself proposes a management strategy to deal with the estimation and even if the PM does not follow a management methodology, this agile practice can guide him/her.

4.3.7.11 Micro-management
Agile Practice: Planning poker
Impact: Contributes to solve the anti-pattern

Detail: Planning poker by itself proposes a management strategy to deal with estimation. Involves all team members experience and skills, this avoids that the PM takes tasks beyond his/her responsibilities due to estimations are based on everyone expertise.

4.3.7.12 Mushroom management
Agile Practice: Planning poker

Impact: Contributes to solve the anti-pattern

Detail: Planning poker is a tool for tasks value estimation, during these sessions it can arise doubts about the activities to be performed. To solve those doubts it will be necessary the input of the PO and this generates a communication flow between the development team and the user.

4.3.7.13 Myopic delivery
Agile Practice: Planning poker

Impact: Contributes to solve the anti-pattern

Detail: Planning poker is also part of a learning process so estimations can evolve and therefore the PM can realize about the need to change original delivery dates, avoiding the constant demand on delivering on the first established date.

4.3.7.14 Process disintegration
Agile Practice: Planning poker

Impact: Contributes to solve the anti-pattern

Detail: Planning poker can help to solve the problem of a demotivated team. During these sessions, it is necessary to discuss and work on agreements between the team members, this increase the cooperation and compromise. Besides, the team feels confidence because the estimations are based on their experience.

4.3.7.15 Project mismanagement
Agile Practice: Planning poker

Impact: Does not contributes directly to solve the anti-pattern

Detail: Planning poker is a tool for tasks value estimation executed at the beginning of the sprint, it does not have a direct relation with the software project monitoring and controlling.

4.3.7.16 Proletariat hero
Agile Practice: Planning poker

Impact: Does not contributes directly to solve the anti-pattern

Detail: The technique that should be followed during estimation sessions using planning poker, promote the open discussion and agreements within the complete development team and the PM. If there exist coercion from the management, it can be identified during these meetings and try to find a solution.
4.3.7.17 Rising upstart
Agile Practice: Planning poker

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Planning poker is used during estimation sessions where all the team is working together. People with this profile would try to show up their abilities during these meetings and it would be easier for the PM to identify the problem.

4.3.7.18 Road to nowhere
Agile Practice: Planning poker

Impact: Contributes to solve the anti-pattern

Detail: Planning poker is a useful tool that works as a compliment for the estimation and planification of the activities that will be executed during the following sprint.

4.3.7.19 Size isn’t everything
Agile Practice: Planning poker

Impact: Contributes to solve the anti-pattern

Detail: The estimations with planning poker are performed by all team members. If the PM is interchanging constantly team members the plan will not be accurate and the problem will arise. This agile practice would help to identify the issue.

4.3.7.20 The brawl
Agile Practice: Planning poker

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: If the PM presents a lack of leadership or management experience, he/she will not be capable to lead the process or solve any conflict during planning poker sessions. These meetings can help to identify the problem and try to find a solution.

4.3.7.21 The domino effect
Agile Practice: Planning poker

Impact: Does not contributes directly to solve the anti-pattern

Detail: Planning poker is a tool for tasks value estimation, it does not have a direct relation with the decisions taken by the PM about the activities that the developers should perform and the impact that they might have on the project.

4.3.7.22 Ultimate weapon
Agile Practice: Planning poker

Impact: Contributes to solve the anti-pattern
Detail: The team can trust in just one person, rely on his/her knowledge and think that the estimations proposed by him/her are the correct one. Planning poker help to solve this problem, because the complete team is obligated to participate to establish accurate estimations based on the input and knowledge of the whole team.

4.3.8 Available product owner impact

In this section it will be presented the matching results between the agile practice Available product owner and the anti-patterns. On table 32, can be observed impact of the agile practice over the malpractices and subsequently the explanation of each selected impact.

Table 32. Available product owner – Project management anti-patterns

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>Project Management Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absentee manager</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>All you have is a hammer</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Appointed team</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Detailitis plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dry waterhole</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fire drill</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glass case plan</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Inflexible plan</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Irrational management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leader not manager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Micro-management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mushroom management</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Myopic Delivery</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process disintegration</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Project mismanagement</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Proletariat hero</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Rising upstart</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Road to nowhere</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Size isn’t everything</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The brawl</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>The domino effect</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ultimate weapon</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

4.3.8.1 Absentee manager
Agile Practice: Available product owner

Impact: Does not contributes directly to solve the anti-pattern

Detail: The available product owner is the representation of the customer on agile projects. Even if the PO is available does not mean that would avoid the absence of the manager to deal with other activities.
4.3.8.2 **All you have is a hammer**
Agile Practice: Available product owner

Impact: Does not contributes directly to solve the anti-pattern

Detail: The representation of the customer on agile projects is the product owner who is responsible for managing the backlog. He is not responsible for making decisions or solving problems within the development process or the subordinates.

4.3.8.3 **Appointed team**
Agile Practice: Available product owner

Impact: Does not contributes directly to solve the anti-pattern

Detail: The available product owner is not responsible for integrating the development team, is responsible for managing the prioritization of the backlog. This agile practice does not have direct inference solving the anti-pattern.

4.3.8.4 **Detailitis plan**
Agile Practice: Available product owner

Impact: Does not contributes directly to solve the anti-pattern

Detail: The available product owner is the representation of the customer on agile projects is not part of his/her activities to prepare a plan. This agile practice does not have a direct relation to generating those plans for the development of the project.

4.3.8.5 **Dry waterhole**
Agile Practice: Available product owner

Impact: Does not contributes directly to solve the anti-pattern

Detail: The available product owner is not responsible for setting the requirements for recruiting the development team, is responsible for managing the prioritization of the backlog. This agile practice does not have direct inference solving the anti-pattern.

4.3.8.6 **Fire drill**
Agile Practice: Available product owner

Impact: Does not contributes directly to solve the anti-pattern

Detail: This agile practice does not have a relation solving the possible issues derived from idle times or the performance of the development team during the execution of the project, the available PO is the representation of the client within the project.

4.3.8.7 **Glass case plan**
Agile Practice: Available product owner

Impact: Contributes to solve the anti-pattern
Detail: The product owner is the representation of the customer on agile projects. He/She needs to prioritize and keep a track of the backlog to let know the development team what should be performed on the following sprints. This agile practice complemented with other agile practices solve the problem of lack of track and updating of the plans.

4.3.8.8 Inflexible plan
Agile Practice: Available product owner
Impact: Contributes to solve the anti-pattern

Detail: Having the product owner available allows discussing possible changes in the plan, according to changes in requirements, priorities, etc. This kind of activities promotes the flexibility in both processes and plans, to allow the team to accommodate their activities within the agile scheme.

4.3.8.9 Irrational management
Agile Practice: Available product owner
Impact: Does not contribute directly to solve the anti-pattern

Detail: The wrong actions from the management such as indecisiveness or imposing wrong decisions are malpractices that cannot be solved in this situation. Even if the product owner is available, he/she does not have inference on the actions that the PM might have within the project.

4.3.8.10 Leader not manager
Agile Practice: Available product owner
Impact: Does not contribute directly to solve the anti-pattern

Detail: Having a leader does not mean that this skill will avoid the lack of order within the project. Neither, having an available product owner cannot help directly on the improvement of the management skills that might need the PM.

4.3.8.11 Micro-management
Agile Practice: Available product owner
Impact: Does not contribute directly to solve the anti-pattern

Detail: This anti-pattern is related to the interference of the PM in technical tasks beyond their responsibility. Having the product owner available do not affect or benefit on finding a solution to this problem.

4.3.8.12 Mushroom management
Agile Practice: Available product owner
Impact: Contributes to solve the anti-pattern

Detail: The product owner represents the customer on agile projects. If he/she is constantly available, the communication will flow easily and the development team can discuss any doubt or possible issue. This agile practice is a good option to face the lack of communication between the team and the user.
4.3.8.13  Myopic delivery  
Agile Practice: Available product owner  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: Even if there exist changes from Product Owner side, in this anti-pattern the management is the person who is insisting on accomplishing the original delivery date, affecting the team performance.

4.3.8.14  Process disintegration  
Agile Practice: Available product owner  
Impact: Contributes to solve the anti-pattern  
Detail: Having an available product owner motivates a collaborative environment, facilitating communication and promoting the improvement of the team performance.

4.3.8.15  Project mismanagement  
Agile Practice: Available product owner  
Impact: Contributes to solve the anti-pattern  
Detail: When the PO is available, he/she might need some update about the project to perform their proper control into the backlog. It is necessary to have control and monitoring of the activities that are being performed during the software development, otherwise, the management would not be able to inform correctly the progress to PO. This agile practice help and motivates the PM keep adequate control of the project activities.

4.3.8.16  Proletariat hero  
Agile Practice: Available product owner  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: The wrong actions from the management such as coercion is a malpractice that cannot be solved implementing exclusively this agile practice. Even if the product owner is available, is not responsible for surveilling the performance of the team, he/she does not have inference on the actions that the PM might have within the team.

4.3.8.17  Rising upstart  
Agile Practice: Available product owner  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: Rising upstarts are team members that have an unmerited sense of superiority that might affect the project progress and team performance. However, the problems that arise from these team members and the subsequent actions taken to resolve these issues cannot be avoided even with a PO and this agile practice does not have direct inference in solving them.

4.3.8.18  Road to nowhere  
Agile Practice: Available product owner
Impact: Contributes to solve the anti-pattern

Detail: Having available the PO is responsible for managing the backlog which will facilitate the creation of the working plan for each sprint and the user stories to be developed. This agile practice supports, in addition to other agile practices, to solve the problem of a lack of planning.

4.3.8.19 Size isn’t everything
Agile Practice: Available product owner
Impact: Does not contributes directly to solve the anti-pattern

Detail: The assignation of activities for each team member concerns to the PM, an available product owner does not deal with these activities within the project. Consequently, this agile practice does not have an impact on the solution to this anti-pattern.

4.3.8.20 The brawl
Agile Practice: Available product owner
Impact: Does not contributes directly to solve the anti-pattern

Detail: The lack of leadership or management experience might generate big problems for the project, the implementation of different agile practices can help to identify or solve the problem. An available product owner would not have a direct impact on solving the lack of these management skills.

4.3.8.21 The domino effect
Agile Practice: Available product owner
Impact: Does not contributes directly to solve the anti-pattern

Detail: The product owner is the representation of the customer on agile projects and is the person in charge of managing the backlog. This agile practice does not take responsibility for other activities such as managing team members and assigning them their activities or projects. That is why this agile practice does is not direct support on solving the problem of moving critical resources between the projects that might affect them.

4.3.8.22 Ultimate weapon
Agile Practice: Available product owner
Impact: Does not contributes directly to solve the anti-pattern

Detail: The ultimate weapon is the superstar of the team, the other team members rely heavily on them trusting they are the ones that are going to solve most of the problems. The identification and treatment of the problems that a specific profile might generate, such as an ultimate weapon, do not concern to the PO.
4.3.9 **Single team impact**

In this section it will be presented the matching results between the agile practice Single team and the anti-patterns. On table 33, can be observed impact of the agile practice over the malpractices and subsequently the explanation of each selected impact.

Table 33. Single team – Project management anti-patterns

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>Project Management Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single team</td>
<td>Absentee manager</td>
<td></td>
</tr>
<tr>
<td></td>
<td>All you have is a hammer</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>Appointed team</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>Detailitis plan</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>Dry waterhole</td>
<td>▲</td>
</tr>
<tr>
<td></td>
<td>Fire drill</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>Glass case plan</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Inflexible plan</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>Irrational management</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Leader not manager</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Micro-management</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>Mushroom management</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Myopic Delivery</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>Process disintegration</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>Project mismanagement</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Proletariat hero</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>Rising upstart</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>Road to nowhere</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Size isn’t everything</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>The brawl</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>The domino effect</td>
<td>▲</td>
</tr>
<tr>
<td></td>
<td>Ultimate weapon</td>
<td>✔️</td>
</tr>
</tbody>
</table>

4.3.9.1 *Absentee manager*

Agile practice: Single team

Impact: Does not contributes directly to solve the anti-pattern

Detail: Having a single team where the people who integrate it is flexible and can perform different activities no matter their technical skill set, does not influence directly for the manager to be present and have an active role in the project.

4.3.9.2 *All you have is a hammer*

Agile practice: Single team

Impact: Contributes to solve the anti-pattern
Detail: The single team opens the mindset of the tasks should be performed only by the team members that are specialized on a specific skill set. There exist many possibilities to solve a problem and this practice motivate to find and choose other possibilities and avoid the situation of trying to solve the problems using always the same techniques.

4.3.9.3 Appointed team
Agile practice: Single team
Impact: Contributes to solve the anti-pattern

Detail: Single team motivates the integration between team members because if someone need help to execute an activity, they are supported by their co-workers. Also, there is constant learning within the team, this promotes their confidence in their teammates.

4.3.9.4 Detailitis plan
Agile practice: Single team
Impact: Contributes to solve the anti-pattern

Detail: The fact of having a flexible skilled team that can execute several activities would lead to the creation of a higher-level plan. The planning should be more open to changes in order to update it depending on which person will be performing different activities. If the plan is very complex, the changes will take more time than necessary. This agile practice would contribute to the problem solving of this antipattern.

4.3.9.5 Dry waterhole
Agile practice: Single team
Impact: Especial cases

Detail: Having a single team should be carefully treated. This kind of antipattern might be aggravated due to the idea that it is necessary hiring people who are able to perform any kind of technical activity, setting very stringent requirements. The implementation of this agile practice would be helpful for solving different issues within the company, but it is necessary to analyze all their needs to know if it is suitable for them.

4.3.9.6 Fire drill
Agile practice: Single team
Impact: Contributes to solve the anti-pattern

Detail: Having a team capable to work on different things irrespective to their skill set, promotes the constant performance of activities, avoiding idle times and work accumulation.

4.3.9.7 Glass case plan
Agile practice: Single team
Impact: Does not contributes directly to solve the anti-pattern

Detail: Having a team which is flexible on executing different activities within a project does not have a direct impact on avoiding the lack of tracking or updating the plans.
4.3.9.8 Inflexible plan
Agile practice: Single team
Impact: Contributes to solve the anti-pattern
Detail: The plan that is prepared, can be adapted according to the needs of the sprint thanks to the versatility of the team members. It is possible to be flexible on the plans and the activities that should be performed due to the single team can execute them; avoiding that the activities tend to be stuck because it needs a specific person to execute them.

4.3.9.9 Irrational management
Agile practice: Single team
Impact: Does not contributes directly to solve the anti-pattern
Detail: The single team does not have direct inference avoiding irrational decisions or negative practices from the management. Team members perform technical activities that are not related to general project management.

4.3.9.10 Leader not manager
Agile practice: Single team
Impact: Does not contributes directly to solve the anti-pattern
Detail: Even if it is available a multidisciplinary team that can improve the performance of the sprint activities, if the PM does not follow a management methodology, the project will not have order and control.

4.3.9.11 Micro-management
Agile practice: Single team
Impact: Contributes to solve the anti-pattern
Detail: A Single team where team members take different responsibilities, solving the different issues present in a project, would help the PM create confidence in them. This agile practice aims to eliminate the PM from taking on tasks that do not directly concern him or her.

4.3.9.12 Mushroom management
Agile practice: Single team
Impact: Does not contributes directly to solve the anti-pattern
Detail: A single team promotes communication within the development team, but it does not have a direct impact on solving the lack of communication with the client.

