# MASARYK UNIVERSITY Faculty of Informatics



# Adapting Agile for Research Laboratory

Master's Thesis

Bc. Filip Svoboda

Brno, Spring 2021

# MASARYK UNIVERSITY Faculty of Informatics



# Adapting Agile for Research Laboratory

Master's Thesis

Bc. Filip Svoboda

Brno, Spring 2021

## Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Bc. Filip Svoboda

Advisor: Ing. Leonard Walletzký, Ph.D.

## Acknowledgements

I would like to thank my advisor, Ing. Leonard Walletzký, Ph.D., for his continuous support during my internship at the Laboratory of Service Science and the elaboration of this thesis. I would also like to thank the whole team therein, who have supported the implementation of my proposed agile process.

### Abstract

Application of agile frameworks outside of the Software Engineering industry, where they are now a proven standard, is getting growing traction. Based on study of Agile in general and an in-depth analysis of six relevant case studies, we propose a lightweight agile process applicable in research groups in academia. We then present a case study of the Laboratory of Service Systems at Faculty of Informatics, Masaryk University, where we have successfully implemented it to significantly improve quality of advisory meetings (p=0.011), motivation of students (p=0.033), affinity to actively represent the laboratory (p=0.007), and overall satisfaction with membership (p=0.020), as indicated by anonymous attitude surveys of laboratory members.

# Keywords

agile, scrum, kanban, research, education, collaboration, software engineering, project management

# Contents

Introduction 1							
1	1 Agile in Software Engineering						
	1.1	Software Engineering	3				
		1.1.1 Predictive Approach	6				
	1.2	Agile Overview	7				
		1.2.1 The Agile Manifesto	9				
		1.2.2 Comparison of Predictive and Agile Approaches	10				
		1.2.3 Agile Frameworks	10				
	1.3	Scrum	12				
	1.4	Kanban	15				
	1.5	Extreme Programming	17				
	1.6	Agile Breaks Free of Software Engineering	17				
2	Agil	e in Research	21				
	2.1	Academic Research Context	23				
		2.1.1 Research	23				
		2.1.2 Research Teams	26				
		2.1.3 Challenges of Team Research	27				
	2.2	Case Studies	30				
		2.2.1 Maryland's SCORE Framework	31				
		2.2.2 txtUML's Agile Methodology	34				
		2.2.3 CECAN's Distributed Scrum	37				
		2.2.4 PROMOBILE's Full Scrum	41				
		2.2.5 SLIIT's Undergraduate Research Projects	44				
	• •	2.2.6 Kaiserslautern's XP Seminar	47				
	2.3	Summary of Case Studies	49				
3	Case Study: Agile in Laboratory of Service Science 53						
	3.1	Background of the Laboratory	54				
		3.1.1 Demography	55				
		3.1.2 Recent History	57				
		3.1.3 Motivations for Change	59				
	3.2	Methods	62				
	3.3	Agile Transformation	64				
		3.3.1 Design Considerations	64				

	222	Agila Process Proposal	65						
	5.5.2	Agne Flocess Floposal	05						
	3.3.3	Adoption	68						
3.4	Result	s	69						
	3.4.1	Student Survey Design	71						
	3.4.2	Thesis Sentiment	72						
	3.4.3	Laboratory Sentiment	74						
	3.4.4	Optimal Meeting Frequency	75						
	3.4.5	Engagement	77						
	3.4.6	Summary of Results	78						
3.5	Discus	sion	79						
Conclusion									
Ribliography									
Dibilography									

### **List of Tables**

- 1.1 Comparison of Predictive and Agile Software Development 11
- 2.1 Soft-Skill Improvements at PROMOBILE 43
- 2.2 Student Opinions at SLIIT 46
- 2.3 Summary of Agile in Research Case Studies 51
- 3.1 Tools Used in LabSeS 59
- 3.2 Statistical Significances of Improvements 79

### **List of Figures**

- 1.1 The Waterfall Model 6
- 1.2 Simplified Scrum Workflow 12
- 1.3 Scrum Workflow 15
- 1.4 Kanban Board Example 16
- 1.5 XP Process Lifecycle 18
- 2.1 SCORE Survey Results 33
- 2.2 Scrum Roles at CECAN 39
- 2.3 Kanban Board at CECAN 40
- 3.1 Laboratory of Service Systems 53
- 3.2 Star Team Topology 60
- 3.3 Network Team Topology 61
- 3.4 *Timeline of LabSeS* 69
- 3.5 Thesis Sentiment, May 2020 72
- 3.6 Thesis Sentiment, April 2021 73
- 3.7 Differences of Medians in Thesis Sentiment 73
- 3.8 Laboratory Sentiment, May 2020 74
- 3.9 Laboratory Sentiment, April 2020 74
- 3.10 Differences of Medians in Laboratory Sentiment 75
- 3.11 Preferred Meeting Frequency, May 2020 76
- 3.12 Preferred Meeting Frequency, April 2021 76
- 3.13 Engagement within Laboratory, May 2020 77
- 3.14 Engagement within Laboratory, April 2021 78

### Introduction

Even though research is one of the most integral cornerstones of a developed society, management and coordination of work in academia is often inadequately addressed and not done in an evidence-based manner. Highly unpredictable and innovative work done by research laboratories puts all efforts of faculty to control chaos to a test, and many eventually regress back to infrequent, one-on-one meetings with students with ad-hoc, unplanned agendas.

Team collaboration is critical in research, as teams increasingly dominate solo authors in the production of knowledge across all fields in productivity, quality, and impact. [1] Benefits of research collaboration range from better use of existing resources to more opportunities for knowledge transfer, complexity management, and wider social impact through large-scale research projects. [2]

Scientific knowledge is produced collaboratively from diverse perspectives across academic branches and industries more than ever. This increase in collaboration is driven by a variety of factors, such as the importance of interdisciplinary and multidisciplinary research and gains from division of labor as a consequence of growing specialization. [3] Collaboration occurs across sectors and types of organizations, such as government-based research programs that usually emphasize multidisciplinary and applied research, or university-industry collaboration, which gives companies a competitive advantage from drawing knowledge and innovation from academia. [2, 4]

All of this collaboration growth is happening amidst the disruptive adoption of information technologies (IT). [5] Such technologies allow for decentralization and distribution of research, up to the emergence of so-called *virtual teams*, unprecedented in human history. [6, 7, 8] Moreover, IT has greatly improved access to knowledge, radically altering the whole social landscape and fast-tracking the shift towards a fast-changing knowledge economy. To name a few impactful effects of such development, we have increased transparency (and thus decreased information asymmetry), automation, or the aforementioned ability to distribute co-workers. Arguably most important among its effects is the ability to react swiftly to changing circumstances, on scales both small (such as a small team reacting to scientific findings in another country) and large (such as the massive worldwide reaction to COVID-19 pandemic). [9, 10]

Various approaches have been developed and tested to manage research groups effectively. [1, 11, 12, 13] One of the approaches that seem to be viable is inspired by the so-called *agile paradigm*, originating from Software Engineering. [2, 14, 15, 16, 17]

Agile and its effects will be studied throughout the whole thesis in various contexts.

The first chapter introduces Software Engineering and Agile, which has emerged therein as one of the work organization paradigms which fully embrace such variability and provide sought fast reaction times. Frameworks widely used, such as Scrum, Kanban, and Extreme Programming, are explored.

The second chapter introduces research applications and academic context. Six different case studies of adoptions of Agile for managing research groups are systematically analyzed. The advantages and disadvantages of each approach are discussed and compared.

The final, third chapter presents a case study of designing our own agile way of working at research group at the Faculty of Informatics, Masaryk University, Czech Republic, in the Laboratory of Service Systems. We were able to successfully address low motivations to engage within the laboratory and improve effectivity of the laboratory head's time with the proposed agile process.

The thesis is typeset in IATEX. Pictures were created in *Microsoft Word* [18], statistical analysis and inference were done in programming language *R* [19].

### **1** Agile in Software Engineering

To effectively address highly complex problem of management of a research team in the following chapters, we will draw our inspiration in methods originally described within domain of Software Engineering. This chapter provides essential introduction to the domain, as well as project management<sup>1</sup> practices therein.

The first section provides definition and context of Software Engineering. Concurrently with inception of Software Engineering, predictive processes (most notably *Waterfall*) were formulated. These approaches were based on the old, Taylorist industrial practices (see [20]), and have quickly proven inadequate for all but the most known and repeated problems.

Second section presents overview, motivations, and essential principles of the opposing mindset and project management practice called Agile.

Third, fourth, and fifth sections then describe briefly three most widespread agile frameworks:<sup>2</sup> Scrum, Kanban, and Extreme Programming.

In the final, sixth section, we discuss the industrial and societal context in which Agile has broken free of Software Engineering-only context, becoming a hot topic in many other fields, including research groups, which we will continue to study in the following chapters.

#### 1.1 Software Engineering

A *computer* is a functional unit that can perform arithmetic and logic computations without human intervention. *Software* is a collection of instructions telling a computer how to solve a certain task, and

<sup>1.</sup> Although *Predictive* and *Agile* are most often described as project management practices, both are understood as comprehensive *mindsets*, giving guidance not only on lifecycle of a software projects, but on management, "customer collaboration", and so on.

<sup>2.</sup> A *framework* is a loose and incomplete (minimal) structure that leaves room for creative adaptation and other practices and tools to be included. A *methodology* is a prescriptive and well-defined (maximal) set of values, principles, tools, and practices that can be used to guide processes to achieve goals. [21]

combinations thereof.<sup>3</sup> Associated work products, such as tests, data, and documentation, may also be part of software. [22]

*Programmer* is any person giving such instructions to the computer. They are typically expressed in an artificial *programming language*. The act of writing in a programming language is called *coding* or *programming*. [22]

*Software development* is any process resulting in or modifying a software product, considered from its conception to termination of its use. *Software maintenance* is the part of software development done after its delivery, i.e. any process of modifying completed software, with purposes such as to correct defects, to improve performance, or to adapt the system to a changed environment or changed requirements.<sup>4</sup> Software development usually includes activities such as requirements gathering, requirements analysis, system design, implementation, testing, documentation, and quality assurance. [22]

Software development life cycle is a sequence of events and patterns that reveal themselves in the lifetime of software development instance. A *software development life cycle model* is a general, idealized abstraction, which connects software products sharing similar life cycles.

The simplest of all life cycle models is so-called *code-and-fix model*, predominant in 1950s and 1960s,<sup>5</sup> years characterized by science applications and programmers being the end-users themselves. The life cycle consists only of two activities – coding a program and running it – repeated until the software is done. The running is done to probe for failures; once diagnosed, the programmer tries to fix them during the next coding phase. [23]

The code-and-fix model, however, quickly proved inadequate. [23] The primary preconditions for its success are

1. full transparency of requirements (i.e. no need for analysis);

<sup>3.</sup> The combinations may be, for example, procedures, functions, or programs.

<sup>4.</sup> Software maintenance is sometimes separated from the definition of software development. In this thesis, we comply with the prevailing choice to include it. As the industry shifts more and more from waterfall projects (see Subsection 1.1.1) to agile products (see e.g. Section 1.3), for which the line between development and maintenance vanishes, such a distinction would not be beneficial.

<sup>5.</sup> The model typical even today for novice programmers and tinkerers.

- 2. small-scale and low-complexity software product (i.e. no need for design); and
- 3. single-person (i.e. no need to coordinate).

As such preconditions are extremely restrictive, new approaches were sought. The body of knowledge and practice eventually became broadly called *Software Engineering*. However, as [23] notes, "There are as many definitions of 'Software Engineering' as there are authors." Some of the definitions are presented in the following list.

- The first published definition is due Bauer (1972), as cited in [23]: "the establishment and use of sound engineering principles (methods) in order to obtain economically software that is reliable and works on real machines."
- The systematic nature of engineering and its roots in science can be emphasized, such as in the ISO/IEC 2382-1:1993 Standard definition, as cited in [22]: "systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software to optimize its production, support, and quality."
- Very simple definition is due Parnas (1978), as cited in [23]: "multi-person construction of multi-version software."
- Wang and King (2000), as cited in [23], make the product attributes explicit: "a discipline that adopts engineering approaches such as established methodologies, process, tools, standards, organisation methods, management methods, quality assurance systems, and the like to develop large-scale software with high productivity, low cost, controllable quality, and measurement development schedules."

At the expense of some rigor, the definitions will be terminated here. The remaining terms will be either briefly defined in footnotes, or we will rely on reader's intuition and experience altogether.

#### 1.1.1 Predictive Approach

*Predictive* approach (also called *defined*) to Software Engineering strives to define exactly what is required, design it fully, build it, and then deploy it. This approach may be run once or cyclical, but will always follow this successive pattern. [24]

*Waterfall* was the first of and is the simplest of predictive software life cycles. It assumes that software development process consists of a number of non-repeating phases in a sequence, each starting only after the last phase was completed – see Figure 1.1. The output of one phase becomes the input to the next phase. [23]



Figure 1.1: The Waterfall Model, adapted from [25].

Once information on customer needs is collected, a contract between the parties is used as a reference for the software development. Customer then waits for the product to be made, performing little to no interaction with the developers, and receives the software only at the end of the process. [26]

If employed correctly, effort spent analysing and designing upfront will lead to greater understanding later on. This makes it a robust, rigorous, and predictable approach for small-scale, simple projects. However, as complexity rises, the full up-front planning becomes progressively more unfeasible. Also, due to the requirement of rigorous completion of each stage, any potential change must be propagated backwards and all phases completed again, which makes them miss delivery dates and prohibitively costly.

As such problems continued to be more and more frequent, iterative models such as the *Rational Unified Process* were invented to combat these issues. [27] However, these heavyweight methodologies, with many phases to execute, many disciplines to focus on, and hundreds of pages of instruction did not let go of full-control over the projects and thus did not address the root causes. Drawbacks such as costly changes to requirements, strict planning and budgeting, and unnecessary complexity of development initiatives, continued to haunt the Software Engineering discipline. New approaches were sought and paradigms were developed.

#### **1.2 Agile Overview**

Growing complexity of IT products (as measured by metrics such as lines of code, number of inputs, cyclomatic complexity, and such) carried a wide load of problems, notably late and over-budget delivery, less-than-satisfactory products, and most prominently, lack of responsiveness to change. [24] With the definition of "success" in predictive paradigm being on-time, within-budget, and all-scope delivery, it became accepted that only 10–20 % of all software projects were successful. [20]

As the success rates did not increase, the paradigm assumed that the instructions and plans were not detailed enough. The added focus on the aspect has, understandably, not increased the success rate. Thus, an alternative paradigm to the documentation-driven, cumbersomely rigid, and highly inflexible predictive development was sought. [24]

*Agile* represents a mindset, a team management approach, and a collection of work organisation practices and frameworks that support continuous progress on work priorities, even in the face of changes. Most agile methods try to minimize risks during the execution of a project by developing software in incremental iterations, which usually last from less than a day to one calendar month, so that during each

iteration (resembling a miniature subproject<sup>6</sup>), the team can fix the requirements and work on these. An agile programming project aims to release new software at the end of each iteration, and between each iteration the team reevaluates its priorities. [2]

However, Agile does much more than merely react to changes: it redefines many of the industrial beliefs held for decades. [20]

Despite the critique by some authors that the agile principles (see later) are insufficiently grounded in theory, that it is suitable for small teams but not larger projects, and that it is not a panacea for effective project management, the majority of peer-reviewed papers and other empirical studies highlight the benefits of adopting agile methods. [2] In January 2018, more than 100,000 users of software development website stackoverflow.com (74 % being full-time employed) from 183 countries around the world participated in a 30-minute survey, and more than 85 % of use Agile in their jobs. [29]

As Agile is relatively young, as opposed to predictive or even codeand-fix approach (see Subsection 1.1.1), this massive adoption shows how Agile is valuable to the whole Software Engineering domain. *14th Annual State of Agile Report* [30] gathered 1,121 full survey responses from agile executives, practicioners, and consultants. Out of these, more than 50 % of respondents said that agile transformation improved their customers' satisfaction and business value, more than 40 % of respondents mentioned increased productivity, better business objectives achievement, better quality of products, and better on-time delivery.

