

**LEARNERS
FIRST!**



Vera Baum & Manuel Illi

An Agile Approach to Learning

1.1 Table of contents

1. Introduction

2. What is Agile Learning?

2.1 The Origins of Agility

2.2 The Agile Manifesto

2.3 The Agile Framework Scrum

2.4 The Core of Agility

2.5 The Agile Learning Approach

2.6 Agile Learning in Organisations

2.7 Comparing Agile and Traditional Learning

3. Theoretical Foundations

3.1 Learning Goals and Motivation

3.2 Scripts

3.3 Self-regulated Learning and Metacognition

3.4 Feedback

4. Roles in Agile Learning

4.1 Learners

4.2 Agile Learning Coaches

4.3 Instructors

4.4 Managers

4.5 Customers and Partners

5. Meetings and Processes in Agile Learning

5.1 Kick-Off : “What's it all about?”

5.2 Planning : “Where will the journey take you?”

5.3 Sprint : “Off you go!”

5.4 Review and Retrospective : “Learning never stops”

5.5 Interim Meeting : “How are you?”

5.6 Settings, Preparation and Follow-Up

6. Agile Learning in Practice

6.1 Where to Start

6.2 Gathering Experience

6.3 Gaining Confidence

6.4 Healthy Growth

6.5 Return on Investment

6.6 A Glance Into the Future

7. References

8. About the Authors

1. Introduction

Money can't buy agility, you have to practice it

We are delighted that you are interested in agile learning. Agile principles and methods paired with the right mindset can really improve the way people live and work together – we have experienced this time and again in many projects and teams over the last few years.

Maybe you share this experience because you are already working in an agile environment, or perhaps you are just vaguely familiar with the term 'agility' and want to learn more about new learning approaches. No matter what made you curious about this book, if you are interested in how to make learning processes more exciting, efficient and sustainable, then you have the right book.

We are passionate about agility. From this passion and from years of experience in agile coaching, agile projects, teaching, training and seminars came the idea to also introduce agility to learning and personal development. We have set out to create agile learning oases in very different contexts.

On the way to a truly agile approach to learning, we have done a lot of research, development, testing and finally put our results into practice. With this book we want to share our findings and experiences. It is intended to be a theoretically sound yet also practical introduction to agile learning.

What you will find in this book

You can expect background information from the field of pedagogy and learning theory as well as software development, but above all we would like to give concrete tips for the practical realisation of agile learning. A large part of this book addresses workplace learning, but it can also be implemented in other contexts. We have had very positive results and experiences with agile learning in schools, youth work and non-profit organisations.

To this date, there are relatively few research results on applying agile learning in practice. Therefore, in chapters 2 and 3 we have collected well-documented findings from organisational theory, teaching-learning research, learning psychology and pedagogy and present how they play a role in agile learning.

In chapters 4 and 5 we provide practical tips for designing roles, processes and meetings in the agile learning practice, combining theoretical considerations with our own research results and practical experience. These chapters are particularly interesting if you want to learn agile yourself or support others in their learning efforts as a coach, manager or L&D professional.

The tips and practical advice we present here may seem very detailed and model-based at first glance. The reason for this is quite simple : Agile learning can be challenging at the beginning because its key aspects are quite different from the typical learning environments we grew up in and were influenced by for many years. To trigger changes in our thinking processes, it is helpful to start by describing new roles, processes and meetings in detail. Over time, these detailed 'instructions' can be scaled back and interpreted

more freely. The effectiveness of this approach has been studied quite well in script research (see [chapter 3.2](#)).

In the last chapter we would like to give you practical guidance if you are considering to establish the agile learning concept in your organisation. We show you a viable path from the first steps to a comprehensive, new and inspiring learning culture.

Depending on what you are particularly interested in, you can also read the individual chapters on their own. We made sure to refer from the practical to the theoretical chapters, and vice versa.

The never-ending journey

When we set out to write a book about agile learning, we quickly realised that a book with agile content should also be written in an agile way. But what does it mean? For us it means :

The book of many : It starts with an idea, then evolves through brainstorming in a team, through questions and suggestions of our peers, through lots of typing and proofreading – this book is the result of the incredibly stimulating collaboration process involving many of our fellow QualityMinds, even if there are only two names on the cover. This book was a new kind of writing experience for all of us and it was exciting, insightful and a lot of fun.

Grown organically : We had a rough structure of the content quite early on, yet sections kept changing. Again and again conversations or teams meetings led us to rewritten sections, or delete others or include completely

new ones. This book may be printed, but the process is still going on.

The never-ending journey : We want to continue working on the topics of this book. Thanks to “Book on Demand”, we can incorporate new experiences and studies into an updated edition more easily. And this is where you come into play.

Your insights matter : We look forward to your feedback, suggestions and advice for the further evolution of the agile learning approach and this book.

Do you have experience with agile learning or interesting information you would like to share with future reader and us? Then write to us at : learning@qualityminds.de

Thank you!

