POLITECNICO DI MILANO

School of Industrial and Information Engineering
Master of Science in Management Engineering

Agile Scrum Survey at pagoPA Project: Challenges and Opportunities

Supervisor: Luca Gastaldi (Politecnico di Milano)

Co-Supervisors: Ivan Moro (SIA S.p.A)

Amato Sorrentino (SIA S.p.A)

Master Thesis by:
Fabian Sambrano Aguilar
Matr. 915815

Academic Year 2020-2021

CONTENTS

ABSTRACT

Agile methodologies have been steadily on a rise around the world, especially in the digital business sector where it is quite predominant. A significant amount of companies in Italy has started to implement it for their projects and to institutionalize its principles and values. The Public Administration is no exception to this trend and with its recent creation of pagoPA, a digital payment platform for the public entities; Agile frameworks have come into deep consideration for this context.

In this work, we study the implementation of Agile Scrum in the company that works on the internal management of pagoPA project. To do so, we surveyed the members of both teams working in the business unit and we analysed their answers to convey a comprehensible look at the current situation. The main topics included were Scrum perception during its implementation, knowledge level of Scrum mechanisms, participation at Scrum events, teamwork and team interactions, framework issues and possible ways to improve. Smartworking was also added since it was a major recent change that the company had to undergo in its way of working during the global pandemic outbreak of 2020.

We found out that, even if Scrum is usually regarded as a flexible methodology applicable to many companies, teams tend to adapt the framework into their pre-established practices and principles. The teams frequently view Scrum favourably and look forward for a deeper agile implementation. However, this teams' desire is not always matched with the client's needs or even the company's hierarchy and internal organization. It is possible to achieve a full Scrum implementation, but it requires the effort of not only a well-prepared cross-functional team with a highly acknowledgeable Product Owner and a supporting Scrum Master; it should also be part of the top management decision to implement it effectively on the company. This is the only way to achieve a final and true Agile transformation for any given team in a large company.

Key words: Agile principles, Scrum framework, Waterfall, software development, Sprint, Product Owner, Scrum Master, teamwork, Scrum events, Smartworking

SOMMARIO

Le metodologie Agile sono in constante crescita nel mondo, specialmente nel settore del digital business in cui è fortemente predominante. Un numero significativo di aziende in Italia ha iniziato a implementarle nei loro progetti ed a istituzionalizzare i principi ed i valori legati. La Pubblica Amministrazione non è stata l'eccezione a questa tendenza e, con la recente creazione di pagoPA, la piattaforma di pagamenti digitali per gli Enti pubblici, i framework Agile sono stati presi in considerazione per questo contesto.

In questa opera, l'implementazione di Agile Scrum è stata esaminata nell'azienda che lavora nella gestione interna di pagoPA come progetto. Per fare ciò, è stato elaborato un sondaggio ai membri di entrambi team che lavorano nella business unit e le loro risposte sono state analizzate per trasmettere una visione completa della situazione attuale. Gli argomenti principali trattai sono stati: percezione della Scrum durante l'implementazione, livello di conoscenza dei meccanismi Scrum, partecipazione negli eventi Scrum, collaborazione ed interazioni nel team, problemi del framework e miglioramenti possibili. Lo Smartworking è stato aggiunto dato il cambio recente a cui l'azienda si è sottoposta nella sua forma di lavoro a causa dello scoppio della pandemia globale nel 2020.

È stato trovato che, sia la Scrum usualmente considerata come una metodologia flessibile ed applicabile ad una molteplicità di aziende, i team tendono sempre ad adattare il framework dentro le loro pratiche e principi previamente stabiliti. Tendenzialmente, i membri dei team valutavano favorevolmente la Scrum e guardano avanti verso una implementazione profonda dell'Agile. In ogni caso, questo desiderio del team non è sempre abbinato con le esigenze del cliente oppure con la gerarchia interna dell'organizzazione. Per avere un'implementazione completa dello Scrum, oltre ad un team cross funzionale con un Product Owner ben preparato e uno Scrum Master di supporto; è necessario che la direzione aziendale sia in grado di implementarlo in forma efficace all'interno dell'azienda. È questa l'unica maniera di raggiungere una vera e propria trasformazione Agile per qualsiasi gruppo di lavoro in una grande impresa.

Parole chiavi: principi Agile, framework Scrum, Waterfall, sviluppo software, Sprint, Product Owner, Scrum Master, teamwork, eventi Scrum, Smartworking

INTRODUCTION

To contextualize, in the first chapter we will start looking at the "Piano triennale per l'Italia Digitale" which will allow us to understand how the Public Administration has established a well-defined strategy for digitalizing every region of Italy. This is where pagoPA comes in as a platform to perform any kind of payment towards a Public entity. We will look at pagoPA as a project in managerial and technological aspects where we will discover the Payment Node (NodoSPC) and its main components. We will close the chapter by including a rapid view of project management and the Eisenhower matrix.

In the second chapter, we will introduce the Agile value and principles, announced in the Agile Manifesto. Then we will understand in detail the Scrum framework by looking at its events, artefacts and pillars. This part will allow us to have a good theoretical understanding of the topics discussed in the research done. At the end, we will briefly review the software development lifecycle and its main phases.

The third chapter starts with a more practical approach since we will be introduced to the actual operation of the business unit (BU) that is in charge of delivering and evolving the NodoSPC. This BU is not part of PagoPA Company but is held by SIA S.p.A at the current status of writing this document. For this section, we will understand the internal work division into two Scrum teams: Team Nodo and Team PM. We will also enter in detail of some common issues and challenges that the BU faces using a case study. Finally, we will recognize the influence of Smartworking as a response to keep the company working during the global pandemic stated in February 2020.

In the fourth chapter, we will make a bibliographical review of Scrum-related research. We will appreciate how these authors took practical and conceptual approaches to study some aspects related to Scrum teams. Among these studies we have teamwork models, Product Owner role analysis, comparisons with other Agile and non-Agile methodologies in the IT sector, conceptual layouts for the work environment, governance and scalability of the framework in multiple Scrum teams and finally a view into the issues and the adaption process of Scrum during the Covid-19 outbreak.

Methodology will be the central topic of the fifth chapter. Here we will be introduced to the Scrum Survey 2020, a research project that started in June and yielded some interesting results in October that same year. For this, we will initiate the survey by using a series of open questions that were revised, grouped and re-formulated. We will grasp the general procedure of the survey in order to visualize the results in the sixth chapter. Now, the whole compendium of results will be presented into three topic groups: Scrum, Task and interactions and Smartworking. This whole chapter will explain in detail how the data was obtained and analysed. This part will also include hintful insights to guide the reading and empathize with the working contexts of the teams.

Finally, in the last chapter we will wrap up the data collected into final discussions that will help us to draw some conclusions of the Scrum framework implementation in this case study and will lead us to create our own perspectives and points of view to implement them in other companies with similar case studies. Further research is suggested to continue understanding the effects of implementing Agile in real life.

LIST OF FIGURES

LIST OF GRAPHS

1 A BRIEF VIEW INTO PAGOPA PROJECT

1.1 PIANO TRIENNALE PER L'ITALIA DIGITALE

The Three-Year Plan for IT in the Public Administration (Piano triennale or Plan) is an essential tool for promoting the digital transformation of the country and, in particular, that of the Italian Public Administration. (Agenzia per l'Italia Digitale, 2021)

The 2020-2022 Piano Triennale represents the natural evolution of the two previous Plans. The first edition emphasized the introduction of the IT Strategic Model in the PA and the second edition aimed to detail the implementation of the model. Now, the new Plan focuses on the implementation of the planned actions.

The Covid-19 emergency required the Administrations to proceed quickly with the activation of procedures for widespread smart working, which involved over 75% of employees. The lockdown led to a new and sudden working condition for the employee, which brought out some critical issues in the use of technological tools (previously not detectable given the possibility of obtaining immediate support in the office). In addition, in many areas it highlighted the need to thoroughly review the organization of processes, favouring the sharing of documents and work materials online (Agenzia per l'Italia Digitale, 2020).

The strategy of this plan is aimed at:

- foster the development of a digital society, where services, **such as payments**, put citizens and businesses at the center **through an all-inclusive app**, through the digitization of the **public administration**, which is the engine of development for the whole country.

- promote sustainable, ethical and inclusive development, through innovation and digitization at the service of people, communities and territories.

- contribute to the spread of new digital technologies, such as different and diverse forms of payment, in the Italian production fabric, encouraging standardization, innovation and experimentation in the field of public services.

While in continuity with the previous one, the 2020-2022 Piano Triennale introduces an important innovation with reference to the recipients of the objectives identified for each of the issues addressed. In fact, the individual administrations will have to carry out the actions to achieve the objectives contained in the Plan.

The Plan is also characterized by a strong emphasis on measuring results. The culture of measurement and consequently of data quality becomes one of the main reasons for this approach.

The guiding principles of the Plan are:

- digital & mobile first for services ensuring access via SPID
- cloud first to prevent the risk of lock-in
- inclusive and accessible
- security and privacy by design
- user-centric, data driven and ***agile***
- only ask information once
- public data as a common good
- open code and available.

The 2020-2022 Piano Triennale, licensed by the Minister for Technological Innovation and Digitization, was drawn up by a large working group made up of staff from AgID and the Department for Digital Transformation and saw the contribution of many central administrations, metropolitan regions and cities.

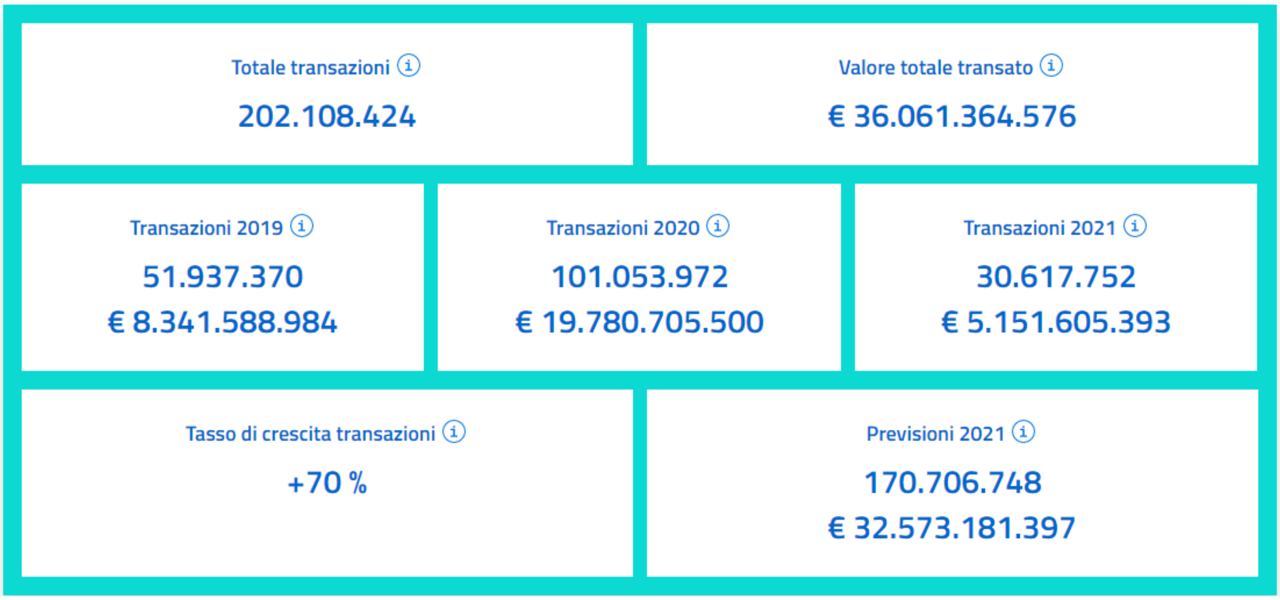
One of the main projects derived from the Piano Triennale is the pagoPA project. Since it is a digital payment platform for the Public administration is completely framed within the digitalization plan for Italy. We will discuss more on the next chapter.

1.2 WHAT IS PAGOPA?

PagoPA is the electronic payment system that makes any payment to the Public Administration (PA) easier, safer and more transparent. Citizens can pay directly on the Institution's website or app or through the online or physical channels of banks and other Payment Service Providers (PSP).

PagoPA allows any Italian citizen to pay taxes, utilities, fees, stamps and any other type of payment to central and local Public Administrations (PA), but also to other institutions, such as companies with public participation, schools, universities, and ASLs. (PagoPA S.p.A., 2020)

As described in the Piano Triennale, PagoPA grows as a project of digital transformation in a context where digital services should increase its coverage and impact, while the payments and duties for a regular citizen are facilitate and agilized. During recent years, the project has had an important increase in its number of transactions which is expected to keep growing (Fig 1). Meanwhile the most frequent Public entities and Banks – Payment Service Provider are shown in Fig 2 with their number of transactions.



Totale transazioni ⓘ
202.108.424

Valore totale transato ⓘ
€ 36.061.364.576

Transazioni 2019 ⓘ
51.937.370
€ 8.341.588.984

Transazioni 2020 ⓘ
101.053.972
€ 19.780.705.500

Transazioni 2021 ⓘ
30.617.752
€ 5.151.605.393

Tasso di crescita transazioni ⓘ
+70 %

Previsioni 2021 ⓘ
170.706.748
€ 32.573.181.397

Fig 1 Total number of transactions over the past few years and the expected growth factor (PagoPA S.p.A., 2020)

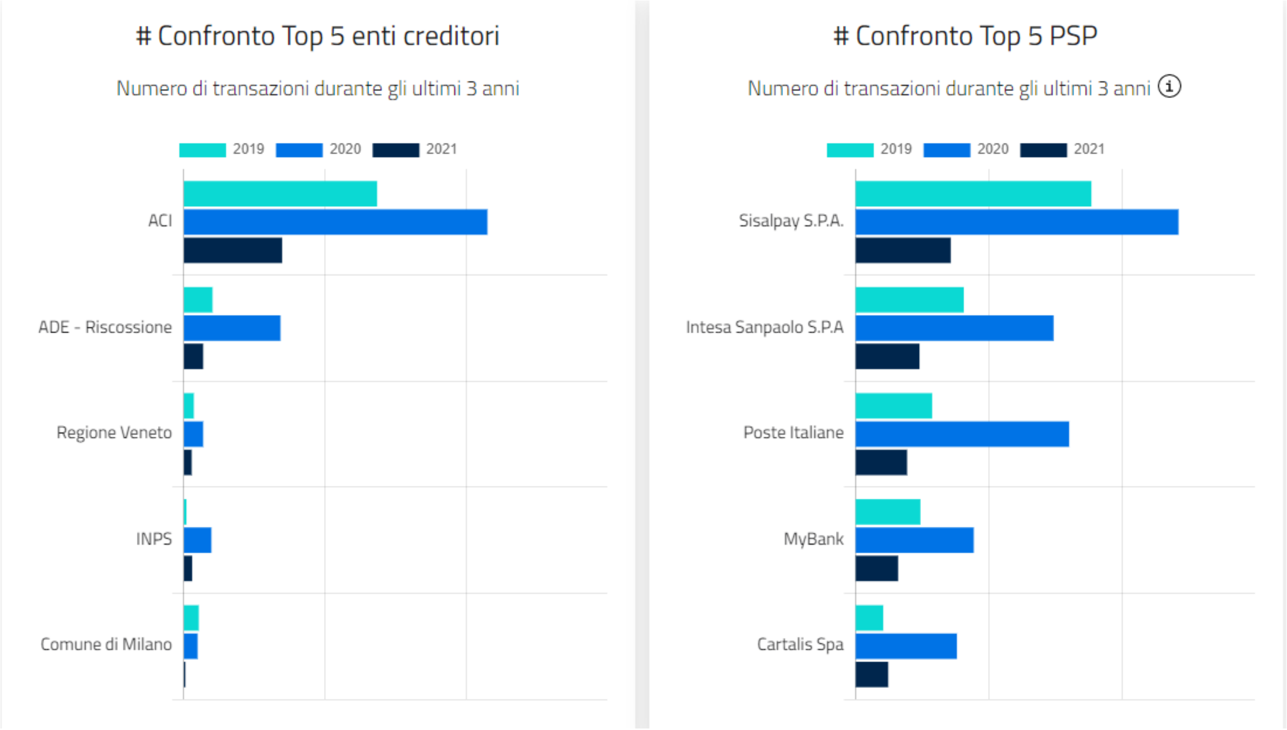


Fig 2 Comparison of number of transactions for the Top 5 Creditors and the Top 5 of Banks and other Payment Service Providers

1.3 NodoSPC

The Payment Node for the Italian Public Connectivity System (referred to as NodoSPC) is a unique technological platform that offers a system to ensure the interoperability between the PA and the PSP. It is the technological infrastructure of pagoPA and is available for each creditor institution.

The operating model of the System refers to the principles of the "Four Corners" model (Fig 3) defined by the European Payment Council in which the infrastructure constituted by NodoSPC enables a communication channel with payment service providers. (Agenzia per l'Italia Digitale, 2020)

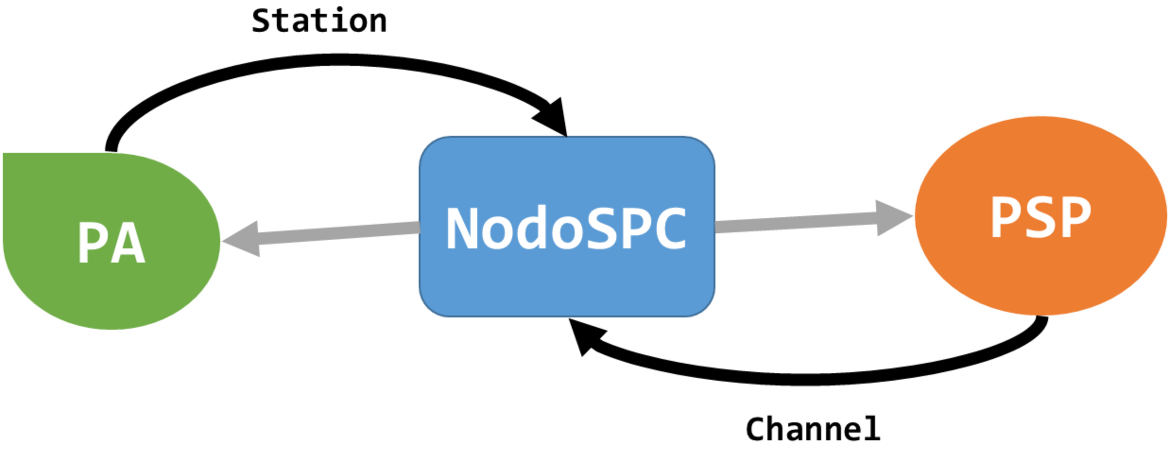


Fig 3 NodoSPC and its main interactions, part of the 4 corners model

1.3.1 NodoSPC Components

Here we mention the main components of the NodoSPC and their general purpose in the pagoPA software ecosystem (Fig 4).

- **NodoSPC:** Platform that connects information between the Public Administration (PA) and Providers of Payment Services (PSP).
- **Payment Manager (PM or WISP):** Part of the pagoPA system that interacts directly with the user (citizens) to help them choose the payment method and safely pay with their account or card data.
- **PDA:** Anagraphic stand-alone data system that receives accession request from various PA and from its technological intermediaries. In the future, it is expected to include as well PSP.
- **WFESP:** Tool that allows the payment fulfilment by redirecting the user to a landing page
- **Wrapper – MyBank:** Handles the necessary path conversions between NodoSPC and the Seller Bank, making it possible to return the payment outcome to the Buyer Bank.
- **LMI:** Backend application that handles the connection interface towards PA and PSP.
- **Cruscotto:** Control Panel that allows a constant monitoring of transactions done, functionalities called and resources deployed.
- **AdminPanel:** Tool used for deploying user assistance.
- **WebBO:** App that visualizes payment entities such as telematic payment requests (RPT) or telematic receipts (RT).

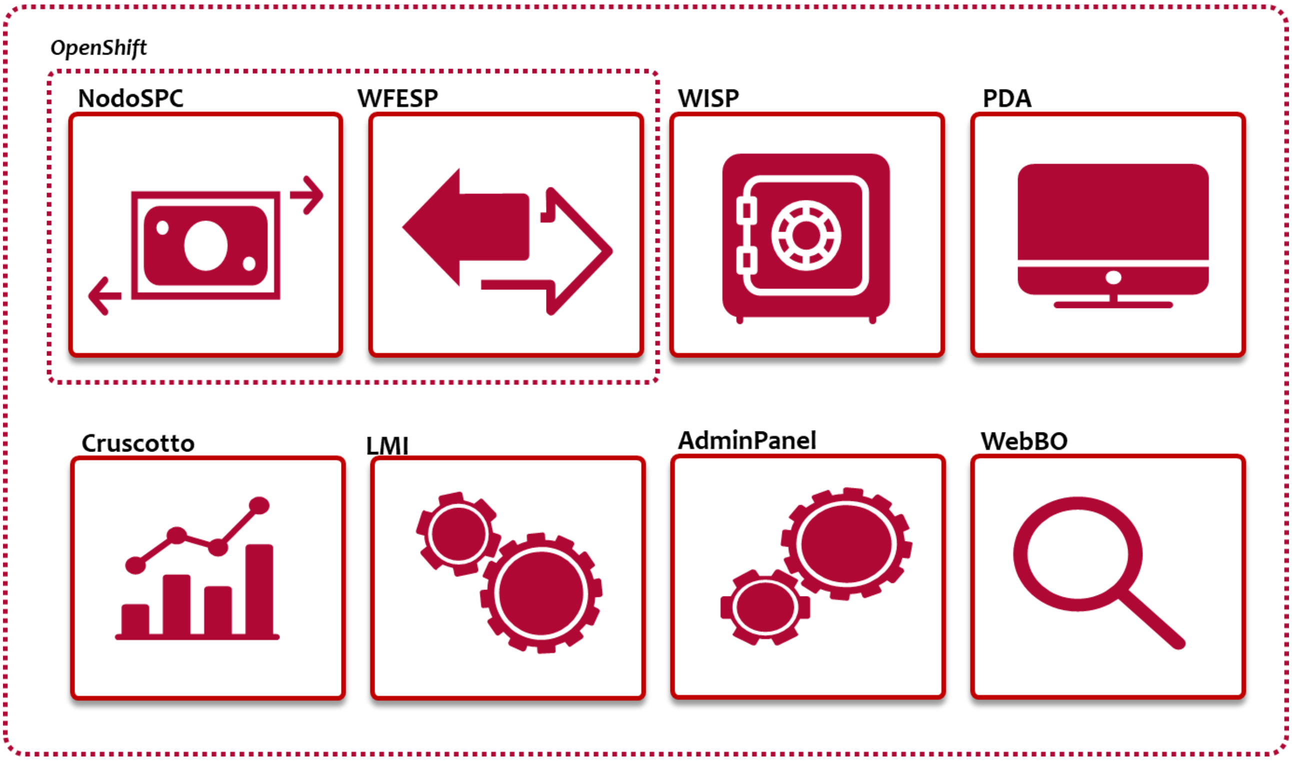


Fig 4 NodoSPC and its components (Crisciullo, 2020)

1.4 Project Management

Projects include people who work cooperatively together toward a common goal, within an established timeframe, and within an established budget to produce identifiable deliverables. For over 50 years, project success has been defined by the criteria of time, budget, and deliverables (Atkinson, 1999).

If a project is to be successfully completed, both planning and execution must be properly implemented. Poor planning will not allow appropriate execution and control processes or achievement of the project's targets (Globerson, 2002).

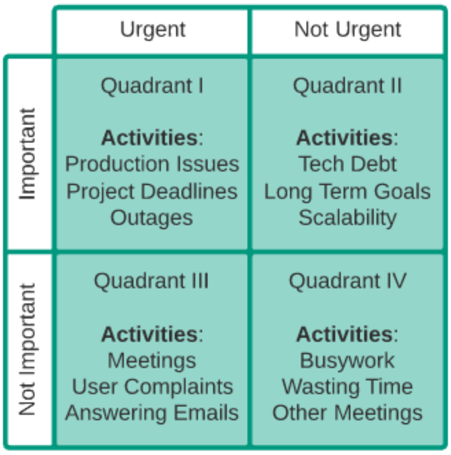
Usually within the company, risk management and communications as the processes with the lowest planning quality. Project managers may not possess the formal tools and techniques for dealing with communications and functional managers are not equipped with the tools and techniques that will allow them to effectively contribute to the risk management process. Improving quality planning processes requires the development of new tools in areas such as communications, as well as organizational training programs designed for the functional managers (Henrie, 2005).

1.4.1 Projectuality and Time Management

Most of the projects at the company have a rationalization process a posteriori due to the agile approach taken. Overall, few schemes pre-conceived exist and new process are created. Creation starts from client requirements in a dynamic but un-projectual way.

Projectuality ensures that each individual can anticipate its own responsibility, instead of having to improvise each time a new project arrives. It is not only a matter of dividing the responsibility areas but also allocating the economical, human and temporal resources.

The project manager should plan the use of time and resources given the project’s scope. This includes defining how important and how urgent any given task might be. The Eisenhower matrix (Fig 5) shows how to take a given decision according to its urgency and importance.



	Urgent	Not Urgent
Important	Quadrant I **Activities**: Production Issues Project Deadlines Outages	Quadrant II **Activities**: Tech Debt Long Term Goals Scalability
Not Important	Quadrant III **Activities**: Meetings User Complaints Answering Emails	Quadrant IV **Activities**: Busywork Wasting Time Other Meetings

Fig 5 Eisenhower matrix with its four quadrants. Each quadrant shows some sample activities such as production issues or busywork for a given project. (Globerson, 2002)

Project management tools are quite important for any given team. However, it is important to keep in to establish some basic principles and a framework that guides our day to day work. This is where Agile comes into play. We will focus more on Agile and especially on the Agile Scrum framework in the next chapter.