Somewhat surprisingly, 33 % of respondents of the report cited even improved predictability over previous predictive approach. Other sources also nots of this phenomenon: agile projects have often better success rates than predictive ones *even* when the definition of success is based on the previous paradigm of on-time, within-budget, and allscope; when using more agile criteria, with focus on active customer collaboration and frequent delivery of value, they perform even better. [20]

<sup>6.</sup> Miniature *agile* project as part of the final *agile* project, delivering a small, independently valuable slice of functionality. [28]

#### **1.2.1** The Agile Manifesto

Even though such alternative methodologies which promised innovative, early, and continuous delivery through embracing variability and complexity, have been evolving since late 1980s, they have not gained enough recognition yet in the 20th century. Over three days in February 2001, 17 software development gurus from different fields and method frameworks came up with *Manifesto for Agile Software Development*, a simple and concise declaration which has laid foundations for such an alternative to predictive approach. [24]

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

The Agile Manifesto states "That is, while there is value in the items on the right, we value items on the left more." This is critical: none is saying do not have processes, documentation, and so on – but if given a binary choice, agile mindset would prefer the items on the left.

The Agile Manifesto also defines 12 ancillary principles, detailing out the four primary values of agile development. Brief explanation of the Manifesto follows; deeper analysis can be found in [24] and [32].

**Individuals and interactions over processes and tools.** Human resources are empowered and given most of the autonomy to choose rules and control themselves after mutual agreement. Close working environments are fostered, institutionalized processes and standards are less enforced. [32] Motivation moves from the old industrial practice of monetary incentives more to internal motivation of autonomy, mastery, and purpose. [28]

**Working software over comprehensive documentation.** Value for customer is the only measure of success. Documentation is, unless required by customer, virtual inventory, which often does not bring much value. [33] The developers are encouraged to keep the code simple, straightforward, and technically as advanced as possible, thus lessening the documentation burden to an appropriate level. [32] Still, "Minimum Viable Documentation" is encouraged.

**Customer collaboration over contract negotiation.** Maximizing the customer satisfaction is a top priority. Software is released frequently and changes are harnessed for customer's competitive advantage, so that changing business conditions do not invalidate a software whose creation he has contracted previously. The negotiation process itself is seen as a means of achieving and maintaining a viable relationship. [32] Contracts focus on developer time instead of scope.

**Responding to change over following a plan.** Change is not discouraged and expensive, but welcomed and expected. Both developers and customers are well-informed, competent and authorized to consider possible adjustment needs emerging during the software development life-cycle. [32]

#### 1.2.2 Comparison of Predictive and Agile Approaches

The Table 1.1 shows the essential differences between predictive and agile software development. As can be seen, Agile redefines many essential aspects of software development when compared to the older predictive approaches.

#### 1.2.3 Agile Frameworks

Several agile frameworks have emerged over the years. These frameworks typically prescribe a coherent sets of principles, practices, and tools which, when used together, have been proven to be effective at responding to change and delivering quality software. Table 1.1: Comparison of predictive and agile software development, adapted from [32, 34].

	Predictive	Agile
Team Management	Rigid hierarchy, micromanagement	Flat hierarchy, self-organization
Developers	Forming silos	Collocated, collaborative, empowered
Customers	Contracted at the beginning only	Dedicated, collocated, empowered
Requirements	Knowable early, largely stable	Largely emergent, rapid change
Changes	Expensive, brings project back to first phase	Welcomed, considered key for learning
Planning	Single-point, extensive	Often, roughly
Tools	Fixed	Changed if advantageous
Architecture	Designed for current and foreseeable requirements	Designed for current requirements
Refactoring	Expensive	Inexpensive
Size	Typically larger teams and products	Typically smaller teams and products
Primary objective	High assurance	Rapid value

In the following sections, we will explore Scrum, Kanban, and Extreme Programming. Majority of contemporary agile teams use either of the three. There are a few other agile frameworks one may encounter, such as Test-Driven Development (TDD), Feature-Driven Development (FDD), Dynamic Systems Development Method (DSDM), or Crystal Family of methodologies. [29, 32]

The agile frameworks are not mutually exclusive: many can function together to bring superior results. For example, a survey of Scrum Masters carried out in 2019 found that 81 % of teams use Kanban in addition to Scrum (to improve their flow of value), 34 % use Test-Driven Development with Scrum (to better understand requirements and speed up the delivery), and 27 % complement Scrum with Extreme Programming (in pursuit of better software quality). [35]

#### 1.3 Scrum

Founded in 1990s by Ken Schwaber and Jeff Sutherland, Scrum is the most widespread contemporary agile framework.<sup>7</sup> [2, 29] It is defined and maintained by the *The Scrum Guide*<sup>TM</sup>: *The Definitive Guide to Scrum*: *The Rules of the Game* [36]. Scrum is adaptive, empirical framework for developing, delivering and sustaining complex products. It uses iterative and incremental delivery to optimize value and control risk.

Scrum facilitates breakdown of work into small tasks that can be completed within cycles called *Sprints*. Sprints take typically 1 - 4 weeks and always result in a potentially releasable Increment (usable software, see later) fully additive to all prior Increments. Requirements get fixed for the duration of a Sprint: this practice limits risk to the duration of a single Sprint.



Figure 1.2: Simplified Scrum workflow, adapted from [37].

Scrum, as minimal framework, acts as a *guard rail*, establishing boundaries only by the accountabilities of the 3 roles, goals of the 5 events, and purposes of the 3 artifacts, letting the teams instantiate the processes with their own specific practices. [28]

<sup>7.</sup> The term "scrum" originates from a strategy in Rugby, where it denotes "getting an out-of play ball back into the game" with teamwork. [32]

**Scrum Team.** The Scrum Team consists of one Product Owner, one Scrum Master, and a three-to-nine members large Development Team.<sup>8</sup> The *Product Owner* is a single person responsible for maximizing product value and managing Product Backlog (see later). He is the client representative and is responsible for stakeholder relations. The *Scrum Master* is a servant-leader for Scrum Team and the larger organization, responsible for promoting and supporting Scrum theory, practices, rules, and values. Scrum Master helps optimize interactions inside and outside of the team and facilitates Scrum events (see later) as needed or requested. He is also responsible for removing impediments that the development team is unable to remove on its own accord.<sup>9</sup> Finally, the *Development Team* delivers potentially releasable Increments of the product. The Development Team should be:

- *self-organizing*, i.e. the development team itself chooses how to organize and accomplish its work (Sprint Goal) in specified date and quality and within boundaries of Scrum,
- *cross-functional*, i.e. having, as a whole, all the necessary competencies to accomplish its work, to reduce risk of external dependencies, and
- *homogenous*<sup>10</sup>, i.e. not forming silos of knowledge nor tasks.

The Scrum Team is optimized for flexibility, creativity, and productivity. Its size is optimized to be small enough to remain nimble, yet large enough to be able to independently complete significant amount of work each Sprint. [20, 21, 28, 36, 38, 39]

**Scrum Artifacts.** There are three recognized artifacts in Scrum, optimized for maximization of transparency of key information across the whole team and all stakeholders of the product. *Product Backlog* is an ordered list of everything known to be needed in the product:

<sup>8.</sup> These are *roles*, which must be played by a single, albeit not necessarily dedicated person.

<sup>9.</sup> Such as requests to perform activities not related to the project, problems in the test server, or difficulties with the technology.

<sup>10.</sup> The term is not established in literature; other names are "collaborative" or just "fully cross-functional".

features, fixes, non-functional requirements, and so on. It is the sole source of requirements off which work is done and it is evolving with the product and the environment. *Sprint Backlog* is a subset of Product Backlog items selected for the current Sprint, enriched with plans for the current Sprint. It also contains a "Sprint Goal", a short, naturallanguage coherence binding the work together, flexible enough to adapt to possible changes and eventualities of complex development work. Finally, *Increment* is the sum of all Sprint Backlog items completed in the Sprint, additive to all prior Increments. It must be in *potentially releasable* condition upon culmination of a Sprint. By producing such "Done" Increments, the team verifies their assumptions of product's value against reality. [20, 28, 36]

**Scrum Events.** There are five events recognized in Scrum, optimized for regularity, transparency, and minimization of the need for ad-hoc meetings. *Sprint* (often also called "Iteration") is a container for all other Scrum Events and results in potentially releasable Increment. At the beginning of each Sprint, *Sprint Planning* is held and results in plans for the Sprint with adequate level of detail. Everyday during the Sprint, a 15-minutes-bound *Daily Scrum* (often also called "Daily Standup" or just "Standup") is held to inspect whether the Sprint Goal will be completed and to potentially adapt. However, the typical fashion is for each to answer the following questions:

- "What did I do yesterday?"
- "What will I do today?"
- "Do I have any impediments?"

At the end of a Sprint, *Sprint Review* and *Sprint Retrospective* are held in succession. The former is a formal inspection of the Increment with stakeholders present, while the latter is typically informal inspection of the Scrum Team itself, discussing the last Sprint "with regards to people, relationships, processes, and tools." [15, 36]



1. Agile in Software Engineering

Figure 1.3: Scrum workflow, adapted from [40].

#### 1.4 Kanban

Kanban is a strategy for optimizing "value flow" through a system.<sup>11</sup> It was created in 1940s by Toyota, aiming to balance demand with capacity. Not so different to rhytm-based Scrum, Kanban essentialy tries to attain near-zero rhytm up to continuous flow. [41] We will describe Kanban as it has been adapted with excellent results in software development. For the original Toyota's manufacturing Kanban, please see, for example, [33].

Kanban, along with some lightweight principles and artifacts (such as *Definition of Workflow*), is based around three main practices.

**Defining and Visualizing the Workflow.** Kanban workflow's transparency is typically maximized using a *Kanban Board*, such as the board in Figure 1.4. The workflow is divided into several columns called *states* (less often also *stages*), in the most minimal design as "To do", "Doing", and "Done". [34] Tasks (items) are displayed on the board and move along states to show progress of the work. Explicit policies for such flow should be made transparent on the board. [42] Other possible definitions include division of the board into rows or using different-colored items based on contextual criteria.

The best Kanban boards serve as so-called *information radiators*. Information radiators should be (1) simple, (2) big (i.e. viewable from a distance), (3) physical (i.e. physically movable), and (4) brightly

<sup>11.</sup> The term "kanban" means "visual card" in Japanese.

#### 1. Agile in Software Engineering



Figure 1.4: Kanban Board example, adapted from [43].

colored. Such tangible displays emit visceral reactions in people, as opposed to putting many detailed elements of information into small computer displays. Queues of physical things are much easier for people to perceive and/or to perceive as problems.<sup>12</sup> [33]

Actively Managing Items in the Workflow. Although active management of items can take many context-dependend forms, the following practices are typically most essential:

• Controlling *Work-in-Progress*, i.e. limiting the number of items that can be present in each workflow state. This practice has the most profound impact on workflow efficiency, focus by reduction of oversubscription, and team commitment and collaboration to continuous delivery of value, and for that is often regarded as a stand-alone practice. [34, 42, 44]

<sup>12. &</sup>quot;My goodness, there's a *gigantic* pile of *Stuff* there! Making any money from the pile? Are there defects in there? Does it need to be combined with other stuff before we can ship it? Do we need – and will we make money with – *each and every item* in the pile?" [33]

- Inspecting all queues in the workflow and removing them, for example by swarming or workflow redesign.
- Ensuring items do not age unnecessarily.
- Addressing blocked items.

**Improving the Workflow.** In the light of the belief that "It is much worse to not experiment than to experiment and fail", universal among the agile frameworks, team commits to continuously improve their workflow to further optimize throughput, time-to-market, and quality and all other aspects of its products. [33, 41] A common practice is to have a regular meeting to inspect and discuss the current status quo, similar to the Retrospective event of Scrum (see Section 1.3). [42]

#### **1.5** Extreme Programming

Extreme Programming (XP) was the first agile framework to be called that way. It acted as a "stepping-stone," forming out of common programming principles, but taking them – if they "did get the job done" – to extremes (thus, *extreme* programming). After a number of successful trials, the XP methodology has conceptualized the key principles and practices used.

Some of the essential practices include planning poker, short releases, capturing functional requirements as "User Stories", simple design, continuous refactoring, pair programming, continuous integration, collective ownership, and the autonomy of the team to devise its own rules to follow if they prove advantageous. [32]

Extreme Programming is, however, largely a practical (empirical) method, rather than an academic method. [32] Even though the rigorous adoption of the full life cycle of XP wanes off in use today (see [29]), the specific practices used therein have been adopted by many Software Engineering teams in pursuit of their technical excellence.

#### **1.6** Agile Breaks Free of Software Engineering

Since the inception of Agile Manifesto, Agile has become a cuttingedge approach in Software Engineering. However, in the recent years

#### 1. Agile in Software Engineering



Figure 1.5: Life cycle of the XP Process, adapted from [32].

it has also gained traction in other contexts, such as general IT, operations, marketing, HR, sales, real estate, venture capital, public institutions, education, research, and services, bringing varied improvements over traditional approaches. [2, 30, 45, 46]

Several authors have tried to isolate the practices of agile project management and separate them from Software Engineering. For example, paper [46] cites specifically the following practices:

- 1. use of the "project vision" concept,
- 2. simple communication tools and processes,
- 3. iterative planning,
- 4. developing activities via self-managed and self-directed teams, and
- 5. frequently applying project plan monitoring and updating activities.

The focal aspects of light and agile methods are simplicity and speed. In development work, accordingly, the development group

concentrates only on the functions needed at first hand, delivering them fast, collecting feedback and reacting to received information.

To compare, paper [32] defines Agile as any development that is

- 1. incremental (small software releases, with rapid cycles),
- cooperative (customer and developers working constantly together with close communication),
- 3. **straightforward** (the method itself is easy to learn and to modify, well documented), and
- 4. **adaptive** (able to make last moment changes).

Agile, more so than other management frameworks, emphasizes teamwork by focusing on the social aspects of software development, with collective ownership and collective responsibility as key principles. [47]

"What is new about agile methods is not the practices they use, but their recognition of people as the primary drivers of project success, coupled with an intense focus on effectiveness and maneuverability. This yields a new combination of values and principles that define an agile world view." [32]

More than not, what Agile truly is, in which contexts it is beneficial, and what aspects of it drive its success remain largely unknown to humankind. In the following chapter, we will study usage of Agile in research groups in academic context, the primary focus of this thesis.
## 2 Agile in Research

In pursuit of interdisciplinarity, researchers worldwide continue to tailor previously industry-specific practices to different contexts with varied levels of success, including tailoring of Agile (see Section 1.6). We will examine literature of applying Agile to education and research contexts in this chapter, reviewing both the theoretical works and experience reports.

In Chapter 1, we have seen the motivations for using Agile – such as embracing variability and flexibility – in software engineering. Such preconditions are entirely in line with specifics of research. For example, in software engineering, one the most important elements of unpredictability is the fluidity of requirements.<sup>1</sup> In research, however, the mere existence of requirements is often compromised. This and other specifics of research done in academic institutions will be discussed in the first section of the chapter.

On the other hand, it is natural that agile frameworks have to be extensively adapted (modified for application) to specifics of the field when they are being applied there. [16] This contrasts with some schools of thought, such as that of the strict Scrum.<sup>2</sup> We believe this stance may be explained by the fact that the inherent peculiarities of academia would be regarded as dysfunctions in the industry for which Agile was originally conceived. In particular, study [48] has identified three major features of academic environment that seem to conflict with Scrum rules:

- 1. students do not engage a full working day,
- 2. students have a busy schedule, and
- 3. research projects developed by students tend to be done by an "army of one man".

Although there are lots of empirical case studies of successful collaborative work using Agile, such reports "evolve from personal

<sup>1.</sup> As Albert Einstein put it, "If we knew what we were doing, it would not be called research, would it?"

<sup>2. &</sup>quot;The Scrum framework, as outlined herein, is immutable. While implementing only parts of Scrum is possible, the result is not Scrum." [36]

#### 2. Agile in Research

experience rather than systematic study" [12] and "their suitability in education is still an ongoing debate." [16] Furthermore, it is still a challenge to find a metric to formally evaluate the real gains achieved by application of the agile methodologies. These studies are typically based on informal feedback and surveys of subjective questions, such as satisfaction with the process. Such data are highly prone to various skews – as case-control study execution is very hard in the context, it is next to impossible to isolate various causal and confounding factors influencing the results.

The first section of the chapter explores the context of research and academic environment and introduces terms such as basic and applied research. We study how researchers form into teams to produce increasingly superior research, exhibiting high levels of autonomy and fuziness. The section concludes with an extensive, but due to state-of-the-art shortcomings explained therein not exhaustive list of challenges to effective teamwork in research settings.

In the second section, we present six case studies of Agile adoption in research and education.<sup>3</sup> We have taken this path, as although there have been efforts to enumerate aspects of research context to be answered before creating a tailored agile process (such as [16]), they still mostly do not give adequate guidance on how to afterwards construct such a process.