4.3.9.13 Myopic delivery
Agile practice: Single team
Impact: Especial cases
Detail: A single team can be a bias for establishing delivery dates. The PM can have the wrong idea that fewer people execute all the pendant activities and overload them with work, this might delay the activities execution generating slips on the schedule. It is necessary to have a clear idea that, even when the team has different skill sets and might perform different activities, not only some of them are enough to accomplish the planned objectives.

4.3.9.14 Process disintegration
Agile practice: Single team

Impact: Contributes to solve the anti-pattern

Detail: On a single team scheme, the whole team cooperates and solve different problems. Both the mindset of the team and the processes that are followed for the execution of the tasks change, avoiding the failure within the project and the demotivation of the team.

4.3.9.15 Project mismanagement
Agile practice: Single team

Impact: Does not contributes directly to solve the anti-pattern

Detail: If the PM does not have a proper control or monitoring of the project, having a single team does not have a direct impact on solving or identifying this problem.

4.3.9.16 Proletariat hero
Agile practice: Single team

Impact: Contributes to solve the anti-pattern

Detail: A team who cooperate and work together to solve the problems, motivate the confidence between them and the management. The single team would help the PM trusting on his/her team to avoid the wrong idea that coercion or other malpractices will increase the productivity and the problem resolution.

4.3.9.17 Rising upstart
Agile practice: Single team

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: The single team would help to identify if there exist a person with this rising upstarts profile, trying to either not be supported by co-workers or being intrusive on other activities to stand out.

4.3.9.18 Road to nowhere
Agile practice: Single team

Impact: Does not contributes directly to solve the anti-pattern

Detail: Even when the team is capable to perform a different kind of activities, if they do not have an execution plan, is not possible to know they need to perform and how to organize their execution times. This agile practice is not suitable to avoid or identify the lack of plans.
4.3.9.19  
**Size isn’t everything**  
Agile practice: Single team  

Impact: Does not contributes directly to solve the anti-pattern  

Detail: The fact of having a multidisciplinary team does not mean that the PM will not assume that they are interchangeable affecting their performance and the project results. This agile practice does not have direct inference in solving this antipattern.

4.3.9.20  
**The brawl**  
Agile practice: Single team  

Impact: Does not contributes directly to solve the anti-pattern  

Detail: It is necessary to have a PM with leadership and management skills that can keep under control the activities performed by a multidisciplinary team. If one is present this antipattern, even implementing the single team practice, will not be a direct solution for the main problem that would represent the lack of leadership and management.

4.3.9.21  
**The domino effect**  
Agile practice: Single team  

Impact: Especial cases  

Detail: The fact of having a multidisciplinary team does not mean that the PM will not assume that the critical resources can be moved affecting and blurring project boundaries. This agile practice does not have a direct impact on solving the antipattern.

4.3.9.22  
**Ultimate weapon**  
Agile practice: Single team  

Impact: Contributes to solve the anti-pattern  

Detail: On a single team, each person takes different responsibilities that help on the constant improvement of their skills and knowledge supported by their teammates. This joint cooperation and communication can help to avoid the team relies heavily on the superstar of the team to solve the problems.

4.3.10  
**Frequent releases impact**

In this section it will be presented the matching results between the agile practice Frequent releases and the anti-patterns. On table 34, can be observed impact of the agile practice over the malpractices and subsequently the explanation of each selected impact.

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>Project Management Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequent releases</td>
<td>Absentee manager</td>
<td>✅</td>
</tr>
<tr>
<td>All you have is a hammer</td>
<td></td>
<td>❌</td>
</tr>
</tbody>
</table>
## 4.3.10.1 Absentee manager

**Agile practice:** Frequent releases

**Impact:** Contributes to solve the anti-pattern

**Detail:** Frequent releases should promote PM actions in short periods of times. Besides, it is necessary his/her active participation during each meeting where the release is presented to the client.

## 4.3.10.2 All you have is a hammer

**Agile practice:** Frequent releases

**Impact:** Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

**Detail:** The problem of trying to use the same techniques to solve all the problems affect team performance and product results. This anti-pattern can be identified with frequent releases practice, due to it will be more evident in reduced amounts of time that the problems are not being solved even when a technique is repeatedly implemented to try to avoid it.

## 4.3.10.3 Appointed team

**Agile practice:** Frequent releases

**Impact:** Contributes to identify a problem, but it might be the anti-pattern or any other issue
Detail: Having frequent releases can show in less time the case where the results obtained are not as expected and they are repeatedly wrong. This situation can be identified that the cause of this repetitive mistakes on the project might be due to the lack of integration and this agile practice would help to identify the antipattern.

4.3.10.4 Detailitis plan
Agile practice: Frequent releases
Impact: Does not contributes directly to solve the anti-pattern

Detail: Having frequent releases help to keep a track of the project progress, however, this agile practice does not have inference on how it is planned these releases and their level of detail. Therefore, it would not be useful to solve this anti-pattern.

4.3.10.5 Dry waterhole
Agile practice: Frequent releases
Impact: Does not contributes directly to solve the anti-pattern

Detail: Setting the requirements for the team selection process is an activity that is usually performed before the project begins. Frequent releases is a periodic activity performed during the project that does not deal with the selection process, due to this it would not be helpful to solve the problem of establishing stringent requirements for a job.

4.3.10.6 Fire drill
Agile practice: Frequent releases
Impact: Contributes to solve the anti-pattern

Detail: Frequent releases promote having constant progress on a project for delivering working software to the client in small amounts of time, this avoids idle times and the accumulation of work till the last minute.

4.3.10.7 Glass case plan
Agile practice: Frequent releases
Impact: Contributes to solve the anti-pattern

Detail: Due to the releases are planned for small periods of time, the tracking of what is being performed is easier. Also, it is necessary to have an updated plan, otherwise the PM would not have an idea of what was being performed from each release to present it to the user.

4.3.10.8 Inflexible plan
Agile practice: Frequent releases
Impact: Contributes to solve the anti-pattern

Detail: Each release receives feedback from the user, in case something should be changed the original release plan also need to be modified. This will facilitate to keep track of the activities and avoid inflexibility on the plan.
4.3.10.9 **Irrational management**  
Agile practice: Frequent releases  
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.  
Detail: Irrational management can be identified thanks to the frequent releases. Each release is composed by different sprints, if the PM takes irrational decisions during this period of time, some crisis might arise, and it will be evident that something is not going as expected on the project.

4.3.10.10 **Leader not manager**  
Agile practice: Frequent releases  
Impact: Contributes to identify a problem, but it might be the anti-pattern or any other issue  
Detail: The team needs the guidance and a methodology to achieve each of the planned releases. If the PM does not have experience using management methodologies, many issues might appear in subsequent occasions and it will be evident on each release, this constant failures can help to identify the anti-pattern.

4.3.10.11 **Micro-management**  
Agile practice: Frequent releases  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: Having frequent releases does not deal directly on the PM performance and their attitude about working with their team. Micro-management involves an insecurity sense from the management that promotes that the PM takes responsibility for tasks that does not concern him/her and cannot be avoided by this agile practice.

4.3.10.12 **Mushroom management**  
Agile practice: Frequent releases  
Impact: Contributes to solve the anti-pattern  
Detail: Due to each release must be reviewed and evaluated with the complete group of involved people to identify if the release was what as expected. The communication flow between the development team and the user is necessary for the feedback, then this anti-pattern should be mitigated by the frequent releases.

4.3.10.13 **Myopic delivery**  
Agile practice: Frequent releases  
Impact: Contributes to solve the anti-pattern  
Detail: It should exist a plan for each of these releases with delivery dates. However, having constant releases help to avoid the constant demand of delivering a project on a specific and immovable schedule. Each release shows the progress of the project, allowing the customer see the product growth and know that the team is working on it.
4.3.10.14  *Process disintegration*
Agile practice: Frequent releases
Impact: Contributes to solve the anti-pattern
Detail: Following the frequent releases practice would help to keep the team integrated trusting that the process they are following work as expected. The team can see the constant progress of their work, this helps to motivate the team and cooperate for the improvement and growth of the final product.

4.3.10.15  *Project mismanagement*
Agile practice: Frequent releases
Impact: Contributes to solve the anti-pattern
Detail: For each release, it is necessary to verify if it was performed what was expected on time. This agile practice helps to keep control of the activities performed between each release to have a comparison point and monitor if the progress is correct.

4.3.10.16  *Proletariat hero*
Agile practice: Frequent releases
Impact: Does not contribute directly to solve the anti-pattern
Detail: Proletariat hero involves an insecurity sense from the management that promotes that the PM use malpractices such as coercion. The fact of delivering correctly each of the releases does not mean that the PM is not applying this wrong practice to try to increase team productivity.

4.3.10.17  *Rising upstart*
Agile practice: Frequent releases
Impact: Does not contribute directly to solve the anti-pattern
Detail: Rising upstarts are team members that have an unmerited sense of superiority that might affect the project progress and team performance. Having frequent release does not have a direct influence solving or identifying teammates with this profile.

4.3.10.18  *Road to nowhere*
Agile practice: Frequent releases
Impact: Contributes to solve the anti-pattern
Detail: On frequent releases it is necessary to have a plan for each release, otherwise, the team would not know what will be delivered on each of them. On this way, this agile practice might help to avoid, in complement with other practices, this anti-pattern.

4.3.10.19  *Size isn’t everything*
Agile practice: Frequent releases
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.
Detail: The planning for each release includes how many team members will perform the activities to have an accurate estimation. If the PM is interchanging those people, it will be evident that the progress does not match with the planned releases and the problem can be identified.

4.3.10.20  The brawl
Agile practice: Frequent releases
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: If the PM has a lack of leadership or management experience, it will not be possible to keep the control on the activities within the project and the decision making might not be accurate. The frequent releases will have problems affecting the project progress but helping to identify the problem.

4.3.10.21  The domino effect
Agile practice: Frequent releases
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: The planning for each release includes how many team members will perform the activities based on their expertise to have an accurate estimation. If the PM is exchanging critical resources, it will be evident that the progress does not match and the project boundaries might be blurred by these actions, making easier to identify the problem.

4.3.10.22  Ultimate weapon
Agile practice: Frequent releases
Impact: Does not contribute directly to solve the anti-pattern

Detail: Frequent releases does not have a direct relationship solving the problem of a team relying heavily on the ultimate weapon team member. This agile practice cannot help either to identify people with this profile.

4.3.11 Common working area impact

In this section it will be presented the matching results between the agile practice Common working area and the anti-patterns. On table 35, can be observed impact of the agile practice over the malpractices and subsequently the explanation of each selected impact.

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>Project Management Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common working area</td>
<td>Absentee manager</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>All you have is a hammer</td>
<td>☐</td>
</tr>
<tr>
<td></td>
<td>Appointed team</td>
<td>✔️</td>
</tr>
<tr>
<td>Anti-pattern</td>
<td>Impact</td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>Detailitis plan</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Dry waterhole</td>
<td>❍</td>
<td></td>
</tr>
<tr>
<td>Fire drill</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Glass case plan</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Inflexible plan</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Irrational management</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Leader not manager</td>
<td>❌</td>
<td></td>
</tr>
<tr>
<td>Micro-management</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Mushroom management</td>
<td>❍</td>
<td></td>
</tr>
<tr>
<td>Myopic Delivery</td>
<td>❌</td>
<td></td>
</tr>
<tr>
<td>Process disintegration</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Project mismanagement</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Proletariat hero</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Rising upstart</td>
<td>❍</td>
<td></td>
</tr>
<tr>
<td>Road to nowhere</td>
<td>❍</td>
<td></td>
</tr>
<tr>
<td>Size isn’t everything</td>
<td>❌</td>
<td></td>
</tr>
<tr>
<td>The brawl</td>
<td>▲</td>
<td></td>
</tr>
<tr>
<td>The domino effect</td>
<td>❌</td>
<td></td>
</tr>
<tr>
<td>Ultimate weapon</td>
<td>❍</td>
<td></td>
</tr>
</tbody>
</table>

4.3.11.1  *Absentee manager*

Agile practice: Common working area

Impact: Contributes to solve the anti-pattern

Detail: Due that the team is working in the same area and the main objective is to facilitate doubt solving and decisions making, the PM is one of the main persons that must be present and have active participation.

4.3.11.2  *All you have is a hammer*

Agile practice: Common working area

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: One of the main objectives of common working area practice is to optimize doubts or problems resolution. Even when applying this agile practice, if the management always tries to solve the issues using the same techniques, this anti-pattern can be identified and later try to find solutions for the present issues and the malpractice.

4.3.11.3  *Appointed team*

Agile practice: Common working area

Impact: Contributes to solve the anti-pattern
Detail: Working in a common area facilitates communication, and consequently, the cooperation within the team. The easy doubt solving and support between the team members improve the integration solving these anti-pattern problems.

4.3.11.4  \textit{Detailitis plan}
Agile practice: Common working area
Impact: Contributes to solve the anti-pattern

Detail: All the team works constantly together, this simplifies the knowledge of what should be performed; later on the planification meetings, the team can estimate accurately delivery dates and tasks to be performed. This way the creation of complex schedules should be avoided because its creation will be based on what the team can perform.

4.3.11.5  \textit{Dry waterhole}
Agile practice: Common working area
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Having a common working area can help to identify the different profiles that work at the project. This agile practice helps for future resources selection, having this knowledge it would better the definition of the new hiring profiles.

4.3.11.6  \textit{Fire drill}
Agile practice: Common working area
Impact: Contributes to solve the anti-pattern

Detail: In case of doubts or decision making, having a common working helps to reduce highly administrative or bureaucratic issues. These activities are streamlined and idle times are avoided for the fastest execution of activities.

4.3.11.7  \textit{Glass case plan}
Agile practice: Common working area
Impact: Contributes to solve the anti-pattern

Detail: If the PM and the whole team are working in a common area, everyone has knowledge of what is being performed and it is easier for them to keep tracking on the executed activities and any possible change. All these status should be updated on the original plan, for later have an accurate planification of the following sprint.

4.3.11.8  \textit{Inflexible plan}
Agile practice: Common working area
Impact: Contributes to solve the anti-pattern

Detail: The fact of having a common working area promotes open discussion to make agile decision making and doubt solving. This practice reduces highly administrative or bureaucratic processes, introducing flexibility on them and streamlining issues resolution.
4.3.11.9  **Irrational management**  
Agile practice: Common working area  
Impact: Contributes to solve the anti-pattern  
Detail: In a common working area, the whole team is present to evaluate the problems and everyone can openly give their opinion. This help to avoid indecisiveness or wrong decision making from the PM, since his team supports him and cooperates to make the best decision altogether.

4.3.11.10  **Leader not manager**  
Agile practice: Common working area  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: The fact of having a common working area does not mean that the PM follows or not a management methodology to keep the project progress in order. This agile practice does not have a direct effect on solving or identifying this antipattern.

4.3.11.11  **Micro-management**  
Agile practice: Common working area  
Impact: Contributes to solve the anti-pattern  
Detail: The performance of the team members and the assignation of each planned activity for the sprint can be observed easily by the PM in a common working area. Therefore, this agile practice allows the PM to keep track on what it is being performed, helping him to trust in their team and avoiding that he/she wants to take tasks out of his/her scope.

4.3.11.12  **Mushroom management**  
Agile practice: Common working area  
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.  
Detail: Since many doubts can arise due to the lack of information about the requirements, this problem can be more evident in a common working area if the team keep asking the same questions regularly and stay unsolved. This agile practice can help to identify the lack of communication with the user.