2 AGILE SCRUM AND SOFTWARE DEV

2.1 WHAT IS AGILE?

At the beginning of the 21st century, a group of thinkers gather together to discuss about software development. From this meeting they created "The Agile Alliance" with members of different organizations but same values contained into software development (Beck, 2001). This would be a first step to formalize a new kind of methodologies that would lead the software development industry for the decades to come. This is the Agile Software Development Manifesto.

2.1.1 Agile Values

Through working to uncover better ways of developing software, certain items are valued more than others (Beck, 2001), in particular, for Agile are more valued:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

2.1.2 Agile Principles

The leading principles behind all of the Agile methodologies are the following (Cho, 2008):

1. The highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

2.1.3 Agile approach

As the business environment continues to change in this turbulent environment, we must confront the business need for relentless innovation and forge the future workforce culture. Agile software development approaches, such as extreme programming, Crystal methods, lean development, Scrum, adaptive software development (ASD) and others, view change from a perspective that mirrors today's turbulent business and technology environment (Highsmith & Cockburn, 2001).

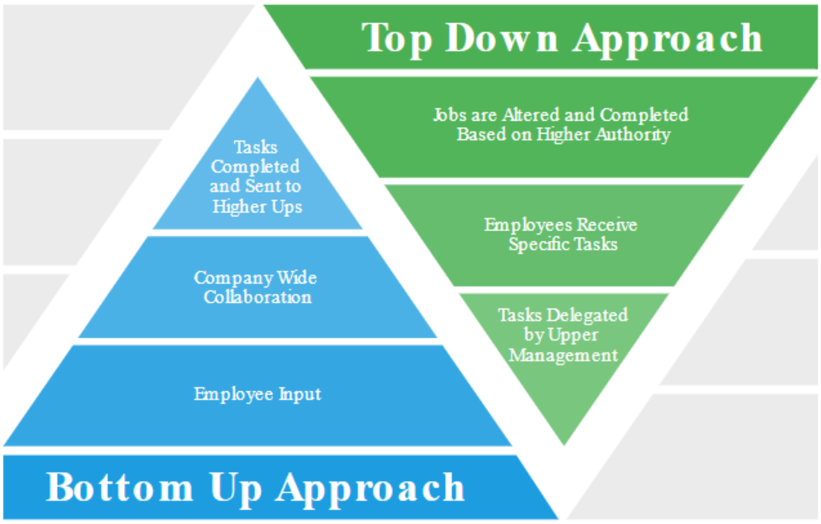

Top Down Approach
Jobs are Altered and Completed Based on Higher Authority
Employees Receive Specific Tasks
Tasks Delegated by Upper Management
Tasks Completed and Sent to Higher Ups
Company Wide Collaboration
Employee Input
Bottom Up Approach

Fig 6 Contrast of Top Down approach with Bottom Up (Cohn, 2010)

In traditional methodologies, such as Waterfall, projects have long and clearly distinct phases with commands that come from the upper management, also known as the Top down approach (Fig 6). Roles are typically well defined and usually fixed from the beginning. Here documentation and reporting are key aspects while authorization is the key to advance within the project (Behutiye, 2020).

Planning, deploying, testing and delivering are the main project phases, which may take place by months or even years. Once a phase is closed, it cannot or should not be re-open again. Each project has one unique staring point and a unique ending point where planning should be millimetrical and carefully followed during deployment (building, assembling, constructing, creating). This approach has been used for years in projects such as civil works and oil refinery plants construction.

Scrum is one of the founding methodologies of the Agile and it is actually opposed to Waterfall. Phases in Scrum are much shorter and are part of project iterations, which may respond to rapid changes in technology or in the market (Perkusich, 2019). Roles are not fixed but fluent and a bottom up approach starts gaining popularity. All of this comes as a contrast of Waterfall's rigurosity and linearity.

Agile has a multiplicity of methodologies, which adapt to specific environments or requirements. Entire chapters may be dedicated to these methodologies, their theory and principles, their structure and their real-world applications. However, in this document we will focus solely on Scrum, one of the most common Agile framework. As a vague exemplification, here there are some other popular Agile frameworks:

- ITIL for service management.
- Prince2 and PMP for project management
- Lean and SixSigma for product management
- eXtreme Programming (XP), Crystal, and Adaptive Software Development (ASD)

2.2 WHAT IS SCRUM

2.2.1 Introduction

Scrum is a lightweight framework based on empiricism (knowledge comes from experience, making decision is based on what it is observed) and lean thinking (reduce waste and focus on essentials) that helps people generate value through adaptive solutions for complex problems. (Schwaber & Sutherland, The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game, 2019)

Scrum employs an iterative, incremental approach to optimize predictability and to control risk.

The Scrum framework is purposefully incomplete, only defining the parts required to implement Scrum theory and various processes and techniques can be employed within the framework (Cohn, 2010).

Scrum methodology is widely used in the IT field due to its rapid essence to overcome different requirements in few time, especially in complex business domain as defined in the Cynefin framework (Browning & Latoza, 2005). There are clear defined roles, and the focus is on the team (Fig 7). At the end, best practices and results are the chosen way to achieve its main goal: producing value (Cho, 2008).

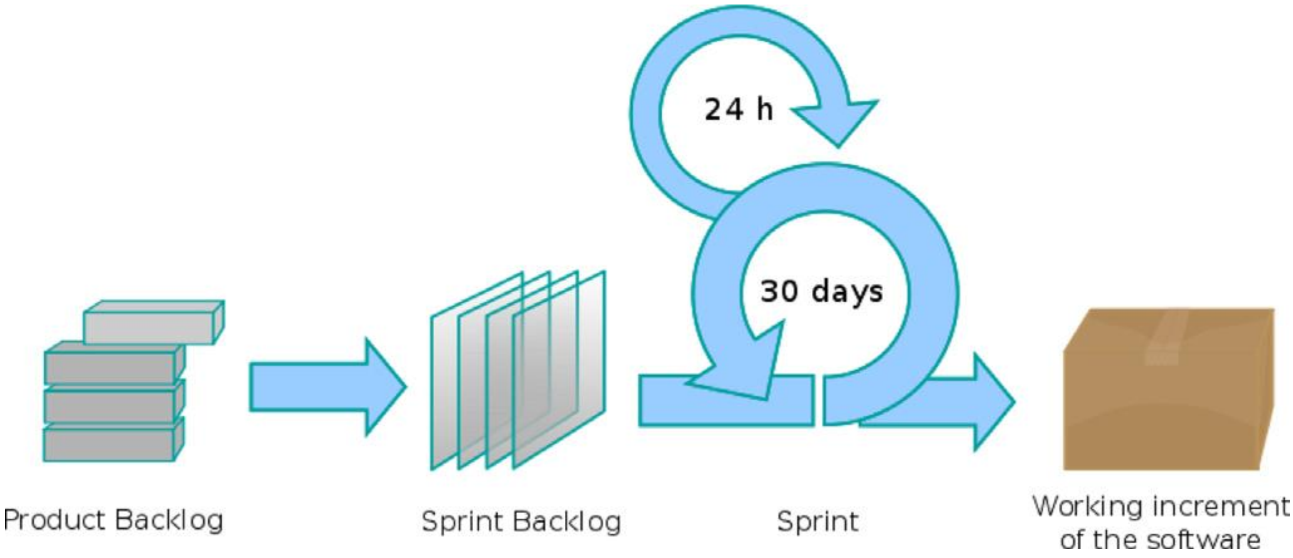


Fig 7 Flow of Scrum Basics

2.2.2 Scrum Team

A Scrum Team is a small team of people, with no sub-teams or hierarchies, focused on one objective at a time (the *Product Goal*). Typically a *Scrum Team* is of 10 or fewer people, if teams become too large they should consider reorganizing into multiple cohesive Scrum Teams each focused on the same product (and hence same *product goal*, same *product backlog* and same *product owner*).

The Scrum Teams are cross-functional (the members have all the skills necessary to create value each Sprint) and self-managing (they internally decide *who* does *what*, *when* and *how*). It is responsible for all product-related activities (stakeholder collaboration, verification, maintenance, development, research ecc) and accountable for creating a valuable, useful *Increment* every *Sprint*.

Within a *Scrum Team* there are three specific accountabilities:

- **Product Owner**:
  Accountable for:
  - Maximizing the value of the product resulting from the work of the Scrum Team.
  - Developing and explicitly communicating the *Product Goal*.
  - Ordering *Product Backlog* items
  - Ensuring that the *Product Backlog* is transparent, visible, and understood.

  In addition:
  - He can also delegate the responsibility but remains accountable.
  - He may represent the needs of many stakeholder in the *Product Backlog*.
  - Those who wants to change the *Product Backlog* can do so by trying to convince the *Product Owner*.
- **Development Team**:
  Committed to:

    - Create any aspect of a usable *Increment* each *Sprint*.

  Accountable for:
    - Creating the *Sprint Backlog*,
    - Adhering to a *Definition* of *Done*,
    - Adapting their plan each day toward the *Sprint Goal*,
    - Holding each other accountable as professional.
- **Scrum Master**:
  Accountable for:
    - Establishing *Scrum* as defined in the *Scrum Guide*
    - Scrum Team's Effectiveness.

  The Scrum Master serves the Scrum Team in several ways, including:
    - Coaching the team members in *self-management* and *cross-functionality*.
    - Helping the Scrum Team focus on creating high-value *Increments* that meet the *Definition of Done* (Partogi, 2013)
    - Causing the removal of *impediments* to the *Scrum Team*'s progress
    - Ensuring that all *Scrum events* take place and are positive, productive, and kept within the timebox.

  The *Scrum Master* serves the Product Owner in several ways, including:
    - Helping find techniques for effective *Product Goal* definition and *Product Backlog* management
    - Helping the *Scrum Team* understand the need for clear and concise *Product Backlog* items
    - Helping establish empirical product planning for a complex environment
    - Facilitating stakeholder collaboration as requested or needed

  The Scrum Master serves the organization in several ways, including:
    - Leading, training, and coaching the organization in its *Scrum* adoption
    - Planning and advising *Scrum* implementations within the organization
    - Helping employees and stakeholder understand and enact an empirical approach for complex work
    - Removing *barriers* between stakeholders and *Scrum Teams*

2.2.3 Scrum events

Sprint

A sprint is a container for all the events with a fixed length of one month or less.

Sprints enable predictability by ensuring inspection and adaptation of progress toward a *Product Goal* at least every calendar month. When a *Sprint's* horizon is too long the *Sprint Goal* may become invalid, complexity may rise, and risk may increase. Shorter Sprints can be employed to generate more learning cycles and limit risk of cost and effort to a smaller timeframe (Schwaber & Beedle, Agile software development with Scrum, 2002).

During the *Sprint*:

- No changes are made that would endanger the *Sprint Goal*
- Quality does not decrease
- The *Product Backlog* is refined as needed
- Scope may be clarified and renegotiated with the *Product Owner* as more is learned

A new *Sprint* starts immediately after the conclusion of the previous *Sprint*.

A *Sprint* could be canceled if the *Sprint Goal* become obsolete but only the *Product Owner* has the authority to cancel the *Sprint*.

Sprint Planning

The Sprint planning initiates the *Sprint* by laying out the work to be performed for the *Sprint*. This resulting plan is created by the collaborative work of the entire *Scrum Team*. Here, The *Product Owner* ensures that attendees are prepared to discuss the most important *Product Backlog* items while understanding how to map them into the *Product Goal*. Meanwhile, the *Scrum Team* may also invite other people to attend *Sprint Planning* to provide advice.

Sprint Planning is *timeboxed* to a maximum of 8 hours for a one-month Sprint (for shorter Sprints it is shorter).

Sprint Planning addresses the following topics:

- Why is this *Sprint* valuable?
  - The *Product Owner* proposes how the product could increase its value and utility in the current Sprint
  - The whole *Scrum Team* then collaborates to define a *Sprint Goal* that communicates why the *Sprint* is valuable to stakeholders.
- What can be done this Sprint?
  - Through discussions with the *Product Owner*, the *Developers* select items from the *Product Backlog* to include in the current *Sprint*.
  - The *Scrum Team* may refine these items during this process, which increases understanding and confidence.
- How will the chosen work get done?
  - For each selected *Product Backlog* item, the *Developers* plan the work necessary to create an *Increment* that meets the *Definition of Done*. This is often done by decomposing *Product Backlog* items into smaller work items of one day or less. How this is done is at the sole discretion of the *Developers*. No one else tells them how to turn *Product Backlog* items into Increments of value.

Daily Scrum

The purpose of the *Daily Scrum or Daily Meeting* is to inspect progress toward the *Sprint Goal* and adapt the *Sprint Backlog* as necessary, adjusting the upcoming planned work. *Daily Scrums* improve communications, identify impediments, promote quick decision making and also eliminate the need for other meetings. However, this is not the only time *Developers* are allowed to adjust their plan as and they can meet throughout the day for more detailed discussion about adapting or re-planning the rest of the *Sprint's* work.

The daily Scrum is a 15-minute event for the *Developers* of the *Scrum Team*. To reduce complexity is held at the same time and place every working day of the *Sprint*. If the *Product Owner* or the *Scrum Master* are actively working on items in the *Sprint Backlog*, they participate as *Developers*.

Sprint Review

The main objective of *Sprint Review* is to inspect the outcome of the Sprint and determine future adaptations. Here, the *Scrum Team* presents the result of their work to key stakeholders and progress toward the *Product Goal* is discussed.

During the event, the *Scrum Team* and stakeholders review what was accomplished in the *Sprint* and what has changed in their environment. The *Sprint Review* is a working session and the *Scrum Team* should avoid limiting it to a presentation. It is *timeboxed* to a maximum of 4 hours for a one-month Sprint (for shorter Sprints the event is usually shorter).

Sprint Retrospective
The goal of the *Sprint Retrospective* is to plan ways to increase *quality* and *effectiveness*.

The *Scrum Team* inspects how the last *Sprint* went with regards to individuals, interactions, processes, tools, and their *Definition of Done*. Then *they* discuss what went well during the *Sprint*, what problems it encountered and how those problems were (or were not) solved. At the end, The *Scrum Team* should identify the most helpful changes to improve its *effectiveness*.

The most impactful improvements are addressed as soon as possible (they may even be added to the *Sprint Backlog* for the next *Sprint*). It is *timeboxed* to a maximum of 3 hours for a one-month sprint (for shorter sprints the event is usually shorter).

2.2.4 Scrum artefacts

Product Backlog

The Product Backlog is an emergent, ordered list of what is needed to improve the product. It contains the *Product Goal:* a future state of the product, the long-term objective that can serve as a target for the *Scrum Team* to plan against and It is the single source of work undertaken by the *Scrum Team*.

The Product Backlog Refinement is the act of breaking down and further defining Product Backlog items into smaller more precise items. This is an ongoing activity to add details, such as description, order, and size. Set of items that contribute to add value to the product. This can include functionalities, requirements, bug fix.

Sprint backlog
A Sprint Backlog is a set of items from the Product Backlog selected by the development team to be completed during a sprint, plus the plan for delivering them. It is a plan by and for the *Developers* updated throughout the *Sprint* as more is learned. It is also created during the *Sprint Planning* and it is the single objective for the *Sprint*. It should have enough details that they can inspect their progress in the *Daily Scrum*. It establishes coherence and focus, encouraging the *Scrum Team* to work together rather than on separate initiatives.

It is composed by:

- *Sprint Goal* (why)
- Set of *Product Backlog* items selected for the *Sprint* (what)
- An actionable plan for delivering the *Increment* (how)

If the works turns out to be different than they expected, the *Developers* collaborate with the Product Owner to negotiate the scope of the *Sprint Backlog* within the *Sprint* without affecting the *Sprint Goal*.

Increment
An Increment is the sum of all Product Backlog items completed during a sprint and the value of the increments of all previous sprints. It is a concrete steppingstone toward the *Product Goal*. In order to provide value, it must be *usable*.

Each increment is additive to all prior increments and thoroughly verified, ensuring that all *Increments* work together. Multiple *Increments* may be created within a Sprint and an increment may be delivered to stakeholders prior to the end of the *Sprint*.

Work cannot be considered part of an *Increment* unless it meets the Definition of Done: a formal description of the state of the *Increment* when it meets the quality measures required for the product. If the *Definition of Done* is part of the standards of the organization, all Scrum Teams must follow it as a minimum. If it is not an organizational standard, the *Scrum Team* must create a *Definition of Done* appropriate for the product. Multiple *Scrum Teams* working together on a product must mutually define and comply with the same *Definition of Done*.

2.2.5 Scrum pillars

- **Transparency**: the emergent process and work must be visible to those performing the work as well as those receiving the work. With *Scrum*, important decisions are based on the perceived state of its *Artefacts* →Artefacts that have low transparency can lead to decisions that diminish value and increase risk. Here Transparency enables Inspection.
- **Inspection**: *Scrum Artifacts* and the progress toward agreed goals must be inspected frequently to detect potentially undesirable variance or problems. To help with inspection *Scrum* provides cadence in the form of its *Events*. Here, instead, Inspection enables adaptation.
- **Adaptation**: if any aspect of the process deviates outside acceptable limits or if the product is unacceptable the process being applied, or the materials being produced must be adjusted as soon as possible. Adaptation is more difficult when the people involved are not *empowered* or *self-managing*.

Given a Scrum team, for each sprint it can be projected a "capacity" which can be measured as total number of "story points". Story points should burn down each day throughout the sprint. The development team should estimate the effort given to a specific task to assign the number of story points (relative estimation).

2.2.6 Additional considerations

The product owner is responsible for maximizing the value of a product. Hence, he/she chooses the order and priority of the items included into the Product Backlog. In general, there is no strict rule of how to decide this, but there are certain advises.

The product owner should address the capacity of the team to work a certain number of story points for the total Sprint run. (Story points = Number of workdays/person).

There are different methodologies and tools to know how long a given item can take (e.g. Planning Poker). Team members might estimate a duration for each task and summing it all together, we could obtain theoretically how many items could we fit inside a sprint. For each task, the team defines a number of Story points. Typically, they do this by using sizes such as XS, S, M, L and XL, which each have an equivalent in workdays per person.

At the beginning there might not be any clue of how long a Sprint should be, the team would have to take a wild guess recurring to self-made estimations. However, after the first Sprint has passed, the team can take action and define a more accurate and empirically based item estimation. This can be considered a trial and error method.

At the end of the Sprint, the product owner can count the total number of story points to calculate the velocity of the team. The question is "How many activities can be completed (done) in a single Sprint?"

The "definition of done" is crucial to properly define a task as complete. Since each Sprint is time-boxed, the team will consider the sprint finished as soon as the time box has expired. For a task to be defined as 'done', it must adhere to the conditions outlined into the "definition of done", e.g.

- Developed with documented coding
- System and User testing finished and successful

Now, we need to understand how the Scrum framework fits into the general lifecycle for Software development. This topic will be discussed in the next section.

2.3 Software Dev

A typical software development project starts from its Setup, then it undergoes the implementation phase, afterwards the team should assure the quality and finally it goes to rollout.

Each product developed usually has at least four stages: developing phase, two testing phases: System integration test (SIT) and User Acceptance Test (UAT) and a final Production phase.

Certain bugs might be found during any phase for any given version. Usually a patch is released to fix bugs found in newer versions. However, if the developers need to solve it fast, a hotfix might be a better solution.

2.3.1 System Development Lifecycle

Today, the traditional approaches to technology system development have been adjusted to meet the ever-changing, complex needs of each unique organization and their users. Below you will find sequential steps to SDLC, but each company will vary in their process. (Smartsheet, 2020)

2.3.2 The Phases of SDLC

The SDLC framework provides a step-by-step guide through the phases of implementing both a physical and software based system.

Analysis/Feasibility: For an SDLC strategy to work there should be a strong idea of what deficiencies exist in the current structure and the goals for the new approach. A feasibility study determines if you can or should accomplish the goals of the plan (Blanchard & Fabrycky, 2006).

Planning/Requirements: A plan can include adapting a current system to meet new needs or developing a completely new system.

Design: A detailed system design can begin that includes features and other documentation. The architects can then build a sample framework.

System Development: An approved design is the catalyst for authorizing development for the new or augmented system .

Testing: Users are brought in to test before deployment to identify areas of concern or improvement.

Deployment: The system is put into a production environment and used to conduct business.

Maintenance: The cyclical nature of SDLC recognizes that the process of change and upgrading are constant. Carry out the replacement of outdated hardware/software, security upgrades, and continuous improvement on a regular basis (Cummings, 2006).

Evaluation: The evaluation process, which supports the continuous improvement of the system. The team continuously reviews what is working and what is in need of improvement.

Disposition/Disposal/End-of-Life: A well-rounded life cycle identifies and decommissions surplus or obsolete assets at the end of their life cycle (Unhelkar, 2016).

Taking into account the phases of the software development life cycle, in the next section we will return to Agile and Scrum to discuss their main problems and issues.

2.4 Agile & Scrum Issues

As mentioned by Cho in his paper there are five different areas in which Scrum Agile has some issues or backslashes. For the author, after comparing and contrasting the traditional waterfall method with Scrum Agile, the main aspects, which we should take into consideration, are documentation, communication, customer involvement and Scrum ceremonies (Cho, 2008).

The idea behind reducing documents in the agile methods is to keep every team members equal by sharing skills and knowledge on the systems. In theory, when a person leaves, there is still a lot of shared knowledge that has gone around among other team members, so it should not be a great problem. However, in reality, this is not feasible and knowledge tends to focus solely on one or two team members. That is why lack of documentation can be detrimental for the continuity of a team in the long-run.

It is well known that ineffective communication is the root of most failures in software products (Parnas, 2006). That is why face-to-face is usually recommended for a constant teamwork and transparency to achieve a good team productivity. Besides, it should be the Product Owner the person in charge of giving clear communication and enabling spaces to give and receive constant feedback (Toivonen, 2020).

For example, the Daily meeting is usually regarded as an effective tool but sometime the team may engage in discussions that are not necessary at that point of time (Toivonen, 2020). It should be a duty of the Scrum Master to ensure the meeting goes into the right direction while discussing only the necessary. After all, the agile methodologies are still on a relatively infancy level in comparison with other centuries-established industries. Agile and Scrum, as any other framework, is subject to holes in its structure, especially if it is not well understood when applied.

Knowing the main theoretical aspects of Agile and Scrum and their possible issues, we can now carry them to a more practical environment. In the next chapter, we will introduce the company in charge of the internal management of pagoPA project that has recently established Agile Scrum on their Working Agreement. It is the Italian company SIA S.p.A.

3 SIA – PAGOPA BUSINESS UNIT

SIA S.p.A is a digital payments company located in southwest Milan. It is the main partner of PagoPA S.p.A, which used to be part of Agenzia per l'Italia Digitale (AgID). SIA works as the software and technology provider, although beyond that SIA manages the internal organization of the project. It is responsibility of SIA to deliver and evolve the current product NodoSPC.

SIA works on the internal management of the platform while PagoPA S.p.A manage the interaction with the final users.

SIA agreed on a contract for the service delivery of PagoPA. Since the main product and its components has already been developed and released in production, the main goals of the current Business Unit at SIA are:

- **Change Management**

This works by complementing and evolving an existing software system through micro activities. The customers and users may demand new characteristics and functionalities. The team needs to keep its platform fresh and up to date to the latest technology and legislation requirements.

- **Application Management**

This relies on reports and defects found. When updated or changed, the software may exhibit new unexpected problems (bugs). It is up a team's duty to detect ways to break the system and attempt to fix them before the users point them out.

3.1 SCRUM TEAMS AT PAGOPA PROJECT

The SIA - PagoPA Business Unit was one of the first to implement this agile methodology in the company. Within the group, there are many people with a lot of experience in the delivery of software projects. However, for a good portion of the team this was the first Scrum experience.

Two Scrum teams have been created in this BU: Team Nodo and Team PM (Fig 8). Each team works on different projects but these are co-dependent (Fig 9). Some of the people in the teams are external to the company, and there are also others who perform tasks for both teams. These and other peculiarities make it interesting to study the comparison between theoretical Scrum and its implementation in real life. The Product owner, which is from PagoPA S.p.A, directs and request new features from both teams. They also have a common Scrum Master and a common Developer Tester. SIA handles the organizational force in the team and external consultants play a key role in the software developing.