These case studies were carefully selected from the literature to cover all three essential agile frameworks (see Section 1.3, Section 1.4, and Section 1.5) in varied sets and settings in academia to allow for a non-trivial comparisons among them in the final section and provide sufficient inspiration for construction of our own agile process for Laboratory of Service Systems in Chapter 3 in an objective and data-driven manner.

<sup>3.</sup> Although research is the focus of the thesis, we have included two academic education studies for comparison. For literature reviews regarding use of Agile in education, please see [34] or [49].

## 2.1 Academic Research Context

In this section, we will study the activity of conducting research<sup>4</sup> done at academic institutions, such as universities and research groups. As we will see, such activities have inherent peculiarities as compared to industry, which ought to be understood before the sections that will follow.

#### 2.1.1 Research

*Research* is "creative and systematic work undertaken in order to increase the stock of knowledge – including knowledge of humankind, culture and society – and to devise new applications of available knowledge." All research satisfies the five core criteria of being

- 1. novel,
- 2. creative,
- 3. uncertain,
- 4. systematic, and
- 5. transferable (reproducible). [50]

Applied research is "original investigation undertaken in order to acquire new knowledge. It is, however, directed primarily towards a specific, practical aim or objective." [50] Its counterpart is *basic research*, defined as "experimental or theoretical work undertaken primarily to acquire new knowledge of the underlying foundation of phenomena and observable facts, without any particular application or use [necessarily] in view." [50]

Research can be *theoretical* (presenting new or established abstract principles) and/or *empirical* (observational or experimental, usually based on real-world data). Such research, which communicates new knowledge arrived at or discovered by the author, is called *primary* or

<sup>4.</sup> We will be using the terms *research* and *science*, as well as their personal nouns *researcher* and *scientist*, interchangeably. Strictly speaking, science is research done in an evidence-based manner, but we have taken the liberty not to differentiate the two. Research is defined in Subsection 2.1.1.

*original research*. Examples include controlled trials, technical reports, and case studies. On the other hand, *secondary* or *review research* is the act of revisiting, reviewing, analyzing, and compiling existing primary research. Examples include narrative reviews, systematic reviews, and meta-analyses.<sup>5</sup> [51]

For example, this thesis – as well as most of the other theses written under the Laboratory of Service Systems (see Chapter 3) – is of applied and empirical nature. Although it employs narrative review of literature<sup>6</sup> in the first two chapters, these chapters only set the already known theory and context for specific goals of the third chapter and the thesis as a whole: evaluating possibility of improving management at Laboratory of Service Systems through Agile and documenting its execution by providing adequate detail of state-of-the-art.

Institutions<sup>7</sup> of various sectors may partake in research, such as academia, business enterprises, government, or private non-profits. [50] This thesis focuses on the first group, i.e. *academia* (or *higher education*), broadly defined as all universities, colleges and other institutions providing formal tertiary education programmes, as well as all research institutes and centres under the direct control of, or administration by such tertiary education institutions. [50] Both public (i.e. under ultimate control by government) and private (otherwise) institutions are considered in this definition. The second institution group, i.e. business enterprises, are also considered in case of direct collaboration with academia.

As we focus on the academic context of tertiary education, differentiation between research and education should be in place. However, as both activities are very closely linked and mostly done by same students and academic staff, on the same premises, with same equipment, and feed its results to one another, defining the borders of the two may

<sup>5.</sup> One may often find so-called *tertiary literature* next to the two, defined as all works indexing or re-compiling data from a broad ranges of secondary research. This third category, comprising of items such as encyclopaedias, manuals, and textbooks, has typically much lower requirements for quality, such as structure, precision, or citations, and is usually not regarded as research.

<sup>6.</sup> *Narrative review* is a type of secondary literature not using explicit and reproducible filtering criteria for the primary and secondary sources it references.

<sup>7.</sup> *Institution* is defined as an established organization of people and resources.

be difficult. Here, the five core criteria mentioned in the beginning of this section may be used to distinguish the two.

There are many stakeholders<sup>8</sup> in research with various categorizations dependent on context and geography. The two most important groups we will be studying in this thesis are faculty and students.

Term *faculty* encompasses everyone with academic *rank*, such as assistant professors, associate professors, and full professors.<sup>9</sup> Academic ranks are awarded for research achievements. Typical research activities of faculty are conducting research, consisting of activities such as identifying research problem, reviewing the literature, collecting and interpreting data, and reporting (in such cases, faculty is often referred to as *academic researcher*) and supervising advanced students on their own novel research (in such cases, faculty is often referred to as *supervisor*), while non-research activities may include teaching courses, supervising bachelor theses without sufficient levels of novelty, or own reading not related to any specific research project. [50, 52]

Term *students* encompasses everyone actively working to obtain an academic *degree*, such as bachelor students working to obtain bachelor's degree, master students working to obtain master's degree, and doctoral students working to obtain doctoral degree.<sup>10</sup> Bachelor and master students are often collectivelly referred to as *undergraduate* students, doctoral students as *graduate* students, and researchers holding undergraduate degree as *postgraduate fellows*. To be awarded a degree, students typically need to write, present and defend a *final thesis* or *dissertation*. The requirements for successful defense of the work vary based on factors such as country, degree sought, school, and major. Although not taxative, it is common that bachelor thesis contains no novelty, master thesis contains some novelty but is still largely dependent on supervisor, and doctoral dissertation is highly novel and adequately independent.

A stakeholder is any entity that can affect or be affected by considered context.
In Czech Republic, these positions correspond to translations *odborný asistent*, *docent*, and *profesor*. However, there are also many other ranks.

<sup>10.</sup> In Czech Republic, the corresponding student levels are called *bakalářský student*, *diplomant* or *magisterský student*, and *doktorand*.

Several other stakeholders of academic research, but not producing it themselves, may be further identified, such as administrative staff, funders, and publishers and their editorial staff. [50, 51].

#### 2.1.2 Research Teams

Teams have been investigated extensively in the corporate world in management literature, as cross-functional project<sup>11</sup> teams are increasingly adopted by companies to manage the rapidly changing and competitive environment. *Team* is defined as "[a group] made up of individuals who see themselves and who are seen by others as a social entity, who are interdependent because of the tasks they perform as members of a group, who are embedded in one or more larger social systems (e.g., community, organization), and who perform tasks that affect others (such as customers or coworkers)." [3]

Teams increasingly dominate solo authors in the production of knowledge across all fields, due to benefits such as better use of existing resources,<sup>12</sup> more opportunities for knowledge transfer,<sup>13</sup> complexity management by work division, and wider social impact through large-scale research projects. [1, 2]

Paper [3] suggests that teams in science can be broadly categorized based on the way of assembly in the following two categories:

- 1. **Formal teams**, such as those formed by academic department or laboratory, which are clearly delimited and more denselyknit, and
- 2. **Collaborative teams**, which are formed ad-hoc, assembled fluidly for the sole purpose of collaborating on a specific scientific project in a decoupled way.

Collaborative teams in science are largely voluntary and based on mutual interests, and scientists have substantial autonomy to create

<sup>11.</sup> In context of research, we define project as any collaborative enterprise of a possibly evolving team. As opposed to traditional project management definition (e.g. [22]), science project may or may not have well-defined start and/or end and it may or may not have co-evolving, rather than pre-defined aim.

<sup>12.</sup> Putting together resources in such a way that all may use the resulting aggregate in a predetermined way is called *pooling* of resources.

<sup>13.</sup> When learning is a by-product of different primary goal, it is called *multi-learning*.

their collaborative teams. [3] Furthermore, experience shows that the clear definition of formal teams' boundaries does not impact the typical autonomy of research teams – laboratory membership tends to be voluntary, innovation challenge or hackathon participation in a company tends to be voluntary, and so on. Compare this with traditional teams in hierarchical corporate structure, in which autonomy to self-assemble is, although not zero, significantly diminished.<sup>14</sup> It is thus questionable whether such categorization is in place, or if *collaborativeness* is a quality lying on a spectrum.

On top of the autonomy to self-assemble, collaborative teams possess high autonomy to restructure or dissolve themselves at will. The unpredictability and innovative nature of research magnify rate of such changes in team compositions, resulting in highly fluid team memberships. Moreover, the creative process (including, at minimum, idea generation) underlying the final team product starts even before a team is assembled. [3]

As various stakeholders<sup>15</sup> participate in the team during various phases of the project with varying levels of involvement and contribution to the final outcome, problem with definition of collaborative team's boundary arises (to take credit and responsibility). [3] It is evident that metrics such as final choice of co-authors or all team members active at the project closure do not capture the full flavor of the team.

#### 2.1.3 Challenges of Team Research

As of today, there are as many methods and methodologies for management of research teams as there are researchers. However, no comprehensive and universal approaches have been developed<sup>16</sup> and most historical and contemporary literature focuses on empirical case studies, from which they sometimes, at most, try to infer some abstract

<sup>14.</sup> As cited in [12], traditional teamwork model "typically emphasise[s] careful selection of participants, shared goals, structured work plans, explicit roles and accountability for products and deadlines."

<sup>15.</sup> Such as consulted specialists, researchers on parallel projects, advisory, management, or former team members, on top of the current main researchers themselves.16. To author's best knowledge.

principles, which, subsequently lack this grounding in coherent theory. [12]

On the other hand, there are scholarly examinations of both success factors and barriers for effective team research. This may be partially due to the fact that such factors may be simple enough to enumerate and thus analyze in systematic reviews.

As there is obvious equivalence between success factors and challenges,<sup>17</sup> we will keep to challenges for a minor gain in simplicity. A rather extensive list of challenges, compiled from papers [2, 12, 13, 15, 48, 51], in no particular order follows. Albeit extensive, it is still not exhaustive and the categories are not necessarily mutually exclusive. For further discussions, please see the respective papers and their lists of sources.

- **Specifics of research work**, such as its unpredictability and innovative nature (see Subsection 2.1.1), manifesting in issues such as the inherent risk of choosing a wrong project or thesis topic.
- **Risks of emergent research teams**, such as the possibility of introduction of team members which will have significant antipathies with other team members.
- Motivational lows pertaining to productivity caused by factors such as too few milestones in sight, daily grind with too few positive feedback mechanisms, and procrastination caused by the perceptions of large amounts of work ahead.<sup>18</sup>
- Other emotional problems that arise during research, such as insecurity (and the related self-consciousness to ask for help or to adapt mentoring style), anxiety (due to high pressure to perform, real or imagined), or even boredom.

<sup>17.</sup> Success factor is a mitigation or removal of a challenge, while challenge is a lack of success factor.

<sup>18.</sup> Please also see the *Dunning-Kruger effect*, e.g. in [53].

- Inadequate research meta-skills, such as inability to select important papers to read or uncritical consummation of information.<sup>19</sup>
- Frictions among team members caused by varying levels of seniority and training. Both technical and methodological inad-equacies may result in ostracism if not handled properly.
- Lack of soft-skills, such as leadership, friendship, and solidarity.
- **Difficulties in maintaining protocols** caused by varying levels of seniority and training.
- **Inaccessibility of team members**, as caused for example by professional travel, overly demanding individual research, or simply low commitment to the particular project.
- Lack of transparency in process,<sup>20</sup> such as frequency and content of meetings, choice and usage of tooling, quality agreements, release deadlines, or data definitions.
- Lack of transparency in people, such as implicitly, rather than explicitly defined roles, implicit, rather than explicit and overtly stated motivations, values, and beliefs of all stakeholders, or inability to capitalize on skills of other team members.
- Lack of transparency in research work products, such as inadvertent duplicate work, inability to capitalize on previously solved problems, or difficulties within the project management triangle of time, scope, and cost, in case of funded research and development.
- Lack of transparency in status, such as insufficient information about activities and results of others or unnecessary delays

<sup>19.</sup> For example, to understand the differences between various levels of evidence, such as theoretical validity, case study, and a large and well-designed case-control study.

<sup>20.</sup> *Transparency* is a quality of information to be available to and understandable by all stakeholders.

#### 2. Agile in Research

between a student starting to struggle and its identification by his supervisor, caused by too infrequent or too ineffective synchronizations.

- **Delays** of various causes, such as waiting times for approvals in hierarchical structures.
- **Problems with distributed initiatives** and interdisciplinary researches, or even simple student-advisor relationships, as lack of communication may create problems.
- Lack of training in interdisciplinary collaboration, manifesting in problems such as frictions occuring during discussions of various discipline viewpoints and their importances.
- **Conflicting theoretical frameworks**, such as basic, hypothesisdriven, and experience-driven research approaches.
- **Cultural differences** and/or **managerial politics** may prove to be barriers to communication and collaboration.
- Ethical dilemmas, such as defining individual boundaries, balancing individual and team autonomy, or distributing limited resources among team members.
- **Inadequate funding**, further exacerbated in cases of interdisciplinary initiatives.
- **Time schedules of students**, who typically have other commitments on top of the single research project in their studies.

As we dive deeper in the following sections, these challenges should be continuously kept in mind and compared against proposed frameworks. The agile approach that follows tries to address many of the issues.

## 2.2 Case Studies

We will present the following case studies in this section:

1. Maryland's SCORE Framework, page 31

- 2. txtUML's Agile Methodology, page 34
- 3. CECAN's Distributed Scrum, page 37
- 4. PROMOBILE's Full Scrum, page 41
- 5. SLIIT's Undergraduate Research Projects, page 44
- 6. Kaiserlautern's XP Seminar, page 47

The comparison of their respective advantages and disadvantages will be provided in Section 2.3.

The case studies are introduced and discussed systematically for the following aspects. For each case study, we begin by presenting the *background* of the research group with its context (such as parent institute) and brief overview of content of its research. We then highlight the *motivation*(*s*) present for agile transition as well as researchers' expectations of the benefits. In cases where reported, previous style of management is compared. After motivation follows the specific *adaptation* of Agile used in the study, examining its elements such as roles, events, and artifacts. *results* discussion follows as by the original author, with its data drawn typically from interviews and observations. Finally, we have a *discussion* the full case study: we try to isolate the elements used from the specificities of the research group, we highlight their value, and reassess situations which are advantageous for adoption of such elements, both individually and in synergy.

#### 2.2.1 Maryland's SCORE Framework

Based on *Adapting Scrum to Managing a Research Group* [15], this case study details out how authors have proposed and implemented agile process called SCORE ("Scrum for Research") in their research group at Department of Computer Science at University of Maryland with superior results.

**Background.** A research group consisting of 2 advisors and about 10 doctoral students. Students have had weekly one-on-one meetings with their advisor prior to the Agile implementation.

**Motivation.** Maximize intra-group transparency. Foster understanding of the results of others. Improve opportunities for collaboration. Decrease meetings load on faculty.

**Adaptation.** Very lightweight – defines only two key elements. *Status meetings* are a three-times-per-week, all-hands-on meetings, similar to Daily Scrum. The goal is to get everyone updated on the current situation (status). All students and faculty take turns, summarizing in 1–2 minutes the following three points:

- 1. "What have I done since the last meeting?"
- 2. "What results have I achieved and which obstacles I faced?"
- 3. "What do I plan to do by the next meeting?"

As with Daily Scrum, the status meeting is kept to 15 minutes. This time-slot is a good balance to give everyone chance to talk, yet discourage delving into technical details. For discussion of such technical details, the second element of SCORE, *on-demand* meetings are scheduled. These are not regularly scheduled, but only when the need arises. Therefore, the required length of the meeting can be estimated much more precisely.

There are a few additional optional elements to be found in SCORE, such as on-demand short talks (for example, summarizing important conferences, recently submitted papers, or instructing others on a tool a student is familiar with), or informal social outings such as a common lunch or a reading group to foster a sense of community.

**Results.** The data sources to assess the implementation were twofold. First, the faculty (authors) personal experience included the following:

- 1. improving student motivation to work consistently,
- 2. keeping up-to-date with how students are doing and much faster identification of a student struggling,
- 3. lower schedule fragmentation and schedule flexibility, and

4. much better productivity of meetings, as meetings now had (by definition) always a clear purpose.

The second data source was an informal opinion survey using Likert items administered among students year and a half after the SCORE adoption. Out of the 13 students who have completed the survey, 8 have experienced the original research group structure and have been asked to answer the same questions about the previous structure as they remembered it. The questions and results are summarized in Figure 2.1.



Figure 2.1: Boxplots of SCORE adoption survey results, adapted from [15].

The change to SCORE has unambiguously positively (or nonnegatively) impacted every considered aspect of student experience in every student's perception. The most improved areas were studentstudent interactions (a median jump of 2.5 points) and interactions with other faculty (a median jump of 2 points). More suprisingly, student-advisor interactions also have slightly improved (a median jump of 0.5 points), despite the fact that they have no longer held weekly one-on-one meetings.