4.3.11.13  **Myopic delivery**  
Agile practice: Common working area  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: Having a common working area promotes that the team works together and the communication flows in a better way. However, this does not mean that the management will not insist on the initial delivery date, provoking that this agile practice does not have a direct impact on solving this antipattern.
4.3.11.14 Process disintegration
Agile practice: Common working area
Impact: Contributes to solve the anti-pattern
Detail: The cooperation and communication are better when everyone is located in the same working area. This agile practice helps to improve the processes and the project execution, avoiding failures that might lead to a demotivated team.

4.3.11.15 Project mismanagement
Agile practice: Common working area
Impact: Contributes to solve the anti-pattern
Detail: Due to the PM is working in the same area as the whole team, it should be easier for him/her to have proper monitoring and control of the activities that are performed during the software development process.

4.3.11.16 Proletariat hero
Agile practice: Common working area
Impact: Contributes to solve the anti-pattern
Detail: A common working area is a great opportunity for the PM to be integrated and work side by side their teammates. This will promote that he/she trust in their abilities and their performance in order to avoid the wrong and unnecessary situation where coercion is applied to increase productivity and have a healthier working environment.

4.3.11.17 Rising upstart
Agile practice: Common working area
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.
Detail: Having a common working area can help the PM to identify the different profiles that are working together on the team and those who might cause trouble within the project. This would be the case of a rising upstart who would try to show up his/her abilities, affecting the progress or the relation with their teammates.

4.3.11.18 Road to nowhere
Agile practice: Common working area
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.
Detail: If there is a lack of plan the team might feel disoriented without a guide to follow and execute. Having a common working area can evidence this anti-pattern due to de constant communication and the probable situation where the team openly share their disorientation.

4.3.11.19 Size isn’t everything
Agile practice: Common working area
Impact: Does not contributes directly to solve the anti-pattern

Detail: The common working area does not have a direct impact to solve the problem of interchanging the resources having the idea that if there exist more people to develop the product it would be finished on time. This undesirable situation can still happen even when the team is working in the same area.

4.3.11.20 The brawl
Agile practice: Common working area

Impact: Especial cases

Detail: It might occur the unfortunate situation where there exists any conflict within the team members. In case the agile practice common working area is not well followed, it is possible that the problem might be aggravated due to there is not a manager with leadership that helps to solve the conflicts.

4.3.11.21 The domino effect
Agile practice: Common working area

Impact: Does not contributes directly to solve the anti-pattern

Detail: The common working area does not have a direct impact on solving the problem of exchanging critical resources, blurring the boundaries of the project. This undesirable situation can still happen even when the team is working in the same area.

4.3.11.22 Ultimate weapon
Agile practice: Common working area

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: This agile practice allows having an overview of the people coexisting and working day by day; the behavior with their teammates, such as the case of an ultimate weapon in which the team would try to over-rely their work affecting the progress of the project. Having a common working area can help the PM to identify the different profiles that are working together on the team.

4.3.12 Product roadmapping impact

In this section it will be presented the matching results between the agile practice Product roadmapping and the anti-patterns. On table 36, can be observed impact of the agile practice over the malpractices and subsequently the explanation of each selected impact.

Table 36. Product roadmapping – Project management anti-patterns

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>Project Management Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product roadmapping</td>
<td>Absentee manager</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>All you have is a hammer</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Appointed team</td>
<td>X</td>
</tr>
</tbody>
</table>
4.3.12.1 Absentee manager
Agile practice: Product roadmapping

Impact: Does not contributes directly to solve the anti-pattern

Detail: Preparing a Roadmap is an activity that is performed at the very beginning of the project. Even if the PM participate in its creation, does not mean that he/she will keep co-operating during the rest of the project.

4.3.12.2 All you have is a hammer
Agile practice: Product roadmapping

Impact: Does not contributes directly to solve the anti-pattern

Detail: The product roadmapping is a meeting where is discussed and selected the main features of the product to be developed. On these meetings is not treated the possible issues that might arise during the project development and the selected techniques to solve them.

4.3.12.3 Appointed team
Agile practice: Product roadmapping

Impact: Does not contributes directly to solve the anti-pattern

Detail: The team selection and their integration is a topic that is not treated during the creation of the product roadmap; this meeting deals with the creation of a general idea of the product to be
created based on their general features. This agile practice does not have direct inference in solving this antipattern.

4.3.12.4  **Detailitis plan**  
Agile practice: Product roadmapping  
Impact: Contributes to solve the anti-pattern  

Detail: The roadmap is a general idea of the product separated by features, there is no need to go into great detail and generate a complex schedule that is not realistic and easy to fulfill. This planning will be going on deeper level based on the planned releases and sprints.

4.3.12.5  **Dry waterhole**  
Agile practice: Product roadmapping  
Impact: Does not contribute directly to solve the anti-pattern  

Detail: Even when the establishment of the requirements for the team selection and the creation of the roadmap are activities performed at the very beginning of the project, they are not a direct relation between them. The antipattern of establishing stringent requirements for the team selection cannot be solved or identified with the creation of the roadmap.

4.3.12.6  **Fire drill**  
Agile practice: Product roadmapping  
Impact: Does not contribute directly to solve the anti-pattern  

Detail: The dates estimated for the roadmap are very general and the plan created on this practice is at a very high level. Due to this even when the team would have this general plan, it would not have the expected impact solving malpractices such as idle times and the accumulation of work.

4.3.12.7  **Glass case plan**  
Agile practice: Product roadmapping  
Impact: Contributes to solve the anti-pattern  

Detail: The product roadmapping is a guide for the team to look at their progress and calculating the number of features that should be developed. Depending on the advance of product development, this roadmap needs to be tracked and updated. The implementation of this agile practice complemented by other practices would help to solve the lack of tracking within the project.

4.3.12.8  **Inflexible plan**  
Agile practice: Product roadmapping  
Impact: Contributes to solve the anti-pattern  

Detail: The product roadmapping is one of the tools that help the team to review the progress of the project. This tool needs to be essentially flexible and open to changes, due to it is prepared at the beginning of the project but that does not mean that the development of the product will be as it is established on it from the beginning.
4.3.12.9  **Irrational management**  
Agile practice: Product roadmapping  
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: What is established on the product roadmap is a common agreement between the customer, the development team and the PM. This agreement needs to be guided by the PM; in case this takes more time or is not accurate as expected due to the management decisions, the anti-pattern can be identified.

4.3.12.10  **Leader not manager**  
Agile practice: Product roadmapping  
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: The creation of the roadmap should be led by the PM. All the treated topics and agreements about the project are part of their responsibilities; in case these activities are not as accurate as expected because of the lack of a management methodology, this practice can help to identify this antipattern.

4.3.12.11  **Micro-management**  
Agile practice: Product roadmapping  
Impact: Does not contributes directly to solve the anti-pattern

Detail: The product roadmapping is a meeting performed at the beginning of the project to have an overview of the features that will conform the final product. During this meeting is not treated the assignation of activities or who is going to perform them, therefore, this agile practice does not have direct impact solving the problems generated from the micro-management.

4.3.12.12  **Mushroom management**  
Agile practice: Product roadmapping  
Impact: Contributes to solve the anti-pattern

Detail: For the preparation of the product roadmapping, it is necessary the action of all the involved people on the project in order to discuss technical or administrative aspects. Generate a product roadmap is one of the first activities that promote the communication between the user and the development team.

4.3.12.13  **Myopic delivery**  
Agile practice: Product roadmapping  
Impact: Contributes to solve the anti-pattern

Detail: Having a roadmap gives an overview of how big and difficult is a project. The complete team will have knowledge of what should be performed, and all the involved people should be
clear that the estimations will not be definitive. Therefore, this agile practice will give to the management a guide for future accurate estimations of the delivery dates.

4.3.12.14 Process disintegration
Agile practice: Product roadmapping
Impact: Does not contributes directly to solve the anti-pattern

Detail: The creation of the product roadmap is planning meeting related to product creation. It does not have a direct impact on solving the problems derived from failed processes during the product development and the consequences that might generate within the team.

4.3.12.15 Project mismanagement
Agile practice: Product roadmapping
Impact: Contributes to solve the anti-pattern

Detail: Due to this is one of the first plans, the product roadmap works as a guide and reference to know the progress of the project. This agile practice can work as a tool to facilitate the monitoring and control of what is being developed, avoiding the lack of tracking on the project.

4.3.12.16 Proletariat hero
Agile practice: Product roadmapping
Impact: Does not contributes directly to solve the anti-pattern

Detail: The wrong activities performed by the PM such as coercion to increase team productivity, are not affairs specific to the roadmap planning. During these meetings is treated related topics to the product size and the features that will conform it and does not have direct inference avoiding management malpractices.

4.3.12.17 Rising upstart
Agile practice: Product roadmapping
Impact: Does not contributes directly to solve the anti-pattern

Detail: People with a profile such as rising upstarts try to skip learning stages or exaggerate showing up their abilities. The product roadmapping does not have a direct impact or help to identify people with this profile, is a meeting meant to create a general overview of the product that will be created.

4.3.12.18 Road to nowhere
Agile practice: Product roadmapping
Impact: Contributes to solve the anti-pattern

Detail: The product roadmap in addition to other plans helps to solve this anti-pattern. The product roadmapping is the general overview of what should be developed, it is not the definitive plan but it works as a guide to compare what is expected vs the reality according to the progress of the project on each sprint.
4.3.12.19  
**Size isn’t everything**  
Agile practice: Product roadmapping  
Impact: Contributes to solve the anti-pattern  
Detail: The roadmap must be prepared at the beginning of the project; this helps to dimension the size of the project and the number of people that should shape the team. From the technical and administrative point of view, it is a good technique to identify how many team members will be needed instead of adding or moving them once the project begins.

4.3.12.20  
**The brawl**  
Agile practice: Product roadmapping  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: The product roadmapping is a meeting where is identified which features will have the product that will be developed. The lack of leadership or management experience of the PM is malpractice that does not have a direct relation with this agile practice and cannot be solved or identified using it.

4.3.12.21  
**The domino effect**  
Agile practice: Product roadmapping  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: The stakeholders should have in mind that the product roadmapping is not a definitive plan to follow and some changes can be made on it. Unfortunately, the implementation of this agile practice, will not avoid that the management moves critical resources between projects that will be blurred the original project boundaries.

4.3.12.22  
**Ultimate weapon**  
Agile practice: Product roadmapping  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: The product roadmap does not help to solve or identify the problem that might generate an ultimate weapon. They can do great things within a project, however, this kind of people can fall into the arrogance affecting themselves and the project. The roadmap is oriented to have an overview of the project structure and the features that should be developed and does not deal with topics about team members and their profiles.

4.3.13  
**Story mapping impact**

In this section it will be presented the matching results between the agile practice Story mapping and the anti-patterns. On table 30, can be observed impact of the agile practice over the malpractices and subsequently the explanation of each selected impact.
### Table 37. Story mapping – Project management anti-patterns

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>Project Management Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Story mapping</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Absentee manager</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>All you have is a hammer</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Appointed team</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Detailitis plan</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Dry waterhole</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Fire drill</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Glass case plan</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Inflexible plan</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Irrational management</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Leader not manager</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Micro-management</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Mushroom management</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Myopic Delivery</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Process disintegration</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Project mismanagement</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Proletariat hero</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Rising upstart</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Road to nowhere</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Size isn’t everything</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>The brawl</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>The domino effect</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Ultimate weapon</td>
<td>✗</td>
<td></td>
</tr>
</tbody>
</table>

#### 4.3.13.1 Absentee manager

**Agile practice:** Story mapping

**Impact:** Contributes to solve the anti-pattern

**Detail:** During the story mapping, it is necessary for the active participation of the whole team and the PM. It is needed the experience of both parts to define the work to be done and prioritize it, this activity will help to promote the presence and action of the PM.

#### 4.3.13.2 All you have is a hammer

**Agile practice:** Story mapping

**Impact:** Does not contribute directly to solve the anti-pattern

**Detail:** Story mapping is an activity related to backlog organization. This agile practice does not have a direct impact on avoiding the scenario of management always using the same techniques to try to solve issues related to the subordinates and the project.

#### 4.3.13.3 Appointed team

**Agile practice:** Story mapping
Impact: Contributes to solve the anti-pattern

Detail: Story mapping sessions are performed periodically, which promote communication and cooperation to achieve common agreements. This practice also encourages people to work together, helping to the integration of the team members.

4.3.13.4 Detailitis plan
Agile practice: Story mapping

Impact: Does not contributes directly to solve the anti-pattern

Detail: The story mapping is an activity that seeks to keep simple the understandability of the requirements, separating the backlog into features and going deeper into different activities that should be performed. This practice does not have a direct impact on the planification of the schedule of the project and how complex it is.

4.3.13.5 Dry waterhole
Agile practice: Story mapping

Impact: Does not contributes directly to solve the anti-pattern

Detail: The story mapping is an activity to organize and understand in a better way the user requirements and it is performed during the project development. This agile practice does not have inference on the team selection process or the requirement establishment for each profile.

4.3.13.6 Fire drill
Agile practice: Story mapping

Impact: Contributes to solve the anti-pattern

Detail: The story mapping avoids the complexity and helps to understand the user requirements. This agile practice facilitates planning and helps the team to keep in mind what they need to perform in order to have an accurate estimation and avoid idle times.

4.3.13.7 Glass case plan
Agile practice: Story mapping

Impact: Contributes to solve the anti-pattern

Detail: The story mapping is a periodic session for refining and updating the backlog as the project progress. This way the team will know what they need to perform and work a complement for the constant track of what should be performed.

4.3.13.8 Inflexible plan
Agile practice: Story mapping

Impact: Contributes to solve the anti-pattern

Detail: Due to the story map should be refined and updated based on the changes performed on the backlog, it is needed the flexibility on this planification. Otherwise, the planned activities on the map might be wrong or any update might be forgotten.
4.3.13.9  Irrational management
Agile practice: Story mapping

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: During the creation of this story map, it is necessary the active participation of the team and the PM. The decisions that are taken during these sessions should be evaluated and agreed on by all of them. It can be identified this anti-pattern in case that the PM proposals are not the best option, or he/she shows indecisiveness, even if their team is there to discuss it.

4.3.13.10  Leader not manager
Agile practice: Story mapping

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: What is established on the story mapping is a common agreement between the development team and the PM. This agreement is about the functionalities and tasks that should be performed and need to be guided by the PM. In case his/her actions and performance are not as accurate as expected due to the lack of a management methodology, this anti-pattern can be identified.

4.3.13.11  Micro-management
Agile practice: Story mapping

Impact: Does not contributes directly to solve the anti-pattern

Detail: The story mapping is a classification and organization technique for the product backlog. This agile practice does not have a direct impact on the performance of the PM and their lack of knowledge in any specific topic that generates insecurity on him/her, taking responsibility of tasks that are out of their scope.

4.3.13.12  Mushroom management
Agile practice: Story mapping

Impact: Contributes to solve the anti-pattern

Detail: It is recommended the action of at least one member of each department that is involved in this project. They need to be working side by side for a better understanding of the requirements, defining accurately the activities to be performed. The creation of the story mapping can help to solve the problem of lack of communication between the user and the development team.

4.3.13.13  Myopic delivery
Agile practice: Story mapping

Impact: Does not contributes directly to solve the anti-pattern

Detail: The sessions to generate the story map are meant for the understanding and refining of the backlog. The result from these sessions is the specification of features and tasks that will be split
on each sprint. Even when this agile practice is part of the planning activities, does not have a direct inference on setting delivery dates.