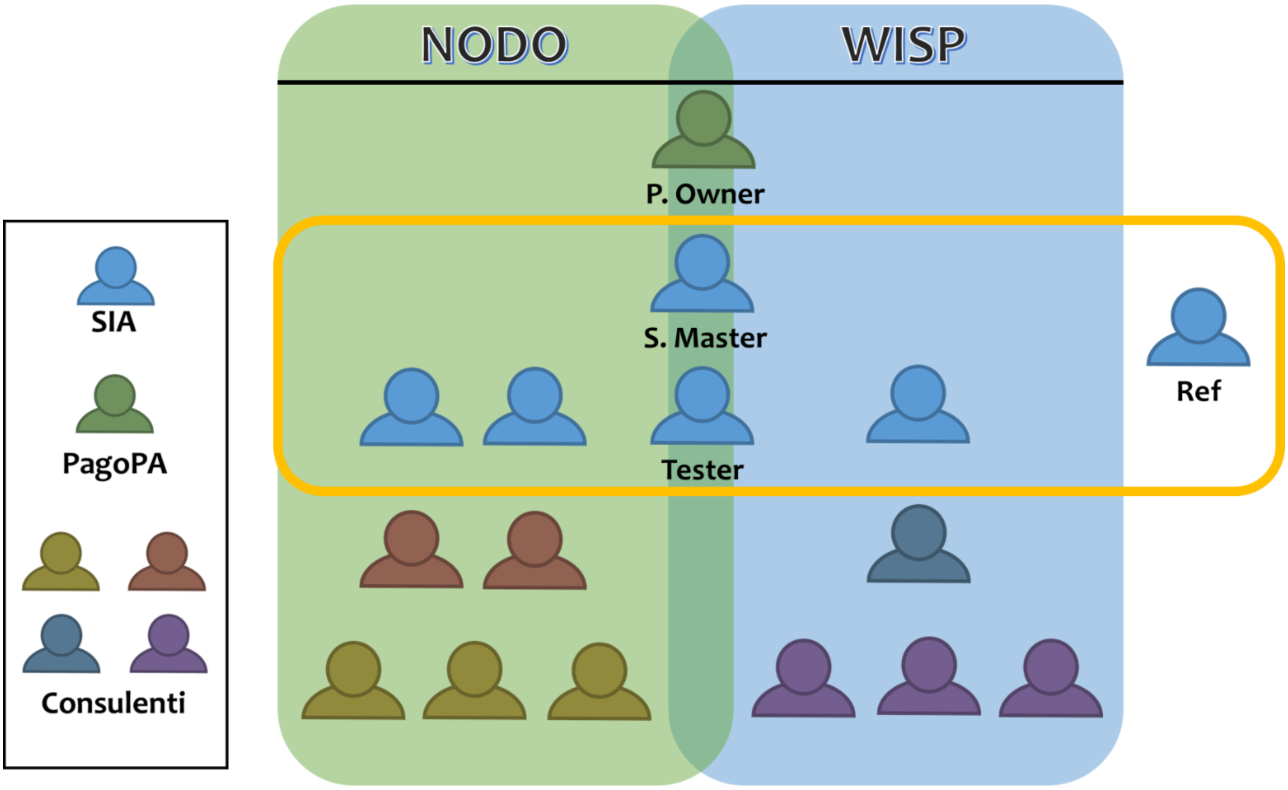


Fig 8 Scrum teams for SIA-PagoPA

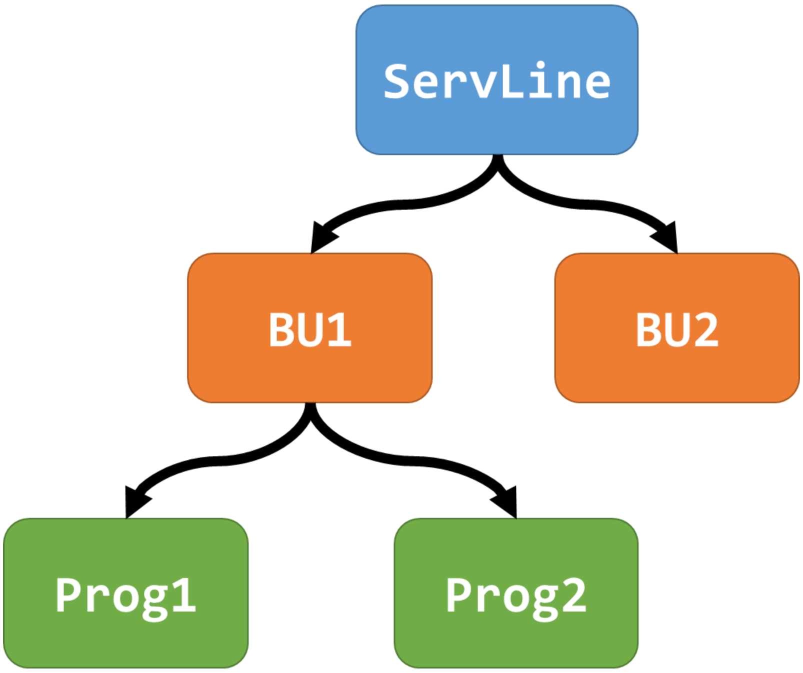


Fig 9 Business Units at SI having two projects and sharing a single Service line

3.2 PagoPA (BU) Issues

The contract establishes that SIA-PagoPA will work using the Scrum framework. This approach differs from the hierarchical waterfall approach that SIA S.p.A uses for most of the company.

As a Business Unit (BU), SIA-PagoPA is not autonomous since it relies directly from PagoPA S.p.A. The Product Owner is at the same time the client, therefore certain dynamics that are typical of any given vendor-customer interaction may not be beneficial in a teamwork.

There is a lack of vision for the long run. Only the product owner knows up to which point he wants its team to arrive with the project. It would be helpful for the team to know the intention behind the tasks given. There is a lack of project management.

Subdivisions among the Scrum team are not advisable but anyway tend to arise due to the complexity of the project. For example, the product owner is located in a different city, Rome; meanwhile SIA – PagoPA works in Milan. He might not have a deep knowledge of all the functionalities of the components of NodoSPC given its external nature.

How can the team properly compute a cost for each Sprint? It involves how to bill and how to define a proper fare. This cost estimation might be easier in a waterfall method due to its more rigid nature. Is there a rigorous control of the Story Points for any given Sprint?

Where is the thinking process? Should SIA – PagoPA BU focus only on service deliver or should it go beyond that?

It is not completely coherent under the Scrum framework to have two separate product backlogs for the same product, being it NodoSPC presented as a unique product with different extensions and components that ultimately serve the same purpose.

PagoPA might ask to SIA to keep determined KPI's by the company created through the Working agreement. In any case, now we will focus our attention to a real-world case study that exemplifies some of the issues mentioned earlier.

3.2.1 Case study: A new functionality

The product owner asks the creation of a new functionality. The Product Owner explains the idea to create an extension of a current component.

Scrum applied in these circumstances creates a particular situation. The team has no way of knowing how this project evolves in the future. It might be a small stand-alone functionality but it could also slowly evolve into a fully functional product such an App.

In any given vendor – customer interaction, the vendor will attempt to sell more and to show unique solutions to the customer. On the other hand, the customer will look for the bargaining strategy to try to receive as much as possible for the lowest price. These all depend on the power of negotiation and it is intrinsic to the nature of the selling process.

However, this negotiation process has some different implications within a team that works together for a common purpose. Due to each company's internal policy, certain measures might be taken to control which data can be shared or transmitted. In economics, this is called information asymmetry. It is normal for it to occur in a transactional environment, nonetheless, for a Scrum team, this can affect key Scrum values in unexpected ways.

At the same time, the developing team, being external to the company, may attempt to use some of these vendor strategies with SIA. As usual in business strategy, Porter's five competitive forces affect in this scenario.

3.3 SMARTWORKING DURING A GLOBAL PANDEMIC

Beside the previously mentioned topics, there was an additional element that came by surprise. Since this research project took place in the midst of COVID-19 quarantine, virtually all of the team members were working remotely from their homes, or as it is commonly known nowadays, Smartworking.

At the start of the pandemic in Italy, the company took strong measures to avoid the spread of the virus within its employees. Since February 24, 2020 the company decided to restrict the access to all the people working inside the company's offices, including consultants. Every single employee had to take their computer home and keep working on their projects as usual.

Most of the work typically done in meeting rooms and open offices had to then rely entirely on business communication platforms such as Microsoft Teams and issue tracking platforms for bug tracking and agile project management such as Atlassian Jira. Fortunately, the team already used quite a lot of these tools and within the company, it had always been one of the "first movers" for embracing new technologies.

Many companies re-open their doors for employees after the strict lockdown in April. However, this was not the case for our company since the directives decided to keep Smartworking during the whole year. Looking in retrospective, this was a very good decision since the work has mostly been uninterrupted by the virus outbreak or governmental restrictions on public spaces. Only the most crucial activities were allowed in the company buildings, such as server maintenance. Nowadays, Smartworking has remained as the main working mode in 2021 and it is very likely that it will keep a profound impact even after the COVID crisis finishes.

We should remember that during the pandemic, not all industries were hit equally. In the case of SIA and PagoPA, both companies boosted their numbers, increased their revenues and their influence on the market. It is worth mentioning that the digital payment sector had some of the biggest advantages in comparison to other kinds of industries, especially those related to civil works, oil and gas, and manufacturing in general.

Understanding the current situation of the company and the business unit is important. Now we can search for similar cases found in literature and compare them to give us some additional hindsight. In the next chapter, we will present and analyze important extracts from a bibliographical research to see how they elaborated their studies and explore how to relate them to our own investigation project.

4 STATE OF THE ART

There have been various studies done about Agile and Scrum frameworks applied in different contexts. These authors have proposed different methodologies and ideas to study the effects on real-life examples in various companies and Scrum teams. Here are some extracts from some authors that we will use as a reference model:

4.1 TEAMWORK MODEL FOR SCRUM

In Scrum, it is usually argued that the teams should self-manage. However, most of the times this concept does not include clear guidelines of how it should be performed. Changing from a one-way linear leadership environment to a multiple leader environment is not easy and most people tend to return to the former. Self-management is an idea that needs some proper definition and deepening to allow the team to truly appreciate and enjoy its benefits, while avoiding its common pitfalls and drawbacks.

Moe et al. conducted extensive fieldwork to provide a better understanding of the nature of self-managing agile teams and its corresponding teamwork challenges. For 9 months, they undergo research in a software development company that introduced Scrum (Fig 9). They focused on the human sense making and on how the group understood the mechanisms of teamwork. They described the project through a conceptual teamwork model, focusing on the interrelations between essential teamwork components. (Moe, Torgeir, & Tore, 2010)

In this work, Moe et al. argue that teamwork is a concept that carries a set of values such as encouraging listening, constructing responses from others views, providing support and recognizing the interest and achievements of others (Katzenbach & Smith, 1993). However, it is highly emphasised that members of a software team should be all together responsible for the end product while they must develop and share mental models to enable themselves to negotiate and understand both teamwork and its correlation to tasks (Levesque, Wilson, & Wholey, 2001).

“It is not sufficient to put individuals together in a group, tag them ‘’self-managing”, and expect that they will automatically know how to coordinate and work effectively as an agile team.” - (Moe, Torgeir, & Tore, 2010)

As described by the author, processes and components of teamwork have been discussed in multiple studies but without arriving to a compelling consensus concerning its conceptual structure (Rousseau, Aube, & Savoie, 2006). This was true until Dickinson and McIntyre identified and defined seven core components of teamwork, which are: communication, team orientation, team leadership, monitoring, feedback, backup, and coordination. All of these elements are connected into a learning loop (Fig 10Fig 10) where teams are characterized as adaptable and dynamically changing over time (Nerur & Balijepally, 2007).

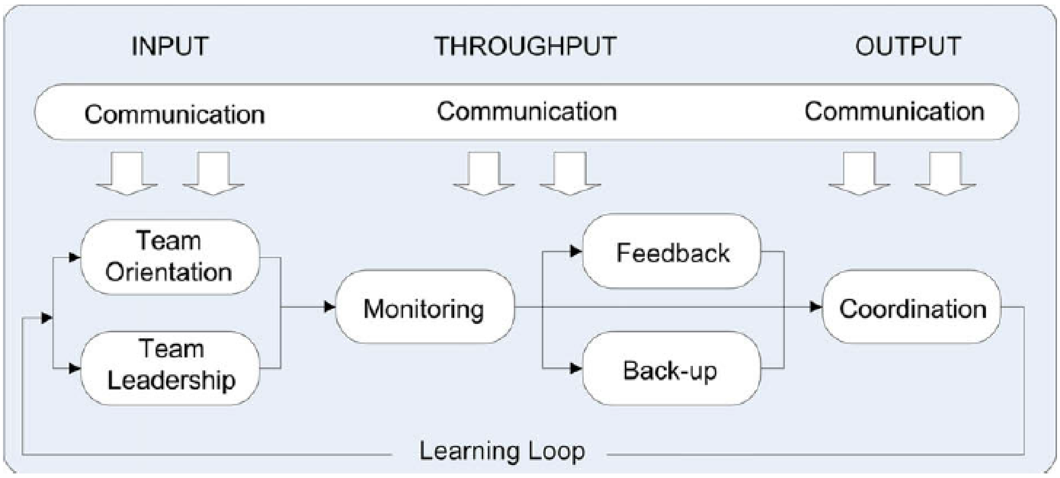


Fig 10 The Dickinson and McIntyre teamwork model used by (Moe, Torgeir, & Tore, 2010)

As a conclusion of the investigation done by Moe et al, they found that transitioning from individual work to self-managing teams requires a reorientation not only by developers but also by management. Making such changes takes time and resources, but it is a prerequisite for the success of any kind of agile method based on self-management. It can be noticed that Scrum has several mechanisms in place for supporting the recommendations of the conceptual teamwork framework (Dickinson & McIntyre, 1997), but that many of these mechanisms are not easy to implement in practice (Moe, Torgeir, & Tore, 2010).

4.2 PRODUCT OWNER ROLE IN SCRUM

Since the Product Owner is the most distinct and important role in the team (Partogi, 2013), it is worth elaborating a whole article based on them. In this research, the objective was to clear up how Scrum is applied from the role of the product owner. This included to study as well as the knowledge and skills he is required to have, according to the Scrum method. This is compared to the perception of a number of actual project owners by gathering information with a limited number of product owners that have had that role for at least 1 year so they could consider a comparable experience.

Sverrisdottir et al. collected data through semistructured interviews with five active product owners that work according to Scrum within five organizations (Fig 11). Their objective with the interviews was to find out how the participants define the role and responsibility in the management of projects, were Scrum is applied. (Sverrisdottir H.S., 2014)

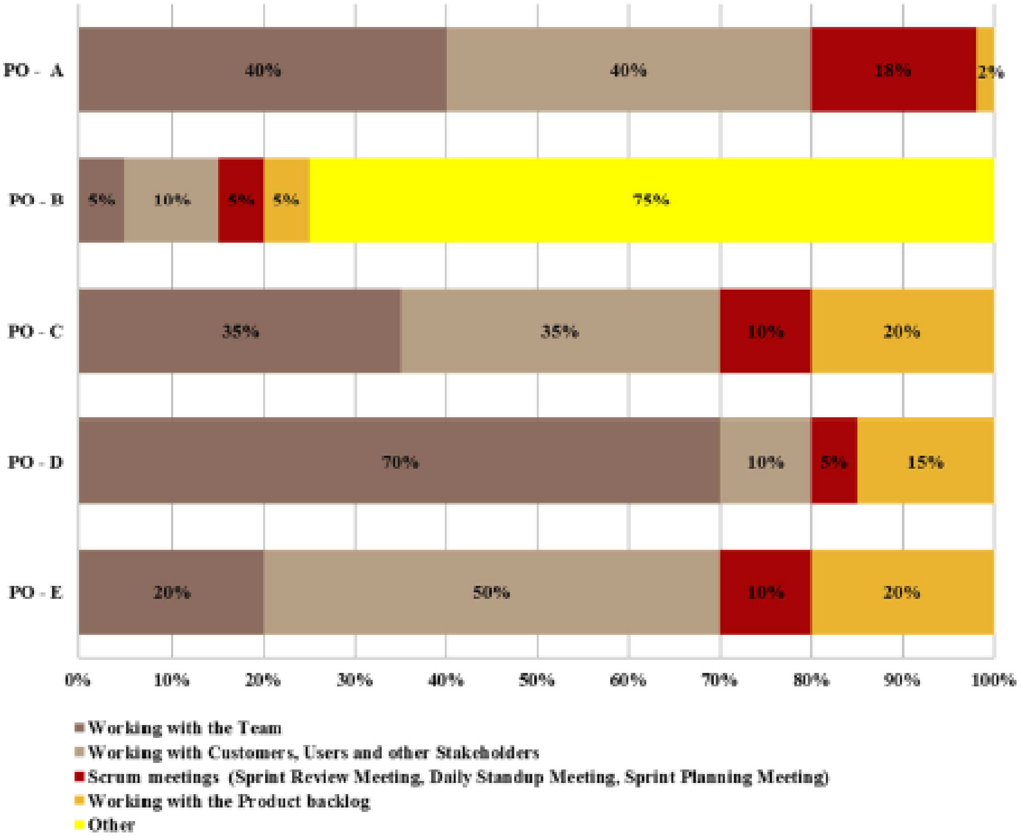


Fig 11 Task of the product owners, and their relative weight in the total time. (Sverrisdottir H.S., 2014)

The results show that the majority of the participants in the survey are using different project management methods. They apply the methods that best fit their own operations. The understanding of the role and responsibility of the product owner is quite different between organizations but seldom in perfect conformance with the official Scrum method (Sverrisdottir H.S., 2014). Cases were reported where there are two product owners for the same product. One is then responsible for business aspect but the other is responsible for technical aspects of the product. Scrum has a strong position in software development with its defined roles, collaboration emphasis, understanding, visibility, effective process and fast development.

4.3 Statistical Analysis for Scrum and Kanban

Kanban and Scrum are two powerful Agile project management approaches in software development.The objective of Scrum and Kanban is achieved by optimizing the development process by identifying the tasks, managing time more effectively, and setting-up teams (Project Management Institute, 2003). A review of the literature reveals that there is a lack of statistical evidence to conclude which methodology is more effective in dealing with the traditional project management factors of budget handling, risk control, quality of the project, available resources, having clear project scope, and schedule handling (Awad, 2005).

Lei et al. performed numerical analysis based on survey responses. They analysed past experiences from those that used Scrum and Kanban methods (Fig 12). They analysed how both Scrum and Kanban lead to the development of successful projects considering various project management factors. (Lei, 2015)

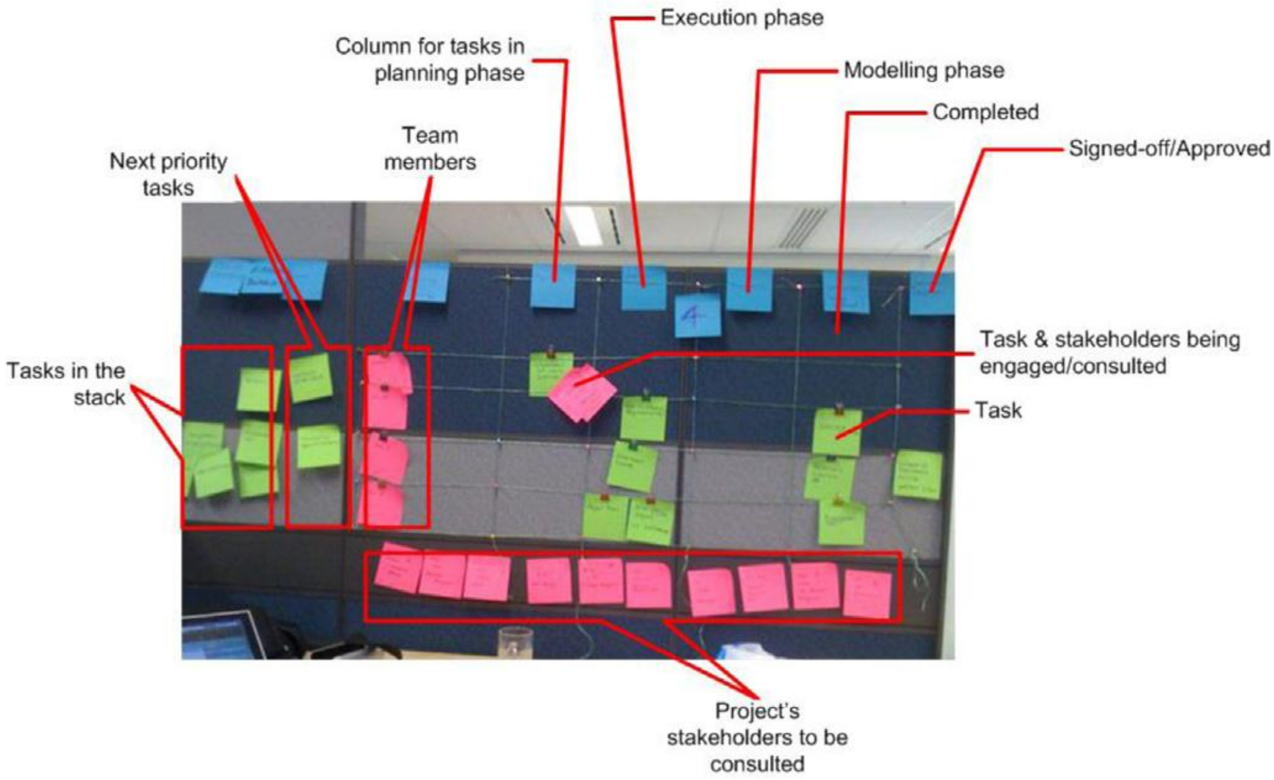

Execution phase
Column for tasks in planning phase
Modelling phase
Completed
Signed-off/Approved
Team members
Next priority tasks
Tasks in the stack
Task & stakeholders being engaged/consulted
Task
Project's stakeholders to be consulted

Fig 12 Kanban card wall example used in a project (Lei, 2015). A similar technique may be used for Scrum purposes

The research team performed additional analysis examining the effect of company size on the performance of Scrum versus Kanban for the Schedule factor. Survey scores for all questions pertaining to the Schedule factor (questions 1.1, 1.2, and 1.3) are averaged separately for respondents working on Scrum-and Kanban-based projects for company sizes of less than 50, 50–100, 100–500, and more than 500 (Lei, 2015).

This analysis showed that Kanban and Scrum have considerable similarities that rely mainly in the values and having a self-organized team (Skarin & Kniberg, 2009). Both are more efficient than the Waterfall methodology due to its feedback and improvement mechanisms while accounting for project scope and value (Keogh, 2011).

At the end of the day, if we have two similar methodologies, it is more important to focus on the implementation rather than their particular differences. After all, each Agile team can choose the methodology that fits the best to their project. However, it is up to them to apply it and follow it in the most fruitful way that allows the team to reach their goals.

4.4 CONCEPTUAL LAYOUT OF THE WORK ENVIRONMENT FOR THE SCRUM TEAM.

Work environment can greatly influence the way a team interacts and behaves. In addition, many studies suggest that the quality of office spaces and their layout has an effect on the health, comfort, satisfaction and productivity of office workers (Mahdavi & Unzeitig, 2005). This is the reason why, the architectural layout design should be a fundamental activity in which the designer or architect arranges compartments to satisfy specific requirements for the team. In this case, we will focus on the requirements for a Scrum team (Regateiro, Bento, & Dias, 2012).

The main purpose of this article is to propose a conceptual model of office layout for the needs of IT project teams working according to the Scrum method. This will have a positive effect on the efficiency of the execution of Scrum projects (Rola, Kuchta, & Kopczyk, 2016). The authors base this conceptual model on requirements that should be fulfilled by offices destined to have Agile working teams. To test the model, it was implemented in a specific real-world project and team, where it was validated and, with further research, should be replicable and scalable to other projects as well.

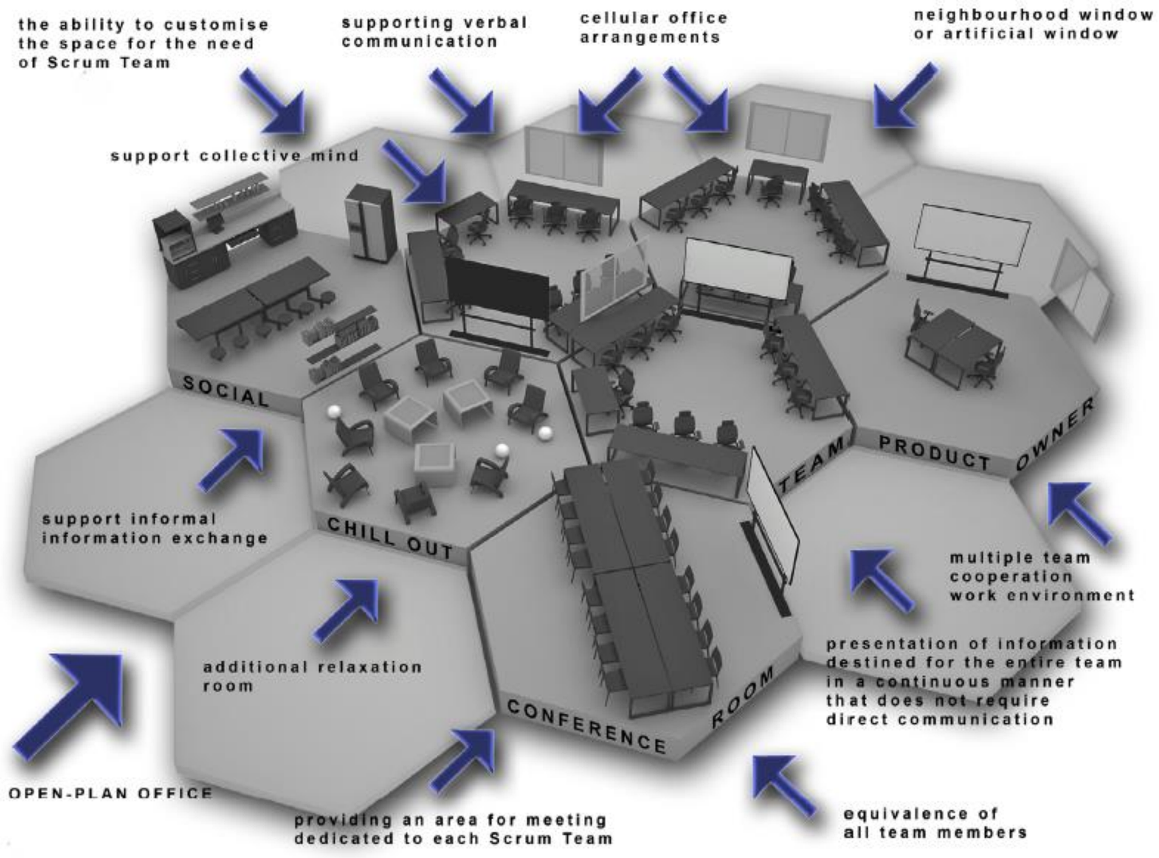

the ability to customise the space for the need of Scrum Team
supporting verbal communication
cellular office arrangements
neighbourhood window or artificial window
support collective mind
SOCIAL
CHILL OUT
TEAM
PRODUCT OWNER
support informal information exchange
multiple team cooperation work environment
additional relaxation room
presentation of information destined for the entire team in a continuous manner that does not require direct communication
CONFERENCE ROOM
OPEN-PLAN OFFICE
providing an area for meeting dedicated to each Scrum Team
equivalence of all team members

Fig 13 Conceptual office arrangement for Scrum teams (Rola, Kuchta, & Kopczyk, 2016)

This article showed some interesting concepts related to the office arrangement for any Agile team that we could implement in any IT project. There was a degree of viability in suggesting this kind of solutions into our office environment. However, after the global pandemic stroke, the company has not returned into presential work. Thus, solutions related to office layout and spaces will not be further considered in this document and it is still unknown if Smartworking trends will stop any time soon.