All in all, there were no scores as low as 1 in the SCORE results, and all but the research quality questions (answered probably with some humility) had median of 4. In contrast, the recollection of the old way has a median score of 4 only in a single aspect – quality of interactions with the advisor.

On top of these, students were surveyed for short answers to elaborate on the benefits and drawbacks of SCORE. Terms such as "research community" and "sense of belonging" were often cited, along with ability to follow other people's daily research routines, quick identification of people who can offer methodological or technical advice, and group-oriented approach which catalyzes greater diversity of ideas and general fun in doing research.

**Discussion.** We like the simplicity of the approach. As research work is highly novel and uncertain (see Subsection 2.1.1), it makes little sense to plan on a cadence. However, even with that, we feel that a continuous improvement dimension of Agile is missing from SCORE. We would welcome a Retrospective event to be held from time to time.

Another issue is with how the data were gathered. Using recollections of the past is inherently very vulnerable to a skew. Nonetheless, SCORE is still one of the best studies data-wise.

On the other hand, the reported perceived benefits were universally positive, and that on a relatively large margin. Fostering the sense of community by often bringing all students face-to-face and having the faculty report in the same fashion most probably helped. The latter point regarding the faculty is important not to encourage overreporting progress and underreporting problems by students.

#### 2.2.2 txtUML's Agile Methodology

Research and Development team of project txtUML adapted one of the more custom and prescriptive approaches. Based on *Create your own agile methodology for your research and development team* [16], author also lists several aspects of research groups which should be taken into consideration when constructing an agile process. **Background.** A research group at Eötvös Loránd University in Budapest, Hungary, whose subteam has adopted Agile. The subteam was working on a single software product "txtUML" and consisted of 9 undergraduate students, one doctoral student, and one teacher. The team was having an inefficient weekly meetings, during which many technical issues involving only two or three members were discussed.

**Motivation.** Improve meeting efficiency. Improve motivation. Train new members fast and preserve knowhow, as undergraduates fluctuate rapidly.

**Adaptation.** Similarly to Scrum, the prescriptive methodology at txtUML team defines specific roles, events, and artifacts.

There are five roles in the methodology. The *project leader* represents the goals of the whole research team towards the university and provides direction, similarly to the Scrum's Product Owner. *Scrum Master* is a non-technical role, responsible for protecting the team's operational values, facilitating better meetings, coaching in self-organization, and reporting of impediments. Those who execute the research and development itself organize into hierarchy of 3 layers of a single *technical leader*, several *subgroup leaders*, and the rest of *developers*. All decisions must be confirmed with the higher layer and the higher layers provide guidance and distribute tasks to do.

The group holds one two-hour meeting each week. Its function depends on the time of the semester; txtUML recognizes four different phases. The first 2-3 weeks are *preparation phase*, during which main research and development directions are discussed, new developers get to know the system, and tasks are assigned. *Weekly routine* takes majority of the semester and is divided into 5 sections: news (pertaining to the whole laboratory), weekly Scrum (with the same format as Daily Scrum), Agile Topic of the Week (presentation and training of one agile value), and technical discussions of the whole team and then in smaller groups (to address technical questions raised with affected or concerned subsets of the group). *Retrospective* is held once in the middle of the semester, during which team members reflect on the quality of teamwork and formulate directions and methods for

future work. Final event of the semester is the *demonstration*, during which all members summarize and showcase their results.

Lastly, txtUML defines three artifacts over the code base itself. The product has its *product backlog* and all subgroups have their respective *issue boards*, implemented as Kanban boards with workflow states "to do", "in development", "under testing", "pull request", and "done". In addition, internal *weekly reports* are compulsory, even though most team members "repeatedly expressed their disapproval."

**Results.** Similarly to SCORE framework report, the case study used both perceptions of authors and survey among team members one semester into adoption. Benefits as observed by the faculty included the following:

- 1. increased efficiency and dynamics of weekly meetings,
- 2. increased quantity of communication among team members,
- 3. developing group responsibility and increased clarity of individual and collective goals, and
- 4. rapid knowledge flow from more senior team members to newcomers.

The last point was stressed the most, and was also one of the original reasons to adopt Agile.

During the Retrospective in middle of the semester, team members highlighted satisfaction with shorter and more efficient meetings and better knowledge transfer. On the other hand, subgroup leaders (not surprisingly) struggled to balance their research and mentoring of juniors, and weekly reports and process over-formalization were critiqued by many. Faculty has exercised their authority and asserted these points as not open to discussion.

At the end of semester, 7 out of 9 students filled out a Likert item survey. The most important results were the following:

1. "To what extent do you think the value achieved during the semester is related to the project's management?" scored 3.71 in average,

- 2. "In what a measure did the values of the agile methodology prevail during the team work?" scored 3.41 in average,
- 3. "How satisfying was the number and quality of feedback you received regarding your work?" scored 4.57 in average, and
- 4. "How much did you enjoy being part of the team?" scored the highest average at 4.71.

**Discussion.** We feel that the methodology is truly unnecessarily complex. For example, defining 5 distinct roles with specific accountabilities for 11 people does not make much sense, and the three layers of reporting hierarchy will necessarily stifle any self-organization, which we know is key for high-performance. [20]

The hierarchy was further enforced by exercising senior rights during work allocation as well as during Retrospectives.

The survey itself was not comparatory, but in absolute terms. Such Likert items without any effort to normalize have next to zero information value (cf. Section 3.2).

On the other hand, authors were able to achieve rapid knowledge flow among team members and once a week meeting is manageable for undergraduate students.

#### 2.2.3 CECAN's Distributed Scrum

Case study of implementing a Scrum-like process at a big distributed research initiative in the United Kingdom follows. The study *Adapting the scrum framework for agile project management in science: case study of a distributed research initiative* [2], although building more off Scrum principles rather than Kanban, is the only study in which activity on online boards was analyzed.

**Background.** Centre for the Evaluation of Complexity Across the Nexus (CECAN), a distributed research centre in UK, spanning more than 50 members, academic researchers of varying seniority, working in 14 different academic institutions. Agile was perceived as potentially effective way of management to be experimented with, rather than an answer to current problems.

**Motivation.** Improve autonomy, self-organisation, flexibility, and adaptivity. Unlock ability to decide based on "bottom-up process of dialogue."

**Adaptation.** The adoption was a commonly agreed transition due to the perceived potential value of Agile. It was based on an evolving interpretation of agile principles and on experimentation in an "explorative, self-taught approach."

The final process at CECAN consists, in essence, of one role on top of what has been there previously, of Kanban boards, and two principles for development work.

The defined role was that of a Scrum Master, a facilitator role played by a member of the research team. The role was conceived as a coordinator for case studies, a link between specific tasks and objectives and other collaborator researchers, as well as liaison with policymakers and representatives from government agencies. As opposed to the Scrum Master role as defined in Scrum (see Section 1.3), the CECAN's adaptation of the facilitator role also combines elements of the Product Owner, such as facilitating connections (with stakeholders and else) and of "senior" developers, such as coordination and providing guidelines for specific research. The Scrum Masters were seen as "connectors of expertise" and coordinators; as intermediaries in interdisciplinary projects, linking expert to expert. However, as opposed to a more traditional approach to management, Scrum Masters have started to lead and facilitate, rather than manage and control. The role was designed to provide much-needed transparency and guidance.

Kanban boards were adopted for both tracking to-do work and documenting already done work. Web-based Kanban tool, replicating Post-It notes, was used. The boards were accessible by everyone, including occasionally by external collaborators and stakeholders, and were managed mainly by Scrum Masters. Each new initiative or discussion was eventually translated into a board. This has represented a novel way of accessing updated and valuable knowledge for the entire organisation about the progress of projects.

The final two principles adopted were *Development Sprints* and *Incremental Development*. Both were only guiding, rather than being



Figure 2.2: Diagram of roles at CECAN, adapted from [2].

explicitly and universally followed. Some teams used fixed cadences, while others perceived them as arbitrary and unnecessary. Some gathered feedback continuously and improved collaboration with stakeholders, some did not.

**Results.** The data obtained come from three sources. First, online activity in the Kanban boards was numerically analyzed. The resulting statistics are presented on Figure 2.3.

Number of active users on a board is positively correlated with the number of cards and activity, such as moving among workflow states and content editing, but not with number of comments and attached documents. Authors argue that this confirms that Kanban boards were used primarily for visualizing workflows. Moreover, although some researchers were highly active proponents of the online tool and it was generally perceived as practical, analysis has shown that only minority of researchers were in fact active, despite the entire organization having full access, often perceiving is as unnecessarily complicated. Nonetheless, a larger number of participants adopted more of a passive role, viewing the boards to have a quick overview of what is happening and what other people are doing.



Figure 2.3: Statistics of cards on each Kanban board at CECAN, adapted from [2].

The second data source came from semi-structured interviews with 10 people, out of which 6 acted as Scrum Masters, and 7 additional researchers not involved with CECAN acted as controls. Data from these interviews were systematically analysed using advanced methods. Based on this analysis, 4 clusters of conditions for and 7 clusters of challenges for adopting Agile in distributed research organisations were identified.

Finally, the third data source was perceptions of the authors, as based on observation and data of previous sources. The adoption seemed successful overall: it facilitated new dynamics in collaboration within the large organisation. However, the process was challenging and had many limitations.

**Discussion.** There are no data on the general sentiment about Agile adoption. Answers in the interviews and authors' perceptions are ambivalent, and the focus on conditions and challenges, rather than real adoption results, reinforces that.

We believe that more than agile frameworks not being suitable for distributed research initiatives, as authors argue, the ad-hoc, incremental, and unguided implementation was to blame for the poor results. We conjecture that if all had committed to a process, its effectivity could be evaluated much better.

#### 2.2.4 PROMOBILE's Full Scrum

The Scrum implementation in paper *Rapid improvement of students' soft-skills based on an agile-process approach* [48] is the first one of the presented to apply the full Scrum framework with only very minor modifications on research projects.

**Background.** Large Scale Qualitification Program on Mobile Technologies (PROMOBILE) at Institute of Computing, Federal University of Amazonas (UFAM), in Manaus, Brazil, hosting 23 projects in 4 research areas involving a total of 34 undergraduate, master, and doctoral students. Such projects initially deliver software in the PRO-MOBILE program, but some of them later become parts of research projects and final theses.

**Motivation.** Improve students' soft-skills, such as proficiency in languages, communication skills, oral presentations, punctuality, and leadership by involvement in a systematic project development experience. Motivate students and inspire them to take responsibility. Decrease loads on senior researchers without sacrificing advisory quality.

**Adaptation.** The adaptation at PROMOBILE closely follows the original Scrum framework (see Section 1.3). Each of the four research groups acted as a team. Faculty took the roles of Product Owners; most senior student of each group, on top of his own research objectives, typically took the role of the Scrum Master. In each group, there were several stand-alone projects, each tackled by up to four students, though one student per project was most frequent.<sup>21</sup> All

<sup>21.</sup> Authors call students elaborating each project a "development team" and say that several development teams share Product Owner and Scrum Master. We believe

#### 2. Agile in Research

Scrum events were observed as described with Sprints being one or two weeks long, with the only slight modification of Daily Scrums happenning only thrice a week, similarly to SCORE Framework, due to busy and often conflicting schedules of students. This down-scaling happened after a period of daily Daily Scrums in which many students struggled. Finally, artifacts such as Product Backlog and Sprint Backlog were observed too and filled with tasks such as investigation elements, experimentation phases, and reporting.

**Results.** The study is the only one out of the presented studies that implemented its agile process incrementally (cf. "Shu-Ha-Ri' principle, [39]). This adoption had three phases. The first phase was designated for training and coaching in Scrum. It was executed rigorously, and with help of an experienced Agile coach, all committed to observing Scrum rules as closely as possible until its effects could be measured, before bending its rules. During the second phase, members started to understand the underlying process. Only after that reservations about Scrum become suggestions for improvement. The final, third phase, was the time when participants reached maturity and became able to improve it themselves. The research group becames self-managed and self-improving and nominated Scrum Masters self-managed to discuss improvements of the whole process and its practices.

The program was running for two years before publication of the study. Data to measure results came primarily from three sources. First, recurring feedbacks from Retrospectives were gathered. Second, students have filled out a questionnaire, but the study is unclear on when it was carried out, how many students participated, and what was its design. Finally, as usual, authors comment on their perceptions.

One of the main advantages brought on was the enhanced visibility of the research process and enhanced communication thanks not only to usage of User Stories. Students also became more open and honest, such as in not claiming an undone item done.

Increased motivation of students was another highly welcomed aspect. Sprint Reviews push students to demonstrate working results, which motivates them to be well prepared, as presenting working

this is a slight misrepresentation. What actually happens is only "formation of silos", not forbidden, only discouraged in Scrum (see [36]).

pieces of research to the whole group is highly satisfying. Furthermore, after a couple of Sprints, students became confident in reporting and sharing results openly even after a failure to deliver what has been promised, using the experience to improve for next Sprints.

Finally, the Table 2.1 summarizes improvements in various softskills, as reported by students in the mentioned questionnaire.

Soft-Skill	Improved	Exercised in	Commentary
Writing skills	70.4 %	Reports, User Stories	Grammar, ability to condense information
Oral skills	70.4 %	Scrum events	Discussion, presentation
Punctuality	100.0 %	Delivering, being on-time	Not stealing time of others
Group activities	100.0 %	Scrum events	Responsibility, cooperation
Leadership	66.7 %	Scrum Master role	Or support of the role
Decision-making skills	88.9 %	Delivering	Execution of work

Table 2.1: Percentages of students who report improvements in various soft-skills, adapted from [48].

**Discussion.** We think that implementation of Scrum at PROMOBILE was very rigorous, as opposed to other case studies presented. One of the possible explanations is the hiring of a professional Agile coach to help with the transformation. The methods used to gather data could be improved – as usual – but we do not think that they invalidate the positive results.

It is evident that students enjoyed the agile way of working and did not perceive the many meetings as too bothering. One of the reasons for that could be how projects were grouped by research areas and as well that many projects were elaborated by more than one student.

Further investigation in the area should be done to identify necessary conditions for students not to be bothered by the meeting overhead, and actually enjoy it and manage it.

### 2.2.5 SLIIT's Undergraduate Research Projects

Moving to the border between research and education, the paper *Improvements for agile manifesto and make agile applicable for undergraduate research projects* [54] describes how about 200 undergraduates worked within a modified Scrum framework to deliver group projects.

**Background.** Software Engineering Project course at Sri Lanka Institute of Information Technology (SLIIT), designed for 3<sup>rd</sup> and 4<sup>th</sup> undergraduate years, spans 2 semesters 14 weeks each. Total 50 groups of 4 members took part, each group choosing either a client-oriented project aimed at solving existing industrial issues or a research project driven with an innovating idea of flexible scope. Both of these types of projects were represented roughly equally.

**Motivation.** Be adaptive to changing requirements or new findings. Foster feedback in client-oriented projects. Experiment with a new way of conducting the course.

**Adaptation.** The agile process at SLIIT comprised of 3 roles, 4 events, and one artifact.

Each one out of the 50 groups was formed by 4 *students*, who elaborated the full project. Each group had a designated internal academic *supervisor*, who monitored progress during the whole project and helped with issues. Moreover, a *panel of supervisors* gathered for start and end of each iteration.

Over the course of 2 semesters, there were 7 *iterations* in total. Each iteration was 3–4 weeks long. At the beginning of each iteration, each student group met with the panel of supervisors to promise the workload for next iteration. The panel would consider weight of user stories and decided on workload for each student. Every week during the iteration, the team met with their supervisor for a *weekly standup meeting* to discuss progress and issues faced. They are not done daily, as that was judged not practical for undergraduates. At the end of the iteration, a *presentation* of the team's work to the panel of supervisors, where their work was evaluated, based on criteria such as individual contribution, team contribution, velocity, oral presentation,

and completeness of the product and research understanding. The first iteration of the first semester was for gathering requirements and documenting them as User Stories. In the last iteration of each semester, an integrated and tested working product was required. At the end of each semester, students conducted a *semester retrospective* to discuss their failures and successes and were asked to document them.

There was a single artifact: the *product backlog*, an output of the first iteration. The product backlog was then referred to by students and supervisors throughout the project. Students had to continuously update the data of product backlog in an online tool of their choice.

Moreover, students were asked to always keep in mind and uphold Agile principles of the Agile manifesto (see Subsection 1.2.1).

**Results.** After the projects were over, students were given questionnaire on their opinions on appropriateness of several principles and practices they have experienced. Although the questionnaire itself was formed by inquiries in both nominal and ordinal scales, the reported results focus mostly on binary answers "Appropriate" – "Not appropriate". Judgment sampling method was used.