We would like to thank all our supporters, helpers, friends and colleagues who made this book a reality. Thank you Carina, Elena, Guna, Joona, Veronika, Rick, Markus, Sanne, Susanne, Tine and the entire QualityLearning team for thinking and developing together. Thank you Madeleine and Rosetta for the typesetting and graphic design ; Bettina, Eva, Tamara and Ralph for the input and proofreading. Thank you Michael and Robert for the unconditional support, critical questioning and creative freedom. Massive thanks to the whole QualityMinds community. You are simply wonderful!

Vera Baum and Manuel Illi, 2022

2. What is Agile Learning?

Learning something and then using it right away, for example a new language, can be very motivating. Being able to react to situations in a dynamic way, even in completely new and challenging situations, is also motivating and more and more important. This chapter therefore takes a look at how learning and agility can be combined and thus creating something completely new – something new that can even change our understanding of how we learn, live and work (Gehlen-Baum & Hoppenz, 2018).

A good starting point is the illustrious word 'agility'. The English adjective 'agile' derives via French from the Latin 'agilis', which means 'quick-moving, nimble, active'. The idea of agility widely discussed today in business management and organisational development, however, is much more recent.

2.1 The Origins of Agility

Scientifically, the concept of 'agility' was developed and researched in sociology in the early 1950s. In their study “Working Paper in the Theory of Action” (1953), the authors Bales, Shils and Parsons examined group behaviour which Parsons (1978) later on incorporated it in his theoretical AGIL paradigm.

The broader interest in agility, however, began in the 1990s. That was the time when not only the 'personal computer' entered many households but also the complexity of

computer programs started to increase rapidly. This is well illustrated by the example of Microsoft Windows. While the first version (NT 1.0) had about 4 to 5 million lines of code in 1993, Windows XP (NT 5.1) in 2001 had already ten times more, containing about 40 million lines of code (O'Brien, 2005).

Traditional production methods that were established in other contexts, for example in automotive manufacturing, could not keep up with this complexity. In 1991, the Iacocca Institute of Lehigh University called for new forms of production under the slogan of 'agility' and defined this term to be “A manufacturing system with extraordinary capability to meet the rapidly changing needs of the marketplace [...] ideally in real-time response to customer demand” (quoted from Hopper et al., 2001, p. 632). The same institute also founded the Agile Manufacturing Enterprise Forum (AMEF) shortly afterwards to advocate the required dynamic reactivity.

In the late 1990s, this flexible reaction was combined with aspects of proactive behaviour, customer expectations and process adaptability (Förster & Wendler, 2012). Software companies quickly took notice of this since they had to develop their products in shorter cycles and react to suddenly emerging requirements all along. There is hardly anything worse than an application that, although well-engineered and stable, is already outdated at the time of its delivery or fails to meet the needs of its users.

Then, in February 2001, a group of 17 leading software developers met in the snow-covered mountains of Utah to pool their experiences into a common approach. The result was the “Manifesto for Agile Software Development” (or in short : “Agile Manifesto”). The values and principles described in this manifesto are still having unforeseen effects.

ects until today, in widely different contexts such as business management or, as in our case, learning.

2.2 The Agile Manifesto

The Agile Manifesto is based on the realisation that many factors determine the success of a complex process and that therefore a well-considered prioritisation is vital. These influencing factors can be anything from interaction among team members and collaboration with customers, to tools, functioning software and documentation up to reactions to changes, contract negotiations, planning, and so forth.

The authors of the Agile Manifesto were driven by the desire to clearly prioritise these factors to make the highly complex processes of software development more transparent, more efficient and more pleasant for everyone involved while still delivering value. The Manifesto is based on four basic values which, as a matter of fact, are a list of priorities :

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value :

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

That is, while there is value in the items on the right, we value the items on the left more.” (Beck et al., 2001)
Based on these four statements, the authors of the manifesto developed twelve agile principles that are

intended to be a practical guide for the implementation of these values, for example “Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done” or “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.” (Beck et al., 2001)

The authors of the manifesto wanted agility to help find a better balance between technical, strategic, market-related and customer-oriented requirements in the context of software development. “We want to restore a balance” is their credo. And it continues : “We embrace modeling, but not in order to file some diagram in a dusty corporate repository. We embrace documentation, but not hundreds of pages of never maintained and rarely used tomes. We plan but recognize the limits of planning in a turbulent environment.” (Beck et al., 2001)

The authors were not intending to declare agility to be a miracle cure or a panacea. Rather, the values and principles were intended to inspire programmers, testers, software architects, managers, project leaders, customers – ultimately all stakeholders in the process – to work together in a more transparent, efficient and rewarding way.

Agility, as many state (including Hofert (2018) in her book “Das agile Mindset”) is not a method but a mindset. An important part of which is actually doing things yourself – principles can all too quickly lose their value if they are not practiced by those who preach them. Part of the Agile Manifesto is therefore also about sharpening the focus on what is essential and departing from unhelpful patterns in every-day interactions. For implementing agility in practice, there are a number of specific frameworks and methods. One such process framework is briefly presented in the

following chapter, as it can be a valuable guiding principle for agile learning as well.