4.5 Scrum and COBIT, possible challenges

COBIT, which stands for Control Objectives for Information and Related Technology, is itself a de-facto standard providing information technology (IT) governance model *(not to be confused with COVID-19).* It is possible to find cases in which companies used to COBIT framework should integrate their practices with Agile Scrum principles. Ozkan (2015) analysed this case study in his paper “Risks, Challenges and Issues in a Possible Scrum and COBIT Marriage” where he discusses the difficulties of merging to different work methodologies in a given IT project team.

To understand better why this frameworks are so diverse between them, we can consider the following points of reference discussed by some authors:

- While COBIT is process-centric and designed to standardize people to the processes, Scrum relies on people and their creativity rather than processes (Cockburn & Highsmith, 2001)

- Instead of over-optimized processes targeting 'best-practices' in COBIT, 'just enough' principle is enacted in Agile, to reduce any unnecessary effort towards over-optimization (Ktata & Lévesque, 2009)
- COBIT regards documentation as a means of storing, sharing, conveying, replicating and backing-up knowledge, planning, codifying and standardizing for practice, and creating logs for further use. Agile methods discourage heavy documentation and prefer to invest time in producing working software rather than in producing comprehensive documentation (Khan, Qureshi, & Khan, 2011)
- Planning and control accomplished by a command and control style of management are practices of COBIT, but agile teams are characterized by self-organization and intense collaboration, within and across organizational boundaries (Cockburn & Highsmith, 2001).
- Instead of managing risks with high assurance in COBIT, agile development confronts risks to tackle them empirically (Ktata & Lévesque, 2009).

Ozkan analysed both Scrum and the IT Governance model COBIT by doing an exhaustive literature review. As a conclusion of his research, he proposes several avenues of the future work for researchers (Ozkan, 2015). Some of these are:

- Developing extensions to Scrum framework to find possible solutions from the theory for the items addressed in their work
- Investigating issues in practice faced by organizations that adopt Scrum within COBIT environments
- Investigating issues in practice faced by organizations that adopt Scrum and matching the issues with COBIT practices
- Finding practical solutions for the issues coming from Scrum practices in COBIT environments and assessing their effectiveness
- Developing a COBIT maturity assessment of Scrum processes

As mentioned by the author, embracing two frameworks with different principles and values is not an easy task. However, it is still possible to get the best practices from both methodologies and put them together in a coherent way, according to the requirements of each single team. This situation may appear not only to Scrum-COBIT interactions, but also with any kind of juxtaposing frameworks, such as Waterfall and Scrum.

4.6 Governance Framework for Chains of Scrum Teams

Metioned in chapter 2, we know that Scrum is usually intended to work with groups from 3 to 9 people. But what about larger teams and interactions between different Scrum teams? This is an interesting scalable factor that poses risk for agile implementation into big corporations. How can a company handle multiple Scrum teams at once or Scrum teams with more than 10 people? This is when the governance framework for chains of Scrum teams comes into play.

Vlietland et al. used a qualitative approach with transcripted interviews from three case studies that were coded and analyzed to identify the issues (Vlietland J., 2015). They based their research method on Eisenhardt work which presents a process of building theory from case study research that has the following steps:

(1) selecting cases
(2) crafting instruments and protocols,
(3) entering the field,
(4) analyzing data,
(5) shaping hypotheses,
(6) enfolding literature and

(7) reaching closure

The researchers identified six issues in chains of codependent Scrum teams, which are coordination, prioritization, alignment, automation, predictability and visibility (Fig 14). The synthesis of these issues with existing theory resulted in nine propositions that were combined into a conceptual model.

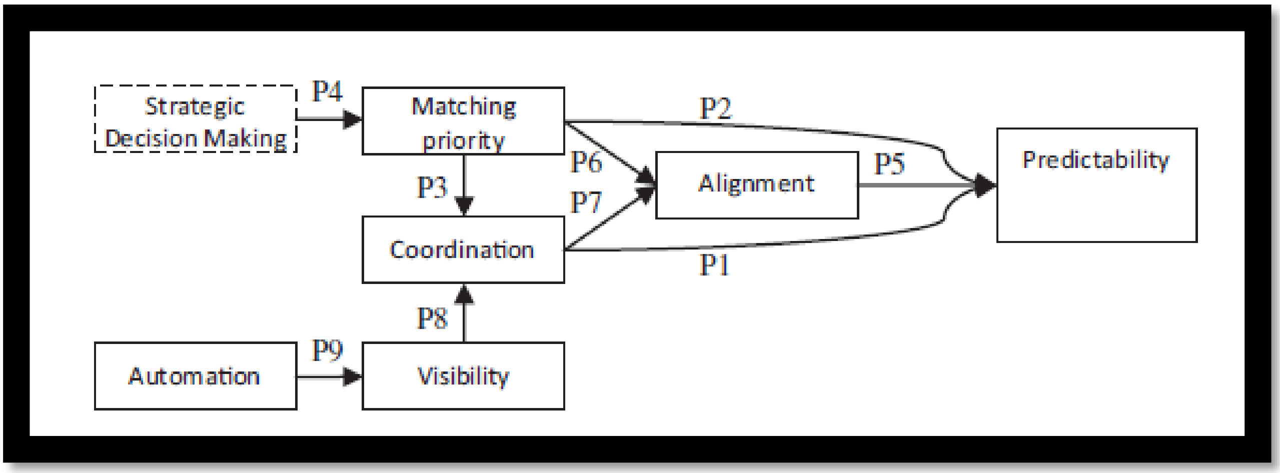


Fig 14 Resulting conceptual model (Vlietland J., 2015)

This study gives a unique perspective on how to manage multiple Scrum teams, something that can be applied to other companies in similar situations. As further research, it would be interesting to analyze and highlight governance within teams with different methodologies, such as Scrum and Waterfall, to understand how could we frame them and how could we theorize about these type of interactions, which are common in large companies.

4.7 Scrum Software Team adapting to Remote Work during Covid-19

In early 2020, a global pandemic has shifted the world unexpectedly into remote work and its effects are still outgoing. As many medical articles and papers about COVID-19 emerged in the midst of the pandemic, there has been also a good amount of new articles talking about its effects in white-collar jobs and, as no surprise; it has affected the IT sector projects as well.

Griffin (2021) points out how the heavyweight nature of Scrum is leading to fatigue within teams, as the remote nature of video conferencing and asynchronous communication bring an overhead not experienced before. As shown in Fig 15, phone and video conversations can still be effective, but not as much as face-to-face interactions in terms of richness of the channel.

The author argues that this result is a weakening of the Scrum fundamentals in teams, with modifications being made to accommodate the new normal that teams are experiencing. However, he concludes that Lean principles can still be applied to Scrum with minor adjustments and it is possible to obtain a variant of Scrum designed with remote teams in mind (Griffin, 2021).

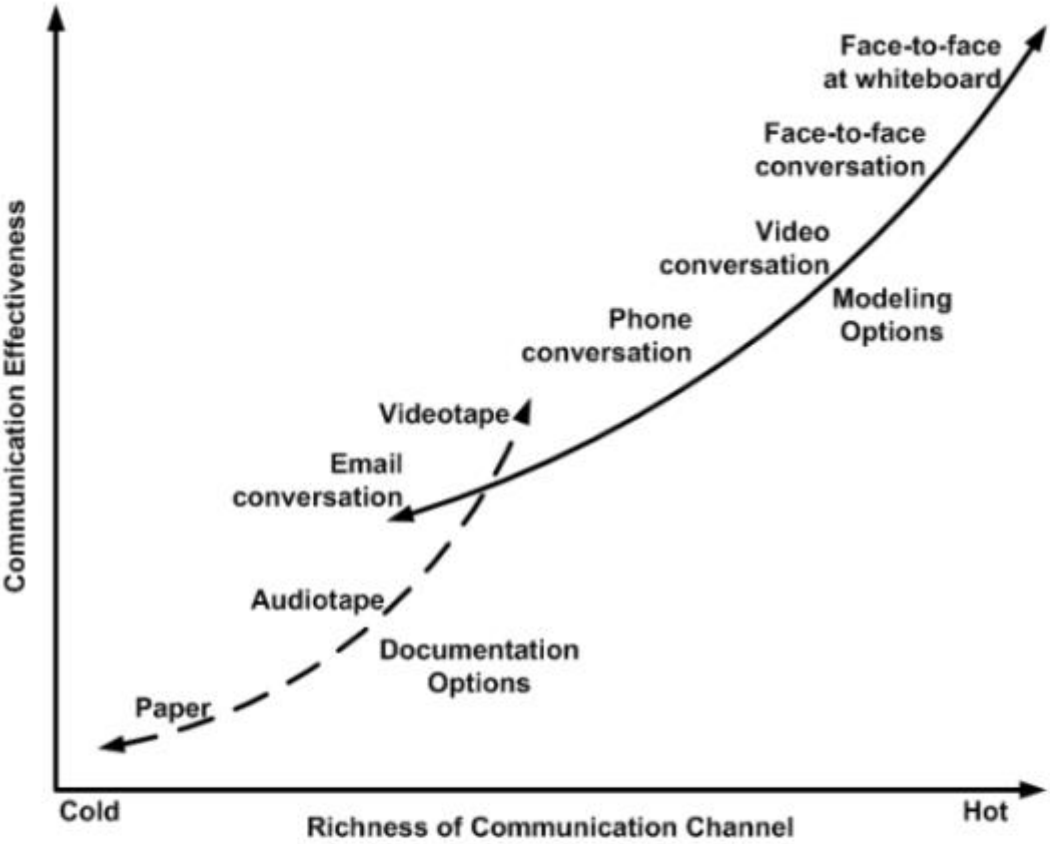


Fig 15 Graph that show different communication options in function of their richness and effectiveness. Personal interactions tend to prevail in effectiveness over impersonal ones (Geekbot, 2020).

This paper gives a true insight of the current paradigm shift to work which has led to a strain on teams like no other experience in their lifetime. Unfortunately, for the sake of this document, this paper was released on 2021 while the methodology and the results were established and obtained during 2020. Nonetheless, further investigation in this topic is highly advised and encouraged as this global situation evolves.

Using the previously presented cases and studies, we can pass to build our own Methodology for our research purposes. This will be presented in the next chapter.

5 METHODOLOGY

As a research project, the idea is to gather empirical information about how the implementation of Scrum is done. These can lead to several interesting scenarios since this team has quite a good amount of unique traits such as having the client being the Product Owner or having the rest of the company working in a non-agile methodology.

The main objective is to qualitatively collect data from employees and analyze it through a set of defined criteria to identify key issues and points of view perceived by the team members. Using this analysis, we can develop a plan to improve the implementation of Scrum in the team and facilitate collaboration, decision-making and teamwork inside the organization.

We expect to highlight interesting results due to the interactions already mentioned before and we hope this would be a contribution to the more practical side since this methodology is still quite recent and there is a lot of room to explore with the Scrum framework in real life cases.

5.1 INTRODUCTION TO THE SCRUM SURVEY 2020

The Scrum Survey 2020 at SIA PagoPA was held between the months of July and August. Each member of team Nodo and team PM was interviewed with a series of questions related to the Scrum methodology, the activities carried out, the working relationships with the group and the effect of Smartworking during the lockdown period due to COVID-19. The reported results were anonymized to ensure the possibility of sharing them within the group without any possible judgments from other team members or team leaders.

The survey methodology had a similar approach to the method used by Cho in his paper where he collected data through a formal face-to-face interview the team members and then audio-taped, transcribed, and coded raw data searching for patterns to identify team's challenge and issues (Cho, 2008).

The Scrum Survey 2020 at SIA, in contrast to Cho's methodology, had a well-established set of questions within predefined categories that enabled a focus on the most important issues. At the same time, questions were analysed and shown in various ways, including some numero-statistical methods to gain a deeper insight into the research and opinions while finding certain common trends. It is worth mentioning that this Survey included an important part concerning its current situation of transition during a global pandemic and the changes in working methodologies from remote, commonly known as Smartworking.

5.2 FORMULATING THE QUESTIONS

To create the survey, some base questions were proposed. These questions include different aspects within the team, such as roles, tasks, teammates interactions and of course Scrum topics. These questions, as discussed with the team leader, should be able to generate open discussions, which might lead to the discovery of different perspectives or issues that may not be seen easily by the upper commands.

It is important to highlight the origins and purpose of these questions. Starting by the interactions and team side; questions related to teamwork, communication and leadership are based on *"A teamwork model for understanding an agile team"* (Moe, Torgeir, & Tore, 2010). In this case, our objective is to understand how the team members work and interact with each other and who leads them. By the latter we can have a qualitative sense of how self-managing the group is.

Considering the quarantine panorama during which this Survey was held, some additional questions regarding Smartworking were included into the final questionnaire.

- ✓ How would you define your role and your contributions to the team?

- ✓ How do you estimate the duration of a task?
- ✓ How important is projectuality and long-term vision for the development of the project?
- ✓ How well do you know Scrum?
- ✓ How appropriate is the task distribution inside a Sprint?
- ✓ How would you define the interaction with consultants and external team members?
- ✓ How do you encourage the Scrum practices within your area?
- ✓ How would you describe teamwork and communication within your team and its challenges?
- ✓ How is the interaction with other BU / work areas?
- ✓ How would you define the leadership in your team?
- ✓ Which are the effects of Smart working and how would you describe them?
- ✓ Did you feel a strong change in workload during the time of the COVID?

5.2.1 Domande di esempio

Since the whole team is Italian and all of its operations are done in Italian, the sample questions were translated into this language to facilitate the reach to the people and generate a more casual feeling of the Survey conversation.

- ✓ Come definiresti **il tuo ruolo e i tuoi contributi** al team?
- ✓ Come **stimare la durata** di un'attività?
- ✓ Quanto è importante **la progettualità e la visione a lungo termine** per lo sviluppo del progetto?
- ✓ Conosci bene la **Scrum**?
- ✓ Quanto è adeguata la **distribuzione dei task** all'interno di uno **Sprint**?
- ✓ Come definiresti l'interazione con i consulenti e i **membri esterni del team**?
- ✓ Come **incoraggi le pratiche Scrum** nella tua area di lavoro?
- ✓ Come descriveresti **la comunicazione e il teamwork** col gruppo e quali sono le sfide?
- ✓ Come è **l'interazione con altre BU** / aree di lavoro?
- ✓ Come definiresti la **leadership** nel tuo gruppo?
- ✓ Quali sono gli effetti dello **Smartworking** e come li descriveresti?

5.3 GROUPING AND RE-FORMULATING QUESTIONS

Starting from the base questions, we notice certain pattern within the questions that could allow us to group them and enter into higher detail for each aspect. We created categories for allowing us an easier data processing. The main groups created were the following:

- Role and issues: This was used to start the conversation and to find common background in the team while discussing some of common issues and ways to improve in the team.
- Interactions: To inquire team members about teamwork, communication, relationship with client and other team members we created this category just for that.
- Scrum: As main topic of the Survey, it encompasses aspects such as knowledge level, implementation in the BU, Scrum events, Scrum artefacts, Scrum roles and alternative methodologies. In general, we try to answer the question: was Scrum implementation a success or a failure?
- Tasks: Everything related to workload, task distribution and team balance was included here. As an important topic, activities performed in the outbound of a Sprint, or Extra Sprint activities, are importantly considered and discussed.
- Smartworking: Given the current global events, we decided to include this part to ask about employees perception of Smartworking, pros and cons and the ideal Work-life balance.

5.4 CATEGORIZED QUESTIONS

Having some feedback from our colleagues, we noticed that certain questions were more useful or valuable than others. That is why certain topics were partially removed while others were augmented and deepened. This is the final list of questions and categories used for the Scrum Survey 2020:

1. **Role and Issues**
   - ✓ Which is your role inside the team?
   - ✓ What is your main contribution?
   - ✓ Is there a well-established long-term vision? If not, what is the team lacking?
   - ✓ Which issues have you perceived inside the group?
   - ✓ What would you recommend to improve? Any suggestions?
2. **Interactions**

   2.1. Teamwork
   - ✓ How are the interactions with your teammates?
   - ✓ How would you describe teamwork and communication within the team?
   - ✓ How are the interactions with consultors and external teams? What about other Business Units?
   - ✓ Who plays a leadership role inside the team?

   2.2. Client
   - ✓ How would you describe the interactions with the Product Owner?
   - ✓ Do you find the client is participating well enough on the planned activities?
   - ✓ Are these interactions good enough? What could it be improved?
   - ✓ How transparent is the client on their decisions and on the information shared?
   - ✓ Is the client well adapted to the Scrum mechanisms?
   - ✓ Is there a perception of confidence by the client to their team?
3. **Scrum**

   3.1. Implementation
   - ✓ What is your knowledge of Scrum mechanisms?
   - ✓ Rank from 1 to 10 your Scrum knowledge level.
   - ✓ Were you aware of these mechanisms before implementing Scrum in our team?
   - ✓ How much has your knowledge of Scrum changed since its implementation?
   - ✓ Do you find Scrum to be a suitable methodology for our project? Why is it so?
   - ✓ Do you encourage the use of Scrum within the company?
   - ✓ How could we improve the knowledge of Scrum tools and mechanisms?

   3.2. Events
   - ✓ How useful are the Scrum events?
   - ✓ Do you find them appropriate for our teamwork?
   - ✓ Which events would be more useful and which ones the less?
   - ✓ How is the Scrum Master's role within the Events?

   3.3. Ideal methodology
   - ✓ Have you heard about other Agile methodologies?
   - ✓ Do you think other Agile methodologies would be suitable for our team?
   - ✓ Do you think there is a better methodology?
   - ✓ Which framework would you propose to implement?
   - ✓ What do you think about going back to Waterfall methodology? Would it be better?
4. **Tasks**

   4.1. Task distribution
   - ✓ How do you estimate the duration of a task? Which methods do you use to assess its time?
   - ✓ How do you perceive the task distribution inside the team? Is it well balanced?
   - ✓ When you consider a task is completely done? Which KPI's do you apply?

4.2. Extra Sprint activities

- ✓ Which tasks are included inside a Sprint?
- ✓ Which activities are done outside of the Sprint planning?
- ✓ Do you participate in Extra Sprint activities?
- ✓ Which percentage of your time do these activities consume?
- ✓ Do you think Extra Sprint activities should have their own framework?

5. Smartworking

- ✓ How is your perception of Smartworking?
- ✓ Do you consider this change has been harsh or not too much?
- ✓ Which are the advantages of Smartworking? Which are the disadvantages?
- ✓ From 10 working days, how many days would you like to stay working at home?
- ✓ What is your ideal Smartworking balance?

5.4.1 Domande dell'intervista

As mentioned before, it was necessary to translate these questions before staring the interview with our team colleagues. During transcription to the Italian language, certain questions were slightly reformulated to fit better into the local idiosyncrasy, although their base contents remained unchanged. Here you can see the Italian version:

1. Ruolo e problemi

- ✓ Parla un po' di te e del tuo lavoro.
- ✓ Qual è il tuo ruolo nel team?
- ✓ Qual è il tuo principale contributo venuto dal lavoro col team?
- ✓ Esiste la progettualità all'interno del team? Percepisce una visione consolidata a lungo termine?
- ✓ Che problemi riscontri all'interno del team?
- ✓ Quali sono i suggerimenti per migliorare?

2. Interazioni

2.1. Gruppo di lavoro

- ✓ Come sono le interazioni con gli altri membri del team?
- ✓ Com'è la comunicazione e il teamwork all'interno dell'azienda?
- ✓ Come sono le interazioni con i consulenti e team esterni?
- ✓ Esiste differenza tra dipendenti e consulenti?
- ✓ È difficile l'interazione con le altre Business Unit dell'azienda?
- ✓ Svolgi un ruolo di leadership all'interno del team?

2.2. Cliente

- ✓ Come descriveresti le interazioni con il Product Owner?
- ✓ Trovi che il cliente stia partecipando abbastanza bene alle attività pianificate?
- ✓ Queste interazioni sono abbastanza buone? Cosa potrebbe essere migliorato?
- ✓ Quanto è trasparente il cliente sulle proprie decisioni e sulle informazioni condivise?
- ✓ Il cliente è ben adattato ai meccanismi Scrum?
- ✓ Senti fiducia da parte del cliente nei confronti del proprio team?

3. Scrum

3.1. Implementazione

- ✓ Qual è la tua conoscenza dei meccanismi di Scrum?

- ✓ Classifica da 1 a 10 il tuo livello di conoscenza su Scrum.
- ✓ Eri a conoscenza di questi meccanismi prima di implementare Scrum nel nostro team?
- ✓ Quanto è cambiata la tua conoscenza di Scrum dalla sua implementazione?
- ✓ Trovi che Scrum sia una metodologia adatta per il nostro progetto? Perché è così?
- ✓ Incoraggi l'uso di Scrum all'interno dell'azienda?
- ✓ Come possiamo migliorare la conoscenza degli strumenti e dei meccanismi di Scrum?

3.2. Eventi

- ✓ Quanto sono utili gli eventi Scrum?
- ✓ Le trovi adatte al nostro lavoro nel team?
- ✓ Quali eventi Scrum sarebbero più utili e quali di meno?
- ✓ Qual è il ruolo dello Scrum Master all'interno degli Eventi?

3.3. Metodologia ideale

- ✓ Hai sentito parlare di altre metodologie Agile?
- ✓ Pensi che altre metodologie Agile sarebbero adatte al nostro team?
- ✓ Pensi che esista una metodologia migliore?
- ✓ Quale framework proporreste di implementare?
- ✓ Cosa ne pensi del tornare alla metodologia Waterfall? Sarebbe meglio tornare a questa metodologia?

4. Attività

4.1. Distribuzione dei task

- ✓ Come stimate la durata di un'attività? Quali metodi usi per valutare il suo tempo?
- ✓ Come percepisci la distribuzione dei task all'interno del team?
- ✓ (Definition of done) Quando si considera che un'attività è complettata? Quali KPI applichi?

4.2. Attività extra Sprint

- ✓ Che tipo di attività sono incluse in uno Sprint?
- ✓ Quali attività vengono svolte al di fuori della pianificazione dello Sprint?
- ✓ Partecipi ad attività Extra Sprint? È frequente che ne partecipi?
- ✓ Che percentuale del tuo tempo consumano le attività Extra Sprint?
- ✓ Pensi che le attività Extra Sprint dovrebbero avere un framework diverso?

5. Smartworking

- ✓ Qual è la tua percezione dello Smartworking?
- ✓ Ritieni che questo cambiamento sia stato duro o non troppo?
- ✓ Quali sono i vantaggi di Smartworking? Quali sono gli svantaggi?
- ✓ Da 10 giorni lavorativi, quanti giorni vorresti restare a lavorare a casa?
- ✓ Qual è il tuo equilibrio Smartworking ideale?

5.5 SURVEY PROCEDURE

With each one of the team members, nine participants from Team Nodo and seven participants from Team PM, we planned an appointment on our calendar. Since we were all working from home, each Survey was done through a call using Microsoft Teams or Skype. Calls usually lasted for about one hour and questions were asked in no particular order. Each conversation was recorded, transcripted and kept in organized records according to the topic: Scrum, tasks and Smartworking.

We decided to use the Survey as a means of conversation and a way to open a discussion. In this way, the main objective was to encourage a detailed talk about Scrum mechanisms in play and issues surrounding the current framework. This way was preferred over a closed questionary to get a better idea of public perceptions of the aforementioned topics.

5.6 Survey Analysis

According to the information obtained through the surveys, different analyses were performed. Different kinds of data processing were done according to the type of data and the desired way to encompass information:

- Numerical values (in a given scale)
- Trend analysis
- Data meta-analysis
- Statistical report (PIE charts)
- Open related opinions and expert's opinions
- Word clouds with data analysis

The compendium of the results was finally translated and reported in three different ways: a long and extensive presentation with all the details, a short and didactic presentation to show as a summary for workmates and this document's version with further explanations and analyses, which is presented in the next chapter.

It is worth noticing that the results are shown in various ways depending on the questions formulated on the Survey. This means that some answers were used to denote opinions while others were transformed into values and statistical trends. Anyhow, this representation allows us to have a clearer concept and a full view panorama of the situation while commenting and extracting the most important parts involved.