The authors mostly focus on correlations of the answers. To simplify and compare with other case studies presented in this chapter, we have extracted absolute frequencies in Table 2.2 from reported contingency tables; two rows do not differentiate between client-oriented and research-oriented project answers, as these could not be inferred from the data presented in the original report.

Some of the results have been commented by the authors. The highlights include:

- Scrum is judged to be appropriate for both client-oriented and research-oriented projects.
- Small teams are not as good for client-oriented projects as for research-oriented projects.
- More than 75 % judge short iterations and frequent progress monitoring as appropriate, irrespective of project type.

#### 2. Agile in Research

Table 2.2: Frequency table of student opinions at SLIIT, adapted from [54]

% of students who believe the practice was appropriate for their project	Client projects	Research projects
Scrum as a whole	86.7	72.3
Short iterations	85.3	75.4
Small teams (4 members)	78.0	86.2
Frequent progress monitoring	86	.4
Weekly standups	96	.2
Embrace requirement changes	79.4	86.2
Team communication	92.6	98.4

- As most are saying yes to weekly standups, daily standups are not deemed necessary.
- Irrespective of the team size, communication is important; this importance is, however, higher for research-oriented projects than the client-oriented projects.

**Discussion.** Overall, the results show a successful application of Agile. Authors included Agile manifesto as a reference which may have improved understanding of the teams and their decision-making ability. Furthermore, weekly standups are attainable from our experience for undergraduates, and most report their appropriateness.

Nonetheless, we dislike several things with the Scrum adoption. First, the projects look like a typical waterfall, with phases masked in iterations: they start with "documentation iteration" and end with "integration and testing" iteration, and so on. This means no working software products are potentially releasable each iteration, rendering students unable to deliver business value and gather real feedback continuously (cf. [28]) Another activity actively discouraging business value is the evaluation by velocity, discouraging transparency.

Self-organization, one of the key benefits of Agile, is similarly actively discouraged by several factors such as task estimates not done by students, too frequent reporting to supervisors, and evaluation of individual contribution.

#### 2.2.6 Kaiserslautern's XP Seminar

Extreme Programming (XP) was the first agile framework, but wanes off use today (see Section 1.5). Some of the critiques of XP in the industry include it being software engineering centric, illogical to non-developers, high prescriptive in practices, and requiring high investment in training developers to be effective using XP. [55]

The final case study traces how a practical, project-oriented Extreme Programming course was taught in Germany in 2002/2003, as described in book entry *Agile Methods in Software Engineering Education* [56].

**Background.** Practical seminar at University of Kaiserlautern, Germany, with 11 master and doctoral students, spanning four weeks, eight hours of work a week.

**Motivation.** Teach XP and provide hands-on experience working with XP. Document feasibility of running a student project with XP, namely test whether novice developers can handle an XP process.

**Adaptation.** All traditional XP roles have been assigned to students: developer team had 8 members, two students took the role of on-site customer, and one student was assigned the role of a tester and tracker. Supervisors acted as an advisory board. As much responsibility and freedom, in accord with XP, has been left to students; supervisors intervened only rarely in cases of serious problems, such as misunderstanding of a practice or infrastructure problems.

On-site customer interacted with developers in the planning game and with the tester in writing acceptance tests. New pairs in pair programming were created daily. Code was commonly owned and integrated several times daily. Adapted programming standards were used. Testing was done manually. Simple design and small increments were encouraged.

During the four week project, there were three releases of working software.

**Results.** A full-day tutorial, teaching students the basic principles and practices of XP, has been performed at the start of the seminar. After it, students themselves developed their systems for planning and recording time needed and spent in their projects.

Data sources were then twofold. First, students prepared their "Lessons Learned" which they then presented to the group and all voted for or against them to assess whether they are only of personal nature or if they reflect opinions of others. These feedbacks were rather practical, with examples including items such as guessing of requirements, improvable communication with customer (due to him being in other room), or slight changes to user stories.

The second data source were observations of the authors (teachers). These, once again, revolved mostly around technical aspects of the process

The pass ratio of the intended acceptance tests was rising from release to release as the students became more experienced with XP and improved their estimation skills. Although functionality was implemented, the final system was of low quality (as gauged by authors), as other attributes were neglected, mainly maintainability and usability. Little refactoring took place as the developers were eager to implement new User Stories.

The authors note two major problems with the process. First, inexperienced teams struggle to give precise estimations for tasks. Second, undisciplined and/or inexperienced development teams tend to ignore process restrictions and requirements. As authors of [56] note, these are not specific to XP, but are amplified by agile process characterized by lack of guidance.

Authors conclude that although XP is easy to teach, it is much harder to perform a project with XP successfully.

**Discussion.** It can be seen that the results focus primarily on technical aspects of software development, touching less on the work organisation aspects which we nowadays understand as core of agile frameworks. This is in line with how XP was conceived (see Section 1.5).

Similarly, it was long presumed that Agile requires highly disciplined and senior developers to be effective. Authors expect that before the study starts and unsurprisingly confirm it after the study ends. As this was all in the dawn of Agile, we believe this results from not sufficient understanding of self-organization and the knowledge worker motivations. Without discipline and senior developers on board, predictive models would not bring any better results. Nonetheless, an inclusion of a process-oriented support such as the role of a Scrum Master would go a long way.

Finally, it can be inferred that overall, students again did not dislike the Agile experience, but such an inference is not adequately corroborated with data.

## 2.3 Summary of Case Studies

The presented research-adapted frameworks draw from several software engineering agile frameworks (such as Scrum and Kanban), in varied amounts of "lightness" (i.e. how prescriptive they are), and have been implemented in a wide range of types of research groups and initiatives.

Most authors of the explorative studies of application of Agile in research propose that integrating agile methods and practices for interdisciplinary collaboration requires significant flexibility. [2, 15] Agile may be thought of as a large set of practices, some of which work well for a specific team by combining them creatively. [16] Nonetheless, some teams adopt the frameworks whole, not bending them without enough experience and understanding, often for superior results. [48]

Key aspects of the case studies presented in this chapter are summarized in Table 2.3. The results and improvements are varied and of varied intensities, but studies universally seem to be generally positive (also cf. Subsection 2.1.3).

It is hard to compare the results on general terms, as there are as many agile processes as case studies. We can, however, rate the best performing processes in various specific aspects:

- Student-centric research group context: SCORE, txtUML, possibly PROMOBILE
- Education context: SLIIT, Kaiserlautern

- Simplicity: SCORE, CECAN, possibly txtUML after leaving out the hierarchical roles
- Rigorous approach: PROMOBILE, possibly SLIIT
- Low overhead: SCORE, txtUML
- Collaboration improvement: SCORE, CECAN
- Student satisfaction: SCORE, PROMOBILE, possibly txtUML
- Self-organization, decreased load on faculty: SCORE, txtUML
- Technical focus: Kaiserlautern, possibly SLIIT

Of course, many other scaled could be devised.

To choose a process or design a new one, it will be necessary to adequately study the context in which it will be applied and the primary benefits we will seek. Only based on such findings we can choose potentially adequate studies to serve as a baseline.

Name	Source	Primary demography	Elements	Primary results	Met. <sup>a</sup>	Advantages	Disadvan- tages
SCORE	Hicks; Foster (2010)	Doctoral students	2 events	Transparency, interactions, consistent work	* * *	Simplicity, effectivity	No Retro's
txtUML	Ilyés (2019)	Undergrad. students	5 roles, 4 events, 3 artifacts	Rapid knowledge flow	* *	Weekly meeting	Prescriptive, complex, hierarchical
CECAN	Hidalgo (2019)	Academic researchers	1 role, Kanban, 2 principles	Dynamic collaboration	*	Kanban usage	Ad-hoc approach
PROMO- BILE	Valentin et al. (2015)	Undergrad., master, and doctoral students	Full Scrum	Transparency, motivation, soft-skills	*	Rigorous process, incremental adoption	Possibly too large overhead
SLIIT	Manamen- dra et al. (2013)	Undergrad. students	3 roles, 4 events, 1 artifact	Student enjoyment, adaptability	* *	Weekly standups, focus on principles	Frequent reporting
Kaiser- lautern	Bunse et al. (2004)	Master and doctoral students	Full XP	Incremental deliver, not best for novices	*	Focus on technical excellence	No take on work organisation
a. Credibil	ity of methods ions, and "*" de	(as judged by us enotes a study w	), where "***" de ith sionificant da	enotes a study with ita-wise limitations	ı high da	ta quality, "**" der	notes a study wit

Table 2.3: Summary of Agile in research case studies presented in this chapter.

<sup>2.</sup> Agile in Research

# **3** Case Study: Agile in Laboratory of Service Science

The first two chapters have provided basic review of literature on Agile in general (Chapter 1) and how have applications of similar methods emerged in research initiatives across academia (Chapter 2).

We have not only seen that there are many heterogenous flavors of Agile, such as Scrum, Kanban, and Extreme Programming, but also that there are as many adaptations of each of these in academic settings as there are authors writing the case studies. Not only that, it also seems that there is no recognized universal set of guiding principles and/or elements from which the processes could be constructed (see Section 1.6).

Although the results vary, the trend in literature is clearly that introducing Agile *does* in fact improve several aspects of running a collaborative venture, such as engagement, efficiency of communication, productivity, and others.



Figure 3.1: Laboratory of Service Systems.

This chapter of the thesis adds another entry into the pool of case studies by testing whether we could reproduce some of the benefits of Agile, as inspired primarily by the SCORE framework (see Subsection 2.2.1) and process at txtUML team (see Subsection 2.2.2) after executing it remotely and slightly tailoring it to our context at *Laboratory of Service Systems* (LabSeS)<sup>1</sup> located in Brno, Czech Republic,

<sup>1.</sup> We will generally mean LabSeS when referring to a "laboratory" or simply a "lab" throughout the chapter.

under the administration of Faculty of Informatics, Masaryk University, in which I have interned in Spring 2020 and Fall 2020 semesters.

The first section provides detailed background of the laboratory, such as information about the affiliated university, its research focus, brief historical timeline, its personal composition, or industrial partners of the laboratory. Motivations behind our need for change are explored, providing reasons for the hypotheses stated in the Methods section.

The second section describes the methods used in sufficient detail for the reader to assess validity and reproducibility of our results. The hypotheses are stated and data gathering mechanisms are explained in succession. Finally, statistical methods used to decide determine success versus failure are elaborated.

The third section describes design considerations following from the laboratory background which have led to a proposal of an agile process which we believed would be beneficial for our laboratory, and details the proposed process itself. We also trace down how we have implemented it in the laboratory in 2020 and 2021.

The fourth section presents the results based both on our perceptions and on quantitative data we have gathered from perception surveys. We also test hypotheses as stated in Methods section. Our lengthy discussions with management are summarized.

Finally, the fifth section summarizes all previous sections and formulates answers to our hypotheses.

## 3.1 Background of the Laboratory

*Faculty of Informatics* (FI)<sup>2</sup> at *Masaryk University* (MU) employs over 280 staff members, out of which nearly 50 are professors and associate professors, and educates more than 2000 students at a given time. The faculty was founded in 1994 as the first informatics-only faculty in the Czech Republic. In addition to quality teaching, top research is carried out in almost twenty laboratories, often in cooperation with industrial partners of the faculty or companies based in the science and technology park directly on the FI campus. [58]

<sup>2.</sup> See https://www.fi.muni.cz/index.html.en.

Laboratory of Service Systems is one of these laboratories.<sup>3</sup> The main research topic of the laboratory is a concept of "Smart Cities" – cities equipped with non-trivial implementation of information technologies improving efficiency of processes within the city, its habitability, and sustainability of its development. [60]

The laboratory, in pursuit of multidisciplinarity, cooperates with many partners at MU and foreign academic institutions and with industry partners in both private and public sectors. Internal academic partners include Studio of Graphic Design and Multimedia, Faculty of Economics and Administration and Faculty of Social Studies; external comprise of faculty renowned in the field of Service Science of Universities of Salerno, Naples, Bucharest, and Paris; municipalities in Czech Republic collaborating on specific Smart City service products include Žďár nad Sázavou, Dačice, Ořechov u Brna, and Lelekovice; and finally, ATTIS software s.r.o. is our leading private sector partner. [59]

Diploma theses include, among others, the open data portal, dashboards, web marketing and the citizen portal. Other, more methodological theses then generally deal with, for example, the Smart Cities strategy or implementation processes. [59]

The scope of the laboratory was previously broader. Smart Cities were just one section of three. Second section, "ERP Systems", studying various implementations and applications of integrated software for planning resources such as *iDempiere* or *Microsoft Business Central*, is currently frozen. Finally, the third section "Data and Service Quality" focused on various aspects of data, has recently been disbanded due to its leading faculty resigning.

#### 3.1.1 Demography

Core of the laboratory is formed by bachelor, master, and one or two doctoral students at FI MU. Although members fluctuate rapidly<sup>4</sup> as they quit academia and enter the workforce or move abroad in an academic mobility program, the size of the laboratory has been

<sup>3.</sup> See https://seslab.fi.muni.cz/.

<sup>4.</sup> Median membership length is two semesters. This is typical for undergraduate student research initiatives (cf. txtUML team in Subsection 2.2.2).

steadily growing. As of Spring Semester 2021, there are two active bachelor students, ten master students, and one doctoral student.

All students writing their theses under supervision of Ing. Leonard Walletzký, Ph.D. are expected, by design, to engage within the laboratory. The level of engagement varies among students, but was universally very low as of Spring semester 2020. The typical student member spends 5 - 8 hours weekly at the laboratory, counting in the actual work on his or her final thesis. The number of hours is not, however, compulsory.

A special type of laboratory engagement is research internship, a part of some study programs at FI. The author of this thesis was an intern in Spring 2020 and Fall 2020 semesters and continues to participate actively, as the laboratory provides basis for this master thesis and, hopefully, one of more case studies for the following dissertation.

Since Spring 2020, there have been, on average, two master students interning at each given moment. The commitment to the laboratory for an intern works out to be about 18 - 20 hours weekly for two semesters, not including time spent on writing a final thesis if he or she is working on one (but all interns have been, as of this day). The internship at LabSeS is typically comprised of the following three broad categories of activities:

- managerial and administrative tasks, such as organizing meetings and municipality visits, facilitating meetings, or managing laboratory-wide task-lists;
- 2. **technical support**, helping with implementing and maintaining technical solutions such as *Microsoft Excel* [61], *ATTIS* [62], or *VMware* [63], and supporting other laboratory members in their usage; and
- 3. **research**, an overarching category serving as a space for all both collaborative and independent research activities for interns who wish to gain deep understanding of the Smart City concepts and literature.
#### 3.1.2 Recent History

The laboratory was founded in 2008 by RNDr. Zdenko Staníček, Ph.D., a key persona of Faculty of Informatics. In 2013, Ing. Leonard Walletzký, Ph.D. has took over management of the laboratory. As both the name and the history imply, main focus was on Service Science – the change to Smart Cities came gradually around the year 2016. The name was not changed, however, as the topic is still largely viewed from the services viewpoint.<sup>5</sup>

The important history starts around early 2020 with a large personal growth of the laboratory, demanding new and more adaptive approaches to organisation and management (see Figure 3.4 in Section 3.4). Moreover, we have gone into a full-remote way of working with outbreak of COVID-19 pandemic.

**Vision and Strategy.** It was imperative to create strong, explicit overarching direction and goals, which would unify the laboratory, before any meaningful change was to occur. For that, we have defined a Vision statement, i.e. a coherent description of long-term ambitions of the whole system, normally expressed through a single sentence. Vision statements are key drivers in commitment and motivation of members and competitive success. [64]

As designed, our Vision states:

"The goal is to build from the Laboratory of Service Systems in 4 years a renowned facility, collaborating with municipalities and companies dealing with Service Science (on the fields of Smart City and ERP systems), and assure lasting transfer of research results to applications and practice, including adequate grant financing." [59]

To further foster culture of teamwork, explicit "Values and Rules" were defined:

"We cultivate team collaboration of all laboratory members, always seeking synergies, how to improve our work

<sup>5. &</sup>quot;The general aim of the Laboratory of Service Systems (LabSeS) is to study possibilities of services provision and their IT support (which is similarly provisioned as a service), so that they bring maximal value to their users. [...]" [59]

together. Bi-weekly meetings of the whole laboratory, during which we discuss such opportunities, help assure that. We follow the motto 'Success of an individual is the success of the laboratory and vice versa.' [...]"

The first step towards an unified laboratory, under whose name its members would be proud to publish, was communicating these verbalizations within the laboratory. We tried to create a compelling vision and values with which students could identify, to possibly reach a buy-in of the team.