4.3.13.14 Process disintegration
Agile practice: Story mapping
Impact: Contributes to solve the anti-pattern
Detail: For the generation of the story map, it is necessary the overall cooperation of the team. This practice promotes the confidence and support within the team, avoiding the failures on the project derived from a demotivated team.

4.3.13.15 Project mismanagement
Agile practice: Story mapping
Impact: Contributes to solve the anti-pattern
Detail: Story mapping is a periodic activity that works as a guide in which the team can monitor and analyze the progress for refining the following functionalities that need to be developed. This agile practice promotes the control and monitoring of the software project to have a proper refinement of the backlog.

4.3.13.16 Proletariat hero
Agile practice: Story mapping
Impact: Does not contributes directly to solve the anti-pattern
Detail: The meetings where the story map is created are meant for the refinement and understandability of the backlog. This session does not have a direct impact on solving or identifying the problem of malpractices executed by the management, such as coercion, to try to increase team productivity.

4.3.13.17 Rising upstart
Agile practice: Story mapping
Impact: Does not contributes directly to solve the anti-pattern
Detail: People with profiles such as rising upstars, are team members that might be complicated to deal with and can provoke bad impact within the project. Story mapping sessions do not have a direct relation identifying and solving the arising problems generated from this kind of people.

4.3.13.18 Road to nowhere
Agile practice: Story mapping
Impact: Contributes to solve the anti-pattern
Detail: The story mapping is a support or a complementary practice to solve the problem of lack of planning. The identification of principal features and tasks can help to create an accurate plan of what should be developed and their priorities.
4.3.13.19  Size isn’t everything  
Agile practice: Story mapping  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: Story mapping sessions are meant to organize exclusively the requirements. This agile practice does not solve or identify the problems derived from projects where it is expected project progress to be proportional to the number of people assigned to it.

4.3.13.20  The brawl  
Agile practice: Story mapping  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: Having a project manager with no leadership or management experience cannot be directly treated with story mapping meetings, this is a session for the understanding and refinement of the backlog, not for dealing with the management skills that should be improved.

4.3.13.21  The domino effect  
Agile practice: Story mapping  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: Story mapping sessions deal with the refinement and understanding of the backlog. This agile practice does not have a direct solution to avoid the exchange of critical resources between the projects the issues generated by this malpractice.

4.3.13.22  Ultimate weapon  
Agile practice: Story mapping  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: The understanding and refining of the backlog is the main objective of the story mapping meetings. These sessions do not have a direct relation on solving the problem of a team relying heavily on the superstar of the team, trusting on their knowledge and expecting their proposals as a unique solution to the treated topics.

4.3.14  Agile portfolio planning impact  

In this section it will be presented the matching results between the agile practice Agile portfolio planning and the anti-patterns. On table 38, can be observed impact of the agile practice over the malpractices and subsequently the explanation of each selected impact.

Table 38. Agile portfolio planning – Project management anti-patterns

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>Project Management Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile portfolio planning</td>
<td>Absentee manager</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>All you have is a hammer</td>
<td>❌</td>
</tr>
<tr>
<td></td>
<td>Appointed team</td>
<td>✗</td>
</tr>
</tbody>
</table>
4.3.14.1  Absentee manager
Agile practice: Agile portfolio planning
Impact: Contributes to solve the anti-pattern
Detail: The PM has to be on the lookout that the performed activities are updated on the board. Also, the agile portfolio planning can help the management to have a better overview of the progress during the sprint and, if there exist any slip within the project, find what is the cause of it.

4.3.14.2  All you have is a hammer
Agile practice: Agile portfolio planning
Impact: Contributes to identify a problem, but it might be the anti-pattern or any other issue
Detail: The agile portfolio can help to identify the persistence of problems within the project. If there is a constant failure, even when is implemented a solution that is supposed to solve the problem, might be the used techniques to solve them the ones that are not working as expected.

4.3.14.3  Appointed team
Agile practice: Agile portfolio planning
Impact: Does not contributes directly to solve the anti-pattern

<table>
<thead>
<tr>
<th>Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detailitis plan</td>
<td>✓</td>
</tr>
<tr>
<td>Dry waterhole</td>
<td>✗</td>
</tr>
<tr>
<td>Fire drill</td>
<td>✓</td>
</tr>
<tr>
<td>Glass case plan</td>
<td>✓</td>
</tr>
<tr>
<td>Inflexible plan</td>
<td>✗</td>
</tr>
<tr>
<td>Irrational management</td>
<td>✗</td>
</tr>
<tr>
<td>Leader not manager</td>
<td>✗</td>
</tr>
<tr>
<td>Micro-management</td>
<td>✓</td>
</tr>
<tr>
<td>Mushroom management</td>
<td>✗</td>
</tr>
<tr>
<td>Myopic Delivery</td>
<td>✓</td>
</tr>
<tr>
<td>Process disintegration</td>
<td>✗</td>
</tr>
<tr>
<td>Project mismanagement</td>
<td>✓</td>
</tr>
<tr>
<td>Proletariat hero</td>
<td>✗</td>
</tr>
<tr>
<td>Rising upstart</td>
<td>✗</td>
</tr>
<tr>
<td>Road to nowhere</td>
<td>✗</td>
</tr>
<tr>
<td>Size isn’t everything</td>
<td>✗</td>
</tr>
<tr>
<td>The brawl</td>
<td>✗</td>
</tr>
<tr>
<td>The domino effect</td>
<td>✗</td>
</tr>
<tr>
<td>Ultimate weapon</td>
<td>✗</td>
</tr>
</tbody>
</table>
Detail: An agile portfolio planning does not have a direct impact on integrating the team, due to it is a tracking tool for the activities executed within the sprints.

4.3.14.4 Detailitis plan
Agile practice: Agile portfolio planning
Impact: Contributes to solve the anti-pattern

Detail: Having an agile portfolio would be a perfect option to keep tracking on the activities, instead of creating a complex schedule. This kind of tools shows in a graphic way what is being performed, which is clearest for the team instead of a very specific plan that might confuse them.

4.3.14.5 Dry waterhole
Agile practice: Agile portfolio planning
Impact: Does not contribute directly to solve the anti-pattern

Detail: An agile portfolio is a tool that helps the team to have an overview of the project progress. This tool does not have a direct relation to the specification of the profiles that will form the development team.

4.3.14.6 Fire drill
Agile practice: Agile portfolio planning
Impact: Contributes to solve the anti-pattern

Detail: The agile portfolios are a visual and clear tool where it is possible to see the day by day progress of the project. Having this kind of tool will help to see the progress, have clear if it is necessary to work harder during the sprint to achieve the planned goals for it and avoid idle times.

4.3.14.7 Glass case plan
Agile practice: Agile portfolio planning
Impact: Contributes to solve the anti-pattern

Detail: The agile portfolio planning shows in an easy, graphic and understandable way the progress of the project. It helps to keep tracking of the planned activities for the sprint and store important data about the progress for future updates of the planned sprints.

4.3.14.8 Inflexible plan
Agile practice: Agile portfolio planning
Impact: Does not contribute directly to solve the anti-pattern

Detail: The agile portfolio presents the progress of what is performed during the sprint. If there exist any slip on the schedule that might affect the project, it will be presented on the board, but the tool does not probe directly that this problem is caused by the lack of flexibility within the project.

4.3.14.9 Irrational management
Agile practice: Agile portfolio planning
Impact: Does not contributes directly to solve the anti-pattern

Detail: There exist activities or situations such as team members behavior that cannot be reflected in this kind of tools. The agile portfolio can help to observe the technical progress, but not directly the people attitude. Even with the implementation of this agile practice will not avoid that the management makes wrong decisions or show indecisiveness.

4.3.14.10 Leader not manager
Agile practice: Agile portfolio planning

Impact: Does not contributes directly to solve the anti-pattern

Detail: An agile portfolio is a tool that works as a support for tracking the day by day planned activities. However, having this tool will not have any impact avoiding a disoriented team that will not know what they need to execute if the PM does not follow a management methodology.

4.3.14.11 Micro-management
Agile practice: Agile portfolio planning

Impact: Contributes to solve the anti-pattern

Detail: Having a general vision of what is being performed and the progress of the project within the sprint, can help the PM to trust on his/her team. An agile portfolio plan is a useful tool to share the executed activities and their status, with this, the PM can avoid taking responsibility for activities out of his/her scope.

4.3.14.12 Mushroom management
Agile practice: Agile portfolio planning

Impact: Does not contributes directly to solve the anti-pattern

Detail: The lack of communication between the user and the development team cannot be avoided using this tool. It is mostly used by the development team and the PM to keep tracking on the activities executed during the sprints.

4.3.14.13 Myopic delivery
Agile practice: Agile portfolio planning

Impact: Contributes to solve the anti-pattern

Detail: An agile portfolio planning let the PM observe in a graphic and clear way the progress of the team every day. This tool helps to have a vision of what is ready to be delivered and to establish better-estimated delivery dates.

4.3.14.14 Process disintegration
Agile practice: Agile portfolio planning

Impact: Contributes to identify a problem, but it might be the anti-pattern or any other issue
Detail: An agile portfolio planning can help to identify if there is a problem with the progress of the activities planned for the sprint following the current processes, demotivating the team and affecting the results of the project.

4.3.14.15  Project mismanagement
Agile practice: Agile portfolio planning
Impact: Contributes to solve the anti-pattern

Detail: An agile portfolio planning is a useful tool for the PM, it supports he/she to keep track and control the performed activities by the team during each sprint.

4.3.14.16  Proletariat hero
Agile practice: Agile portfolio planning
Impact: Does not contributes directly to solve the anti-pattern

Detail: There exist activities or situations such as team members behavior that cannot be reflected in this kind of tools. An agile portfolio is a tool is a tracking tool that does not have a direct impact solving PM attitude or wrong actions that they can take over their teams, such as be coercive to increase team productivity.

4.3.14.17  Rising upstart
Agile practice: Agile portfolio planning
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Rising upstarts are people who are always trying to perform more activities than expected, to demonstrate their abilities that not necessarily are always correct. Due to an agile portfolio planning is a tracking tool that shows what is each team member performing, it might help to identify any team member with this profile.

4.3.14.18  Road to nowhere
Agile practice: Agile portfolio planning
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: If there is a lack of a plan, the agile portfolio planning can help to track the performed activities, but it will not be possible to compare if the progress of what is being performed is correct. Consequently, the anti-pattern can be identified.

4.3.14.19  Size isn’t everything
Agile practice: Agile portfolio planning
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.
Detail: If the management is interchanging the team members, the agile portfolio would show that what was planned does not match. This tool can help to identify that moving resources affect the progress of the project.

4.3.14.20 The brawl
Agile practice: Agile portfolio planning
Impact: Does not contributes directly to solve the anti-pattern

Detail: An agile portfolio planning is a tool that tracks the performed activities within the sprint. It does not help to directly help to solve or identify the lack of leadership or management experience.

4.3.14.21 The domino effect
Agile practice: Agile portfolio planning
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: If the management is moving critical resources between the projects, the agile portfolio would show that what was planned does not match with the obtained results within the sprints. This tool can help to identify the problem and how progress can be affected.

4.3.14.22 Ultimate weapon
Agile practice: Agile portfolio planning
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: The ultimate weapon is people who have an outstanding performance which is not wrong, the problems arise when the team relies heavily on this person. This situation can be reflected in the agile portfolio planning, where is possible to see who was working on each activity. This tool can help to identify if there exist a person with this profile.

4.3.15 Agile UX impact
In this section it will be presented the matching results between the agile practice Agile UX and the anti-patterns. On table 39, can be observed impact of the agile practice over the malpractices and subsequently the explanation of each selected impact.

Table 39. Agile UX – Project management anti-patterns

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>Project Management Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile UX</td>
<td>Absentee manager</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>All you have is a hammer</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Appointed team</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Detailitis plan</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Dry waterhole</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Fire drill</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Glass case plan</td>
<td>✓</td>
</tr>
</tbody>
</table>
### Agile Practices as Solutions for Software Project Management Anti-patterns

<table>
<thead>
<tr>
<th>Anti-pattern</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inflexible plan</td>
<td>✔️</td>
</tr>
<tr>
<td>Irrational management</td>
<td>✗</td>
</tr>
<tr>
<td>Leader not manager</td>
<td>✔️</td>
</tr>
<tr>
<td>Micro-management</td>
<td>✗</td>
</tr>
<tr>
<td>Mushroom management</td>
<td>✔️</td>
</tr>
<tr>
<td>Myopic Delivery</td>
<td>✔️</td>
</tr>
<tr>
<td>Process disintegration</td>
<td>✔️</td>
</tr>
<tr>
<td>Project mismanagement</td>
<td>✔️</td>
</tr>
<tr>
<td>Proletariat hero</td>
<td>✗</td>
</tr>
<tr>
<td>Rising upstart</td>
<td>✗</td>
</tr>
<tr>
<td>Road to nowhere</td>
<td>✔️</td>
</tr>
<tr>
<td>Size isn’t everything</td>
<td>✗</td>
</tr>
<tr>
<td>The brawl</td>
<td>✗</td>
</tr>
<tr>
<td>The domino effect</td>
<td>✗</td>
</tr>
<tr>
<td>Ultimate weapon</td>
<td>✗</td>
</tr>
</tbody>
</table>

#### 4.3.15.1 Absentee manager

**Agile practice:** Agile UX

**Impact:** Contributes to solve the anti-pattern

**Detail:** The integration of the user experience into the development of a product requires feedback meetings where the client help to improve the usability experience. To achieve this, it is necessary the PM active participation and his/her leading during the meeting, promoting their action during all the project stages.

#### 4.3.15.2 All you have is a hammer

**Agile practice:** Agile UX

**Impact:** Contributes to solve the anti-pattern

**Detail:** Implementing agile UX on a project can help to find many different points of view for proposing a solution to the problems that are present on the system designing. The received feedback from the user and the discussions generated from those sessions can serve as different options to manage problems instead of trying to solve them with only one technique.

#### 4.3.15.3 Appointed team

**Agile practice:** Agile UX

**Impact:** Contributes to solve the anti-pattern

**Detail:** Implementing agile UX implies the action of the whole team to help the UX team with new ideas for designing. Due to the constant cooperation and open participation, it helps to integrate the team.

#### 4.3.15.4 Detailitis plan

**Agile practice:** Agile UX
Impact: Contributes to solve the anti-pattern

Detail: The agile UX process needs several feedback iterations with the client. This promotes constant changes in the design and functionalities of the system and, in consequence in the working plan. Therefore, is better to not have a very detailed plan, since there will be modifications within it.

4.3.15.5  *Dry waterhole*
Agile practice: Agile UX

Impact: Does not contributes directly to solve the anti-pattern

Detail: Agile UX is a designing process where the whole team and the customer help to improve the usability of the product, in this process are not included activities related to the team selection or the establishment of requirements for their profiles.

4.3.15.6  *Fire drill*
Agile practice: Agile UX

Impact: Contributes to solve the anti-pattern

Detail: The revision and usability test are performed periodically in agile UX, the received feedback is taken to improve the interfaces and the functionalities of the system. These constant modifications do not allow the team to have idle times and, in consequence, avoid delays with deliveries.