6 Scrum Survey Results

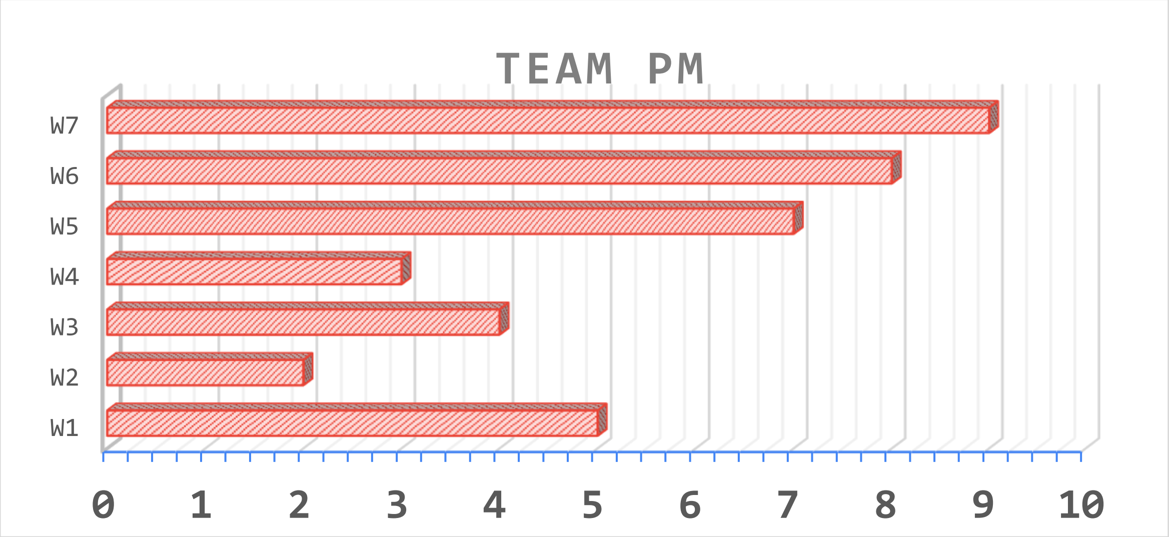
6.1 Scrum

6.1.1 Knowledge level

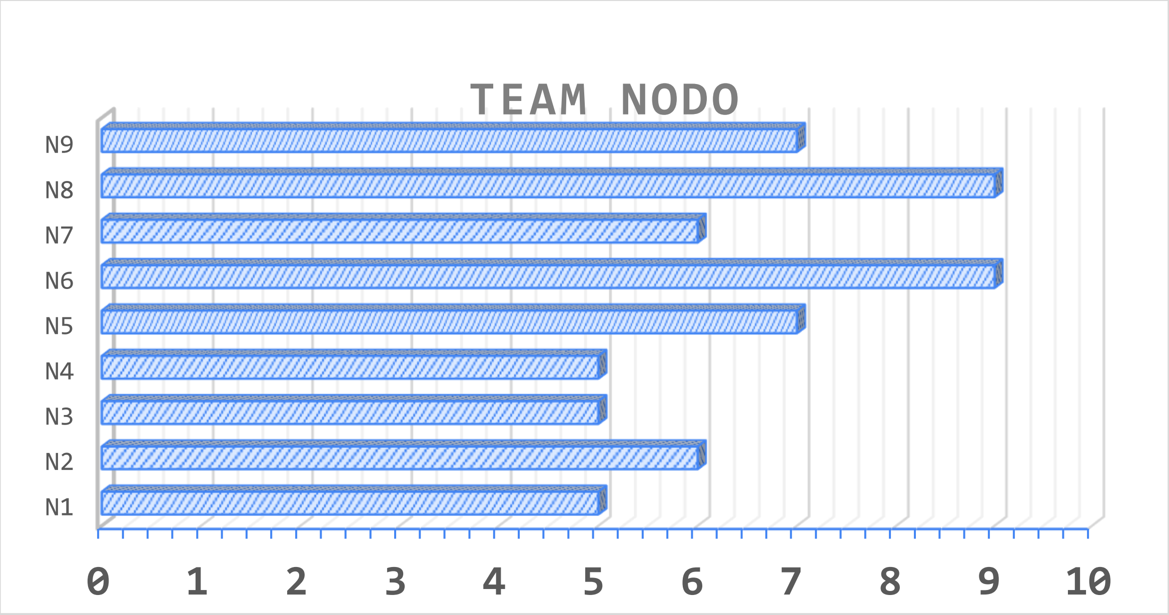
Team members self-assessed their level of knowledge of the Scrum methodology. Where, 10 means maximum competence and 1 means very little knowledge of Scrum mechanisms. A ranking for measuring someone's level in Scrum can be divided into Theorical experty and Empical experty on the field. Participants usually referred to their knowledge just into the theorical side. However, Scrum Experts tend to underestimate their knowledge since they attempted to communicate an average between their theorical knowledge and their experience. That is the reason why the question was reffered exclusively into the theorical part to avoid ambiguities and homogenize the scale of reference where zero meant absolutely no knowledge in Scrum and 10 meant excellent theorical management of Scrum mechanisms.

Team Nodo evaluated itself with an average of 6.6 (Graph 2); a slightly higher average than the PM team with an average of 5.4 (Graph 1). This different may be purely coincidental, but together with other factors it could be a symptom of a team which knows more about Scrum than the other. However, after these analysis it is reasonable to question if this difference translates strongly into the practical field.

The disparity for Team Nodo seems considerably smaller than Team PM. While team Nodo ranges from 5 to 9, team PM range wider from 2 to 9. According to the answers given, team Nodo has a more homogenous team in terms of Scrum knowledge, which could potentially be a great advantages to follow the events and the artifacts.



Graph 1 Level of Knowledge for Scrum - Team PM



Graph 2 Level of Knowledge for Scrum - Team Nodo

Here there are some mixed opinions about the knowledge of Scrum and the experience for team members. As an important event, we can consider that in the beginning, there was an immersion with a coach Scrum for learning the basics and guiding the process of methodology changing.

What is your opinion about the team's knowledge of Scrum? How could we improve it?

- *I have never looked at Scrum mechanisms even for general knowledge.*
- *On a practical level, that of the Scrum that you need to know we know.*
- *Unfortunately, time is wasted knowing what Scrum predicts or not.*
- *The dive with coach Scrum was lighter.*
- *Before starting I was at zero. I only have seen it academically at university.*
- *Either you have to train me or get a specialized person.*
- *I knew practically nothing about Scrum, already empirically in the field I understood many things.*
- *You should learn the practical side because you can study the theoretical side.*
- *I have a minimum, elementary level because I'm working, I'm not doing Scrum.*
- *What I know is more than enough for my current role.*

As seen by certain employees, not everyone is willing to increase their Scrum knowledge level. Some find it as a waste of time while others think they already know enough for what is required at job.

6.1.2 Scrum Implementation:

In the wordclouds we'll find the most frequently used words for topic. For Scrum Implementation we can some of the most mentioned concepts are "method", "project", "knowledge", "course" and "team". Some less obvious words for this topic could be "adapted", "visibility", "client", "product", "principles" and "functional" (Fig 16).

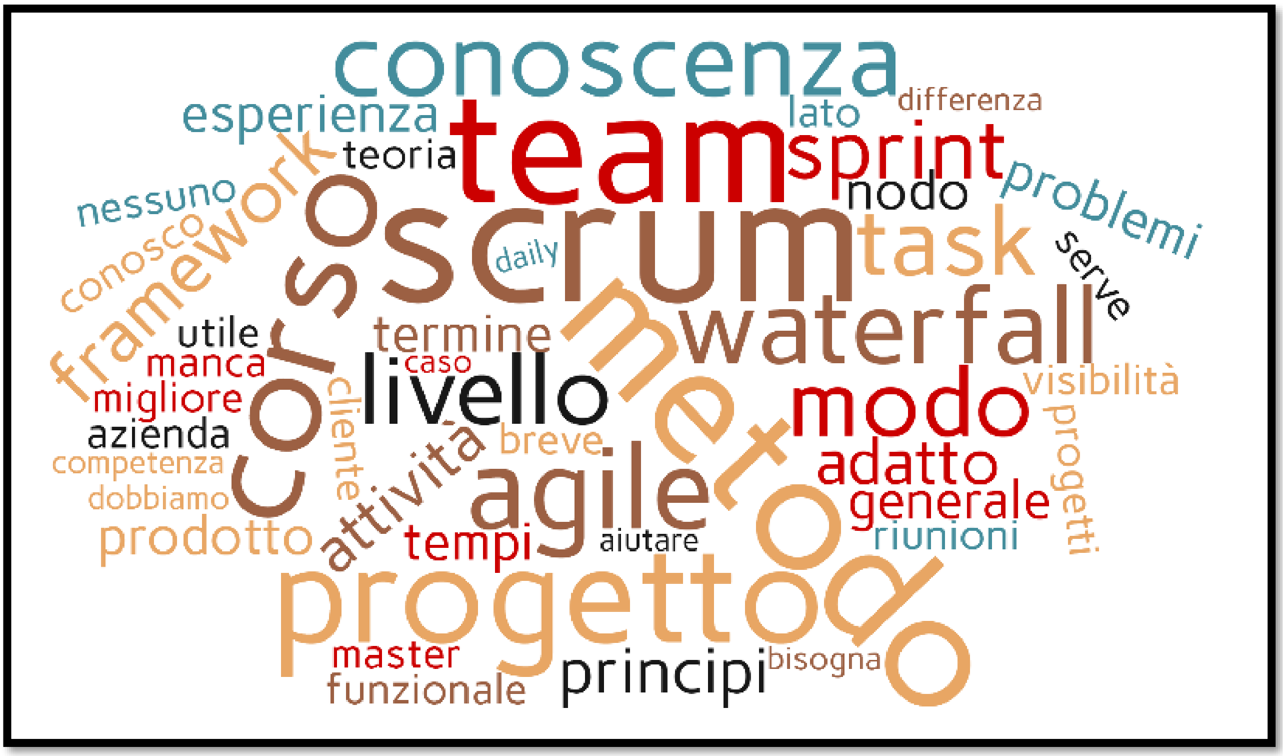


Fig 16 The Scrum implementation word cloud

What is your view about the Scrum implementation?

- *It's okay for the whole team to attend meetings, increase visibility on what needs to be done.*
- *I perceive a general level of dissatisfaction in my colleagues.*
- *So far, I have to say that the Scrum Master has always handled the situation well.*
- *I work in a group that uses these things but it's not my job to do Scrum.*
- *What we do is a fairly agile Scrum.*
- *It was my first experience working in a Scrum environment.*
- *From literature, Scrum serves in a complex scenario. This is not the case in our case.*
- *Before only the analyst knew what to do, now the developer also has an idea.*
- *We worked with this methodology only for this project.*
- *This methodology is very stimulating because we have more communication with the customer.*

Not many team members are used to the current methodology for working. Scrum environment is a new experience for most of them. In the case of the roles, they seem to be mostly well defined with some particular effects on the framework.

Is there a need to take a course?

- *I don't have the extra time to devote to taking a course.*
- *I feel like it's a bit late to take a course now.*
- *I don't know if a Scrum course is needed but you might like it.*
- *I took courses and I think I have deepened enough.*
- *I don't think that on a professional level a course serves to add that much*
- *In my opinion, you don't need a Scrum certificate*
- *I apply the things that are asked of me but I have not taken courses, I know the elementary principles.*

- *In reality, it is a matter of time to master.*
- *I took a course and got certified. It was a voluntary choice.*
- *We did mini courses with a Scrum guru.*

The idea of doing a Scrum course does not seem particularly popular among the team members. It's usefulness of possible advantages are clearly not been perceived due to the present environment the team faces. Perhaps the theoretical appropriation of Scrum does not seem as an attractive idea.

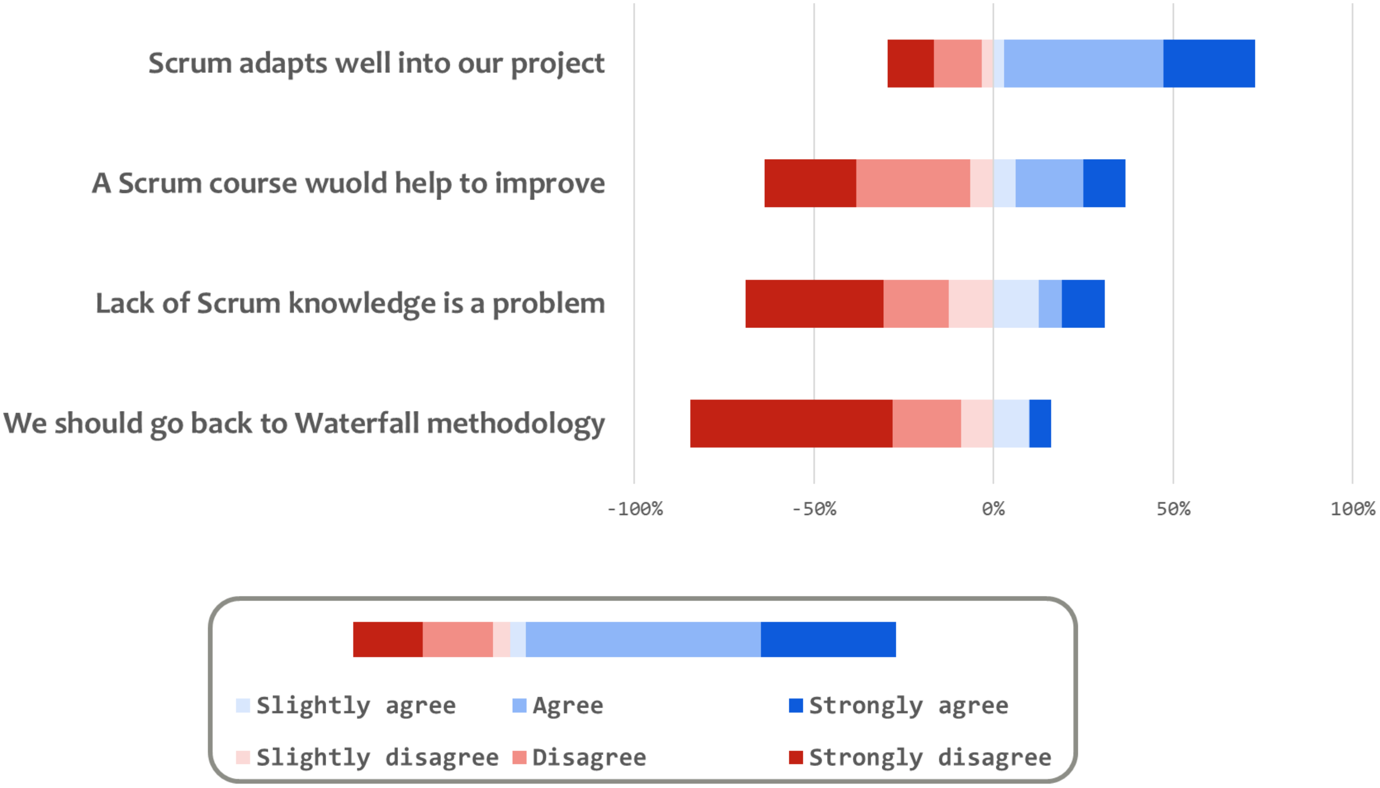
Fortunately, for both teams, the Scrum Master is considered as a Scrum expert thanks to his preparation in Scrum mechanisms, understanding artifacts and Events and how the Principles should be adapted into the project. The Expert's opinion is crucial in this context to guide the discussion and provide a much more detailed view of the panorama.

Expert's opinion:

<< The team is already used to working in an" Agile "way. Unfortunately, the team is not aware of or interested in the principles of the Agile manifesto and the principles and pillars of Scrum. The result is a " zombie " application of the Scrum framework : team members participate in Scrum ceremonies like any other activity that needs to be done because " so it was decided to do ". In general, they do not show interest in deepening the contents of Scrum ceremonies or in using typical tools such as user stories, story points, velocity, burn-down charts, etc. >>

Scrum trends

Each interviewee valued how strong they agreed with the following affirmations (Graph 3). Blue colors to the right represent to agree, red colors to the left represent disagree. These bars can give a general idea of how different opinions are distributed across the team.



Graph 3 Scrum trend bars

Overall, the general idea of Scrum seems to catch the eye on the team members compounding with the rejection to turning back to Waterfall methodology, both ideas go in hand. Strangely enough, a lack of knowledge on Scrum mechanisms or having a Scrum course do not seem as important concerns for the team, probably to a highly practical approach rather than theorical.

Which are the benefits on the use of Scrum?

- ❖ *With Scrum you can work in parallel and the tasks are smaller.*
- ❖ *A small indication is enough to start working.*
- ❖ *Scrum is a tool like many others and allows you to change your vision and direction.*
- ❖ *The best thing about Scrum are the increments.*
- ❖ *The level of agility allows us to manage our tasks better.*
- ❖ *With Scrum we can release evolutives continuously.*
- ❖ *Scrum is more for projects that constantly have multiple changes in short amount of time.*
- ❖ *You can apply Scrum wherever you want. For unknown stuff, the Scrum approach is more suitable.*
- ❖ *You have the opportunity to monitor systematically what is happening in the project.*
- ❖ *For me, having short-term goals is best.*

Most of the team members can find advantages in Scrum such us its agility, short-increments and adaptation to rapid changes. It offers a more instant approach that seems to be well received in the teams.

Is the most adequate framework to implement? Why do you think so?

- ❖ *It is adequate for how the client wants to interact with the team (immediately).*
- ❖ *Since the project is in its final phase, I would say that for this specific project it is a bit too much.*
- ❖ *As for the development for our project, perhaps it is not the 100% indicated methodology.*
- ❖ *Yes to the framework, we can adapt to use it but we could do it much better.*
- ❖ *Scrum is not doing particularly well simply because we do not believe in it.*
- ❖ *It would have been better to apply it from the beginning of the developments. I would apply further the Sprints logic.*
- ❖ *There are no big deadlines for Scrum and when you use Scrum, you start arguing.*
- ❖ *Scrum works if there is not too much stuff to do.*
- ❖ *I don't know if Scrum is the most suitable framework, maybe other methodologies are*
- ❖ *I think agile is very suitable for the PM team because it is a very inter functional team.*

Inside the team, there are nonetheless various doubt on whether Scrum is the most adequate methodology to implement. Some argue that it is already late to implement it or that simply there are better frameworks out there while others claim is a matter of the team who does not want to adopt it within their working process. In this discussion, time of implementation, type of project and faith from the team seem to play important factors.

Which problems have you found during Scrum's implementation?

- ❖ *I don't know if it's due to Scrum, our application of Scrum or it's us who aren't capable.*
- ❖ *Maybe it could be something more organized.*
- ❖ *It is a bit of a mixed bag because sometimes there are few tasks and other times there are more.*
- ❖ *I don't think the problems are related to Scrum knowledge.*
- ❖ *I have the impression that the approach is in the meantime, then if it doesn't work we change.*
- ❖ *We cannot think of a change and apply it immediately.*
- ❖ *Not enough previous thought has been given. All that we have done is to go backwards.*
- ❖ *The fact that ideas change so quickly is not good.*
- ❖ *We try to run more than we are able, more certainties would be needed to do so.*

- *There is no one in the organization who wants to implement Scrum.*

Team members for this question might be somewhat unsure whether the blame on the issues found should be given to the Scrum methodology or not. It is up to each member but specially duty of the Scrum Master to judge whether the framework resides useful or not much at all.

Expert's opinion:

The Scrum framework ***is suitable*** *for a team like that of the PM because:*

1. *the team is cross-functional*
2. *the team deals with the "continuous development" of a product that is already in production*
3. *all activities are represented by additions / changes / corrections to the product performed incrementally*
4. *the product is used by customers (B2C) and therefore it is possible to follow the principles of "user-centric design" (eg: user stories)*

The Scrum expert enumerates the reasons why Scrum is a good option as a framework, at least for team PM, in the case of Team Nodo the answer might be more on a traditional level.

What has changed since Scrum was implemented?

- *Things change names but in the end the work is more or less the same.*
- *What I do hasn't changed much.*
- *I'm enjoying the way meetings go.*
- *Before, the work was more uncertain. It's better with the Scrum methodology and I don't see any drawbacks.*
- *At this point I feel we work in very short waterfall.*
- *If like us we are a small team, I have not seen such a big difference.*
- *I feel like we are a small waterfall model within a sprint.*
- *Before we had no confrontation with the OP. Now the tasks are opened directly by him.*
- *Releases are more frequent, and that's better.*
- *Now the whole team has general visibility of everything thanks to meetings.*

About half of the people observe important changes with the Scrum methodology. However, the other half has a harder time trying to recognize them. That can be a huge challenge for any team trying to adopt a new framework or methodology.

Would it be better to change the methodology?

- *I am not saying to go back to the old framework, but it lacks some organization to the current methodology.*
- *I feel that the methodology hardly changes my work within the team.*
- *With Scrum you never get bored, always in two weeks something new comes to you.*
- *The agile methodology allows you to reduce time by overlapping activities.*
- *Specifically, I don't know if there is a better methodology, but the Waterfall is not.*
- *We can do it all with Scrum. Scrum lends itself well to uncertainties.*
- *If we were to go back to the waterfall methodology I would have no problem.*
- *We can go back to the classic one with Project Management from PMBOK.*
- *I don't know if the other methodologies are better or worse than this one.*
- *The project is too dynamic that using the classic Waterfall would be really difficult.*

Some members show concern in coming back to the previous framework, however there seems to be members which don't find any try usefulness to maintain or even preserve the Scrum method since its benefit may not be immediately received.

Three different points of view given by different team members, each one describing how useful they find the methodology:

- ✓ *I'm sorry but I don't see the usefulness in the methodology. We have technical and practical problems to solve, and methodology doesn't seem to me to be the tool that can make a difference.*
- ✓ *I think that right now for the level of the team and the product this is enough. The team has found its own dimension of Scrum which will not be that of literature but which is very effective. Pure Scrum in real life may not even exist.*
- ✓ *The team must look for an appropriate methodology for the project. There are so many agile modes that the team could implement. However, good use of Scrum by the whole team, including the OP, can substantially help this project and others to come.*

6.1.3 Scrum Events

In the Scrum events wordcloud we can see the main words tend to be related mostly to the Scrum ceremonies. However, they are not all talked about with the same frequency. "Planning" and "daily" tend to take the lead, followed by "retrospective" and "refinement". "Review" seems to be one of the least discussed about (Fig 17).

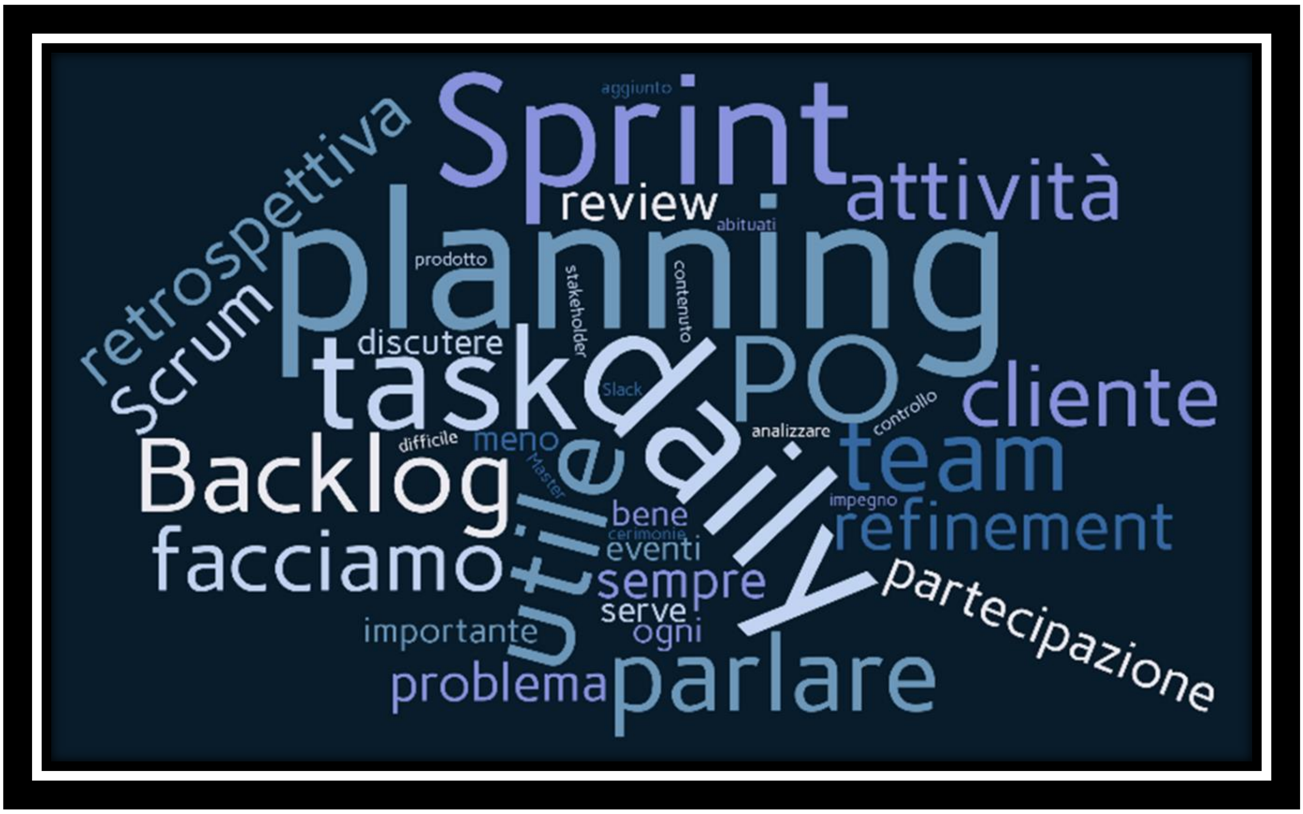


Fig 17 Scrum Events wordcloud

What is your general overview about Scrum events?

- ❖ *So many ceremonies for 4 little things seems to me a little too much.*
- ❖ *Scrum events are just a name for me. They didn't turn my life upside down.*

- *Maybe we weren't used to talking to everyone.*
- *The number of micro meetings should be reduced as there is often nothing to say.*
- *Now we are at the end in practice, or there are no big tasks to do.*
- *It was decided to exclude part of the team so as not to steal too much time.*
- *I have no added value to bring because I am not carrying out activities within the Sprint.*
- *To say there is nothing to discuss means that the team has not focused on any task.*
- *They are useful but it is heavy that they are always with the customer.*
- *The tasks are calculated so that everything flows well, even if we feel like after 3 days.*

The team overall has a mixed opinion about the Scrum ceremonies. Some argue it may be excessive, others find them to be time consuming. Lastly, there are some team members which blame the team itself for not finding them useful.