**Tool Support.** The second element still missing in the laboratory was standardized toolchain, which would support communications and endeavors in the laboratory. As can be seen in e.g. Subsection 2.2.3, tooling is a critical element, which, if inadequate, may render many of Agile benefits futile. The list of software tools we have adopted follows.

- 1. **Intranet.** Previously, file-sharing and group chat capabilities of *Microsoft Teams* [65] were used as intranet. However, this solution was unscalable. We have thus migrated it in February 2020 to *Microsoft Excel* [61], a more robust tool with support of matrix user permissions, widget creation, wikis, and more.
- 2. **Group calling software.** We have kept the *Microsoft Teams* [65] for group video-calling. Nonetheless, we have made it much leaner by reducing the number of different Teams to two one for all members and one for management only and removed all functionality other than chat and call.
- 3. Virtual Machines. We have used *VMware* [63] before for administration of our virtual machine farm, and continued to do so. The difference was again in standardization (naming conventions, explicit responsible person, wikis in intranet, ...) and slimming down (several machines with arcane names and only legacy applications were shut down).
- 4. **Presentation.** Laboratory had minimal external marketing and presentation, mostly in the form of a subpage of webpages of

one study program at FI and promotions in courses led by Ing. Leonard Walletzký, Ph.D. In spite of this, it was growing (see Subsection 3.1.3). However, this growth with inherent responsibilities warranted a better and more formal presentation. For that cause, our webpages https://seslab.fi.muni.cz/ [59] were created. They contain description of the laboratory, its history and research foci, lists of publications and theses, contacts for all members, a newsfeed, and more.<sup>6</sup>

Key attributes of the tools used are summarized in Table 3.1.

Tool type	In 2019	In 2020+	Actions	
Intranet	<i>Microsoft Teams</i> [65]	<i>Microsoft Excel</i> [61]	Migrated	
Group Calling SW	Microsoft Teams [65]	Microsoft Teams [65]	Removed redundancies, scheduled calls	
Virtual Machines	VMware [63]	VMware [63]	Standardized naming, created wiki in intranet	
Presentation	None	Webpages [59]	Created from scratch, newsfeed	

Table 3.1: Tools used in Laboratory of Service Systems.

## 3.1.3 Motivations for Change

The laboratory connects academicians and students sharing research field of interest in Service Science and Smart Cities. Nonetheless, our goal was more than "just" connecting people and aggregating their research results under an unifying name: the goal for the laboratory was *to be itself the collaborating facility*, i.e. to produce research itself in such a way that the outputs are superior to the sum of individual research initiatives (cf. Subsection 2.1.2). We have found that laboratory has had mostly star-shaped team topology centered around person of Ing.

<sup>6.</sup> Furthermore, the laboratory was getting marketed and promoted in a much more thorough and systematic manner since 2021, such as on Open days (see [66]), on conferences, trade fairs, social media, and so on.

Leonard Walletzký, Ph.D.: several students have been writing their own solo theses and minimalized their participation in the laboratory, which they regarded only as a necessary evil to receive the advisory. The potential value of the laboratory was thus compromised.

The Figure 3.2 represents the original topology. Factory and City symbols stand for laboratory business partners and partner municipalities, respectively. Lines display collaboration.



Figure 3.2: Star team topology (original).

As the laboratory was steadily growing over the years from a small group to over 15 core members not including external collaborators (see Section 3.1), the time and attention requirements placed on the laboratory head grew accordingly. To publish consistently and to manage several students writing their theses is a difficult job, but publishing, advising over 10 students, and collaborating with several

private and public sector stakeholders is nearly impossible to manage. The sheer amount of work has put strong constraints on time schedule of Ing. Leonard Walletzký, Ph.D., as well as made it hard to provide individualized feedback in adequate amount and quality.

We thus had to find a new way of managing the research laboratory which would foster collaboration among its members, engagement and enthusiasm to work *for and within* the larger laboratory rather than independently, and distribute part of the workload from the central element among the laboratory. Not only that, this would improve collective creativity and enable knowledge transfer (see Section 2.1). Thus, the desired laboratory team topology was a much more enmeshed network, as depicted in Figure 3.3.



Figure 3.3: Network team topology (desired).

#### 3. CASE STUDY: AGILE IN LABORATORY OF SERVICE SCIENCE

However, this task would not be so simple. Perhaps the most elementary reason was a very low motivation of students to engage anymore than necessary to graduate. Proactivity in choosing more projects or collaborating was virtually nonexistent. Laboratory held all-hands-on meetings, but irregularly, and a significant portion of the laboratory had always "other obligations" to tend to, as participation was not compulsory.

Furthermore, as there was this minimal motivation to start, the laboratory has been struggling to win projects and grant programs, which would allow for the work to be executed collaboratively. This has proven to be a self-fulfilling cycle.<sup>7</sup> We thus wished to break out of the cycle by finding possible synergies enabling collaboration with what we had at hand to begin with, i.e. within the advised theses.

Indeed, there were repeating themes across the works which we could uncover and leverage. For example, technologies used, such as *ATTIS*, *CKAN*, or *Metabase* were prominent in several theses. Similarly, strategical partners of the laboratory such as specific municipalities often collaborated on a sequence of theses, rather than on a single one.

It thus became a priority to create an environment which would support the laboratory members in finding such synergies and motivating them to execute on them. The benefits would be profound and two-fold: first, there will be more informal knowledge transfer, improving motivation and satisfaction; second, workload will be lifted off Ing. Leonard Walletzký, Ph.D. as the single point of failure, a requirement we started off with.

## 3.2 Methods

At Laboratory of Service Systems, we are interested in testing whether we can reproduce, in our context, the following primary results of SCORE study (cf. Figure 2.1) by designing our own agile process:

1. improved perception of student-student interactions;

<sup>7.</sup> This happens when there is a positive feedback loop (i.e. amplifying itself) of a negative outcome. In this case, when we do not have enough motivated personnel to execute on grant programs, we do not win such programs. This further diminishes the appeal of such laboratory work and amplifies the lack of human resources.

- 2. improved advisor schedule and availability; and
- 3. improved productivity of meetings with advisor.

Various other process improvements will also be measured.

The data of the case study come from the following two primary sources:

- 1. personal perceptions of the author and laboratory head, and
- 2. on-line surveys having both quantitative ordinal Likert items and free-text sections.

As I have been performing a research internship in the laboratory with the agile transformation as one of the goals, we have closely interacted on an informal basis with the laboratory head and advisor of this thesis, Ing. Leonard Walletzký, Ph.D., throughout my stay.

For measurement of attitudes and opinions, Likert items and scales are one of the most popular choices nowadays since its conception in 1932 by Rensis Likert. *Likert items* present statements or factors to be rated by respondents on an ordinal scale, such as "Disagree", "Neutral", and "Agree". *Likert scales* are then combination of results of all of the related Likert items, most commonly as a simple sum or average after encoding the answers as numbers. This, however, requires us to make not easily tenable assumption that all items have same weights. [67, 68, 69]

The described scale is a representative of an *ordinal* type of scale. Ordinal scales allow for ordering, but not comparing degrees of difference. <sup>8</sup> There is no attempt to make the intervals of the scale equal in terms of some rule. For example, statements "*Excellent* student is three times as good as a *Not bad* student" or "*Rarely* and a half" make no sense at all (cf. Subsection 2.2.2). Thus, the real differences between adjacent ranks may not be equal. [67, 68]

Since the numbers of this scale have only a rank meaning, the appropriate measure of central tendency is the median. Arithmetic

<sup>8.</sup> To compare, less number-like are *nominal* scales, i.e. categorical only, where any numerical encoding would be name only and best central tendency is the mode; while the next more number-like are *interval data*, allowing for differences and arithmetic means, and *ratio data*, allowing also for ratios and geometric means.

mean would make no formal sense, as well as the corresponding parametric tests (such as t-test). One of the most appropriate tests is the *Wilcoxon-Mann-Whitney U test*. The justifiability and robustness of "forgetting" the non-normal origin and using parametric tests on large enough samples is much debated in the literature. [68] We will err on the side of caution with the more rigorous approach of U test.

On-line platform *Survio* [70] was used for creating the surveys and gathering the results. The statistical analysis and graph generation was performed in the statistical software R [19].

## 3.3 Agile Transformation

The description of the agile process as we have conceptualized it based on SCORE framework (see Subsection 2.2.1) and process at txtUML team (see Subsection 2.2.2) follows, including the inputs we have taken into account when designing it and the actual implementation process.

### 3.3.1 Design Considerations

When selecting case studies from Chapter 2 as inspiration for Laboratory of Service Systems, we have used these three main criteria:

- 1. simplicity of the process;
- 2. improved student motivation and decreased load on faculty as primary results; and
- 3. research laboratory context with undergraduates as primary demography.

As follows from the comparisons made in Section 2.3, SCORE and processes at CECAN and txtUML best address the first point, SCORE, txtUML, and PROMOBILE best address the second point, and both SCORE and txtUML best address the third point. The choice was clear: synthesize SCORE and txtUML processes (as described in [15] and [16]) and tailor them to best fit our context.

The SCORE framework is based around frequent status reports, scheduling technical discussions only on demand, and creating sense

of community and a flat hierarchy, as described in Subsection 2.2.1. The benefits of increased flexibility of faculty schedules and increased insight into works of others is exactly what we seek. However, we will need to down-scale the status reports, as meeting thrice a week would not be attainable in our context, where undergraduate students often engage just a few hours a week.

For the txtUML process, as described in Subsection 2.2.2, we have identified several major limitations with its conception of roles and artifacts. Nonetheless, we highly appreciate how they approach meetings with the weekly meetings divided into several phases, which, when used together, create a better and more comprehensive report of current status. Similarly, usage of a Retrospective is seen as very adequate for an agile process, though we would welcome Retrospectives being a bit more frequent.

#### 3.3.2 Agile Process Proposal

The process we have constructed has one scheduled event and one on demand event, accepts three roles we have started with but defines only a single new role, and recognizes no artifacts. The events are constructed to enable synchronization and tracking status between all lab members (as opposed to pairs advisor–student) and optimize time commitments to technical advisory. We did not specify any artifacts not only to keep the framework lean, but also due to the inherent heterogeneity, uncertainty, and novelty of research (see Section 2.1), rendering such constructs somewhat arbitrary in our largely nondevelopmental context.

**Biweekly Sync.** This is the only regular event we hold, with the goal to update everyone on current situation and synchronize. It occurs each second Wednesday between 5:00 PM and 6:00 PM and all laboratory members are expected to, but not required to attend. It is conducted through *Microsoft Teams* with everybody having his or her camera on to simulate face-to-face interaction. The event is often facilitated by the Scrum Master (author of the thesis, see later) and has the following agenda.

#### 3. CASE STUDY: AGILE IN LABORATORY OF SERVICE SCIENCE

- Laboratory News. All news that are important to the group as a whole are presented, typically, but not exclusively by Ing. Leonard Walletzký, Ph.D. The section typically takes between 10 – 15 minutes.
- 2. **Scrum.** Taking turns, all members answer the following questions:
  - "What did I do over the past two weeks?"
  - "What am I going to do over the next two weeks?"
  - "Is there anything that impedes or blocks my progress?"

The Scrum is timeboxed to 15 minutes to encourage all contributions to be as high-level as possible. This helps the members to focus on key points and speak in non-technical way so that everybody can follow their progress. Moreover, it encourages the students to prepare their entry and distill the most important aspects before speaking. If an issue worth discussion arises, it is either postponed after the meeting if of manageable estimated complexity (say, 30 minutes), or an on-demand meeting is scheduled.

- 3. **Retrospective.** The last formal part of the meeting is discussion about the process, team, communication strategy, tooling, or anything else worth addressing from the "how we work" viewpoint. We have expected up to 20 minutes in this part, though it has been very often neglected as we were unable to provoke enough participation.
- 4. **Closing.** Finally, we thank everybody for joining and end the meeting. If there were issues identified during Scrum worth addressing at additional, but manageable detail, they are discussed in a smaller virtual rooms.<sup>9</sup>

<sup>9.</sup> Disinterested parties leave, possibly to a parallel room if they have their own issues at hand. This is not to waste each other's time.

**On-demand Technical Meetings.** All issues that require deeper and more focused discussion get an on-demand meeting scheduled, with only a small number of attendees involved with the issue. These are most often, as expected, the standard one-on-one advisory meetings between a student and Ing. Leonard Walletzký, Ph.D. However, many such meetings have also emerged as meetings of two or three students themselves, discussing a specific topic, helping themselves, or skill-sharing.

**Roles.** We define no new roles than what we have started with: standard members, interns, and faculty. We try to bring everyone on the same level within the laboratory (cf. Section 1.3), so that any difference between, if any, is implicit. The full palette of academic titles and ranks all members come to the laboratory with, we presumed, is discriminating enough as it is.

The role of *faculty* (or *laboratory head*), as played by Ing. Leonard Walletzký, Ph.D., is the functional responsible for the laboratory as well as the only one with the formal authority to advise students in writing their theses. To limit discrimination,<sup>10</sup> I have advised him to act as much as a standard member in the meetings as possible, such as reporting in the Scrum as well, not immediately reacting to all issues presented from authority standpoint, and so on.

The role of *interns* is that of support service to others and has been thoroughly explained in Subsection 3.1.1.

Finally, my part as an author was regarded as a new role of a *Scrum Master*, as I was proposing, implementing, and facilitating the new process. Furthermore, the role had the standard responsibilities as described in Scrum: teaching Agile, facilitating meetings, coaching others in non-technicalities, and facilitating the removal of impediments. However, the influence of the role quickly faded out as our group quickly became self-organized and self-sufficient, not in need of a "designated" facilitator.<sup>11</sup>

<sup>10.</sup> Significantly distinguishing the role inhibits students' self-organization and inclination to report transparently on their problems and failures, among others.11. As the greatest Scrum Master is the seemingly unnecessary and dispensable one. [71]

**Other elements.** Inspired by both the SCORE and txtUML reports, we searched for ways to further foster collaboration and sense of community throughout the group. As co-locating the office or common restaurant visits were impossible due to the COVID-19 situation, we had very limited options. Nonetheless, we were able to leverage a *Thesis Presentation* for good results.

During the 1 - 2 hour long workshop, all members get a time-slot to present their thesis or work-in-progress and then receive feedback both on the content and delivery from everyone else. Students ready to graduate get a longer time-slot of about 25 minutes to mock the real thesis defense, while more junior students get a shorter slot to present what they have been working on, experimenting on, or reading, even if it is not yet in a form of a coherent writeup.

#### 3.3.3 Adoption

Once we had vision, strategy, and tooling in place (see Subsection 3.1.2) and the first survey done (see Subsection 3.4.1), we were ready for the transition. This aligned very well with start of Fall 2020 semester, the new process was thus adopted at the time, i.e. beginning of November 2020 – see Figure 3.4. The adoption here generally refers to explanations of changes and scheduling of the regular meeting.

Our motivations and reasons for change were explained during the first meetings of the semester. Afterwards, all members were instructed in a series of short sessions about the new process. The sessions were kept light-weight and emphasized direct communication as the primary way how to get information or solve problems. However, the amount of instruction in total did not exceed about two hours, as we did not wish to have results skewed by some expectations of what "should happen."

At the same time, the regular biweekly meetings were scheduled. They were still not made compulsory, however their importance in building collective excellence was emphasized on every occasion. Furthermore, it was communicated as an essential element of receiving thesis advisory, and if one wants to receive adequate quantity and quality, one should respect the common schedule.

The meetings were running by the agenda described, but we quickly found we have very limited inputs for the "News" and "Retrospec-



Figure 3.4: Timeline of the Laboratory of Service Systems.

tive" items. The first quickly degenerated into status report of Ing. Leonard Walletzký, Ph.D. We have not addressed this anymore, as it became apparent that his actions do in fact typically involve most of the laboratory and thus fit the section.

The second point, routine skipping of "Retrospective" after a few runs in which we were unable to facilitate any inputs, needed to be addressed. We agreed that we do not interact as a group often enough to have any meaningful inputs, and chose to have one full-fledged Retrospective on 7<sup>th</sup> April 2021, just as they did at txtUML.

## 3.4 Results

The framework, as we have described it, is conceptually very simple, thus we have not expected too much. The original SCORE report did present an unambiguous improvement in all measured aspects, even if they were partially skewed by the way of collecting the ratings, and the txtUML report was also generally positive. As we had much lower commitment of laboratory members and had no face-to-face interactions nor community buy-in to begin with, we were skeptical.

However, the observed effects were massive. Students quickly embraced self-organization and proactively sought out resources and capacities which would help them solve their struggles. They started identifying synergies between themselves and organized discussions between themselves naturally, without guidance or presence of us. Furthermore, the regular reporting of progress remarkably improved the group's accountability, as it was necessary to report progress in an incremental fashion, if one did not wish to give off an impression of slacking.