4.3.15.7  *Glass case plan*
Agile practice: Agile UX

Impact: Contributes to solve the anti-pattern

Detail: On agile UX is very important to have the feedback of the user in order to identify those things that are not easy to understand or learn from the system. All this feedback is used to implement modifications on the interfaces that may affect also the functionality, and consequently, it is needed to modify the initial plan and keep track of those changes.

4.3.15.8  *Inflexible plan*
Agile practice: Agile UX

Impact: Contributes to solve the anti-pattern

Detail: The received feedback from the reviewing sessions and usability tests are used to implement modifications and improve the system. All these activities generate changes within the originally planned project, for which it is necessary to have both a flexible plan and processes.

4.3.15.9  *Irrational management*
Agile practice: Agile UX

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.
Detail: If the PM is imposing wrong decisions it might affect the design process among other activities; these issues will be reflected on the final product. Due to agile UX has periodic revisions, these wrong decisions can arise on them and the practice can help to identify this problem.

4.3.15.10  Leader not manager  
Agile practice: Agile UX  
Impact: Contributes to solve the anti-pattern  
Detail: Agile UX by itself propose a management strategy to deal with user interfaces design and project development. Following this proposal would help the PM to start implementing management activities to improve their skills and keep the project under control.

4.3.15.11  Micro-management  
Agile practice: Agile UX  
Impact: Does not contributes directly to solve the anti-pattern  
Detail: The implementation of agile UX is a practice that seeks the integration of usability in the developed product. These activities do not represent a direct solution to solve the problem of a PM trying to be involved in many different activities beyond their responsibility.

4.3.15.12  Mushroom management  
Agile practice: Agile UX  
Impact: Contributes to solve the anti-pattern  
Detail: Agile UX defines that it is required the active involvement of the user within the process for the improvement of the final product. Therefore, using this agile practice the communication will flow and allow a constant interaction for a better understanding of the requirements and the product.

4.3.15.13  Myopic delivery  
Agile practice: Agile UX  
Impact: Contributes to solve the anti-pattern  
Detail: The constant feedback from the client in agile UX, sometimes represent changes in the product. These activities should give an overview of the project status to the management; therefore, the PM should be prepared to changes on the development and the schedule, for the new establishment of delivery dates.

4.3.15.14  Process disintegration  
Agile practice: Agile UX  
Impact: Contributes to solve the anti-pattern  
Detail: Agile UX implies constant demos and usability test where most of the team members are involved. It is a cooperative process that promotes team integration to work jointly and avoid overall demotivation.
4.3.15.15  Project mismanagement
Agile practice: Agile UX

Impact: Contributes to solve the anti-pattern

Detail: In Agile UX the customer gives feedback and based on this, some changes should be performed to improve the product. It is necessary that the management monitor those performed changes to not lose control over its creation or modification and deliver quality software.

4.3.15.16  Proletariat hero
Agile practice: Agile UX

Impact: Does not contributes directly to solve the anti-pattern

Detail: Agile UX practice helps for the interface product improvement to make it easy to learn and understand for the final user. This agile practice does not help directly to solve the problems of the PM performance or their taken actions to increase team productivity.

4.3.15.17  Rising upstart
Agile practice: Agile UX

Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: On agile UX meetings, it is usually performed brainstorming where the team members are required, to propose more ideas and different point of views. During these sessions can be possible to identify if there exist a team member with a rising upstart profile, trying to show up their knowledge and impose their ideas in a presumptuous way.

4.3.15.18  Road to nowhere
Agile practice: Agile UX

Impact: Contributes to solve the anti-pattern

Detail: The main objective of agile UX is to integrate its practices into agile schemes, for which it is necessary to have an integration plan. This agile practice can help to introduce the planning activities into the process to keep order while the integration of the methodology is performed during the project development.

4.3.15.19  Size isn’t everything
Agile practice: Agile UX

Impact: Does not contributes directly to solve the anti-pattern

Detail: Agile UX helps for the improvement of the creation of the user interfaces and their understandability. This agile practice does not have a direct impact on solving the problems related to the management wrong decisions on the assignation of the team activities or the number of people working on a problem.

4.3.15.20  The brawl
Agile practice: Agile UX
Impact: Contributes to identify the anti-pattern or the problems derived from it but does not contribute to solve it.

Detail: Introducing agile UX practices represents the integration of different working areas and teams such as UX, development, management, and users. This agile practice requires management skills to organize and lead the integration within all these different profiles, in case there exist a lack of these skills, this practice would help to identify the problem.

4.3.15.21 The domino effect
Agile practice: Agile UX

Impact: Does not contributes directly to solve the anti-pattern

Detail: Agile UX does not have direct relation solving or identifying problems related to the exchange of critical resources and the problems that these actions might generate between the projects. This agile practice is related to the improvement of the product following the UX practices, not the management activities.

4.3.15.22 Ultimate weapon
Agile practice: Agile UX

Impact: Does not contributes directly to solve the anti-pattern

Detail: This antipattern is related to specific profiles with great abilities that might provoke that the team heavily relies on them and lose their path on their assignations affecting the results on the project. Agile UX is a practice that deals with the improvement of the product, it does not have direct inference solving team issues.
Chapter 5:

Discussion
5 Discussion

In this section we are going to discuss some remarks and data obtained from the matching process performed for identify the impact and the possible solutions that the agile practices would represent to the software project management anti-patterns.

The obtained results from the matching process gave us important information that we are going to explore and review on this section for a better understanding of the impact that the agile practices might have over the project management activities more specifically the anti-patterns and their derived issues.

After the execution of this process we obtained different data that was classified into three different perspectives, this give us an overview of the general process that allow us to compare the impact from different angles. This classifications were by total number anti-patterns that impact each agile practices, total number of agile practices impacted by each anti-pattern and total number of impacted anti-patterns classified by their activities.

Apart of this classification of the results, we include a table (Table 41) where is possible observe the final matching of the Agile Practices impacting the Project Management Anti-patterns. On each column is presented the selected impact that, after evaluating them, we consider was the appropriate for the practice. We use the same iconography selected for represent the impacts, detailed on Chapter 4 in section 4.3 Consolidated list.

On this table is possible to see the classification of Project Management Anti-patterns by impacted activity, it is represented by color as follows:

- Controlling
- Motivating
- Planning
- Scheduling
- Staffing

Activities Controlling and Motivating are impacted by the anti-patterns
<table>
<thead>
<tr>
<th>Pattern</th>
<th>Daily standup</th>
<th>Sprint planning</th>
<th>Retrospectives</th>
<th>Sprint review</th>
<th>Short iterations</th>
<th>Release Planning</th>
<th>Planning Poker</th>
<th>Available product owner</th>
<th>Single team</th>
<th>Frequent releases</th>
<th>Common working area</th>
<th>Product roadmaping</th>
<th>Story mapping</th>
<th>Agile portfolio planning</th>
<th>Agile UX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fire Drill</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Glass Case Plan</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Mushroom Management</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Myopic Delivery</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Project Mismanagement</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>The Domino Effect</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Absentee Manager</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Irrational Management</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>The Brawl</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>All You Have Is a Hammer</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Micro-Management</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Process Disintegration</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Proletariat Hero</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Rising Upstart</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Ultimate Weapon</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Inflexible Plan</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Leader Not Manager</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Road to Nowhere</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Detailitis Plan</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Size Isn't Everything</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Appointed Team</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Dry Waterhole</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
</tbody>
</table>

Table 40. Final compilation of the matching process of agile practices impacting project management anti-patterns by activity
On the other hand, we include Table 40 it is exposed a little summary obtained, which present the total number of each agile practice impact over the anti-patterns by each of the impact classifications.

The matching process was performed on 15 agile practices matched with 22 anti-patterns, a total of 330 cases were studied on this project. Thus, it was found that less than half of the cases with 42% the agile practices contribute to find a solution to the anti-patterns.

The previous remark does not mean that the rest of cases will not have a positive impact. In 22% of the cases, the agile practices contribute to identify the Project Management malpractices, or their consequences. Joined, the cases where the agile practice does not contribute to solve the anti-patterns represents 35%.

Only 1% on the matches are special cases where there is not a solution and the implementation of an agile practice over an antipattern might provoke the aggravation of the problems anti-patterns. Those cases are: **Single team** over **Dry waterhole** (Section 4.3.9.5), **Single team** over **Myopic delivery** (Section 4.3.9.13), **Single team** over **The domino effect** (Section 4.3.9.21), and **Common work area** over **The brawl** (Section 4.3.11.20).

We can highlight that joining the cases where the agile practice contributes solving the anti-patterns and the cases where they contribute identifying the anti-patterns or their derived problems, more than the half of the cases have a positive impact over the malpractices. In consequence, the implementation of agile practices can bring important benefits to solve the issues generated by inadequate management practices, but it is necessary to perform an adequate analysis to select the right practices and find better solutions to reduce the problems.

<table>
<thead>
<tr>
<th>Impact</th>
<th>Number of matchings anti-patterns / agile practice</th>
<th>Percentage of matching Agile Practices / Anti-patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contribute to solve some Anti-pattern</td>
<td>138</td>
<td>42%</td>
</tr>
<tr>
<td>Do not contribute to solve some Anti-pattern</td>
<td>117</td>
<td>35%</td>
</tr>
<tr>
<td>Contribute to identify some Anti-pattern or the problems derived from it</td>
<td>71</td>
<td>22%</td>
</tr>
<tr>
<td>Especial cases</td>
<td>4</td>
<td>1%</td>
</tr>
</tbody>
</table>

### 5.1 Number of anti-pattern related to agile practices

On Figure 7 is possible to observe the first data taken from the number of anti-patterns impacted by agile practice. Most of the anti-patterns can be improved with the implementation of agile practices. We can notice that **Inflexible plan** is the most positive impacted anti-pattern, with 13 agile practices (in this case, **Daily standup**, **Sprint planning**, **Sprint review**, **Short iterations**, **Release planning**, **Planning poker**, **Available product owner**, **Single team**, **Frequent releases**, **...**).
Common working area, Product roadmapping, Story mapping, Agile UX) that can help to avoid it.

Most of the agile practices that have an impact on this antipattern are related with planning activities, which make sense due to Inflexible plan is an anti-pattern related to the lack of flexibility within the processes and the plans. Moreover, it is possible to highlight that the agile practices promote flexibility in many aspects of their implementations such as processes, management, tracking, planning, etc.

Dry waterhole is the least impacted anti-pattern with 13 agile practices (Daily standup, Sprint planning, Retrospectives, Sprint review, Short iterations, Release planning Planning poker, Available product owner, Frequent releases, Product roadmapping, Story mapping, Agile portfolio planning, Agile UX) that does not have a direct impact in solving or identifying it. Likewise, there is one agile practice that helps to identify the Dry waterhole anti-pattern which is Common working area.

Also, there exist one special case where the Single team might aggravate the problems generated by this malpractice. The reason why it was not possible to propose an agile practice as a solution for this anti-pattern using the agile practices, is because the malpractices that are related to it are performed during the team selection process or before. The agile practices are activities executed during the development process and at this point, they do not have inference on previous activities such as recruitment.

The anti-pattern that was mainly impacted with its identification by implementing some agile practices is Rising upstart. The 11 agile practices that contribute to identify it are: Daily standup, Sprint planning, Retrospective, Sprint review, Release planning Planning poker, Single team, Common working area, Agile portfolio planning and Agile UX. This anti-pattern, even when it agile practices might help to detect it, is not easy to be solved by the implementation of those agile practices, due to it is related to the performance and the profile of team members and it does not have direct relation with the developing process.

Most of the agile practices that help to identify the Rising upstart anti-pattern are more related to promote the teamwork, cooperation and constant interaction between the co-workers, which facilitate the identification of the profiles and problems related to the team performance.

The domino effect, The brawl, Rising upstart and Dry waterhole are anti-patterns that does not have an agile practice that contribute to their solution or avoidance. Nevertheless, all of them at least have one agile practice that can help to identify them. Therefore, the use of agile practices can bring some benefit to them, contributing in some way to avoid their derived problems.
Figure 5. Number of anti-patterns related to the agile practices
5.2 Number agile practices related to anti-patterns

On the other hand, on Figure 8 we can see the results from the point of view of the anti-patterns related to each agile practice. In this case we can have the opposite overview of how many anti-patterns are impacted on each of the classifications by the agile practices.

First, we can observe the most active practices that contributes to solve anti-patterns, or the issues generated by them are Sprint planning and Planning poker with 15 anti-patterns each one. Sprint planning is contributing on solving Absentee manager, Appointed team, Detailitis plan, Fire drill, Glass case plan, Inflexible plan, Irrational management, Leader not manager, Mushroom management, Myopic delivery, Process disintegration, Proletariat hero, Road to nowhere, Size isn’t everything and Ultimate weapon. Planning poker impact almost the same anti-patterns, it just varies between All you have is a hammer and Micro-management instead of Fire drill and Glass case plan.

Part of the main problems on management are derived from the lack of planification, tracking, updating or activities related to the administration of the plans. This two agile practices are oriented to deal with this kind of problems and have control over the planning activities.

In addition, the practice that has less impact solving the project management malpractices is Retrospectives having 4 contributions solving anti-patterns cases (Absentee manager, All you have is a hammer, Appointed team, Process disintegration). However, it is the most useful practice to identify anti-patterns or their derived issues; in total it helps to identify 17 anti-patterns the problems that might be caused by them which are: Detailitis plan, Fire drill, Glass case plan, Inflexible plan, Irrational management, Leader not manager, Micro-management, Mushroom management, Myopic delivery, Project mismanagement, Proletariat hero, Rising upstart, Road to nowhere, Size isn’t everything, The brawl, The domino effect and Ultimate weapon.

Retrospectives are essentially applied to identify the weakness areas on a project and the problems within the development process. For this reason, even when this agile practice does not solve the problems, at least contribute identifying them in order to propose possible solutions and keep track on them until the processes have improved.

Finally, we have found that Available Product Owner is the practice that less contribute on solving or identifying project management anti-patterns. This practice presents 16 cases where it is not suitable to identify or solve the problems derived by the malpractices. This situation can be generated due to even when the Product Owner should be available, he/she is not in charge of more activities related to the team or the development process management.

Even though, it helps to solve Glass case plan, Inflexible plan, Mushroom Management, Process disintegration, Project mismanagement and Road to nowhere. These anti-patterns are related to the client interaction and the planning of the backlog which are part of the main activities performed by the Product Owner and are also the kind of activities more promoted to be executed on agile schemes.
Figure 6. Number of agile practices related to anti-patterns
5.3 Number of impacted anti-patterns classified by their activities

Finally, during the research process of the anti-patterns list, we kept their classification per activity in order to take them as a reference and know more about the project management anti-patterns impact based on their classification. As well these activities help us to have a better understanding in which management areas the agile practices would have more influence.

For this part of the discussion, we separate the analysis per activity. The numbers presented on figures 9 to 13 are the agile practices impacting the anti-patterns classified by the influenced activities by the malpractices.

On figure 14, we can see the antipattern activity that has better results with the implementation of the agile practices is Controlling, it counts with 67 cases where the implementing agile techniques can contribute to solve or avoid the antipattern. Furthermore, the activity that has less anti-patterns solved is Staffing with 9 coincidences.

It is important to mention that Controlling is the most impacted activity by the anti-patterns but also is one of the activities that have more relation to them. It is impacted by Fire drill, Glass case plan, Mushroom management, Myopic delivery, Project Mismanagement, The domino effect, Absentee manager, Irrational management and The brawl. This result can lead us to the understanding that most of the anti-patterns that impact this activity are related to agile practices oriented to the planning and control of the software process.