Ranking for Events per team

Each team member has identified which event contributes most to the working group for them and at the same time which event is the most redundant or, the one that fits the least in the current implementation of the Scrum methodology. Here we should the results:

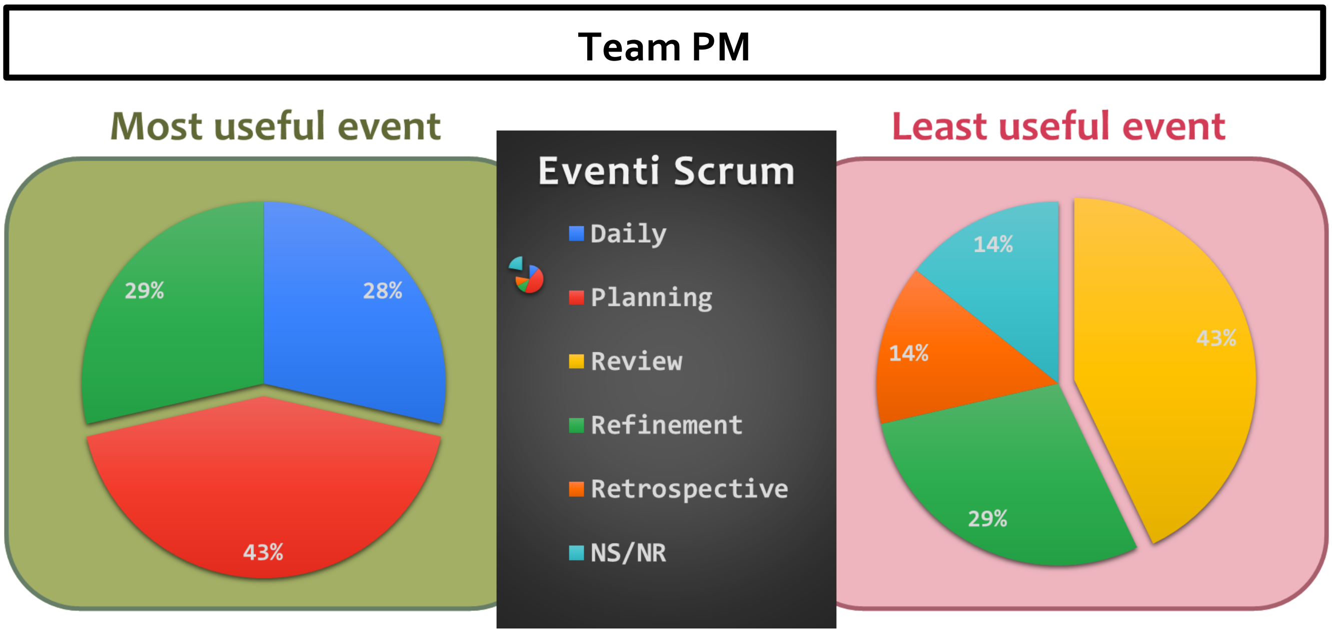


Fig 18 Team PM answers to most and least useful Scrum events

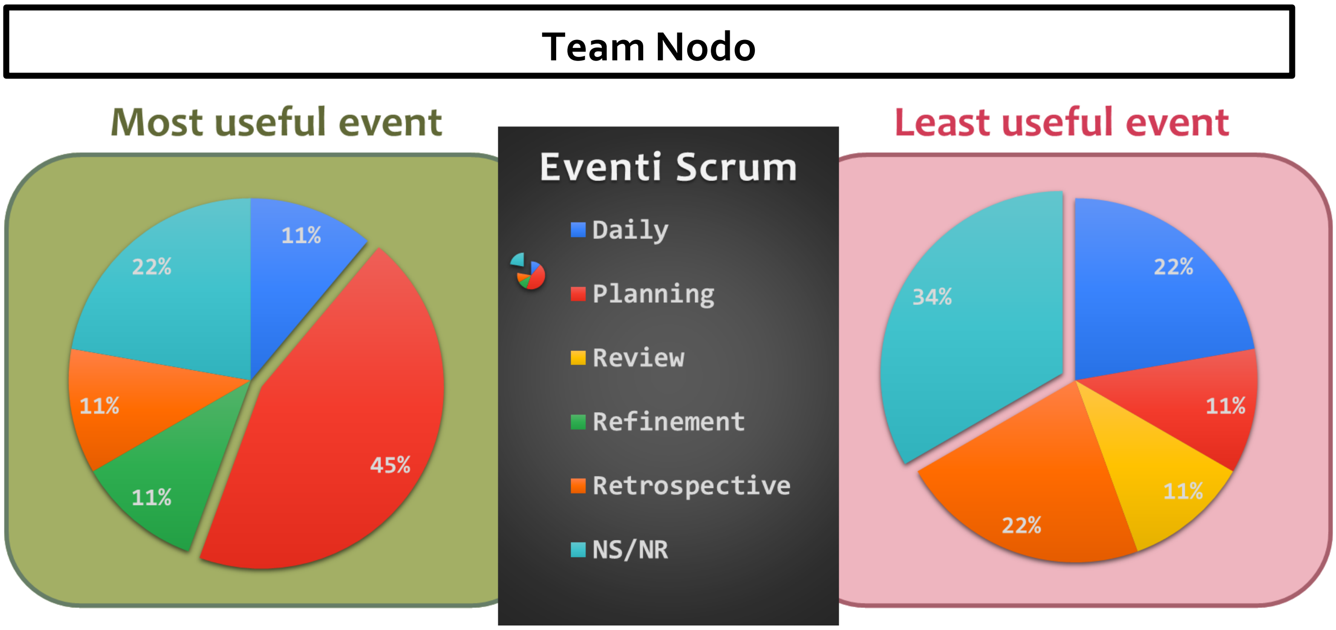


Fig 19 Team Nodo answers to most and least useful Scrum events

We can see that both teams have regarded the Sprint planning as the most useful event while there is not consistent choice from the less useful event. Nonetheless, we can notice some patterns such as the Daily Meeting in Team PM where it is considered quite useful (Fig 18) while it is not the same for Team Nodo, which had a higher rate of not answering the question at all (Fig 19). To summary, some general trends: While Sprint Planning and perhaps Backlog Refinement tend to have a positive opinion; Sprint Review and Sprint Retrospective tend to not be seen very value adding.

Opinions for Scrum Events

In this section, we collect all the opinions that make reference to a particular Scrum event for both teams. In principle, both teams have a similar way of carrying out the events (especially by the fact that they share Product Owner and Scrum Master), however in practice this perception can vary widely from event to event.

Daily meeting:

- *With the daily meeting we understand who needs help and we talk to each other to understand who does the activity or not.*
- *The most useful is the daily because people were not used to talking and confronting each other every day.*
- *The dailies are essential because we all keep up to date.*
- *Now the daily is exploited properly, we bring everything inside, everything we need.*
- *The OP is almost never present during the dailies, but I don't know if according to the Scrum framework he is required to participate.*
- *The daily in my opinion is superfluous. I find it more like an excuse not to contact us right away.*
- *I find that, in the daily meeting, sometimes there is nothing to say and we could avoid it.*
- *Sometimes the daily is skipped, however we always have the situation under control.*
- *The useful time is significantly less than the utility, so the utility is so high.*
- *The daily is certainly useful to align ourselves every day with what we do.*

Most of the team members find the Daily Meeting as a useful tool update the team and find where to focus the daily task. However, we can highlight as a red flag the fact that it is sometimes skipped even though they are just 15 min. This may be a signal it is not always carried out properly or there's a lack on the desire for team communication.

Sprint Planning:

- *Sprint planning is not done in an optimal way.*
- *The customer throws out tasks without the team being prepared to collect them.*
- *In planning, everyone's participation is very useful.*
- *In Sprint Planning we are never precise.*
- *Sometimes we do damage because we say things can be done but then we can't.*
- *In planning we have a vision of everything there is to do.*
- *Doing a good planning leads you to do a good Sprint.*
- *I would like to discuss more during the Planning. With the other half hour we could discuss the most difficult things.*
- *It let us all decide together the tasks to carry in the next Sprint. I find it very useful.*
- *Analyzing much better what you do during planning would be helpful.*

Both teams regard the Sprint Planning as an event that they cannot miss since it contains a lot of value for everyone. Some team members even argue that it should be longer. Critiques in this event tend to be more on the Product Owner which may not precede the meeting as he should.

Sprint Review:

- *In the review we show the customer what we have done.*
- *There is no inspection to find out if what has been done actually meets the needs of the stakeholders.*
- *There is no adaptation because nothing is said about how that fact contributes to the evolution of the product.*
- *In the Review, the OP can receive feedback and discuss possible changes in the roadmap.*
- *The review shows the increase, so everyone can decide whether to bring it into production.*
- *I participate in the review a few times and also for the relationship with the client.*
- *The least useful event is the review.*
- *In theory, it is used to discuss the progress of what was produced in the last Sprint.*
- *We should talk about the plan and what to do in the future.*
- *I also like the fact that we do the reviews and refinements. They work well.*

In the case of the Sprint Review, not many people discussed about it. In fact, most of the opinions presented here come from the Scrum Master, which was one of the few that understood the event and saw its value. It seems that not many team members have a hint of what is the Sprint Review intended for.

Backlog Refinement:

- *The analysis should be done in the refinement backlog.*
- *Refinement is the one that can be most useful. It is more an analysis of the service, not just the task.*
- *4 hours a week of Backlog refinement, I would never take them away!*
- *The planning and the Backlog refinement are the most useful.*
- *Few do the Backlog refinement, only analysts participate.*
- *I also find the Backlog refinement useful for the other phases of the Sprint.*
- *I find the refinement would be the less important event since, in my opinion, we could do it very well in planning.*
- *We do this because the Scrum Master has set it as a commitment on Slack.*
- *I haven't done many because usually others participate.*
- *It is not okay for the OP on Planning to insert different tasks to those of the refinement backlog.*

For this event, we start grasping the problem of inequality inside the teams. To "save time", it was regarded as unnecessary the participation of the developers in the Backlog Refinement, allowing only analyst to participate as mention one on the opinions. This happens also for some other events but it basically breaks an important premise of Scrum methodology as a team and perpetuates the equilibrium between its members.

Sprint Retrospective:

- *The retrospective seems to me very useful but the format seems to me a bit similar to the ine used by anonymous alcoholics.*
- *You are obliged to say certain things between us and with the client. What went fine and what went wrong.*
- *The retrospective takes too long. We dedicate one hour every two weeks.*
- *We always say the same things, the organization of work. What was the biggest problem.*
- *I find the retrospective useful because it helps to solve negative things.*
- *It leads you to improve the planning of the next Sprint.*
- *It's a bit superfluous and doesn't last long. I feel the situation is well monitored on a daily basis.*
- *The retrospective is not so useful because it is not strictly necessary.*
- *Planning is obviously the most important. Then the retrospective, quite important.*
- *Planning is the most useful because it is well done. Instead, for the Sprint Retrospective we don't arbitrarily place tasks but we decide how many resources are necessary.*

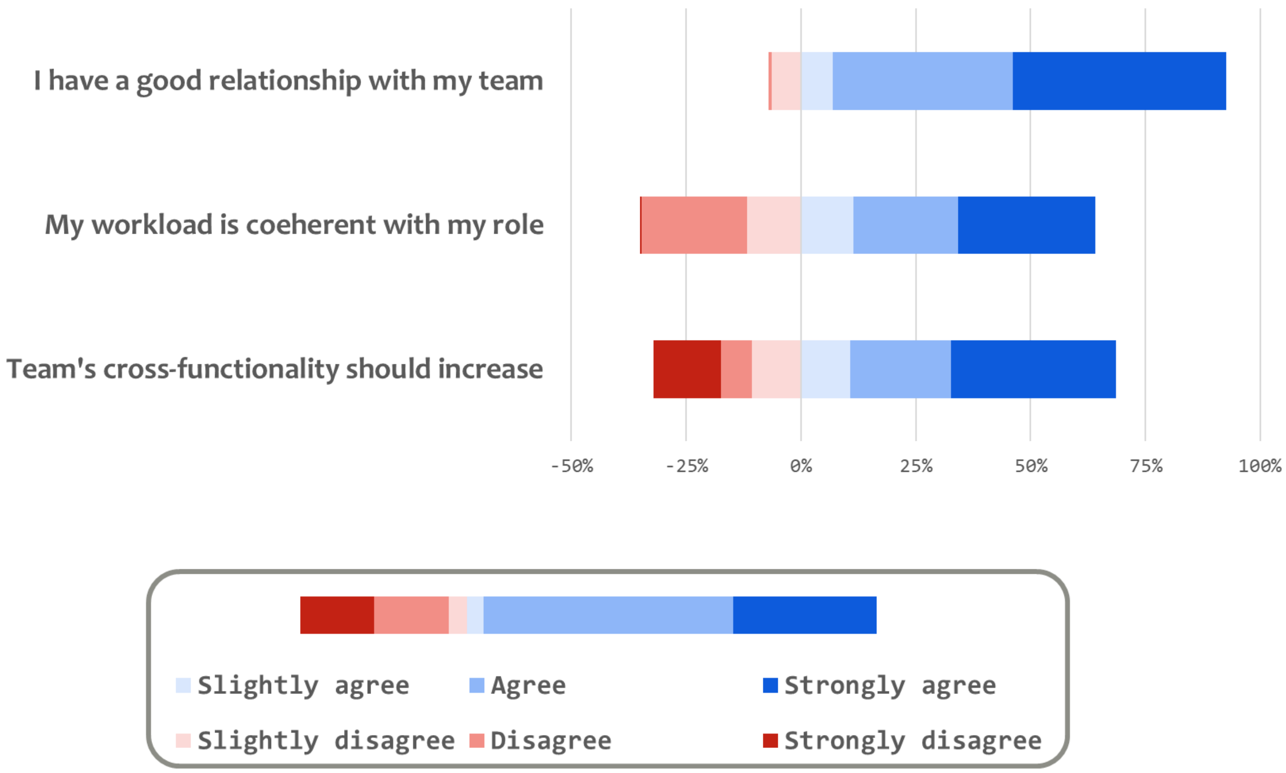
Taking into account that this event is mostly considered as a feedback for the whole Sprint, opinions are quite divided whether they find it useful or not since it can sometimes be repetitive and superfluous.

Overall, the Scrum events need to be revised in both teams to strive for higher standards and more rigorous time scheduling. Likewise, an basic knowledge of the Scrum events should be carried out by everyone as an equal participation on every event. In this way the Scrum principles can be followed in a better way while fomenting the framework way of thinking with the help of a committed Product Owner and a helpful Scrum Master. At the same time it is quite important that the team members put a part of their effort into increasing the curiosity for Scrum events and all other Scrum artifacts.

6.2 TASKS AND INTERACTIONS

Trends

Following trend bars, as shown in the previous section, this time we analyze aspects regarding to interactions, teamwork, communication, roles and tasks. The results show there a good and welcoming working environment since none of the employees reported having trouble with other team members. Taking about how coherent are tasks related to their roles, things start getting a little debatable. A more controversial idea that arose was about team's cross-functionality. Even though this is considered a pillar thought within Scrum ideals, up to a 30% percent of the team does not agree with it.



Graph 4 Task and interactions trend bars

6.2.1 Team interactions

Some opinions about the interactions were recorded for each team between team members and the Product Owners. This interactions may include opinions on teamwork, communication, roles and leadership.

Team NODO

How are the interactions between the work groups?

- *We are an old and self-organized team.*
- *I already know how to intervene with my teammates.*
- *It tries to be replaceable by one among the other. Now the knowledge is always divided in the team.*
- *There are developers talking to each other, analysts talking to each other.*
- *Communication between us is easier when it is informal than when it is formal.*
- *In my opinion there shouldn't be a leader within the team. We have a Scrum Master.*
- *I don't perceive the difference between employees and consultants and I feel quite involved.*
- *The development guys have been here for a long time and there is mutual trust and a good relationship has been built.*
- *Everyone on the team is very competent, they are very smart.*
- *I talk to colleagues every day. If there is any problem, we can talk to you on Skype.*
- *There is a habit of communicating with individuals but the information is not being shared.*

The team Nodo has been working for a long time and is it used to work in a more informal way. Communication inside the team has turned into a matter of closeness between teammates with a similar role. However, this communication model is detrimental to the interdisciplinary cross-functionality approach intended for Scrum.

Expert's opinion:

<< All this depends on the complexity of the service. We are divided in this way because we have a fairly large service. Then it's clear that the Scrum framework says the team should be more cross functional. It should be allowed to go beyond the limit. It is a question of finding the right balance.

The PM team are better at going beyond the limits. We have more demarcated roles and we are stuck in our roles for various reasons. The service has become more complex and in my opinion the balance is found with more defined roles. >>

The Scrum experts analyses a common deviation in performance where one of the teams has grown more independent than the other.

Team PM

How are the interactions between the work groups?

- *Periodically we met in the room to discuss the activities to be done.*
- *The team is well organized and we are the whole team together.*
- *I feel good with the team because there are daily meetings. There is ongoing mutual support.*
- *After the meetings on Slack we have a more technical meeting to divide the tasks to be done.*
- *Good degree of assortment between people. We all work together and there is support.*
- *The team is really a very cool team, there is a lot of collaboration.*
- *You can work better if we were all blended better and not divided into many different small sections.*
- *Interactions are great and they were also good before.*
- *We do not give details of who did what, unless asked.*
- *Within the Scrum team, I don't express myself very well.*

Most of Team PM members have a positive assessment over their team interactions and performance. This is an indicator that they are going into the right path for enabling a Scrum process.

Expert's opinion:

<< The interactions with internal and external colleagues are generally good: the team has always worked in an Agile way (even if not formally) and is used to dialogue in order to obtain a shared understanding of the activities to be performed in each sprint. Unfortunately, in the "refinement" phases the internal colleagues are inhabited in a 1-to-1 dialogue which, in many cases, does not allow all team members to be aware of what has been said or decided. Likewise, during the sprint, external colleagues refer to their own representative who centers all communications on themselves. Consequently, it is not possible to have a transparent and shared vision of the progress of the activities during the sprint. >>

Communication methods show improve to avoid information clusters reserved for only some team members. This transparency issue tends to be hard to handle when people of different companies compose the team. Each one with their own objectives in mind.

Interactions wordcloud

For this cloud the words change considerably in comparison to the Scrum wordclouds (Fig 20). Here, the main words are "client", "team", "task" and "communication". Some Scrum Sprint words also appear here, while the most interesting words from a teamwork perspective are "together", "involved", "difficulties", "equal", "leader", "visibility", "dialogue", and "close to".

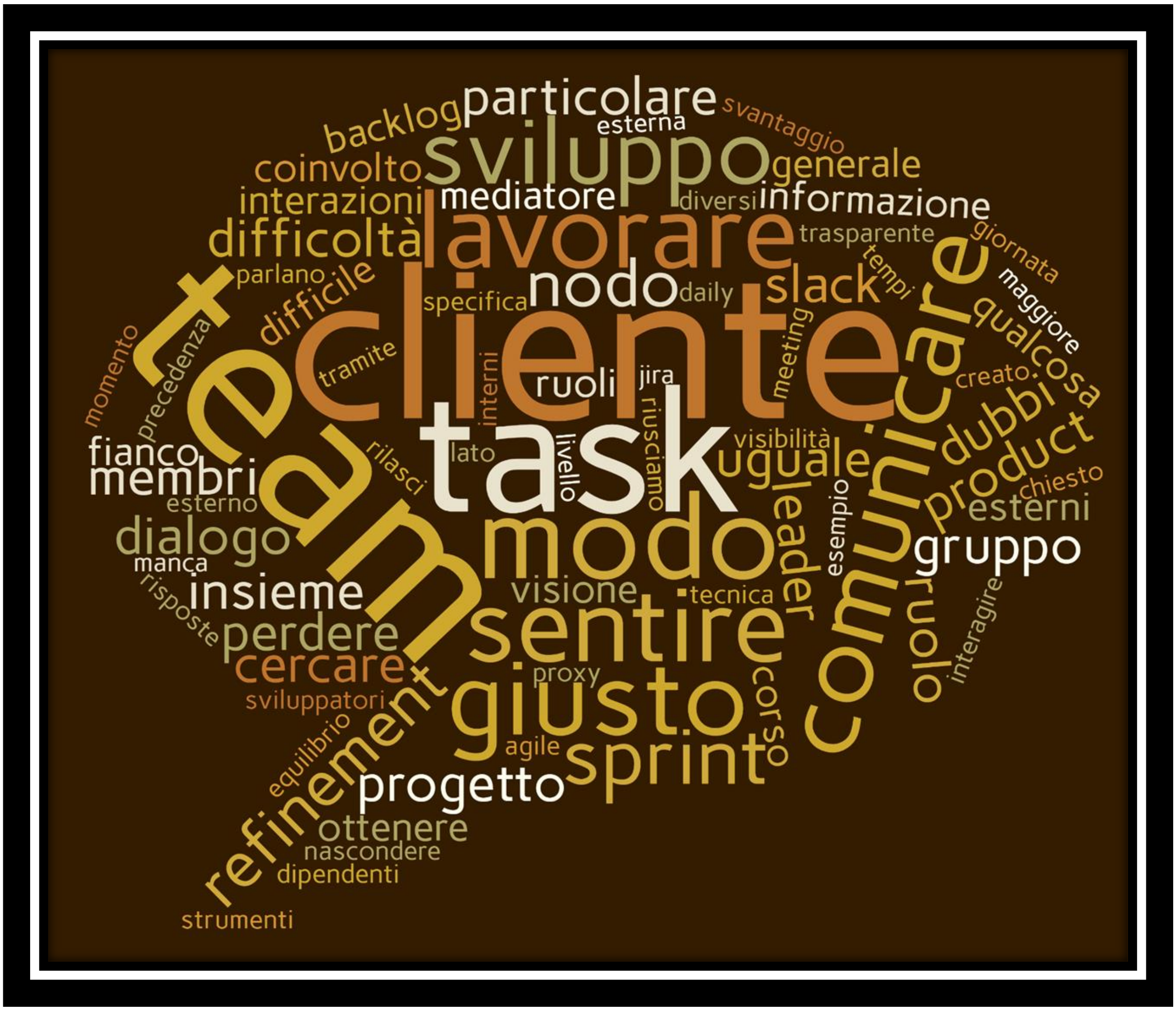


Fig 20 Team interactions wordcloud in dialog shape

Overall aspects with other entities

How is the interaction with external teams?

- *With the other BUs you must always interact with the old methodology.*
- *The approach with the other BUs is a bit difficult. There are no channels, there are no procedures.*
- *We are not self-consistent, we need the rest of the company.*
- *There isn't much interaction with the company. The most commercial relationships are through them.*
- *There are procedures due to the fact that it is a large company.*
- *We rarely talk to the guys outside our BU, maybe a little bit with the guys from team Nodo.*
- *Whenever an external BU is needed, we waste time.*
- *If you save us time, it would be better to do it ourselves.*
- *Sometimes I interact with the Nodo team but it is a limited interaction.*
- *It has often happened that days are lost to exhaust external requests*

Communication with external teams, as usual, tends to be complex and highly ineffective. There is no existing framework or methodology that connects the different teams related to the project. To worsen the situation, both teams PM and Nodo are surrounded by teams that work with a different methodology. A company strategy is needed to surpass these kind of problems.

Concerns related to the relationship with the client

What percentage of team members have expressed a concern related to these aspects with the client?

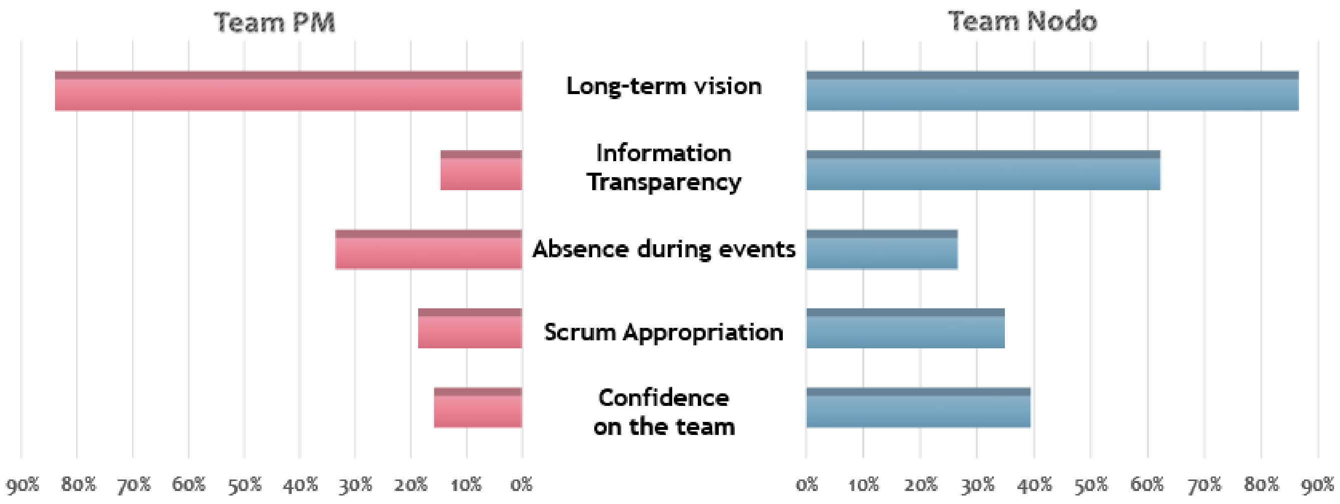


Fig 21 Trends related to concerns with the client

It is evident that long-term vision has become one of the biggest concerns related to the product owner. It seems that the focus and the destiny is far from clear for both teams. Other considerable aspects might include lack of transparency, lack of confidence on the team's capabilities and his absence during Scrum events.

*Which are the **positive** aspects of the relationship with the customer?*

- ❖ *We don't need to face the customer because we have the tools*
- ❖ *We usually get some pretty quick answers.*
- ❖ *From a technical point of view, he makes himself available to give us directions*
- ❖ *Customer participation is fair. You don't need to be with him every day.*
- ❖ *We understand the requests more specifically or we can convince the customer himself.*
- ❖ *Maybe his idea is very abstract so it could be revised and adjusted.*
- ❖ *I see no disadvantages in seeing the client within the team or having constant interactions with him.*
- ❖ *We reduce development time and we have clearer requirements*
- ❖ *The customer seems to me totally involved on the Node and on the PM.*
- ❖ *The relationship we currently have is more than enough, going further would be too much.*

Both teams used to have an intermediate figure in charge of passing information from the client to the developers. Now, this figure is no longer used and the team receives tasks directly from the customer, which has become he full-righteous product owner. This aspect has been highly appreciated by the team.