The product as observed by the laboratory head was no less impressive. From the time-saving perspective, schedule of Ing. Leonard Walletzký, Ph.D. has became much freer, as it became apparent that the need for in-depth technical advisory meetings was much lower than previously understood – the regular collective biweekly cadence supplied enough synchronization not to require regular one-on-one cadences. This allowed for better preparation for advisory meetings (including the fact that meetings each now had pre-agreed, specific purpose), driving up the quality of the meetings as well as decreasing stress and cognitive load of frequent context-switching. More available time also translated to larger buffer for acquisition of new projects and continued laboratory growth. As per Ing. Leonard Walletzký, Ph.D. own words, "I still have little time, but without Agile, it would all be utterly impossible."

From the work quality perspective of our advisees, we have also observed a big shift. The regular cadence enabled much shorter timeframe between a student struggling and its identification by us, while the freer schedule enabled shorter timeframe between this identification and scheduled problem-solving technical meeting. Finally, to Ing. Leonard Walletzký, Ph.D.'s delight, there was a clear distinction between previous theses presentations he has been part of and those which had mock presentations delivered in the Thesis Presentation workshop. Furthermore, all of these benefits were virtually free. After the initial launch, which required some capacity of the Scrum Master, the train continued to accelerate by itself, quickly phasing the role out.

#### 3.4.1 Student Survey Design

We have chosen to administer a survey first to assess the original situation before we would start implementing any changes. The survey was inspired by, but not identical, to the survey administered in the original SCORE report. This was designed as such to allow for a high degree of comparison between our and their studies, while allowing us to draw out some additional information appropriate for our laboratory.

Students<sup>12</sup> were asked in May 2020 to think about the last month before going fully remote due to the emergent COVID-19 pandemic, i.e. about February 2020 (please refer to Figure 3.4). At the time, we had no idea how the situation will develop. During the survey, 5 student lab members out of 7 participated.

We have not really escaped this predicament up to this day. As we were unable and also mostly highly unwilling to have a face-to-face meeting since March 2020, all Agile had to be consequently conceptualized, executed, and evaluated in the virtual (remote) world. The second survey in April 2021 thus asked about the remote work in the past three months in 2021, i.e. about January through March 2021. During the survey, 8 student lab members out of 10 have participated. One of them was not writing a final thesis, thus questions about thesis were filled out only by 7 respondents.

The survey was administered in Czech language in the first run; the second run was translated to English. We have also increased the level of detail in a few of the questions – see two new questions in Figure 3.9 and greater detail in Figure 3.14.

The survey was short (mean fill-out time was 3 minutes and 34 seconds) and anonymous. It contained rating questions, choice questions, and free-text; however the free-text questions were mostly left unanswered. The few answers we got in the first survey mostly re-

<sup>12.</sup> Three collaborators from Faculty of Economics and Administration also participated in the first survey run. However, it became apparent from their answers that they do not identify as laboratory members. As the thesis focuses mostly on internal affairs of the laboratory, the answers were subsequently discarded.

volved around laboratory head not having enough time and his long response rates.

#### 3.4.2 Thesis Sentiment

The results for the thesis questions in the first run of the survey, depicted in Figure 3.5, in which students were asked to rate quality of interactions with and availability of their advisor, fellow students, and other consultants, and various attributes of their theses, were mostly in line with our observations and expectations: productivity and motivation had lowest medians at "Not bad", while whiskers of quality of interactions with advisor and their frequency bottomed out at "Very poor" ratings, based on the inundation of the advisor stemming from the large laboratory and overhead. However, the comparatively high perceptions about availability and quality of interactions with fellow students surprised us.



Figure 3.5: Boxplot of thesis sentiment. May 2020, n = 5.

Although we expected some improvements in the second run of the survey, as we have seen improvements across many of process aspects and gathered some informal feedback, the real amplitude of sentiment difference surprised us when comparing the answers. Not a single median was lower in 2021 than in 2020 – see Figure 3.6 and Figure 3.7 – closely resembling the results of the SCORE report, what we did not think would be possible attaining. The largest differences observed were with quality of interactions with advisor and with motivation to write.



Figure 3.6: Boxplot of thesis sentiment. April 2021, n = 7.



Figure 3.7: Differences of medians of thesis sentiments.

#### 3.4.3 Laboratory Sentiment

To compare, perceptions about functioning of the laboratory were less varied, mostly converging around the medians.



Figure 3.8: Boxplot of laboratory sentiment. May 2020, n = 5.



Figure 3.9: Boxplot of laboratory sentiment. April 2021, n = 8.

74

We have not included the first two questions – rate how lab helped with increasing your skills and quality of your thesis – in the first survey, thus the missing data in Figure 3.8.

All of the questions had higher medians after the Agile transformation, with the largest difference of 2.5 points in the overall experience – see Figure 3.9 and Figure 3.10. Once again, these differences were by far out of expected proportions.



Figure 3.10: Differences of medians of laboratory sentiments.

#### 3.4.4 Optimal Meeting Frequency

We have inquired students about how often they would optimally like to meet, with results shown in Figure 3.11. The answers before the implementation had covered all choices between once a week to once per semester. It was obvious that these answers were based on the lower motivations to engage and we could not take them into account too much when designing our process – they clearly were not advising us on attainable meeting frequency.

This phenomenon can however be explained by the simple fact that students were asked such questions much earlier than they were instructed what benefits more frequent synchronizations could bring, and thus saw no value in them.



Figure 3.11: Preferred meeting frequency. May 2020, n = 5.

However, the new opinions from April 2021 in Figure 3.12 about preferred meeting frequency showed completely different data. The implemented biweekly frequency had overwhelming 75 % representation, as can be seen in Figure 3.12, while the only remaining 25 % all asked for even more frequent meetings (!), all without us setting the attendance compulsory. This suggests that either we have planned, executed, and facilitated the meetings very well, or that the mere consistency and rhytm was beneficial.

All in all, the answers for this single question backed up our hopes that students enjoyed working in the new process.



Figure 3.12: Preferred meeting frequency. April 2021, n = 8.

#### 3.4.5 Engagement

Perhaps the most impressive of all the sections of the survey were the interests of students in participation in laboratory activities. Where before none were interested in advising fellow students, authoring a paper, or participating in lab presentation (see Figure 3.13), we now had 29 % of respondents advising fellow students, 57 % co-authoring or interested in co-authoring a paper, and whopping 86 % of respondents interested in helping present the laboratory, as presented on Figure 3.14. Especially the latest fact of interest in presenting laboratory was not yet leveraged, left until now only to Ing. Leonard Walletzký, Ph.D. and interns.

In the first survey, we did not differentiate between "interested in" and "participating". This more detailed choice was added only later, when we saw many students engaging and curious.



Figure 3.13: Participation or interest in different activities within laboratory. May 2020, n = 5.





Figure 3.14: Participation and interest in different activities within laboratory. April 2021, n = 8.

#### 3.4.6 Summary of Results

As we established in Section 3.2, we will be using Wilcoxon-Mann-Whitney U test to test whether answers in the first and second surveys had different distributions. Furthermore, we are interested only in testing whether we have in fact improved (as medians imply), thus we will be using single-tailed variant of the test.

The Table 3.2 summarizes tests of all previously presented results. In the "Engagement" group we have not differentiated between "participating" and "interested in", rather we have grouped these together and tested against "not interested."

The distributions of quality of advisor interactions, advisor availability, quality of thesis, productivity, motivation, and overall experience of writing a thesis, as well as overall experience with being a member of the laboratory differed statistically significantly. Furthermore, inclination to co-author a Smart City paper differed statistically significantly and to represent laboratory differed highly statistically significantly.

Factor	Group	$\mathbf{D}\mathbf{M}^{a}$	Signif. <sup>b</sup>	p-value
Advisor Quality	Thesis	2.0	*	0.011
Advisor Availability	Thesis	1.0	*	0.042
Student Quality	Thesis	1.0		0.398
Student Availability	Thesis	1.0		0.274
Consultant Quality	Thesis	1.0		0.151
Consultant Availability	Thesis	0.0		0.242
Quality of thesis	Thesis	1.0	*	0.015
Productivity	Thesis	1.0	*	0.035
Motivation	Thesis	2.0	*	0.033
Overall	Thesis	1.0	*	0.044
Transparency	Laboratory	1.0	•	0.051
Fulfillment of Expectations	Laboratory	1.0		0.053
Overall	Laboratory	2.5	*	0.020
Write thesis	Engagement	_		1.000
Advise fellow students	Engagement	_		0.147
Get involved with municipalities	Engagement	_		0.088
Smart City paper co-authorship	Engagement	-	*	0.041
Different paper co-authorship	Engagement	-		0.147
Participate in lab presentation	Engagement	-	**	0.007

Table 3.2: Statistical significance of improvement in attitudes between May 2020 and April 2021 in accordance to various factors.

*a*. Difference of medians

b. Significance level (single-tailed) of different distributions

## 3.5 Discussion

We have been able to statistically significantly improve perceptions of student-advisor interactions, which corresponds to schedule observations of Ing. Leonard Walletzký, Ph.D. himself. However, we were not able to improve student-student interactions. This is probably due to

the fact that these were rated unexpectedly high even in the first run of the survey.

Furthermore, many other aspects of Laboratory of Service Systems have improved due to, or in parallel with, the new agile process, such as motivation and productivity during thesis writing, as well the overall experience. We have also succeeded in motivating students to co-author papers or participate in laboratory presentation.

Overall, the results were a magnitude larger than we have expected, while the changes to our work were relatively minor: meet a bit often, have a facilitator, observe the process. As maintenance of the process has only low overhead and we are now used to it, we do not see any benefit today in changing or discontinuing the observance of the process.

## Conclusion

The goal of the thesis was to review contemporary research on the application of agile frameworks originating in the Software Engineering industry and to implement a tailored agile process at Laboratory of Service Systems, Faculty of Informatics, Masaryk University.

After the necessary introductions to Agile in the industry provided in the first chapter, we have analyzed and compared six case studies of the application of Agile in academic context: SCORE at Maryland University, Agile Methodology at txtUML, Distributed Scrum at CE-CAN, Full Scrum at PROMOBILE, Undergraduate research projects at SLIIT, and finally Kaiserlautern's XP seminar. These case studies captured the full flavor of contemporary approaches to Agile in academia, trying to solve many challenges therein.

Based on our findings, we have synthesized an agile framework to be applied in research initiatives with low compulsory commitments, as was the case of Laboratory of Service Systems. The requirements of such an environment are clear: be simple, lightweight, and enjoyable to engage in. Expected benefits of such a framework are to distribute load from a central element of the team topology throughout the group and to foster a sense of community, motivating the group to engage together and commit together.

The agile process was implemented at our laboratory and the results were measured by short anonymous surveys of student perceptions and attitudes two semesters after the adoption. Quantitative sections about thesis and laboratory attitudes were based on 5-point Likert scale items, and 12 out of the 13 questions had improved their answer medians. The largest median differences were observed in quality of interactions with the advisor by 2 points, motivation during writing of thesis by 2 points, and overall satisfaction with laboratory membership by 2.5 points.

Answers to these questions – quality of interactions with advisor, motivation, and membership satisfaction – had statistically significantly different distributions as per single-tailed Wilcoxon-Mann-Whitney U test, as furthermore had perceptions of advisor availability, quality of theses, overall writing experience, and affinities to co-author Smart City papers or actively represent the laboratory. On the other hand, we were not able to improve perceptions of student-student interactions, possibly due to relatively high ratings given even prior to the agile transformation.

The transformation was very successful. Further work at Laboratory of Service Systems could focus either on drawing into Agile all collaborators, such as other faculty or municipalities, or on uncovering more opportunities for synergy among team members in a more rigorous fashion so that students would have more incentives to cooperate and further distribute the load throughout the network of the laboratory. Furthermore, the same process could be implemented at one or more other research groups to confirm its effectiveness on a larger set of laboratories.

The fact that application of practices coming from Agile, such as forcing periodic group meetings, gathering feedback, leveling out the student-faculty relationship, and working on cadence, brings on such a wide array of improvements, such as improved transparency, teamwork, consistency, motivation, and communication dynamics shows excellent potential for future use of Agile in academia.

However, what Agile truly is, in which contexts it is beneficial, and what aspects of it drive its success remain largely unknown. Future works should also focus on systematization of the ample amount of case studies that can be found in literature, such as on categorization of processes, listing of possible elements of Agile, and standardization of the forms of data acquisition and analysis. These results could then feed into multidisciplinary collaborations with Psychology, Sociology, and other disciplines to create unifying theories. These will, once the digital revolution is over, fondly remember Agile as the historical precursor to addressing common societal problems in the workforce.

# Bibliography

- WUCHTY, S.; JONES, B. F.; UZZI, B. The Increasing Dominance of Teams in Production of Knowledge. *Science*. 2007, vol. 316, no. 5827, pp. 1036–1039. ISSN 0036-8075, ISSN 1095-9203. Available from DOI: 10.1126/science.1136099.
- 2. HIDALGO, Enric Senabre. Adapting the scrum framework for agile project management in science: case study of a distributed research initiative. *Heliyon*. 2019, vol. 5, no. 3, e01447. ISSN 24058440. Available from DOI: 10.1016/j.heliyon.2019.e01447.
- 3. WANG, Jian; HICKS, Diana. Scientific teams: Self-assembly, fluidness, and interdependence. *Journal of Informetrics*. 2015, vol. 9, no. 1, pp. 197–207. ISSN 17511577. Available from DOI: 10.1016/j. joi.2014.12.006.
- SCANDURA, Alessandra. University-industry collaboration and firms' R&D effort. *Research Policy*. 2016, vol. 45, no. 9, pp. 1907– 1922. ISSN 00487333. Available from DOI: 10.1016/j.respol.2016. 06.009.
- JIROTKA, Marina; LEE, Charlotte P.; OLSON, Gary M. Supporting Scientific Collaboration: Methods, Tools and Concepts. *Computer Supported Cooperative Work* (*CSCW*). 2013, vol. 22, no. 4-6, pp. 667–715. ISSN 0925-9724, ISSN 1573-7551. Available from DOI: 10.1007/s10606-012-9184-0.
- FEITOSA, Jennifer; SALAS, Eduardo. Today's virtual teams: Adapting lessons learned to the pandemic context. Organizational Dynamics. 2020, p. 100777. ISSN 00902616. Available from DOI: 10.1016/j.orgdyn.2020.100777.
- MALHOTRA, Arvind; MAJCHRZAK, Ann; ROSEN, Benson. Leading Virtual Teams. *Academy of Management Perspectives*. 2007, vol. 21, no. 1, pp. 60–70. ISSN 1558-9080, ISSN 1943-4529. Available from DOI: 10.5465/amp.2007.24286164.
- KERAMINIYAGE, K.; AMARATUNGA, D.; HAIGH, R. Achieving success in collaborative research: The role of Virtual research Environments. *Electronic Journal of Information Technology in Construction*. 2009, vol. 14, pp. 59–69.

- WANG, C. Jason; NG, Chun Y.; BROOK, Robert H. Response to COVID-19 in Taiwan: Big Data Analytics, New Technology, and Proactive Testing. *JAMA* [online]. 2020, vol. 323, no. 14, p. 1341 [visited on 2020-07-06]. ISSN 0098-7484. Available from DOI: 10. 1001/jama.2020.3151.
- KEESARA, Sirina; JONAS, Andrea; SCHULMAN, Kevin. Covid-19 and Health Care's Digital Revolution. *New England Journal of Medicine*. 2020, vol. 382, no. 23, e82. ISSN 0028-4793, ISSN 1533-4406. Available from DOI: 10.1056/NEJMp2005835.
- HALL, Kara L.; VOGEL, Amanda L.; HUANG, Grace C.; SER-RANO, Katrina J.; RICE, Elise L.; TSAKRAKLIDES, Sophia P.; FIORE, Stephen M. The science of team science: A review of the empirical evidence and research gaps on collaboration in science. *American Psychologist*. 2018, vol. 73, no. 4, pp. 532–548. ISSN 1935-990X, ISSN 0003-066X. Available from DOI: 10.1037/amp0000319.
- O'SULLIVAN, Patricia S; STODDARD, Hugh A; KALISHMAN, Summers. Collaborative research in medical education: a discussion of theory and practice: Collaborative research in medical education. *Medical Education*. 2010, vol. 44, no. 12, pp. 1175–1184. ISSN 03080110. Available from DOI: 10.1111/j.1365-2923.2010. 03768.x.
- DESJARDINS, Marie. How to succeed in graduate school: a guide for students and advisors. *XRDS: Crossroads, The ACM Magazine for Students*. 2008, vol. 14, no. 4, pp. 1–9. ISSN 1528-4972, ISSN 1528-4980. Available from DOI: 10.1145/1375972.1375975.
- BIESIALSKA, Katarzyna; FRANCH, Xavier; MUNTÉS-MULERO, Victor. Protocol and tools for conducting agile software engineering research in an industrial-academic setting: a preliminary study. In: *Proceedings of the 6th International Workshop on Conducting Empirical Studies in Industry - CESI '18*. Gothenburg, Sweden: ACM Press, 2018, pp. 29–32. ISBN 978-1-4503-5736-4. Available from DOI: 10.1145/3193965.3193970.
- 15. HICKS, Michael; FOSTER, S. Jeffrey. Adapting Scrum to Managing a Research Group [online]. 2010 [visited on 2020-04-23]. Tech. rep. UM Computer Science Department. Available from: http: //www.cs.umd.edu/~mwh/papers/score.pdf.