2.3 The Agile Framework Scrum

Scrum is next to many other frameworks and methods, such as Kanban or extreme programming, the currently most widely used framework to implement agile processes. The English word 'scrum' (short for scrumming) comes from Rugby and originally describes the joining together of the team in a “tight organised formation” where “the ball gets passed within the team as it moves as a unit up the field” (Nijland 2020). Jeff Sutherland and Ken Schwaber, who were also involved in the Agile Manifesto, created with the “Scrum Guide” a sophisticated process model in the early 2000s that takes up the Manifesto ideas (Sutherland & Schwaber, 2013).

Scrum is based on three empirical pillars : transparency, inspection and adaptation, through which the actual progress of complex processes becomes visible and manageable. Sutherland and Schwaber believe that the Scrum team is at the heart of the Scrum process, in which a clear division of roles ensures transparent responsibilities (Gloger, 2016) :

The **Product Owner** is responsible for the product vision, for creating value and the big picture. Their tasks include the prioritisation of requirements and tasks as well as the communication with stakeholders (e.g. management, customers, users). This also means that the PO is the intermediary between the stakeholders and the development team.

The **Development Team** develops the software and is interdisciplinary. While in classic (software) development different team members have fixed tasks and responsibilities such as software architect, developer, tester, etc., where each reports to their own department, in Scrum the team members work together cross-functionally. As a team, they are jointly responsible for what is developed in which way in a sprint. This means each member combines broad general qualifications with deep specialised skills and takes on tasks as needed.

The **Scrum Master** has to make sure that the development team can work as efficiently as possible without external obstacles and disturbances. They also ensure that the Scrum process works, rules are followed and open communication takes place.

Besides these roles, Scrum also defines a clear time-based structure. Well-defined process phases and events help to make reliable forecasts about future progress. The risk can be calculated because tangible results are produced at short intervals.

Sprint : The sprint is a fixed, time-limited period in which a usable and potentially deliverable product increment is completed (max. 30 days, although the trend is now increasingly towards one- or two-week sprints). Sprints should generally have the same duration and structure, though there are occasionally times when it makes sense to vary sprint lengths.

Planning : At the beginning of the sprint, the Scrum team sets the goal(s) for the current sprint together. For this purpose, the product owner prepares an organised list of requirements for the product (i.e. the product backlog) and is responsible for the content and prioritisation. It is

important to define acceptance criteria for each requirement so that it is possible to determine whether the requirement is successfully delivered or not. The development team estimates the complexity, effort and risk of the various requirements in the product backlog and decides how many items can be delivered in the sprint. If necessary, the product owner can specify or clarify individual points. The team then commits to the sprint goal(s) and independently organises how to develop the requirements and who works on which task.

Daily Scrum : The 'daily scrum' is a daily meeting for the development team, lasting maximum 15 minutes, to discuss progress, challenges and the next steps.

Review : At an informal meeting of the scrum team and the stakeholders towards the end of the sprint, the product increments completed are presented to the product owner and, if required, also to the stakeholders and evaluated together. The review is about the 'what' of the sprint, meaning the result of the sprint and necessary changes for future sprints. The development team demonstrates the functionalities that has been implemented. The Product Owner in turn compares the presented functionality with the acceptance criteria agreed upon in the planning.

Retrospective : At the end of the sprint, the scrum team reflects on their way of working and collaborating. The retrospective is about the 'how' of the sprint to learn from it for future sprints and to continuously improve. Just like in the Agile Manifesto, the retrospective in the scrum process makes learning an important part the process.

Product Backlog – Sprint Backlog : In addition to roles and events, the Scrum Guide also describes artifacts such as the 'product backlog' and the 'sprint backlog.' Both are

lists for either the entire project or the current sprint in which the tasks are transparently described and prioritised.

These artifacts can also be applied as helpful elements for self-directed agile learning. Such backlogs help to precisely describe personal learning goals and to keep an eye on general learning topics and areas of growth and development.

Because of the clear responsibilities (roles), time limits (sprint phases) and transparent artifacts, goals, processes and necessary operative steps can be planned but remain at the same time flexible for (short-term) changes, such as for new requirements (Gloger & Margetich, 2014).

It is important to note that Scrum is much more than just an effective structure. It lays the foundation for successful cooperation through certain basic values which have to become part of a shared team culture. According to Schwaber and Sutherland the five most important basic values are commitment, courage, focus, openness and respect (Sutherland & Schwaber, 2013).

2.4 The Core of Agility

Agility as a concept and Scrum as a framework have been very successful and therefore long since spread to other contexts. In self-help and management books numerous ways are presented to make corporate leadership, management and HR more agile and to apply Scrum methods (Gloger & Margetich, 2018 ; Häusling, 2017 ; Hofert, 2016 ; Rahn, 2018 ; Scheller, 2017). First experiences have been made in 'agile public administration' (Bartonitz et al., 2018), Scrum models are tested in schools (EduScrum, Scrum4Schools) and universities design