*Which are the **negative** aspects of the relationship with the customer?*

- ❖ *I don't feel the client present by my side. On several occasions I would need his help.*
- ❖ *There is difficulty in expressing the meaning of what is being exposed. It is not that we can discuss so much.*
- ❖ *The refinement backlog does not always include tasks that end up in the Sprint.*
- ❖ *Other times it is a bit binding. Maybe it's not the best thing to let the customer know.*
- ❖ *It created a bit of confusion within the group.*
- ❖ *They don't think well about the visibility of the whole of what it can impact.*

- *My colleagues cannot express themselves in the same way with me as with the client.*
- *We would like to change some methods but it cannot be done because the customer makes changes.*
- *Having to contact the client with intermediaries is not immediate and blocks our work.*
- *Greater participation of the end customer could help.*

Many team members express a concern on communication channels and inequalities created inside the group. The performance of the product owner may not be regarded as the best but it has kept afloat the project and the teams are moving onwards.

Expert's opinion:

<< A direct dialogue with the customer allowed the team to be faster in carrying out activities: in fact, previously, all communications to / from the customer passed through a" proxy "which acted as a" mediator "between the team and the client. This approach is still valid today in some situations (see the case of refinement activities). The greatest difficulty that I encounter is due to the loss of decision-making autonomy as regards the prioritization of activities and the scheduling of releases in UAT and PROD, which the customer now mainly deals with. In fact, previously, the "proxy" was in charge of negotiating these points with the customer for the benefit / benefit of the team.>>

The expert highlights the diminution of the decision-making autonomy inside the team while giving a small background explanation of how the current situation came to be.

6.2.2 Task distribution

Task wordcloud

For this cloud (Fig 22), we start seeing words such as "team", "sprint", "task", "develop", "test" and "analysis". Words such as "problems", "autonomy", "help", "estimate", "priority" and "plan" may give a rough idea of how we could build the discourse on tasks. A key word in this context is "extra" (referred to Extra Sprint), since as we will see later on, it denotes the portion of task that are not included within the Sprint planning or with the framework in general but are still necessary to have inside the team.

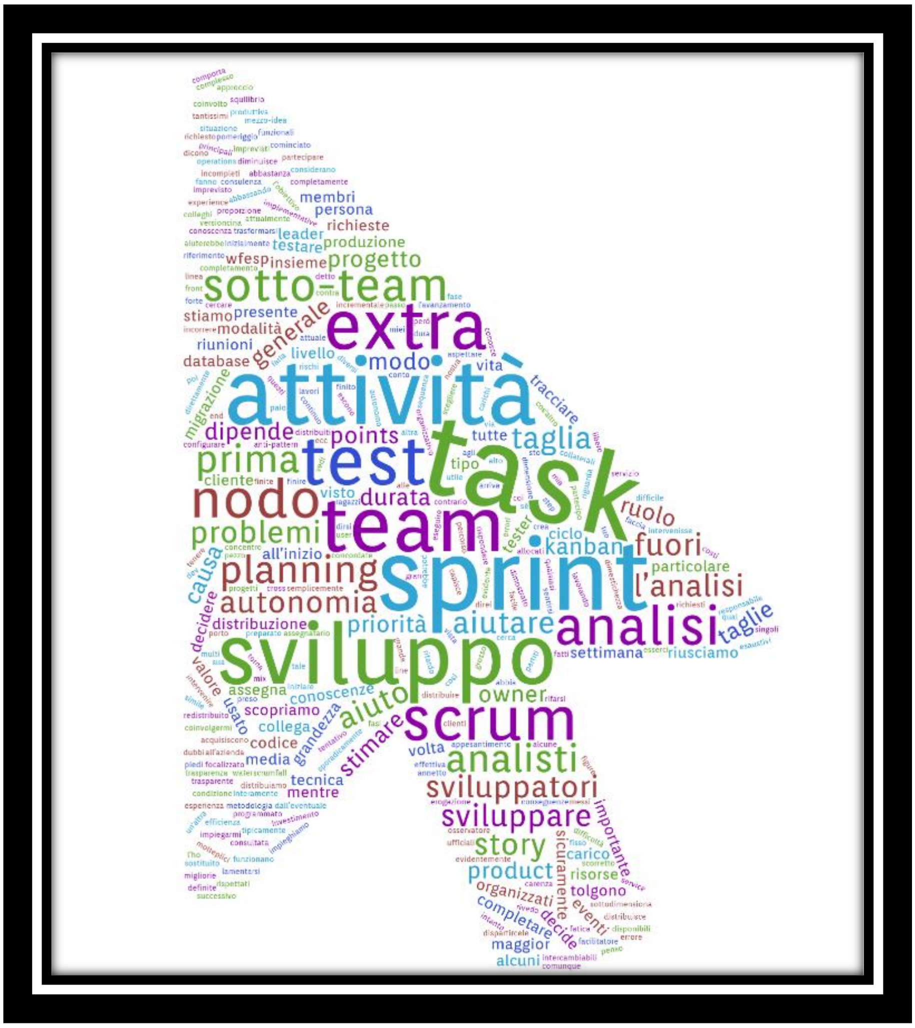


Fig 22 Task distribution word cloud in mouse pointer shape

How is the distribution of the tasks inside the team?

- *I saw too much weight. If an activity is for tomorrow, everyone is going to do it.*
- *An activity that for developers can be M, for analysts it can become L and with the test XL.*
- *If we all have to decide together we do the "poker planning" with story points.*
- *There is no particular method, whoever is free may take one of the available tasks.*
- *Often there are no tasks ready to develop because analysis or testing is still to be done.*
- *I don't know how many activities others do, this goes against transparency.*
- *Many times it is not easy to give a size to tasks and many times we also tend to make mistakes.*
- *It is not clear who can decide whether a task should be performed or not.*
- *It is not possible to estimate well if the analysis must be redone, especially if the task was not prepared in the backlog.*
- *Some Sprints have a lot of tasks for analysts, and others for developers. There is an imbalance.*

The team seems to perceive some disorganization for creating and assigning tasks. Roles are not always clear and team members tend to decide who should do what simply by good will. Not every person is equally charged on work, creating some task imbalance between team members.

Tasks times can be also quite problematic and hard to estimate. Depending on their role, team members may assign a different task time for any given activity. Fortunately, some initiatives such as "poker planning" can be helpful in determining a good approximation for these tasks.

Expert's opinion:

<<The team is divided into two sub-teams: development and testing. The development sub-team autonomously distributes the tasks by referring to its "team leader"; moreover, it does not transparently communicate the progress of tasks, only the completion. When the development sub-team finishes the tasks it "assigns" them to the test sub-team, this results in a Scrum anti-pattern which is called "WaterScrumFall". Due to this approach, in all sprints there are at least a couple of incomplete tasks because "the test sub-team did not have time to complete them". In general, the team members were not able to estimate properly the duration of their team's tasks.>>

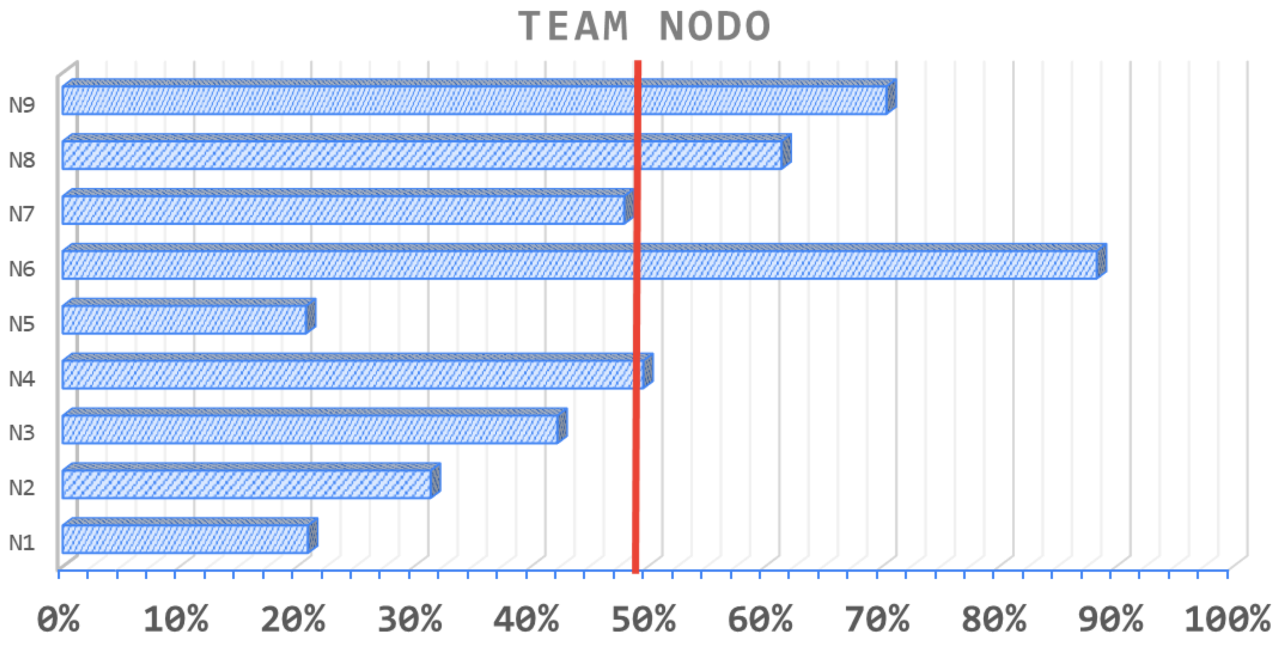
The Scrum Expert refers on how, even applying the Scrum framework within the teams, the team still relies heavily on Waterfall practices for their day-to day work and interactions. One of these cases is the usage of "team leaders" which act as a mediator between the client and the employees, leaving aside one of the main purposes of the Scrum framework.

Percentage of time in Extra Sprint activity

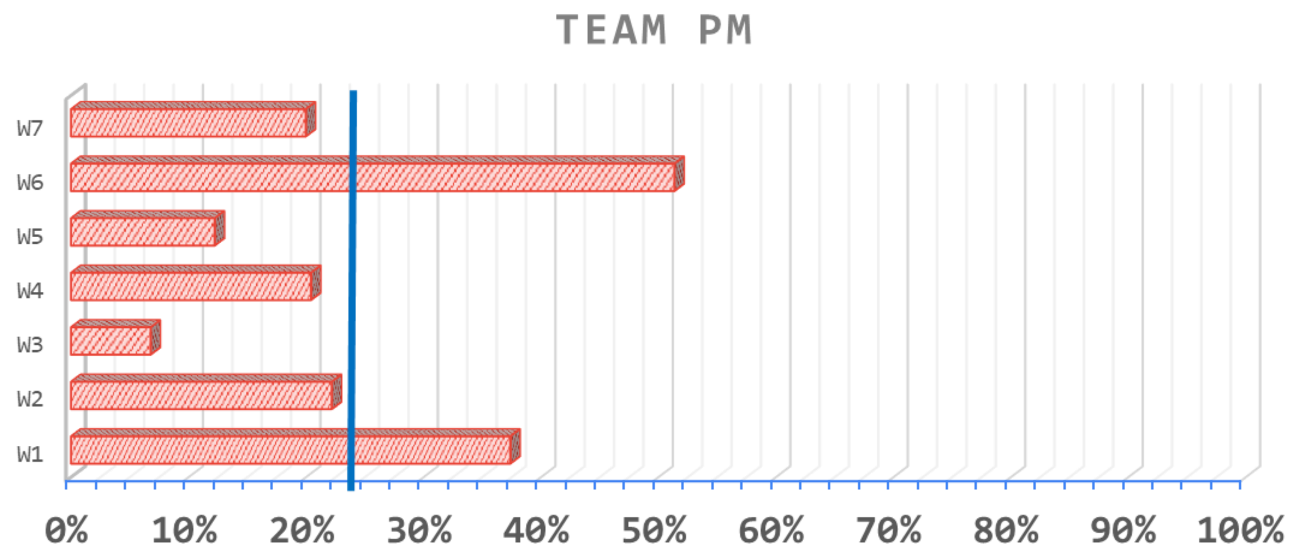
As mentioned before, there are certain activities or task that should be done by the team, but do not enter into any of the Sprint planning processes for either of the teams. These activities are mainly related to the change management processes. Here, the team should resolve information request or incidents reported by the users, the customer or other teams in the company.

Team members estimated the percentage of their weekly time they spend on activities outside the Sprint . Team Nodo members (Graph 5) on average are working more in Extra Sprint activities (48%) while Team PM (Graph 6) has a smaller percentage (24%); thus meaning they have been more dedicated to activities inside the Sprint.

This difference is an interesting sign of showing which team is closer to a full implementation of the Scrum framework. We can notice in this case that some team members even spend more than 50% of their time doing Extra Sprint activities, especially in Team Nodo. It is a clear symptom of a high amount of unframeworked tasks, which could lead the team towards unexpected consequences.



Graph 5 Percentage of time used in Extra Sprint activities - Team Nodo



Graph 6 Percentage of time used in Extra Sprint activities - Team PM

How do you feel your team is coping with Extra Sprint Activities?

- *Any problems in production take me away from the Sprint.*
- *We tried to do a similar Kanban for Extra Sprint activities but it was a bit confusing.*
- *Initially I was able to dedicate 50% - 60% of my time to Sprints. Now I don't dedicate more than 10%.*
- *We are not organized on a project or service basis.*
- *You have a hard time saying how many tasks go into the Sprint.*
- *To keep ourselves busy we work on tasks outside the Sprint.*
- *Extra Sprint tasks are needed but are not directly required; it is more like an investment.*
- *The multiple commitments of individual team members prevent them from thinking thoroughly.*
- *Maybe we end up with all the tasks done, so in the meantime we decide to do tasks in advance.*
- *It is complex to track the true effective capacity of the team.*

As described by team members, a reason for having high Extra Sprint time percentage is due to the change in project dedication for each team member. This means that a certain person might have been quite useful for the Sprint tasks at a given period, but now that contribution has diminished due to the project needs. This causes the team member to start performing external activities, but still engaging into the Scrum events such as Daily meeting and Sprint planning.

This kind of effects are not easy to foreseen within the Scrum framework since they tend to appear in the middle and long run. Some team members may remain in a middle ground where they are less frequently required for Sprint tasks, but still required for some of them. The Product Owner should promptly address this stealthy problem. He should tell whether a resource is required in the project or not.

The biggest issue with Extra Sprint activities, after all, is the fact that they are team activities that do not have a defined framework or methodology. This creates an internal team incongruence where it is not clear for the team how to distribute their time between Sprint and Extra Sprint tasks. As backfire, this also generates trouble in assessing the team's overall capacity and estimate team task times in general.

6.2.3 Improving process

Improve process wordcloud

This wordcloud is unique with respect to others (Fig 23). While other wordclouds tend to solely gather vocabulary related to their topic, this one encompasses vocabulary from all of the areas at ones. Some of the main word seen are "team", "task", "client", "members", "test" and "improve". Interesting words seen in this case are "knowledge", "competence", "functionality", "fast", "plan", "missing", "division" and "workload". It is worth noting that also certain Scrum words such as backlog and refinement also made the cut into this assembly.

However, it is important to distinguish how big certain words appear, since it denotes its relative frequency with respect to other words in the same discourse. For example, "test" and "develop" seem to come more often than "analysis" or "plan". Team members used the word "improve" not so frequently and a t the same time "problems" is used even less frequent. These examples might give us a rough perception of the team's issues and the need to improve them.

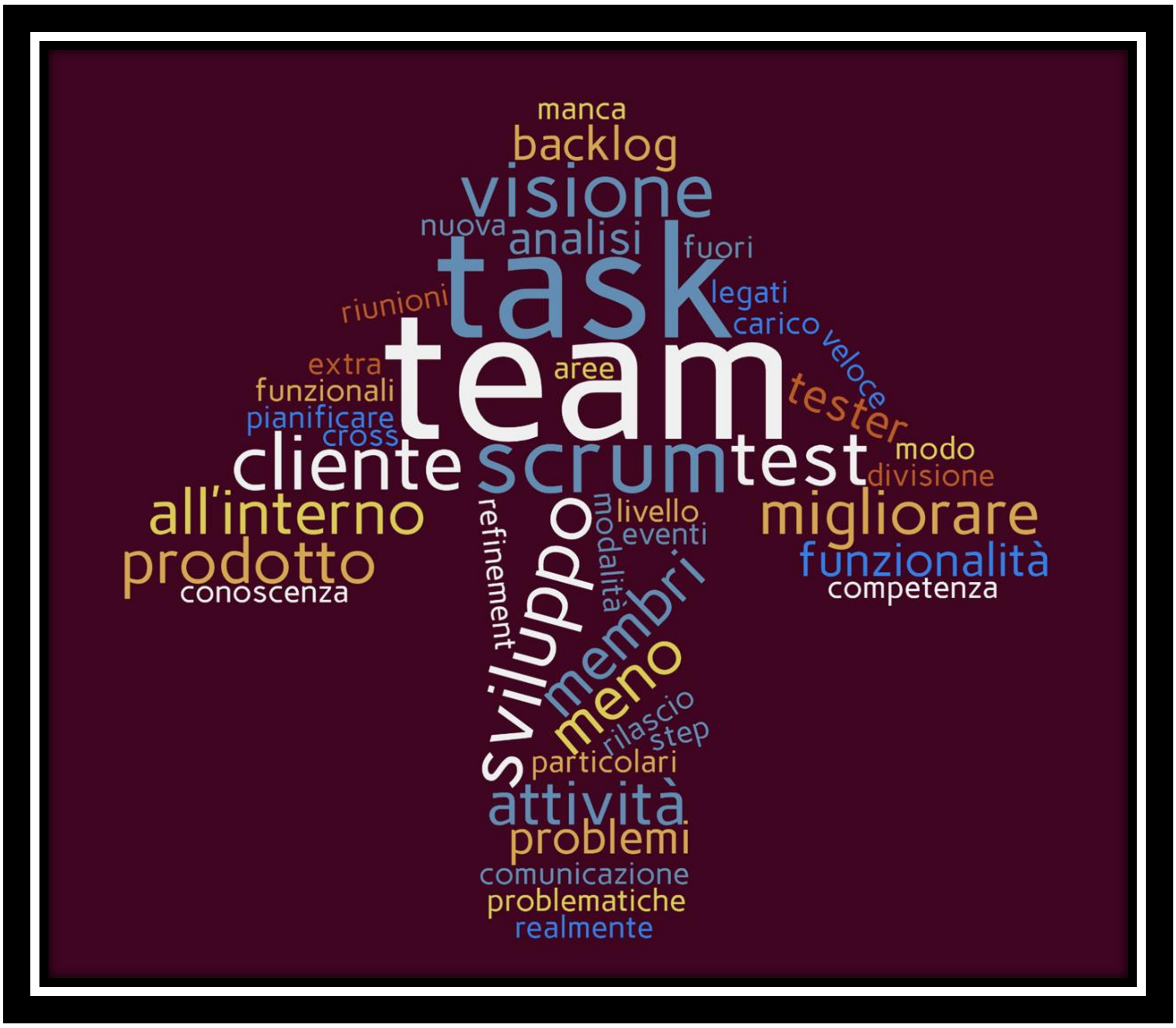


Fig 23 Improving process word cloud in arrow shape

Which are some key aspects at work?

- *Each one of us gets a role by having authority and being authoritative. With Scrum, we are all equal.*

- *In our development environment, we are quite free to do what we want.*
- *We always manage to be on time and we are quite responsive.*
- *I have to pretend to work 100% because I am being asked to do it but I cannot.*
- *The leader of the development team acts as the mediator between the team and the client himself.*
- *We make all together the decisions along with customer.*
- *The team should be cross-functional but it is actually mono functional. That does not help Scrum.*
- *A better overview of the whole product is important.*
- *The team members will obtain expertise from their previous knowledge as the properly apply it to any task given.*
- *You should always try to carry everything to a consolidated praxis.*

Team members may have very particular and detailed perspective from their experience inside the team. Some perceive collaboration in decision-making and freedom to choose what to do. Others feel more constraint on the group's composition or in how they should act in certain situations. In any case, they are all fully aware of the framework mechanics to are used or that should be used in order to achieve a higher Scrum implementation.

Expert's opinion:

<<The Scrum Master within the team has the task of coaching following the three pillars. Individuals must follow the five values with a good level of mastery: focus, openness, respect, courage and commitment.

Team members should all participate actively, for this it would be necessary to: avoid making 1-to-1 interactions between individual members, involve all team members in refinement activities, avoid using a "team leader", track the individual tasks performed by team members in the sprint backlog and be transparent in assigning tasks.>>

The Scrum expert reminds us about the three Scrum pillars and the five Scrum values that each individual should follow. At the same time, he gives some tips to increase the active participation within the team.

Which problems have you seen in our team's work methodology?

- *Often well-defined tasks do not arrive. There is little backlog refinement.*
- *The concept of capacity is lacking and, in theory, teams should be segregated.*
- *We have currently been underpowered for the test part. We are not 100% dedicated.*
- *Not everyone working on the project has access to the same tools.*
- *There is no air of will to make the project go forward. We do not have people dedicated to the Sprint.*
- *Extra Sprint activities take up your time and do not allow you to plan well within Sprints.*
- *From my point of view, everything is working fine. No points for improvement in this team.*
- *Everything has a specific deadline. We are tied to a seasonality.*
- *I cannot think of particular things to improve further.*
- *Missing more people dedicated only to us. We are also not 100% convinced of the framework.*

Problem perception within the team is quite divided and diverse within the team. Some people are concerned about the tasks given which might not be well defined, or may be lacking some test. Others might associate issues perceived with a lack of commitment from the team members, willingly or unwillingly. As mentioned in the task distribution part, some team members criticize how Extra Sprint activities are a great challenge to handle for the team. Oddly enough, some members report no issues within the team while at the same time not finding any need for a possible improvement.

How can we improve?

- *A longer analysis phase could help, or better contextualize.*

- *If there were no Extra Sprint activities but only Sprint tasks, this would be much better.*
- *On the structure side, I feel that I miss the analysis so much.*
- *A Scrum expert may come to point out where we are wrong.*
- *There is no board where we can put all our questions.*
- *We should divide better the percentage of load on each project.*
- *It would be nice to have a debate step immediately after the planning and an analysis step to know what it should include.*
- *Documentation needs to be improved. Make a document in which everything is enclosed and explained in detail.*
- *We should improve the scheduling of events and avoid skipping them.*
- *To improve our team we should have cross-functional, collocated, committed and involved members.*

An important improvement, according to the team, would be on the analysis side of the task, which they perceive to be lacking. Workload, task distribution, Scrum Events and Extra Sprint activities are frequent topics within the team. Curiously enough, some team members mention possible changes from other frameworks, such as using a board, which comes from Kanban, or heavily documenting, which is mainly used in Waterfall methodologies but avoided in Agile ones.

6.3 SMARTWORKING

In February 2020, when we originally conceived this project, we would have never guest a global pandemic would completely disrupt life as we know it. In any case, we did this entire Survey by using our Smartworking tools, the ones the company had recently implemented. Since this could be a one-time opportunity, we took the advantage to our side and added questions related to the recent transformation in our working environment and its effects on the implementation of the Scrum framework. Here we collected the team's perception of Smartworking for the company.

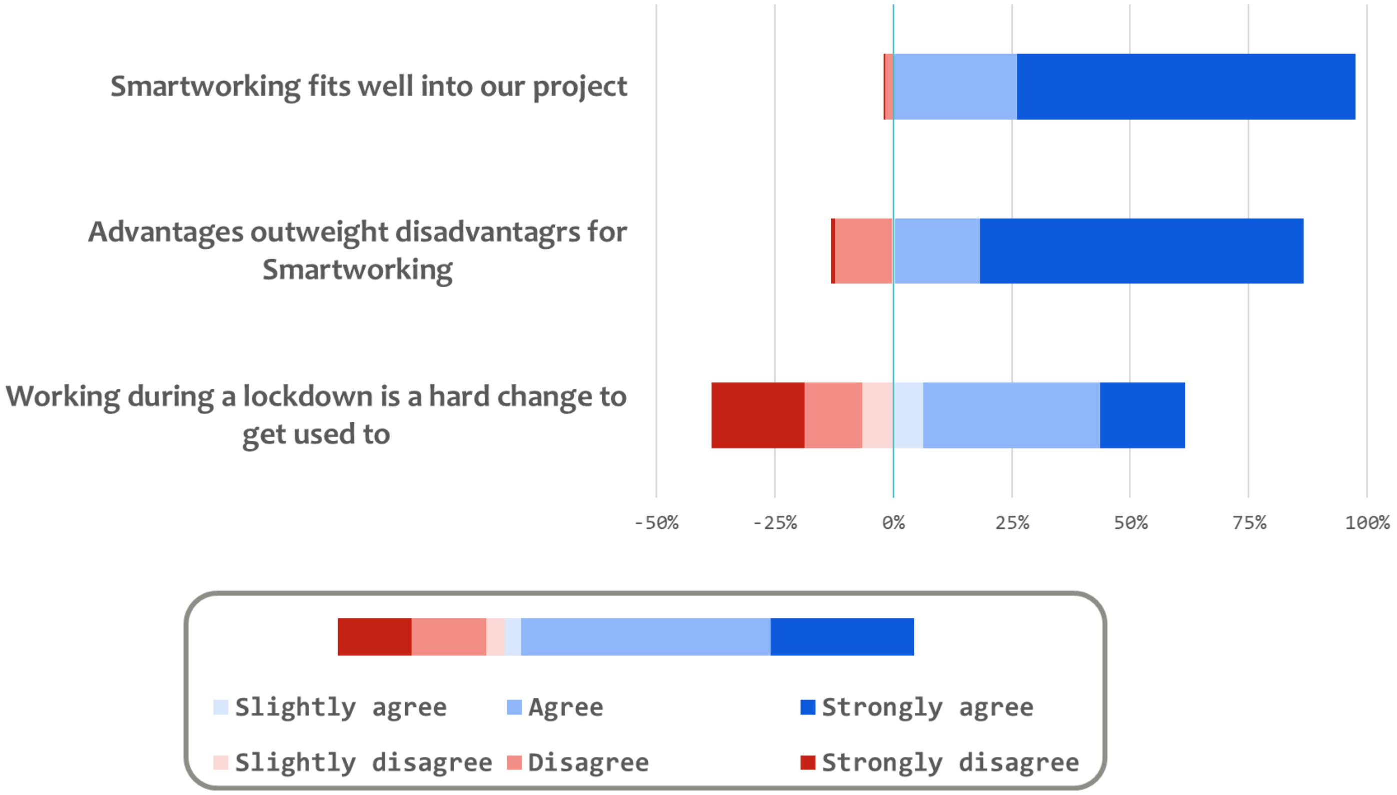
How do you feel Smartworking influences our work?