- ILYÉS, Enikő. Create your own agile methodology for your research and development team. In: 2019, pp. 823–829. Available from DOI: 10.15439/2019F209.
- WAINER, M. Using Agile Management to Share Course Projects Among Multiple Classes. In: *AGILE 2006 (AGILE'06)*. Minneapolis, MN, USA: IEEE, 2006, pp. 405–410. ISBN 978-0-7695-2562-4. Available from doi: 10.1109/AGILE.2006.59.
- MICROSOFT, Corporation. *Microsoft Word* [online]. [N.d.] [visited on 2021-05-17]. Available from: https://www.microsoft. com/en-us/microsoft-365/word.
- R, Foundation. R [online]. [N.d.] [visited on 2021-05-17]. Available from: https://www.r-project.org/.
- VERHEYEN, Gunther. Scrum: a pocket guide : a smart travel companion. 's-Hertogenbosch: Van Haren Publishing, 2019. ISBN 978-94-018-0377-9. OCLC: 1158018221.
- 21. DOSHI, Hiren. *Scrum Insights for Practitioners: The Scrum Guide Companion*. Practiceagile.com, 2016. ISBN 978-0-692-80717-0.
- 22. STANDARDIZATION ISO, International Organization for; (IEC), International Electrotechnical Commission. *Standard ISO/IEC* 2382:2015. 2015-05. Tech. rep. Edition 1.
- 23. MOHAPATRA, Pratap K. J. Software engineering: (a lifecycle approach). New Delhi: New Age International, 2010. ISBN 978-81-224-2721-9. OCLC: 897032746.
- 24. MEASEY, Peter; RADTAC. *Agile foundations: principles, practices and frameworks*. 2015. ISBN 978-1-78017-255-2. OCLC: 904816124.
- ROYCE, W. Winston. Managing the Development of Large Software Systems. 1970.
- SÁNCHEZ, J; RAMOS, B. David; RAMOS, M. M. Illmara; VIANA, de S. Wesley; E SILVA, R. Gabriel; DE OLIVEIRA, H. T. Elaine. On the use of Scrum for the management of research-based projects. In: *Nuevas Ideas en Informatica Educativa*. 2016, vol. 12, pp. 589–594.

- KRUCHTEN, Philippe. *The rational unified process: an introduction*. 3rd ed. Boston: Addison-Wesley, 2004. The Addison-Wesley object technology series. ISBN 978-0-321-19770-2. OCLC: ocm53038793.
- OCKERMAN, Stephanie; REINDL, Simon. Mastering Professional Scrum: A Practicioner's Guide to Overcoming Challenges and Maximizing the Benefits of Agility. Addison-Wesley, 2018. The professional scrum series by scrum.org. ISBN 0-13-484152-2.
- STACKOVERFLOW.COM. Developer Survey Results 2018 [online].
  2018 [visited on 2020-08-15]. Available from: https://insights.stackoverflow.com/survey/2018.
- 30. DIGITAL.AI. 14th Annual State of Agile Report [online]. 2020-05 [visited on 2021-05-17]. Tech. rep. Available from: https:// explore.digital.ai/state-of-agile/14th-annual-stateof-agile-report.
- 31. BECK, Kent; BEEDLE, Mike; BENNEKUM, Arie van; COCK-BURN, Alistair; CUNNINGHAM, Ward; FOWLER, Martin; GRENNING, James; HIGHSMITH, Jim; HUNT, Andrew; JEF-FRIES, Ron; KERN, Jon; MARICK, Brian; MARTIN, Robert C.; MELLOR, Steve; SCHWABER, Ken; SUTHERLAND, Jeff; THOMAS, Dave. *Manifesto for Agile Software Development*. 2001. Available also from: http://www.agilemanifesto.org/.
- 32. ABRAHAMSSON, Pekka; SALO, Outi; RONKAINEN, Jussi; WARSTA, Juhani. Agile Software Development Methods: Review and Analysis. *arXiv*:1709.08439 [*cs*]. 2017. arXiv: 1709.08439.
- 33. LARMAN, Craig; VODDE, Bas. Lean Primer [online]. Version 1.6. 2009 [visited on 2020-12-07]. Available from: https://www. leanprimer.com/.
- LÓPEZ-ALCARRIA, Abigail; OLIVARES-VICENTE, Alberto; POZA-VILCHES, Fátima. A Systematic Review of the Use of Agile Methodologies in Education to Foster Sustainability Competencies. *Sustainability*. 2019, vol. 11, no. 10, p. 2915. ISSN 2071-1050. Available from DOI: 10.3390/su11102915.

- 35. SCRUM.ORG; GMBH, Age of Product People. *Scrum Master Trends 2019.* 2019. Available also from: https://www.scrum.org/resources/2019-scrum-master-trends-report.
- SCHWABER, Ken; SUTHERLAND, Jeff. The Scrum Guide<sup>™</sup>: The Definitive Guide to Scrum: The Rules of the Game [online]. 2020 [visited on 2021-05-17]. Available from: https://scrumguides. org/.
- ANNOSI, Maria Carmela; MAGNUSSON, Mats; MARTINI, Antonella; APPIO, Francesco Paolo. Social Conduct, Learning and Innovation: An Abductive Study of the Dark Side of Agile Software Development: Social Conduct, Learning and Innovation. *Creativity and Innovation Management*. 2016, vol. 25, no. 4, pp. 515–535. ISSN 09631690. Available from DOI: 10.1111/caim.12172.
- LIMA, I.R.; DE CASTRO FREIRE, T.; COSTA, H.A.X. Adapting and using scrum in a software research and development laboratory. *Revista De Sistemas De Informação Da FSMA*. 2012, vol. 9, no. 9, pp. 16–23.
- 39. ŠOCHOVÁ, Zuzana; DANĚK, Milan. *Skvělý ScrumMaster*. 2018. ISBN 978-80-251-4927-0. OCLC: 1078645163.
- SVERRISDOTTIR, Hrafnhildur Sif; INGASON, Helgi Thor; JONASSON, Haukur Ingi. The Role of the Product Owner in Scrum-comparison between Theory and Practices. *Procedia -Social and Behavioral Sciences*. 2014, vol. 119, pp. 257–267. ISSN 18770428. Available from DOI: 10.1016/j.sbspro.2014.03.030.
- ANDERSON, David J; CARMICHAEL, Andy. Essential kanban condensed. Seattle, Wash: Lean Kanban University Press, 2016. ISBN 978-0-9845214-2-5. OCLC: 1050051860.
- 42. VACANTI, S. Daniel. *The Kanban Guide* [online]. 2020 [visited on 2021-10-05]. Available from: https://kanbanguides.org/.
- 43. YERET, Yuval. Limiting Work in Progress (WIP) in Scrum with Kanban - What / When / Who / How [online]. 2018 [visited on 2020-12-07]. Available from: https://www.scrum.org/resources/ blog/limiting-work-progress-wip-scrum-kanban-whatwhen-who-how. Publication Title: scrum.org.

- 44. SCRUM.ORG; VACANTI, Daniel; YERET, Yuval. *The Kanban Guide for Scrum Teams* [online]. 2019 [visited on 2020-09-18].
- 45. CIRIC, D.; LALIC, B.; GRACANIN, D.; PALCIC, I.; ZIVLAK, N. Agile Project Management in New Product Development and Innovation Processes: Challenges and Benefits Beyond Software Domain. In: 2018 IEEE International Symposium on Innovation and Entrepreneurship (TEMS-ISIE). 2018, pp. 1–9.
- CONFORTO, Edivandro C.; SALUM, Fabian; AMARAL, Daniel C.; SILVA, Sérgio Luis da; ALMEIDA, Luís Fernando Magnanini de. Can Agile Project Management be Adopted by Industries Other than Software Development? *Project Management Journal*. 2014, vol. 45, no. 3, pp. 21–34. ISSN 8756-9728, ISSN 1938-9507. Available from DOI: 10.1002/pmj.21410.
- ROBINSON, Hugh; SHARP, Helen. The Social Side of Technical Practices. In: HUTCHISON, David; KANADE, Takeo; KLEIN-BERG, Jon M.; MATTERN, Friedemann; MITCHELL, John C.; NAOR, Moni; NIERSTRASZ, Oscar; PANDU RANGAN, C.; STEF-FEN, Bernhard; SUDAN, Madhu; TERZOPOULOS, Demetri; TY-GAR, Dough; VARDI, Moshe Y.; WEIKUM, Gerhard; BAUMEIS-TER, Hubert; MARCHESI, Michele; HOLCOMBE, Mike (eds.). *Extreme Programming and Agile Processes in Software Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, vol. 3556, pp. 100–108. ISBN 978-3-540-26277-0. Available from DOI: 10.1007/ 11499053\_12. Series Title: Lecture Notes in Computer Science.
- VALENTIN, Eduardo; CARVALHO, Jose Reginaldo Hughes; BARRETO, Raimundo. Rapid improvement of students' softskills based on an agile-process approach. In: 2015 IEEE Frontiers in Education Conference (FIE). Camino Real El Paso, El Paso, TX, USA: IEEE, 2015, pp. 1–9. ISBN 978-1-4799-8454-1. Available from DOI: 10.1109/FIE.2015.7344408.
- RICO, David F.; SAYANI, Hasan H. Use of Agile Methods in Software Engineering Education. In: 2009 Agile Conference. Chicago, USA: IEEE, 2009, pp. 174–179. ISBN 978-0-7695-3768-9. Available from DOI: 10.1109/AGILE.2009.13.

- OECD. Frascati Manual 2015: Guidelines for Collecting and Reporting Data on Research and Experimental Development [online]. OECD, 2015 [visited on 2020-07-06]. The Measurement of Scientific, Technological and Innovation Activities. ISBN 978-92-64-23880-0. Available from DOI: 10.1787/9789264239012-en.
- SHOJA, Mohammadali; ARYNCHYNA, Anastasia; LOUKAS, Marios; D'ANTONI, Anthony V.; BUERGER, Sandra M.; KARL, Marion; TUBBS, R. Shane (eds.). *A Guide to the Scientific Career: Virtues, Communication, Research and Academic Writing*. 1st ed. Wiley, 2019. ISBN 978-1-118-90742-9. Available from DOI: 10.1002/ 9781118907283.
- 52. CRESWELL, John W. *Educational research: planning, conducting, and evaluating quantitative and qualitative research.* 4th ed. Boston: Pearson, 2012. ISBN 978-0-13-136739-5. OCLC: ocn693750954.
- 53. KRUGER, Justin; DUNNING, David. Unskilled and unaware of it: How difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of Personality and Social Psychology*. 1999, vol. 77, no. 6, pp. 1121–1134. ISSN 1939-1315, ISSN 0022-3514. Available from DOI: 10.1037/0022-3514.77.6.1121.
- MANAMENDRA, M.A.S.C.; MANATHUNGA, K.N.; PERERA, K.H.D.; KODAGODA, N. Improvements for agile manifesto and make agile applicable for undergraduate research projects. In: 2013 8th International Conference on Computer Science & Education. Colombo, Sri Lanka: IEEE, 2013, pp. 539–544. ISBN 978-1-4673-4463-0. Available from DOI: 10.1109/ICCSE.2013.6553969.
- 55. PARTOGI, Joshua. 5 reasons why eXtreme Programming isn't popular [online]. 2016 [visited on 2020-07-16]. Available from: https: //medium.com/agility-path/5-reasons-why-extremeprogramming-isnt-popular-83790418b901.
- 56. BUNSE, Christian; FELDMANN, Raimund L.; DÖRR, Jörg. Agile Methods in Software Engineering Education. In: KANADE, Takeo; KITTLER, Josef; KLEINBERG, Jon M.; MATTERN, Friedemann; MITCHELL, John C.; NAOR, Moni; NIERSTRASZ, Oscar; PANDU RANGAN, C.; STEFFEN, Bernhard; SUDAN, Madhu; TERZOPOULOS, Demetri; TYGAR, Dough; VARDI, Moshe Y.; WEIKUM, Gerhard; ECKSTEIN, Jutta; BAUMEISTER, Hubert

#### **BIBLIOGRAPHY**

(eds.). Extreme Programming and Agile Processes in Software Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, vol. 3092, pp. 284–293. ISBN 978-3-540-22137-1. Available from doi: 10.1007/978-3-540-24853-8\_43. Series Title: Lecture Notes in Computer Science.

- 57. FACULTY OF INFORMATICS, Masaryk University. *Faculty of Informatics* [online]. 2020 [visited on 2020-04-25]. Available from: https://www.fi.muni.cz/index.html.en.
- 58. FACULTY OF INFORMATICS, Masaryk University. Faculty of Informatics - Statistics [online]. 2020 [visited on 2020-04-25]. Available from: https://www.muni.cz/o-univerzite/fakulty-apracoviste/fakulta-informatiky/statistika.
- 59. SERVICE SYSTEMS LABORATORY, of. Laboratory of Service Systems [online]. 2021 [visited on 2021-05-17]. Available from: https://seslab.fi.muni.cz/.
- 60. ŠTEVANKOVÁ, Lucia. Analýza Smart City z pohledu IT managementu [online]. 2018 [visited on 2020-04-25]. Available from: https://is.muni.cz/th/lbwle/. Diplomová práce. Masarykova univerzita, Fakulta informatiky, Brno.
- 61. MICROSOFT, Corporation. *Microsoft Excel* [online]. [N.d.] [visited on 2021-05-17]. Available from: https://www.microsoft. com/en-us/microsoft-365/excel.
- 62. ATTIS, software s.r.o. *ATTIS* [online]. [N.d.] [visited on 2021-05-17]. Available from: https://www.attis.cz/software.
- 63. VMWARE, Inc. *VMware* [online]. [N.d.] [visited on 2020-10-06]. Available from: https://www.vmware.com/.
- 64. BOWEN, A. Shannon. Mission and Vision. In: *The International Encyclopedia of Strategic Communication*. 1st ed. Wiley, 2018. ISBN 978-1-119-01072-2. Available from DOI: 10.1002/9781119010722.
- 65. MICROSOFT, Corporation. *Microsoft Teams* [online]. [N.d.] [visited on 2021-05-17]. Available from: https://www.microsoft. com/en-us/microsoft-teams/group-chat-software.

- 66. SCHWARZOVÁ, Zuzana; WALLETZKY, Leonard. DOD Laboratoř servisních systémů FI MUNI [online]. 2021 [visited on 2021-05-17]. Available from: https://www.youtube.com/watch?v=CqQJ-6dADqY.
- SULLIVAN, Gail M.; ARTINO, Anthony R. Analyzing and Interpreting Data From Likert-Type Scales. *Journal of Graduate Medical Education*. 2013, vol. 5, no. 4, pp. 541–542. ISSN 1949-8357, ISSN 1949-8349. Available from DOI: 10.4300/JGME-5-4-18.
- KOTHARI, C. R. Research methodology: methods & techniques. New Delhi: New Age International (P) Ltd., Publishers, 2004. ISBN 978-1-282-10262-0. OCLC: 1109204175.
- BURNS, Alvin C.; BUSH, Ronald F. Basic marketing research: using Microsoft Excel data analysis. 2nd ed. Upper Saddle River, N.J: Pearson Prentice Hall, 2008. ISBN 978-0-13-205958-9. OCLC: ocn173299261.
- 70. SURVIO, s.r.o. *Survio* [online]. Brno, Czech Republic, [n.d.] [visited on 2021-05-17]. Available from: https://www.survio.com/.
- 71. WATTS, Geoff. *Scrum mastery: from good to great servant-leadership*. s.l.: Inspect & Adapt Ltd, 2013. ISBN 978-0-9575874-0-3.