Motivating activity has 54 cases where there is a not direct impact addressing malpractices and being Planning with 8 the activity that has fewer coincidences where the agile practices does not have direct inference solving or identifying the anti-patterns.

Last but not least the activity that has more cases where the agile practices help to identify the anti-pattern, or their derived issues is Motivating with 54 coincidences. Parallel that, Scheduling has only 2 cases where the agile practices help to identify their impacting malpractices.

Agile practices usually are more oriented to improve the processes within software development. These techniques are implemented and executed by the team but not much of them have direct inference dealing with the team performance. However, the agile practices can contribute to identify the anti-patterns that affect these activities to reduce or avoid their produced problems, this way would be easier for the management propose and implement solutions instead of having a growing problem.
Figure 7. Number of agile practices related to anti-patterns Controlling activity
Figure 8. Number of agile practices related to anti-patterns Motivating activity
Figure 9. Number of agile practices related to anti-patterns Planning activity
Figure 10. Number of agile practices related to anti-patterns Scheduling activity
Figure 11. Number of agile practices related to anti-patterns Staffing activity
Figure 12. Anti-patterns activities impacted by agile practices
Chapter 6:

Conclusions
6 Conclusions

The contribution that we sought performing this research project, was to propose possible solutions or recommendations to be followed in order to solve the problems derived from malpractices carried out by the management, throughout the implementation of agile practices within an organization.

For achieving this objective, we established the following questions and performed the appropriate research to answer them:

- Which agile practices can help to prevent, reduce or avoid the impact of Project Management Anti-patterns?
- Which agile practices can help to identify the Project Management Anti-patterns?
- How these agile practices can help to reduce the impact of Project Management anti-patterns?

For the execution of the central part of the project, it was necessary to perform a research and a document analysis in order to execute the systematic mapping for obtaining a consolidated list of fifteen agile practices previously explained on Chapter 2. A similar process was executed for the obtention of a consolidated list of Project Management Anti-patterns, the detail of this part of the research can be found in Chapter 3.

On Chapter 4 it is possible to find the detail of the complete process followed to perform the matching between the agile practices and the Project Management Anti-patterns to propose solutions to them. Additionally, it was necessary to search both the management activities performed on agile schemes and the responsible of them, to establish an agile PM profile. This activity was crucial to complete the analysis of the agile practices as a solution to the Software Management Anti-patterns and is also explained in detail in Chapter 4.

Reviewing the literature it was noticed how trendy it is at this moment the implementation of agile practices and their benefit within an organization [13], [14]. Consequently, after performing this study we observed all the advantages and improvements to the development processes that the use of agile practices can bring the use of agile practices; we are going to discuss in this chapter, the obtained results and some discoveries of this analysis.

The matching process that we performed to propose a solution or support the identification of the Project Management Anti-patterns, accomplish the objectives and expectations to achieve what we established as our awaiting results.

We can highlight that less than half of agile practices contribute to find a solution to the Project Management Anti-patterns. However, other parts of them are useful to identify the anti-patterns or the problems derived from them, that altogether create a good option to solve most of the issues that management areas can face during the development of a project.

The most useful agile practices that help to solve the management malpractices are Sprint planning and Planning poker. In consequence, these agile practices have also an impact on all the activities (Controlling, Motivating, Planning, Scheduling, Staffing) related to the anti-patterns.
The previous result makes sense when we talk about the activities performed during **Sprint planning** and **Planning poker** meetings. Both agile techniques are directly mainly oriented to planning, control and estimation of the activities that should be performed during the sprint, joined with open participation of the team members during the agile process.

Furthermore, the matching process gives us a total of 117 cases where agile practices that do not have a direct impact on solving or identifying the anti-patterns. However, the agile practices that don't offer a solution to the anti-patterns still have value in this study because their appearance allows us to identify problems that exist or they show us what to avoid to decrease anti-patterns.

**Dry waterhole** is the anti-pattern which has the greatest number of cases where the agile practices do not have an impact on solving it, 13 in total. This anti-pattern mostly focuses on the recruitment process, which can't easily be solved by the agile techniques. In cases where there are activities related to the development process where the team has already been selected, the agile practices prove to have less utility.

Last but not least, we identified some especial cases within the implementation of agile practices that we consider to be classified as a different impact on the anti-patterns. These special cases are situations where the team must be cautious following or using this agile practices due to, they might aggravate some anti-patterns or the problems derived from them.

Thus, **Single team** and **Common work area** are practices that should be carefully implemented, being aware of potential risks in relation to several anti-patterns.

**Single team** impact over **Dry waterhole**, **Myopic delivery** or **The domino effect** was established as an special case due to, this agile practice might increase the problems on them. Having a team which can perform activities irrespective to their skill set, might derive in a work overload or the creation of complex plans where delivery dates are immovables.

On the second case, a **Common working area** can aggravate the lack of leadership or management skills that is the case of **The brawl** anti-pattern. It might happen that the person who is responsible for these activities might not fulfill properly the need conflict resolution, provoking worst problems within the team.

Joined, it is very important to remark that the implementation of the agile practices does not imply that the specific Project Management Anti-patterns will be avoided or solved. It is necessary to perform a previous analysis to identify the biggest problems within an organization to select the most suitable techniques to solve the issues.

At the same time, it is needed to consider that some agile practices work as a support, this means that in some cases one practice not necessarily will help to solve the problem alone, also it will be needed the support of other practices that amalgamated, will become the final solution.

Apart of these findings related to the proposal of the agile practices as a solution to the Project Management Anti-patterns, we will add some important discoveries that we recognized during the development of this project.

First, **Agile practices vs engineering practices**. During the research activities, we noticed that most of the authors refer their work to the implementation of agile practices. Even so, most of the
literature mix the terms of agile practices with engineering techniques, which are not the same. The engineering techniques are technical practices to improve the quality of the software while the agile practices are oriented to the management and improvement of the development process[3].

Even when some engineering practices support agility and give quality to the developed software, sometimes it is necessary to separate them to the agile practices because they do not have the same impact on the activities related to managing the processes and the team organization.

During the development of the project, we were able to identify which of these practices were not as useful as expected for this work because it is oriented to the management area. However, it can be proposed as an extension of this project or the execution of future studies, a similar analysis that was performed on this project to propose those engineering techniques as a solution to the Project Management Anti-patterns.

Second, Identification of the Project Manager activities on agile. This study makes a relationship between traditional and agile development topics. The Software Project Management Anti-patterns are oriented to traditional schemes, where most of the management activities are the responsibility of the Project Manager. Joined, we propose as a solution to these malpractices, the implementation of agile practices that are the newest development schemes.

For performing this study, it was necessary to research and review what activities should perform the agile team representatives. It is normal to have this kind of distinction between the representatives in charge of leading agile projects due to, on agile does not exist the role of Project Manager.

On agile, the assignment of management activities depends on the expertise of the traditional PM. We consider that the most accurate role that fit better for this project was the Scrum Master. Nevertheless, we prefer to keep the PM term in order to avoid any confusion, due to we were focusing on agile practices and not in any specific agile methodology.

Finally, extracting all the activities and skills that the Scrum Master should have, we propose an equivalence of a PM in agile that execute most of the activities that the traditional Project Manager should perform. This proposal facilitates the relationship between the anti-patterns and the agile practices and can be considered for future projects that deal with traditional and agile development areas.

Third, Software project management anti-patterns. For the obtention of a consolidated list of anti-patterns, it was necessary to research about this topic to obtain different lists and execute a systematic mapping. Unfortunately, there were not enough research on this area, however, we were able to obtain a single consolidated list that was the selected to be the reference anti-patterns list.

On the other hand, project management is on the day by day activities on the development and other industry areas; besides, project management anti-patterns are as well present in most of the organizations, generating failures that can be treated if they are identified.

Sometimes, the enterprises have an endless list of issues, but the problem is that they are not identified and there is not enough information about these topics that would help to solve them. Thus, it would be enriching to have more research and development about project management anti-patterns and can be proposed topic to be updated and extended.
Even when not all the agile practices are able to solve all the management problems derived from the malpractices, at least there is one proposal for each of them. This work was accomplished positively, covering the expected results.

Carrying out this project allowed me to enrich my knowledge about project management that is part of the skills that I look constantly to improve because I find them fascinating and it is one of the software areas in which I would like to specialize.

Usually, it is most common to learn about the positive way of performing our daily activities as a Project Manager, but it is also important to know about what is incorrect within the management processes. The knowledge about the malpractices will facilitate the identification of what can be the source of the problems that are generating failures within project development.

Moreover, this project not only allows me to increase and improve my management skills but also, learn more about agile practices, their implementation, how they work and the advantages of using them. Nowadays is very important to have the proper comprehension of these techniques due to most of the organizations are interested in it and it is essential for working in software development activities.

Working on this thesis brings me satisfaction because I know that it can be a useful tool that would support the management areas within an organization. They will be able to look easily for recommendations to address potential project management malpractices.
Chapter 7:

References


7 References

7.1 Primary paper references for agile practices


7.2 Primary paper references for Software Project Management Anti-patterns


7.3 General paper references


Annexes
Annex A.- Data extraction forms for agile practices lists

In this annex are presented the data extraction forms from each selected paper that contained an agile practice list.


**Study ID:** Abbas2010  
**Article title:** Using Factor Analysis to Generate Clusters of Agile Practices  
**Author Name:** Noura Abbas; Andrew M Gravell; Gary B Wills  
**Year of publication:** 2010  
**Venue:** 2010 Agile conference

**List of agile practices:**

<table>
<thead>
<tr>
<th>Active stakeholder participation</th>
<th>Daily stand up meeting</th>
<th>Iteration task list</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture specification detailed</td>
<td>Data naming conventions</td>
<td>Iterative development</td>
</tr>
<tr>
<td>Architecture specification high level</td>
<td>Database refactoring</td>
<td>Model document reviews</td>
</tr>
<tr>
<td>Burn down chart</td>
<td>Database testing</td>
<td>Pair programming</td>
</tr>
<tr>
<td>Case tool modelling</td>
<td>Defect reports</td>
<td>Paper based modelling</td>
</tr>
<tr>
<td>Co located team</td>
<td>Defect trend metric</td>
<td>Paper models</td>
</tr>
<tr>
<td>Code inspection</td>
<td>Developer tests</td>
<td>Planning game</td>
</tr>
<tr>
<td>Code refactoring</td>
<td>Evolutionary design</td>
<td>Proved architecture early</td>
</tr>
<tr>
<td>Coding standard</td>
<td>Flexible architecture</td>
<td>Regular status report</td>
</tr>
<tr>
<td>Collective ownership</td>
<td>Gantt chart details</td>
<td>Requirements specification details</td>
</tr>
<tr>
<td>Configuration management</td>
<td>Gantt chart high-level</td>
<td>Simple design</td>
</tr>
<tr>
<td>Continuous code integration</td>
<td>Incremental delivery</td>
<td>Small releases</td>
</tr>
<tr>
<td>Continuous database integration</td>
<td>Independent confirmatory exploratory testing</td>
<td>Source code</td>
</tr>
<tr>
<td>Customer acceptance tests</td>
<td>Initial agile architectural modelling</td>
<td>Static code analysis</td>
</tr>
<tr>
<td></td>
<td>Initial agile requirements modelling</td>
<td>Sustainable pac</td>
</tr>
</tbody>
</table>

Study ID: Diebold2014

Article title: Agile Practices in Practice -A Mapping Study-

Author Name: Philipp Diebold; Marc Dahlem

Year of publication: 2014

Venue: 18th International Conference on Evaluation and Assessment in Software

List of agile practices:

<table>
<thead>
<tr>
<th>Common knowledge</th>
<th>Evolving and hierarchical specification</th>
<th>Refactoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous integration/deployment</td>
<td>Learning loop</td>
<td>Small cross-functional teams</td>
</tr>
<tr>
<td>Continuous specification analysis</td>
<td>Outcome review</td>
<td>Time boxing</td>
</tr>
<tr>
<td>Customer involvement</td>
<td>Planning meeting</td>
<td>Unattached communicative teams</td>
</tr>
<tr>
<td>Daily discussion</td>
<td>Product vision</td>
<td>Validation practic</td>
</tr>
<tr>
<td>Delivering frequent releases</td>
<td>Progress monitoring</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quality check</td>
<td></td>
</tr>
</tbody>
</table>


Study ID: AnnualStateofAgile2018

Article title: 12th Annual State of Agile report

Author Name: CollabNetVersionOne

Year of publication: 2018

Venue: 12th Annual State of Agile report

List of agile practices:

<table>
<thead>
<tr>
<th>Daily standup</th>
<th>Sprint/Iteration review</th>
<th>Planning poker/ team estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint/Iteration planning</td>
<td>Short iterations</td>
<td>Kanban</td>
</tr>
<tr>
<td>Retrospectives</td>
<td>Release planning</td>
<td>Dedicated customer/ Product owner</td>
</tr>
</tbody>
</table>
Single team (integrated dev and test) | Unit testing
---|---
Frequent releases | Coding standards
Common work area | Continuous integration
Product roadmapping | Refactoring
Story mapping | Continuous deployment
Agile portfolio planning | Pair programming
Agile/Lean UX | TDD

Automated acceptance testing
Collective code ownership
Sustainable pace
BDD
Emergent design

**Annex A4. 7th Annual State of Agile Development survey [A4]**

**Study ID:** AnnualStateofAgileReport2013

**Article title:** 7th Annual State of Agile Development survey

**Author Name:** CollabNetVersionOne

**Year of publication:** 2013

**Venue:** 7th Annual State of Agile Development survey

**List of agile practices:**

- Daily standup
- Iteration planning
- Unit testing
- Retrospective
- Release Planning
- Burndown/Team based estimation
- Velocity
- Coding standards
- Continuous integration
- Automated builds
- Dedicated product Owner
- Integrated Dev/QA
- Refactoring
- Open work area
- TDD
- Digital task board
- Story Mapping
- Kanban
- Collective Code Ownership
- Pair programming
- Automated acceptance testing
- Analog task board
- Continuous Deployment
- Agile games
- Cycle time
- BD

**Annex A5. Hybrid Development Approaches in software systems developments [A5]**

**Study ID:** Kurhmann2019
**Article title:** Hybrid Development Approaches in software systems developments

**Author Name:** Marco Kuhrmann, Joyce Nakatumba-Nabende, Rolf-Helge Pfeiffer, Paolo Tell, Jil Klünder, Tayana Conte, Stephen G. MacDonell, Regina Hebig

**Year of publication:** 2019

**Venue:** Survey research

**List of agile practices:**

<table>
<thead>
<tr>
<th>Daily standup</th>
<th>Collective Code Ownership</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration planning</td>
<td>Pair programming</td>
</tr>
<tr>
<td>Unit testing</td>
<td>Automated acceptance testing</td>
</tr>
<tr>
<td>Retrospective</td>
<td>Analog task board</td>
</tr>
<tr>
<td>Release Planning</td>
<td>Continuous Deployment</td>
</tr>
<tr>
<td>Burndown/ Team based estimation</td>
<td>Agile games</td>
</tr>
<tr>
<td>Velocity</td>
<td>Cycle time</td>
</tr>
<tr>
<td>Coding standards</td>
<td>BDD user stories</td>
</tr>
<tr>
<td>Continuous integration</td>
<td>Architecture specifications</td>
</tr>
<tr>
<td>Automated builds</td>
<td>Prototyping</td>
</tr>
<tr>
<td>Dedicated product Owner</td>
<td>Coding standards</td>
</tr>
<tr>
<td>Integrated Dev/QA</td>
<td>Refactoring</td>
</tr>
<tr>
<td>Refactoring</td>
<td>Code Review</td>
</tr>
<tr>
<td>Open work area</td>
<td>Pair programming</td>
</tr>
<tr>
<td>TDD</td>
<td>TDD</td>
</tr>
<tr>
<td>Digital task board</td>
<td>Continuous integration</td>
</tr>
<tr>
<td>Story Mapping</td>
<td>Iteration/Sprint review</td>
</tr>
</tbody>
</table>