- *When we are in person, we talk all day and it seems like doing a continuous daily. We are now more enticed to keep dailies than we did before.*
- *Smartworking makes it easier to maintain Scrum events.*
- *I did not feel any difference because we all continued to work together.*
- *I have not noticed this difference between office work and Smartworking.*
- *Things go on anyway.*
- *In my opinion, it is very manageable from home.*
- *With technology, even being distant, we can all be together.*
- *With the client in Rome, we already had this mechanism. Perhaps from the customer's point of view the situation feels now leveled.*
- *It is a very bold business choice to keep everyone in the house.*
- *We are lucky because we can do our work remotely.*

Overall, team members have perceived this change quite positively. Being a company that focuses on software services, Smartworking has shown to be a good fit. The main comments talk about how work is quite manageable from home and how the project has not been interrupted at all. Some teammates highlight the fact that Scrum events rigor is easier to keep in Smartworking in comparison to its presential counterpart.

6.3.1 Pros and cons

Trends related to Smartworking



Graph 7 Smartworking trend bars

Speaking of Smartworking trends, virtually everyone agrees with its implementation on our company (Graph 7). For sure, there are still some disadvantages but they are still outnumbered by the advantages in most part. The only point with which the team members disagree is about the perception of change of working during a global pandemic, some perceive it as a big change while others do not perceive any significant change to their working habits.

Smartworking wordcloud

Smartworking talk brings a new variety of words into play (Fig 24). Some of the most used ones are "house", "work", "office", "colleagues", and "manage". Certain curious words we could include are "feel", "everyone", "contact", "together", "life" and "never". Have you noticed the house shape of the wordcloud?

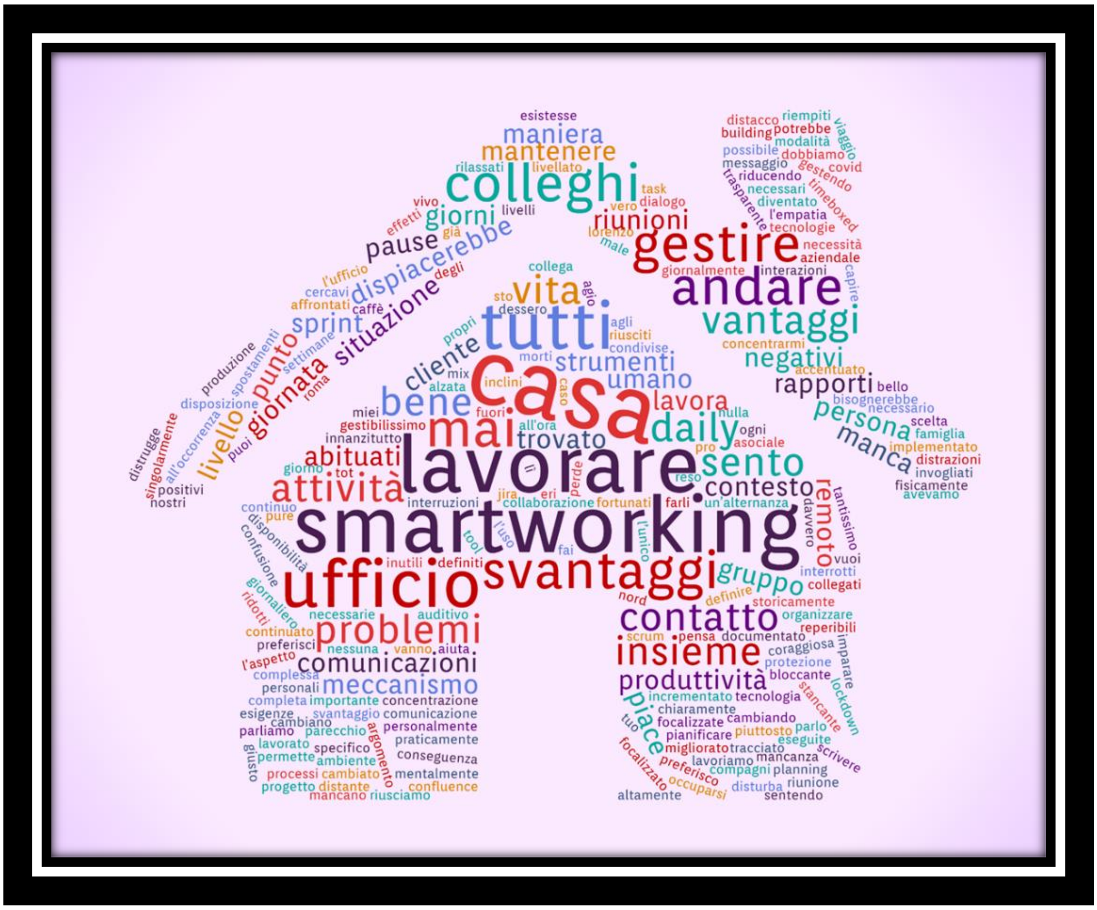


Fig 24 Smartworking wordcloud in house shape

Which advantages do you perceive for Smartworking?

- *I have not seen that much difference. Instead, I saw more availability.*
- *Interactions are now faster and less time is wasted on meetings too.*
- *There are no disadvantages but you should know how to manage well your time.*
- *We use more frequently collaboration tools like Jira and Confluence.*
- *We can document the work done and make it more transparent to others.*
- *I find this as an opportunity to organize better my day.*
- *I can have more privacy and avoid commuting.*
- *We are all more relaxed. You work at your ease. It works even better.*
- *It helps you to concentrate more because there are no distractions. Fewer interruptions.*
- *We can make a better use of time while we get rid of "dead" moments filled with useless activities.*

Team members notice a significant reduction in wasted time, either for unnecessary meetings, commuting, and dead moments throughout the day. In general, they experience more autonomy to handle their day by day, which is highly appreciated.

Which disadvantages have you experience until now?

- *Human contact is missing. I miss it either visually or audibly.*
- *In Smartworking, you are always present; it feels harder to take breaks.*
- *In terms of productivity, I feel we are performing equally but it is a bit tiring.*
- *We work even more than before.*

- *We never take breaks and we need to learn to manage our time better.*
- *Personal relationships are missing. It is harder for you to plug yourself out.*
- *I do not think there are any drawbacks.*
- *The only drawback is the social aspect because we never see colleagues.*
- *Remote working has accentuated the detachment between colleagues by reducing empathy.*
- *I find myself in a highly complex situation because the processes are not well defined.*

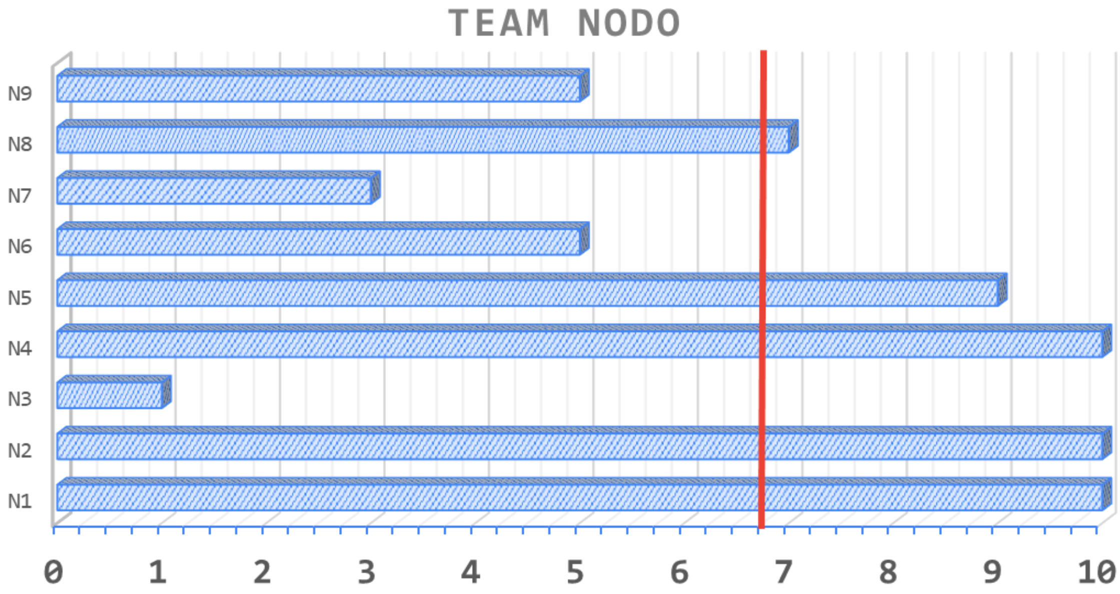
One of the biggest disadvantage for Smartworking is, of course, the lack of human contact and the reduction of interactions, kept to their basics. Someone has even argued it can reduce empathy and create detachment among colleagues.

It is natural that team members experience confusion, especially during these times. At the same time, some members think it could lead to work overload. That is why limits, both company wise and personal wise, should be establish promptly to avoid negative situations.

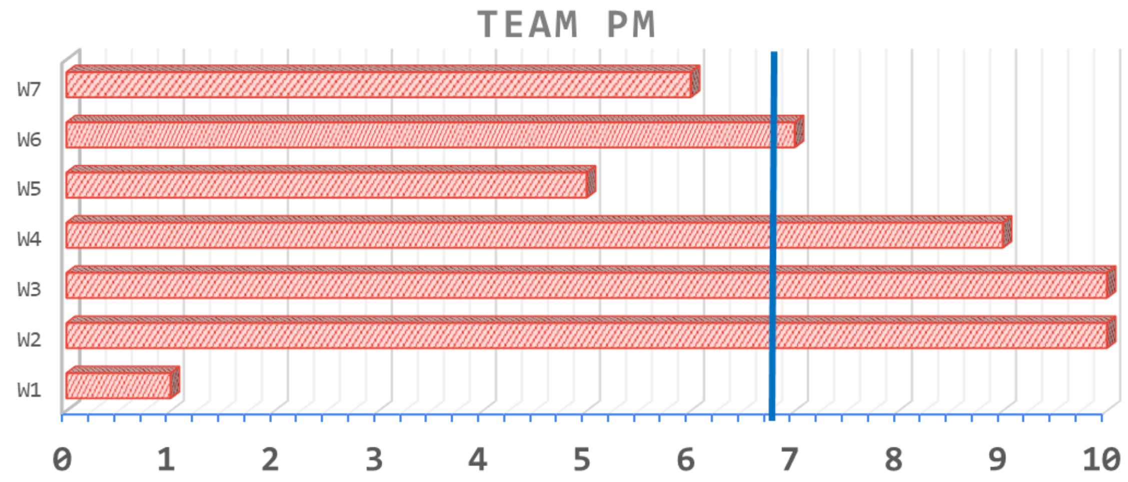
6.3.2 Smartworking balance

Each team member shared the amount of days they would like to do in Smartworking mode from 10 working days. This is showed in Graph 8 and Graph 9.

The majority in both teams would prefer to work multiple days from home. In this case, there was little difference between the averages of the teams, thus obtaining a global average of 6.8 days out of 10 in Smartworking.



Graph 8 Amount of desired days to do Smartworking - Team Nodo



Graph 9 Amount of desired days to do Smartworking - Team PM

Which would be the best Smartworking balance for you?

- *I would not mind a mixed mode. Maybe 1 day at home.*
- *I would rather stay at home all the time; I have always been a bit unsociable.*
- *If there were the possibility of doing Smartworking for life, I would do it.*
- *There are those who prefer the office, but I prefer working from home.*
- *If they gave me the chance to be always at home, I would do it.*
- *For me, if I come to the office once every two weeks it would be fine.*
- *The work is quite similar and you could alternate between home and office, half-and-half.*
- *If, on the other hand, I had to do a certain number of days in the office and others at home, I would prefer to stay in Milan.*
- *If I had my whole family in Milan, I would not mind going to the office.*
- *Working from home is very comfortable. I would do it always.*

Overall, most employees would prefer to enjoy the Smartworking benefits for the long run. Some argue a mixed mode might be a solution if they were in Milan, although it is quite common to find colleagues that work from Sicily, Puglia or Campania or even South America. Therefore, for them I would be easier if the company operations stayed always on remote mode.

Has the pandemic open our eyes towards a new and better form of working for life? It could be the case, but this and so many other topics will be discussed in the next and last chapter.

7 DISCUSSION AND CONCLUSIONS

7.1 RESEARCH IMPACT ON THE SCRUM FRAMEWORK

Studies as the one we are holding might not have been 9 months intense as in Moe et al, 2010. Anyhow, open surveys that follow a similar method can yield great and extensive results with huge similarities within vertical teams of diverse range of companies. After all, Scrum is not very specific on how to establish monitoring in development teams, although this is implicitly a prerequisite for feedback, coordination, and backup (Moe, Torgeir, & Tore, 2010).

Analyzing the teamwork framework (Dickinson & McIntyre, 1997) within both Team Nodo and Team Wallet, it was evident that teamwork as a general concept might be lacking from some strong team orientation and leadership, which in this case should come directly from the Product Owner at PagoPA. This lack in the input of the learning loop may lead to interesting consequences where monitoring and feedback are truly affected and may not be resilient into both teams.

Studies that analyzed the Scrum framework in comparison with other methodologies are usually regarded as compatible or adaptable in their principles. Scrum can be considered alongside similar Agile methodologies such as Kanban (Lei, 2015). It could also be adapted into methodologies that follow apparently opposite principles, such as COBIT or Waterfall (Ozkan, 2015). It is more a matter of understanding how to adapt these principles for each particular project, following the requirements of each team.

7.2 SCRUM ADAPTATION FOR THE TEAM

The situation is clear; the two teams are working exactly as they have always worked. In practice, rather than adjusting their principles and practices to using the Scrum framework tools, both teams adapted the Scrum tools to the principles and practices they had before. To do this, some tools have been "removed". For example, the retrospective is no longer held. Others have been adapted, for example, the sprint backlog is simply the list of activities to be done in the sprint or the backlog refinement is simply the call that was previously made between the client and the analyst to detail the requirements.

The only Scrum ceremony that remains unchanged for the team members is the Daily meeting as it updates and improves communication between team members. This is a similar result obtained by other authors while performing similar research processes in other Scrum teams (Cho, 2008).

Now we could say that both teams are adopting their own "process" with their own "principles" and that they are using some of the "tools" made available by the Scrum framework.

The Scrum framework was never really adopted because considering the three elements:

principles -> practices -> tools.

The process of "Agile transformation" has never really been started, which first of all concerns the principles, which are those of the Agile Manifesto, but only the tools have been introduced (JIRA for the management of the backlog and sprints and the "ceremonies") and reference was given to what practices were to be adopted (Scrum and CI / CD).

7.3 SCOPE OF THE PROJECT: FAILURE OR SUCCESS?

There was no real intent to adopt Scrum because obviously it is a "strategic" type activity that must be properly conceived, approved, planned and then executed as a real project.

A project whose scope is "to adopt Agile principles, practices and tools" has never been conceived, started or executed. So in reality it cannot be said whether we failed or not because it was never defined what the goal was.

• What was the "future state" to reach?

• What was the "current state" to start from?

• What was the strategy for "transitioning" from the "current state" to the "future state"?

This may have been said verbally or is "implied" but it is not clear that there is a clear statement of purpose.

The change in principles and scope must be the starting point for seeking a new methodology. This is what the theory says. There are teams that deal with an "Agile transformation". As a guide we can refer to the IT strategy books.

7.4 Working Agreement and Best Methodology

The process change was based on a "working agreement" document between the customer and the development team as a supplier. If you consider that the use of Scrum ultimately derives from a "contract" with the customer, in the end it can be said that the goal has been achieved. Without prejudice to the content of the contract, we moved on to "negotiate" with the customer how to apply the content of the contract.

The result is what can be observed today for the two teams: it is not Scrum, it is not Agile but it is the application of a software development process to the best of the current negotiation between the team itself and the customer. From this point of view it can be said that both teams use the best methodology taking into account the objectives of the client and the team and how the negotiation has been carried out to date.

We don't know for sure if the customer is satisfied with the current methodology. The objectives of the client or the team manager are also not clear. However, the current methodology obtained after about 12 months of negotiation which, at least on a superficial level, seems to be satisfactory both for the client and for the team, similar consequence found to happen also in the study of multiple Product Owners (Sverrisdottir H.S., 2014). In this situation, what would be the most appropriate way to give recommendations or to suggest which is the best methodology? In the end, there is this underlying situation that really prevents us from judging the methodology.

7.5 Repercussions on the Dev Team

The decisions that the client and the manager make have various repercussions and effects on the daily activity within the development team. We can say that these effects were mainly two:

First, the teams had to integrate into their process, which was initially totally "custom", elements that are instead "standard" (because they belong to Scrum), such as: the sprint as a recurring event of a fixed duration, the User Stories, the Backlog (product and sprint), the ceremonies (planning, daily and review). Obviously, only the most "familiar" and close to "practice" of the standard was considered, which was obviously easier to "understand" and "apply", while everything that was distant from "consolidated practices" was ignored / removed. or in any case from your "comfort zone". The "familiar" elements of Scrum have been somewhat adapted to the existing way of working but, ultimately, their adoption has in all cases "forced" us to make some minor adjustments to the as-is. This made it possible to make the process less "custom" and therefore more easily recognizable, manageable and, ultimately, more "predictable".

Second, the customer, having assumed the role of PO, necessarily had to change his "attitude" towards the team / supplier and the same applies to the team / supplier towards the PO / customer. Here the same argument applies as before: there has not been a radical "transformation", but undoubtedly the behaviors have changed / evolved as a result of the different roles that the customer / PO and the team / supplier had to assume. For example: now the customer / PO directly follows the development / evolution of the system over time, communicates directly with the development team and therefore with the people who carry out the activities, has the opportunity to evaluate at the end of the sprint what has actually been achieved . Likewise, the team / supplier has a better command of the development process which allows him to be autonomous but at the same time confident that what he is developing is in line with the customer's requests. In addition, there is a slight improvement in the level of "engagement" of the team itself, as now more team members can talk directly with the customer without "intermediaries".

7.6 Smartworking during Covid-19

When this project started, no one would have ever expected a global pandemic to disrupt life, as we knew it. However, this was the reality that we came to find in 2020 and onwards. It is true that each sector was stroked differently, and the digital business sector was actually one of the "lucky ones". The digital sector not only manage to stay afloat but also to keep growing during this harsh year that inevitably turned into a global recession.

During this context, business practices changed dramatically, Smartworking became a *de-facto* way of work and existing working methodologies had to adapt to this new reality. Some authors argued about the fatigue caused by the constant video conference environment and the increase of distractions as a compelling factor to work (Griffin, 2021). They also argued about how this weakened Scrum fundamentals, fortunately, adaptation was still quite viable and presented itself as a key solution to the challenges for the next years.

As shown by the Scrum Survey, Smartworking turned out to be quite convenient and beneficial for the team. Not only it allowed a greater sense of autonomy over working time, but it also cut wasted time during idle times; thus creating higher efficiency. Of course, the human factor was one of the biggest drawbacks of Smartworking, nonetheless, it because a well-accepted way of working. Currently, it might not be only a temporal way of working, but it will surely lead the basis for the next generation working in the world.

BIBLIOGRAPHY

Agenzia per l'Italia Digitale. (2020). *Specifiche Attuative del Nodo dei Pagamenti-SPC versione 2*.

Agenzia per l'Italia Digitale. (2021). *Piano Triennale 2020-2022 - Promuove la trasformazione digitale del Paese e della PA*. Retrieved from https://www.agid.gov.it/it/agenzia/piano-triennale

Atkinson, R. (1999). Project management: Cost, time, and quality, two best guesses and a phenomenon, it's time to accept other success criteria. *International Journal of Project Management, 17(6), 337-342.*

Awad, M. A. (2005). A comparison between agile and traditional software development methodologies. *University of Western Australia, 30.*

Beck, K. B. (2001). Manifesto for agile software development.

Behutiye, W. &.-F. (2020). *Management of quality requirements in agile and rapid software development: A systematic mapping study.* Retrieved from https://doi.org/10.1016/j.infsof.2019.106225

Blanchard, B. S., & Fabrycky, W. J. (2006). Systems engineering and analysis (4th ed). *New Jersey: Prentice Hall. p.31*.

Browning, L., & Latoza, R. (2005). "The use of narrative to understand and respond to complexity: A comparative analysis of the Cynefin and Weickian models. *Emergence: Complexity and Organization, 7(3–4): 32–39*.

Cho, J. (2008). Issues and Challenges of agile software development with SCRUM. *Issues in Information Systems, 9(2), 188-195.*

Cohn, M. (2010). Succeeding with Agile: Software Development Using Scrum. *Upper Saddle River, NJ: Addison-Wesley*.

Crisciullo, C. (2020, February). PagoPA System Overview. Milano: SIA S.p.A.

Cummings, H. (2006). Management Information Systems for the Information Age. *McGraw-Hill Ryerson*.

Dickinson, T. L., & McIntyre, R. M. (1997). A conceptual framework of teamwork measurement. *Team performance Assessment and Measurement: Theory, Methods, and Applications; Psychology Press, NIJ*, 19-43.

Globerson, S. &. (2002). The Impact of the Project Manager on Project Management Planning Processes. *Project Management Journal, 33(3)*, 58-64.

Henrie, M. &.-P. (2005). Project Management: A Cultural Literary Review. *Project Management Journal, 36(2), 5–14.*

Highsmith, J., & Cockburn, A. (2001). Agile software development: the business of innovation. *Computer, vol. 34, no. 9, pp. 120-127*, doi: 10.1109/2.947100.

Katzenbach, J. R., & Smith, D. K. (1993). The discipline of teams,. *Harvard Business Review 71 (2) 111–120.*

Keogh, L. (2011). Scrum and Kanban: both the same, only different. *https://lizkeogh.com/*.

Lei, H. &. (2015). A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing. 43.* doi:10.1016/j.rcim.2015.12.001

Levesque, L. L., Wilson, J. M., & Wholey, D. R. (2001). Cognitive divergence and shared mental models in software development project teams. *Journal of Organizational Behavior 22 135–144.*

Moe, N. B., Torgeir, D., & Tore, D. (2010). A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology, Volume 52, Issue 5*, 480-491. doi:10.1016/j.infsof.2009.11.004.

Nerur, S., & Balijepally, V. (2007). Theoretical reflections on agile development methodologies – the traditional goal of optimization and control is making way for learning and innovation. *Communications of the ACM 50 (3)*, 79- 83.

Ozkan, N. (2015). Risks, Challenges and Issues in a Possible Scrum and COBIT Marriage. doi:10.1109/APSEC.2015.29.

PagoPA S.p.A. (2020). *Cos'è pagoPA*. Retrieved from https://www.pagopa.gov.it/it/pagopa/

Parnas, D. (2006). Agile methods and GSD: The wrong solution to an old but real problem. *Communication of the ACM 49(10), 29*.

Partogi, J. (2013). Certified Scrum Master vs Professional Scrum Master. *Lean Agile Institute*.

Perkusich, M. &. (2019). Intelligent Software Engineering in the Context of Agile Software Development: a Systematic Literature Review. *Information and Software Technology*. doi:10.1016/j.infsof.2019.106241.

Project Management Institute. (2003). A Guide to the Project Management Body of Knowledge. *Project Management Proccesses for a Project*, 47.

Rola, P., Kuchta, D., & Kopczyk, D. (2016). Conceptual model of working space for Agile (Scrum) project team. *Journal of Systems and Software, Volume 118, ISSN 0164-1212*, 49-63. doi:10.1016/j.jss.2016.04.071

Rousseau, V., Aube, C., & Savoie, A. (2006). Teamwork behaviours - a review and an integration of frameworks. *Small Group Research 37 (5)540–570*.

Schwaber, K., & Beedle, M. (2002). Agile software development with Scrum. *Prentice Hall. ISBN 978-0-13-067634-4.*

Schwaber, K., & Sutherland, J. (2019). The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. *retrieved May 13, 2020*.

Skarin, M., & Kniberg, H. (2009). Kanban and Scrum - Making the Most of Both. *QCon Plus (May 17-28): Stay Ahead of Emerging Software Trends*.

Sverrisdottir H.S., H. T. (2014). The Role of the Product Owner in Scrum-comparison between Theory and Practices. *Procedia - Social and Behavioral Sciences, Volume 119, ISSN 1877-0428*, 257-267. doi:10.1016/j.sbspro.2014.03.030

Toivonen, E. (2020). 5 Problems With Scrum You Need to Avoid. *Thinking Portfolio OY*. Retrieved from https://thinkingportfolio.com/5-problems-with-scrum-you-need-to-avoid/

Unhelkar, B. (2016). The Art of Agile Practice: A Composite Approach for Projects and Organizations. *CRC Press. pp. 56–59. ISBN 9781439851197*.

Vlietland J., H. v. (2015). Towards a governance framework for chains of Scrum teams. *Information and Software Technology, Volume 57, ISSN 0950-5849*, 52-65. doi:10.1016/j.infsof.2014.08.008