**Study ID:** Masood2018
Article title: Adapting Agile Practices in University Contexts

Author Name: Zainab Masood, Rashina Hoda, Kelly Blincoe

Year of publication: 2018

Venue: Journal of Systems and Software

List of agile practices:

<table>
<thead>
<tr>
<th>Scrum board</th>
<th>Review meeting</th>
<th>Sprint length and weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily stan-up</td>
<td>Release planning</td>
<td>Sprint review</td>
</tr>
<tr>
<td>Self-assignment</td>
<td>Retrospective</td>
<td>Product owner</td>
</tr>
<tr>
<td>Sprint planning</td>
<td>Cross functionality</td>
<td>Pair programming</td>
</tr>
</tbody>
</table>

Annex A7. Systematic literature review on agile practices in global software development

Study ID: Vallon2017

Article title: Systematic literature review on agile practices in global software development

Author Name: Raoul Vallon, Bernardo José da Silva Estácio, Rafael Prikladnicki, Thomas Grechenig

Year of publication: 2017

Venue: Information and Software Technology

List of agile practices:

<table>
<thead>
<tr>
<th>Standup meeting</th>
<th>Continuous integration</th>
<th>Collective Code ownership</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backlog</td>
<td>Scrum of scrums</td>
<td>Simple/ incremental design</td>
</tr>
<tr>
<td>Sprint/iterations</td>
<td>Pair programming</td>
<td>Planning game</td>
</tr>
<tr>
<td>Sprint planning</td>
<td>Coding standards</td>
<td>Refactoring</td>
</tr>
<tr>
<td>Retrospective</td>
<td>Automated testing</td>
<td>Onsite/ proxy customer</td>
</tr>
<tr>
<td>Sprint review/demo</td>
<td>Test driven development</td>
<td>Requirements workshop</td>
</tr>
<tr>
<td>User stories</td>
<td>Burndown chart</td>
<td></td>
</tr>
</tbody>
</table>

Study ID: Jalali2010

Article title: Agile Practices in Global Software Engineering – A Systematic Map

Author Name: Samireh Jalali; Claes Wohlin

Year of publication: 2010

Venue: 2010 International Conference on Global Software Engineering

List of agile practices:

<table>
<thead>
<tr>
<th>Continuous integration</th>
<th>Planning meeting</th>
<th>Planning game</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standup meetings</td>
<td>Short iteration proxy</td>
<td>Close collaboration</td>
</tr>
<tr>
<td>Pair programming</td>
<td>Sprint demo</td>
<td>System metaphor</td>
</tr>
<tr>
<td>Retrospectives</td>
<td>Code standards</td>
<td>Instant messages</td>
</tr>
<tr>
<td>Scrum of scrums</td>
<td>Refactoring</td>
<td>User stories</td>
</tr>
<tr>
<td>TDD</td>
<td>Burndown charts</td>
<td>Code reviews</td>
</tr>
<tr>
<td>Sprint review</td>
<td>Virtual scrum wall</td>
<td>Acceptance tests</td>
</tr>
<tr>
<td>Backlog</td>
<td>Unit testing</td>
<td>Feature driven development</td>
</tr>
<tr>
<td>Automated testing</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Annex A9. A systematic mapping study on the combination of software architecture and agile development [A9]

Study ID: Yang2015

Article title: A systematic mapping study on the combination of software architecture and agile development

Author Name: Chen Yang, Peng Liang, Paris Avgeriou

Year of publication: 2015

Venue: The Journal of Systems and Software

List of agile practices:

<table>
<thead>
<tr>
<th>Backlog</th>
<th>Just enough work</th>
<th>Continuous integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterative and incremental Development</td>
<td>Sprint</td>
<td>Scrum meeting / daily stand up</td>
</tr>
<tr>
<td></td>
<td>Agile architecting</td>
<td></td>
</tr>
</tbody>
</table>
Agile Practices as Solutions for Software Project Management Anti-patterns
Master Thesis

Incremental delivery  
User story  
Rapid and flexible development  
Refactoring  
Retrospective  
Face-to-face communication  
Collaborative sw dev process  
TDD  
Quick and early feedback  
Use case  
Metaphor  
Release planning  
Pair planning  
Effective communication  
On-site customer  
Pair programming  
Trust in teams  
Prototyping  
Self-organized team  
Developer Story  
Informal communication  
Roadmap  
Periodic design Review  
Small team  
Simple architecture  
Goal oriented method  
Incremental re-architecting  
End-user-focused and Value-centric system design  
Poka-yoke  
Small increments and short iteration  
Featured- based and high priority task  
Eliminating waste  
Just-in-time delivery  
Spike solution  
Tacit knowledge


Study ID: Liubchenko2016

Article title: A Review of Agile Practices for Project Management

Author Name: Vira Liubchenko

Year of publication: 2016

Venue: XIst International Scientific and Technical Conference "Computer Science and Information Technologies"

List of agile practices:

Small teams  
Frequent builds  
Low-coupling packets  
Constant testing  
Constant documentation  
Iterative controls  
Ability to declare the project done on time  
Customer is the highest priority  
Provide the best value for money  
Customer participation  
Team effort  
Everything is changeable
Minimalism is essential  Determine technology  Feature growth not size growth

Annex A11.


Study ID: Alam2017

Article title: Impact and Challenges of Requirement Engineering in Agile Methodologies: A Systematic Review

Author Name: Serish Alam; Asim Ali Shah; Shahid Nazir Bhatti; Amr Mohsen Jadi

Year of publication: 2017

Venue: International Journal of Advanced Computer Science and Applications

List of agile practices:

- Direct communication
- Client interaction
- Provision of user stories
- Prioritization of user stories
- Emergence of requirements
- Features update in change management
- Continuous planning
- Requirement analysis pairing
- Shared conceptualization
- Prototyping
- Pre-testing
- Requirement management
- Review meetings


Study ID: Sampietro2016

Article title: Adoption and Evolution of Agile Practices

Author Name: Marco Sampietro

Year of publication: 2016

Venue: 5th International Scientific Conference on Project Management in the Baltic Countries

List of agile practices:

- Short iterations
- Prioritized product backlog
- Stand up/ scrum meeting
- Informal design
- Done criteria
- Task board
- Refactoring of code
<table>
<thead>
<tr>
<th>Iteration reviews/ demos</th>
<th>Stabilization iterations</th>
<th>Sustainable pace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embracing changing</td>
<td>Design inspections</td>
<td>Whole multidisciplinary</td>
</tr>
<tr>
<td>requirements</td>
<td>Emergent design</td>
<td>team</td>
</tr>
<tr>
<td>Complete features testing</td>
<td>Frequent release of</td>
<td>Planning poker</td>
</tr>
<tr>
<td></td>
<td>working software</td>
<td>TDD acceptance testing</td>
</tr>
<tr>
<td>Requirements written as</td>
<td>Negotiated scope</td>
<td>Timeboxing</td>
</tr>
<tr>
<td>informal stories</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release planning</td>
<td>Team documentation</td>
<td></td>
</tr>
<tr>
<td>Burndown chart</td>
<td>decisions rather than</td>
<td></td>
</tr>
<tr>
<td>Collective ownership of</td>
<td>planning</td>
<td></td>
</tr>
<tr>
<td>code</td>
<td>Configuration management</td>
<td></td>
</tr>
<tr>
<td>Features in iteration</td>
<td>Pair programming</td>
<td></td>
</tr>
<tr>
<td>customer valued</td>
<td>Retrospectives</td>
<td></td>
</tr>
<tr>
<td>Team velocity</td>
<td>Automated unit test</td>
<td></td>
</tr>
</tbody>
</table>


**Study ID:** Kropp2016

**Article title:** Agile Practices, Collaboration and Experience

**Author Name:** Martin Kropp, Andreas Meier, Robert Biddle

**Year of publication:** 2016

**Venue:** International Conference on Product-Focused Software Process Improvement

**List of agile practices:**

- Unit testing
- Refactoring
- Automated builds
- Coding standards
- Continuous integration
- Clean code
- TDD
- Continuous delivery
- Automated acceptance testing
- Burndown charts
- Release planning
- Onsite customer
- Iteration planning
- User stories
- Daily standup
- Task board
- Pair programming
- Collective code ownership
- Retrospective
- Open work area
- Kanban Pull system
Annex A14. A systematic literature review on agile requirements engineering practices and challenges [A14]

Study ID: Inayat2014

Article title: A systematic literature review on agile requirements engineering practices and challenges

Author Name: Irum Inayat, Siti Salwah Salim, Sabrina Marczak, Maya Daneva, Shahaboddin Shamshirband

Year of publication: 2014

Venue: Computers in Human Behavior

List of agile practices:

- Face-to-face communication
- Customer involvement
- User stories
- Iterative requirements
- Requirements priorization
- Change management
- Cross-functional teams
- Prototyping
- Testing before coding
- Requirements modeling
- Requirements management
- Review meetings and acceptance tests
- Code refactoring
- Shared conceptualizations
- Pairing for requirements analysis
- Retrospectives
- Continuous planning

Annex A15. Using CMMI together with agile software development: A systematic review [A15]

Study ID: Selleri2014

Article title: Using CMMI together with agile software development: A systematic review

Author Name: Fernando Selleri Silva, Felipe Santana Furtado Soares, Angela Lima Peres, Ivanildo Monteiro de Azevedo, Ana Paula L.F. Vasconcelos, Fernando Kenji Kamei, Silvio Romero de Lemos Meira

Year of publication: 2014

Venue: Information and Software Technology

List of agile practices:
<table>
<thead>
<tr>
<th>Daily meetings/ Stand up Meetings</th>
<th>On-site Customer</th>
<th>Product backlog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing (early)/ TDD</td>
<td>Pair programming</td>
<td>Retrospective</td>
</tr>
<tr>
<td>Continuous integration</td>
<td>Stories/ User stories</td>
<td>Sprint Review Meeting</td>
</tr>
<tr>
<td></td>
<td>Iterative development</td>
<td></td>
</tr>
</tbody>
</table>
### Annex B. Agile practices matching by name or concept

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>[A1]</th>
<th>[A2]</th>
<th>[A3]</th>
<th>[A4]</th>
<th>[A5]</th>
<th>[A6]</th>
<th>[A7]</th>
<th>[A8]</th>
<th>[A9]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily standup meeting</td>
<td>Daily standup meeting</td>
<td>Daily standup</td>
<td>Daily standup</td>
<td>Daily stand-up</td>
<td>Standup meeting</td>
<td>standup meetings</td>
<td>Scrum meeting / daily stand up</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sprint/ Iteration planning</td>
<td>iteration task list/ iterative development</td>
<td>Planning meeting</td>
<td>Sprint/ Iteration planning</td>
<td>Iteration planning</td>
<td>sprint planning</td>
<td>Sprint planning</td>
<td>Planning meeting</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Retrospectives</td>
<td>-</td>
<td>-</td>
<td>Retrospectives</td>
<td>Retrospective</td>
<td>Retrospective</td>
<td>Retrospective</td>
<td>Retrospective</td>
<td>Retrospective</td>
<td>Retrospective</td>
</tr>
<tr>
<td>Sprint/Iteration review</td>
<td>working demoable software</td>
<td>Outcome review</td>
<td>Sprint/Iteration review</td>
<td>-</td>
<td>Iteration/Sprint reviews</td>
<td>Sprint review/ Review meeting</td>
<td>Sprint review/demo</td>
<td>Sprint review/ Sprint Demo</td>
<td>-</td>
</tr>
<tr>
<td>Short iterations</td>
<td>Incremental delivery/ velocity</td>
<td>Time boxing/ Delivering frequent releases</td>
<td>Short iterations</td>
<td>Velocity/ Cycle time</td>
<td>-</td>
<td>Sprint length and weight</td>
<td>Sprint/ Iterations</td>
<td>short iteration proxy customer</td>
<td>Sprint/ Incremental delivery/ iterative and incremental development/ Small increment and short iterations</td>
</tr>
<tr>
<td>Release planning</td>
<td>Small releases</td>
<td>-</td>
<td>Release planning</td>
<td>Release planning</td>
<td>Release planning</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Release planning</td>
</tr>
<tr>
<td>Planning poker/ team estimation</td>
<td>Burn down chart/ Planning game</td>
<td>-</td>
<td>Planning poker/ team estimation</td>
<td>Agile games/Burn down/ Team based estimations</td>
<td>Burn down chart/ Expert estimation</td>
<td>-</td>
<td>Burn down chart/ Planning game/ Estimation meeting</td>
<td>Burn down chart/ Planning game</td>
<td>Trust in teams</td>
</tr>
<tr>
<td>Dedicated customer/ Product owner</td>
<td>-</td>
<td>Product vision</td>
<td>Dedicated customer/ Product owner</td>
<td>Dedicated product Owner</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Featured-based and high priority task</td>
</tr>
<tr>
<td>Single team</td>
<td>-</td>
<td>Small cross-functional teams/ Common knowledge</td>
<td>Single team</td>
<td>Integrated Dev/QA</td>
<td>-</td>
<td>cross functionality</td>
<td>-</td>
<td>close collaboration</td>
<td>Collaborative sw dev process/ Small team</td>
</tr>
</tbody>
</table>

---

Gabriela Castro Flores | 158
### Agile Practices as Solutions for Software Project Management Anti-patterns

**Master Thesis**

<table>
<thead>
<tr>
<th>Frequent releases</th>
<th>Delivering frequent releases</th>
<th>Frequent releases</th>
<th>Open workarea</th>
<th>Model checking</th>
<th>Requirements workshop</th>
<th>Roadmap</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Common work area</strong></td>
<td><strong>co located team</strong></td>
<td>Common work area</td>
<td></td>
<td>Model checking</td>
<td>Requirements workshop</td>
<td></td>
</tr>
<tr>
<td><strong>Product roadmapping</strong></td>
<td><strong>Progress monitoring</strong></td>
<td><strong>Product roadmapping</strong></td>
<td></td>
<td>Model checking</td>
<td>Requirements workshop</td>
<td><strong>Roadmap</strong></td>
</tr>
<tr>
<td><strong>Story mapping</strong></td>
<td><strong>Story mapping</strong></td>
<td><strong>User stories</strong></td>
<td><strong>User stories</strong></td>
<td><strong>User stories</strong></td>
<td><strong>User story</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Agile portfolio planning</strong></td>
<td><strong>Agile portfolio planning</strong></td>
<td><strong>Digital</strong></td>
<td><strong>Limit work in progress/ Definition done/ready</strong></td>
<td><strong>Scrum board</strong></td>
<td><strong>Virtual scrum wall</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Agile UX</strong></td>
<td><strong>whiteboard sketches/sketching modelling/ Simple design/ evolutionary design/ paper based modelling/ Paper models</strong></td>
<td><strong>Agile/ Lean UX</strong></td>
<td><strong>Design reviews</strong></td>
<td><strong>Simple/ Incremental design</strong></td>
<td><strong>Periodic design Review/ Prototyping/ End-user-focused and value-centric system design</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Agile Practice</strong></th>
<th>[A10]</th>
<th>[A11]</th>
<th>[A12]</th>
<th>[A13]</th>
<th>[A14]</th>
<th>[A15]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Daily standup</strong></td>
<td>-</td>
<td>-</td>
<td>stand up/ scrum meeting</td>
<td>Daily standup</td>
<td>-</td>
<td>Daily meetings/ Stand up Meetings</td>
</tr>
<tr>
<td><strong>Sprint/ Iteration planning</strong></td>
<td>-</td>
<td>Continuous planning</td>
<td>-</td>
<td>Iteration planning</td>
<td>Continuous planning</td>
<td>-</td>
</tr>
<tr>
<td><strong>Retrospectives</strong></td>
<td>Retrospectives</td>
<td>Retrospective</td>
<td>Retrospectives</td>
<td>Retrospective</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sprint/Iteration review</strong></td>
<td>Iterative controls</td>
<td>Review meetings</td>
<td>iteration reviews/ demos</td>
<td>Review meetings and acceptance tests</td>
<td>Sprint Review Meeting</td>
<td></td>
</tr>
</tbody>
</table>

Gabriela Castro Flores
<table>
<thead>
<tr>
<th>Short iterations</th>
<th>Timeboxing/ team velocity/ Short iterations</th>
<th>Continuous delivery</th>
<th>Iterative development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release planning</td>
<td>Release planning</td>
<td>Release planning</td>
<td></td>
</tr>
<tr>
<td>Planning poker/ team estimation</td>
<td>Burn down chart/ Planning poker</td>
<td>Burn down chart</td>
<td></td>
</tr>
<tr>
<td>Dedicated customer/ Product owner</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single team</td>
<td>Small teams</td>
<td>whole multidisciplinary team</td>
<td>Cross-functional teams</td>
</tr>
<tr>
<td>Frequent releases</td>
<td>Frequent builds</td>
<td>frequent release of working sw</td>
<td></td>
</tr>
<tr>
<td>Common work area</td>
<td>Direct communication</td>
<td>Synchronous communication</td>
<td>Open work area</td>
</tr>
<tr>
<td>Product roadmapping</td>
<td>Feature growth not size growth</td>
<td>task board</td>
<td>Face-to-face communication</td>
</tr>
<tr>
<td>Story mapping</td>
<td>Provision of user stories/ Priorization of user stories</td>
<td>Requirements written as informal stories</td>
<td>User stories</td>
</tr>
<tr>
<td>Agile portfolio planning</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agile UX</td>
<td>Prototyping</td>
<td>Informal design/ design inspections/ emergent design</td>
<td>Prototyping</td>
</tr>
</tbody>
</table>
Annex C. Software project management anti-patterns extended definitions

Absentee Manager

Anti-pattern name: Absentee Manager

Optional name:

Definition: Any manager who engages in avoidance behavior or is invisible for long periods of time – either hiding on premises or away from the office. No one wants the constant intrusion of management, it indicates a lack of trust in the workforce, but there are obviously times when management must be visible because they are the primary decision makers. Why key managers cannot be found, subordinates are left to make crucial decisions or those decisions are delayed. Either way, the manager's continued absence is hindering, or even damaging the company. Furthermore, it is demoralizing to employees when their boss is not putting in the time.

All You Have Is a Hammer

Anti-pattern name: All You Have Is a Hammer

Optional name: One-trick Pony

Definition: One-dimensional management, where the same techniques are used on all subordinates and in all situations. In reality, if all you have is a hammer, everything is a thumb!

One of the biggest mistakes you can make is to impose your own values and win conditions on someone else. You might be motivated by money, or title, or whatever… someone else may not be, and assuming that everyone holds the same motivations, anxieties, and needs ensures that most of your subordinates are unhappy and unmotivated.

Appointed Team

Anti-pattern name: Appointed Team

Optional name:

Definition: The false assumption that a group of people selected by management will immediately gel and become a team. There are no perfect criteria for screening team members.

However, when teams are appointed according to management insight, needs, and biases, it often results in disempowered teams unwilling to take extraordinary measures to meet project goals.

Detailitis Team
**Anti-pattern name:** Detailitis Plan

**Optional name:** Death by Planning

**Definition:** Excessive planning for software projects leads to complex schedules that cause downstream problems.

Sometimes the solution to effective delivery is regarded as a high degree of control via a continuous planning exercise that involves most of the senior developers, as well as the managers. This approach often evolves into a hierarchical sequence of plans, which show additional (and unnecessary) levels of detail. The ability to define such a high level of detail gives the perception that the project is fully under control. In reality, software projects are rarely fully under control.

---

**Dry Waterhole**

**Anti-pattern name:** Dry Waterhole

**Optional name:**

**Definition:** You get into the habit of specifying stringent requirements for a job when it is not strictly necessary. Over time this habit spreads to other employers, and the pool of available talent dries up as lesser experienced people are denied the opportunity to get experience.

---

**Fire Drill**

**Anti-pattern name:** Fire Drill

**Optional name:**

**Definition:** Airline pilots describe flying as “hours of boredom followed by 15 seconds of sheer terror.” Many software projects resemble this situation: “Months of boredom followed by demands for immediate delivery.” The months of boredom may include protracted requirements analysis, re-planning, waiting for funding, waiting for approval, or any number of techno-political reasons.

---

**Glass Case Plan**

**Anti-pattern name:** Glass Case Plan

**Optional name:** Death by Planning

**Definition:** Often a plan produced at the start of a project is always referenced as if it’s an accurate, current view of the project even if it’s never updated. This practice gives management a “comfortable view” of delivery before the project starts. However, when the plan is never tracked against, nor updated, it becomes increasingly inaccurate as the project progresses.

This false view is often compounded by the absence of concrete information on progress, which often is known only after a critical deliverable slips its schedule.
Inflexible Plan

Anti-pattern name: Inflexible plan

Optional name: Chaos, The More Things Change

Definition: Software development plans and processes cannot predict all possibilities that can occur and therefore cannot be followed to the letter to overcome unpredicted problems. A project manager and other managers need to be flexible to ensure that when risks are identified they can be tackled in a flexible, pragmatic, and effective manner. Once chaos starts it is very difficult to restrain. Flexibility of management is the key to resolving this anti-pattern.

Irrational Management

Anti-pattern name: Irrational Management

Optional name: Headless Chicken, Pathological Supervisor, Short-Term Thinking, Managing by Reaction, Decision Phobia, Managers Playing with Technical Toys

Definition: Habitual indecisiveness and other negative management habits lead to de facto decisions and chronic development crises.

Irrational Management covers a range of commonly occurring software project problems that can be traced back to the personalities of the person(s) running the project. For example, the manager may have obsessive interests in some aspect of the technology or personality limitations that cause them to become ineffective or irrational managers. Irrational management can be viewed as a skewed set of priorities where the manager’s personal priorities, no matter how nonsensical, guide the software project in irrational directions.

Leader Not Manager

Anti-pattern name: Leader Not Manager

Optional name:

Definition: Being a great leader does not necessarily mean being a great manager. This anti-pattern illustrates the problem of having vision but no plan.

Inspiring leadership is very important, but so are the day-to-day operations of an organization, project, or team and to do them effectively requires management acumen that even great leaders can lack. Some might view these tasks as drudgery, but while leaders inspire, managers organize, plan, supervise, and advise.

Micro-Management

Anti-pattern name: Micro-Management
Optional name: Unbalanced People, Technology and Process, Attrition ‘R’ Us

Definition: Many project managers do not understand how to manage people. They are sometimes excellent technical staff who are promoted or line managers who are given a change of role without necessarily having the required set of skills and experience to manage a software development team.

Project managers that over-manage particular aspects of the project, aspects under the responsibility of subordinates, are likely to do so either because they are weak in that particular area and believe that a micro-focus will mitigate risks or because it is the one skill area that they have.

The range of micro-management can vary significantly; a fire drill is an occasional form of micro-management, while managing daily developer tasks is an extreme form.

Often this leads to the loss of staff, and the managers responsible are surprised because they do not understand the negative effect of their actions on the developers’ response.

Mushroom Management

Anti-pattern name: Mushroom Management

Optional name: Pseudo-Analysis, Blind Development

Definition: In some architecture and management circles, there is an explicit policy to isolate system developers from the system’s end users. Requirements are passed second-hand through intermediaries, including architects, managers, or requirements analysts. Mushroom Management assumes that requirements are well understood by both end users and the software team at project inception. It is assumed that requirements are stable.

Myopic Delivery

Anti-pattern name: Myopic Delivery

Optional name: Delivery Zone

Definition: Slipped schedules that are faster than can be controlled, generating project compression.

Project compression creates a gap in the space-time continuum, which allows the incomprehensible to be accomplished in the minds of management. It causes all logical, pragmatic, rational, reasonable thinking to fly out the window. Management will do anything to meet the delivery date. No one’s life is un-expendable. Everyone will suffer. Nothing is more important than the delivery date. This anti-pattern discusses what happens when the schedule slips and management continue to demand the original delivery date.

Process Disintegration

Anti-pattern name: Process Disintegration
Optional name: The Process Club

Definition: The dilemma of how to handle failing processes. Despite its name this anti-pattern is primarily a people-caused anti-pattern that has process consequences. As process advocates in a seeming process-resistant world, we are often faced with the dilemma of just how to handle situations in which a group of professionals is determined to pull defeat from the jaws of victory, and all in the name of principle.

Many times we have stood by helplessly as seasoned professionals managed to disrupt otherwise functional processes, preventing success and souring everyone’s attitude.

Sometimes these people were just corncobs (i.e., difficult people), and when recognized as such could be appropriately handled. But what is going on when everyone seems to have lost it and there is no longer any reasonable hope of continuing without failure? This is where we are likely facing a Process Disintegration anti-pattern.

Project Mismanagement

Anti-pattern name: Project Mismanagement

Optional name: Humpty Dumpty

Definition: Inattention to the management of software development processes can cause lack of direction and other symptoms. Proper monitoring and control of software projects is necessary to successful development activities. Running a product development is as complex as creating the project plan; and developing software is as complex as building skyscrapers, involving as many steps and processes, including checks and balances. Often, key activities are overlooked or minimized.

This anti-pattern concerns the monitoring and controlling of software project. The timeframe for this occurs after planning activities, and during the actual analysis, design, construction, and testing of the software system. Project mismanagement involves mistakes made in the day-to-day running of a project, assuming planning errors (such as Detailitis Plan and Glass Case Plan) have not been made.

Proletariat Hero

Anti-pattern name: Proletariat Hero

Optional name:

Definition: The “everyman” worker is held up as the ideal, when in reality, he is a prop being used to mask inadequacies of management. A form of labor discipline as a means of “motivating” staff that gives an excuse for management to raise output expectations… get more with less.

The basic, flawed assumption at work in the plight of the Proletariat Hero is that people are lazy. Managers who believe this then utilize some form of labor discipline that coerces workers to become more efficient. It assumes that people will work at the slowest rate that is not punished. It treats workers as unthinking cogs in a machine and is a mindset
solely interested in efficiency. Paradoxically, removing individuality and creativity from work can often reduce productivity as people become bored and inefficient.

**Rising Upstart**

**Anti-pattern name:** Rising Upstart

**Optional name:**

**Definition:** Rising upstarts are superstars that cannot wait their time and want to forego the requisite time to learn, mature, and find their place. This can sometimes be through ignorance (they do not know what they do not know), and sometimes it is through impatience (they know what others do not know). The Rising Upstart presents a real challenge to all but the most proficient managers.

Uncontrolled Rising Upstarts can cause an imbalance in an organization, a situation that can be difficult to manage. A happy equilibrium must be found that allows the cream to rise to the top in step with their talents (artificially holding them down will force these valuable assets to leave), but not too quickly that they prematurely crash and burn or offend the still effective old guard.

**Road to Nowhere**

**Anti-pattern name:** Road to Nowhere

**Optional name:**

**Definition:** The lack of a plan causes confusion and a crisis of leadership.

Without a plan, how can you organize to achieve your goals? How can you properly partition the work that must be done and track its progress? How do you get buy-in from your superiors, subordinates, and peers for the direction in which you are taking them? Furthermore, if you somehow manage to meet your goals, even by luck, how will you be able to repeat that success, learn from mistakes, and extract best processes? It is exceptionally difficult to make even the simplest of tactical decisions when a broader plan is not in place – it is like walking in the dark with your hand outstretched.

Even those managers who prefer *ad hoc* techniques and, through their hard work and luck, sometimes manage to be successful, the dissatisfaction and frustration of everyone else that has to work “blind” is palpable. Moreover, how are you going to learn from someone who does not even know why he succeeds himself?

**Size Isn’t Everything**

**Anti-pattern name:** Size Isn’t Everything

**Optional name:** How To Have a Baby in One Month with Nine Women

**Definition:** It is critical to allocate the appropriate number of staff to the different phases of a project; schedule too many staff at the wrong time and the productivity level drops
significantly; schedule too few staff and the deliveries are late. Delivery delays, cost overruns, and technical failure can result from not planning according to the chosen software development lifecycle. The overlap of software lifecycle phases can also have a significant impact on planned deliveries if staffing isn’t increased accordingly.

Originally identified by Brooks in the *Mythical Man-Month* (1979), it involves the assumption that developers are interchangeable and that the number of people working on a project is inversely proportional to development. Brooks discusses how this assumption is false and presents what he calls Brooks’ Law: “Adding manpower to a late project makes it later” [Brooks, 1979].

**The Brawl**

**Anti-pattern name:** The Brawl  
**Optional name:** The Anti-Patton  
**Definition:** There are at least two distinct aspects to being a project manager: leadership and management. It is important that the two are understood and that effective leadership and management skills are applied to achieve success. A poor manager or leader typically results in project failure. However, leadership, more than management, can save a project. Often the battle cry of the engineer or the programmer is, “We need a leader, not a manager,” or “There is no leadership.” When this is the case, development teams muddle about, stagnate, and fail to attain their goals. This festering often results in a brawl where each faction of the development team attempts to position themselves into a leadership role.

Management cliques are very common and always negatively impact those around them because of the politics involved, causing a dysfunctional work environment. Management hierarchies are often based on personal relationships (so-called “old boys’ clubs”) and manager’s favorite views of software development practices regardless of their management skill and experience.

**The Domino Effect**

**Anti-pattern name:** The Domino Effect  
**Optional name:** Crap Rolls Downhill  
**Definition:** Project managers who treat the resources for each project for which they are responsible in a collective manner, blur project boundaries causing a domino effect of problems. The need to continually move the critical resources to cover gaps left by moving them from a previous project leads to software delivery delays, developer frustration, and lack of confidence in management, and in extreme cases, to developer attrition and project failure.

**Ultimate Weapon**

**Anti-pattern name:** Ultimate Weapon
**Optional name:**

**Definition:** Phenoms can be relied upon so much by their peers or organization that they become the conduit for all things.

Ultimate Weapons can do great things, but often they know how great they are and their arrogance and self-righteousness can be just as great. If these individuals are not managed effectively, the harm they cause can overshadow their achievements.

Despite their talents, even a superstar cannot win on their own all the time, and moreover, the Ultimate Weapon’s teammates can fall second-rate and forgotten.

Managers can become so seduced by the prodigious talent of these individuals that they involve them in everything. They become so important to their project, team, or even company, that without them success seems impossible.