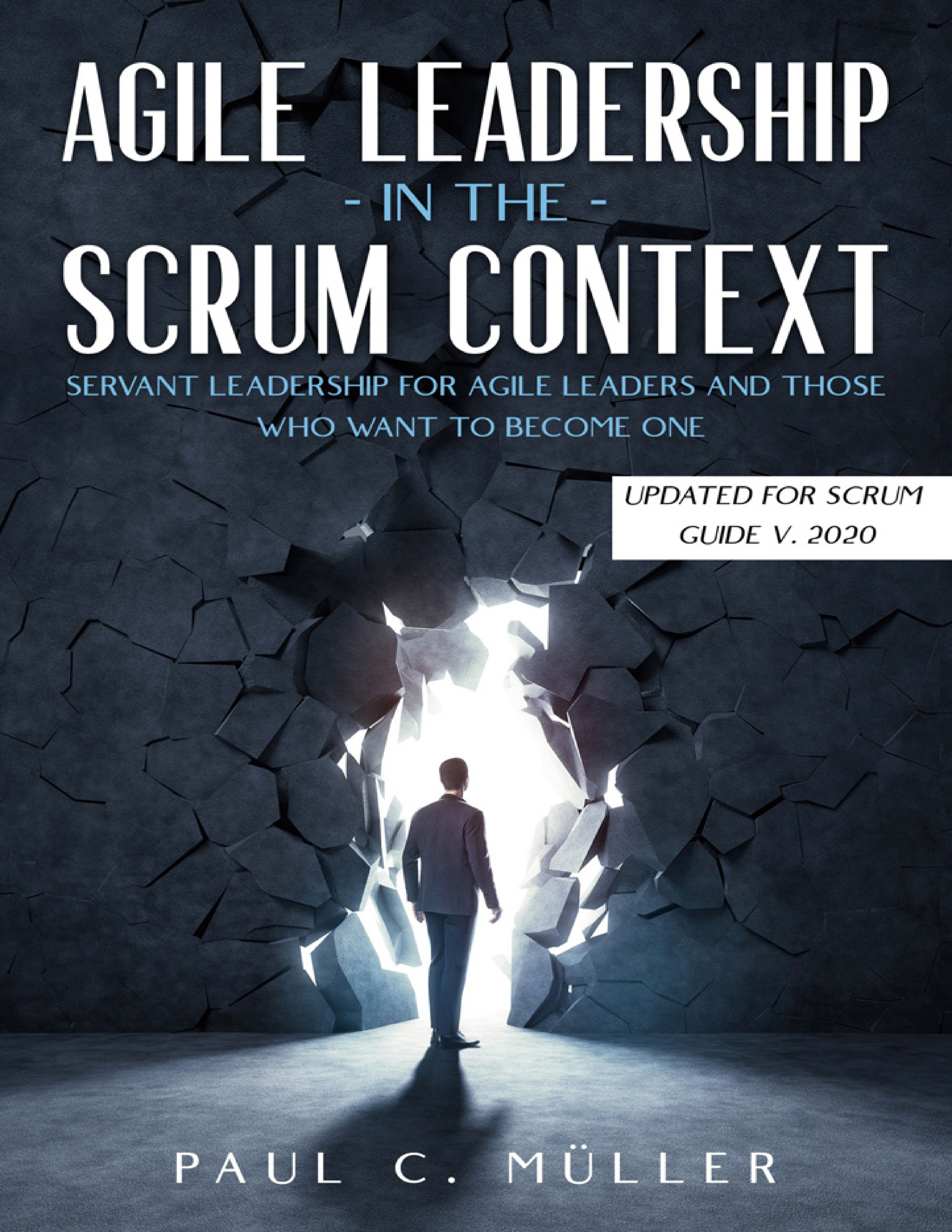
AGILE LEADERSHIP

- IN THE -

SCRUM CONTEXT

SERVANT LEADERSHIP FOR AGILE LEADERS AND THOSE WHO WANT TO BECOME ONE

UPDATED FOR SCRUM GUIDE V. 2020



PAUL C. MÜLLER

Table of Contents

- PRELIMINARY REMARKS
- FOREWORD
- UNDERSTAND AND APPLY THE SCRUM FRAMEWORK
    - The agile manifesto
    - Twelve principles of the Agile Manifesto
    - Empirical process control
    - Scrum values
    - Roles in Scrum
    - Scrum Events
    - Scaling Scrum with Nexus
- DEVELOPING PEOPLE AND TEAMS
    - Self-organized teams
    - Facilitation (moderation)
        - The input
        - The moderated event itself
        - Free and informed decisions
        - The result
    - Agile leadership
- AGILE PRODUCT MANAGEMENT
    - Forecasting and release planning
        - Agile product development
        - The project triangle
        - Release planning
    - Product Vision
        - Vision Boarding
    - Product value

- Corporate strategy
- Stakeholders and customers

PROFESSIONAL PRODUCT MANUFACTURING AND DELIVERY

- Software development
  - Emergent Software Development
  - Continuous Quality
  - Continuous Integration und Delivery
  - Optimized workflow
  - Integrated risk management
- Continuous Quality
  - Test First

BUILD AN AGILE ORGANIZATION

- Organizational design and culture
  - Competing values
  - Overcome Zombie Scrum
  - The task of an agile leader in relation to the organization
- Portfolio planning
- Evidence-Based Management
  - The use of EBM in practice

EXAM PREPARATION "PROFESSIONAL AGILE LEADERSHIP

RECOMMENDED READING

PRELIMINARY REMARKS

Professional Agile Leadership^{TM} (PAL E) is, like the other Scrum certifications mentioned in the book, the property of Scrum.org. This book has been written without influence or commission from Scrum.org as an expression of the author's own engagement with the topics presented. For the sake of easier readability, brands and trademarks have been omitted in the text. However, these are also meant in each case and should be understood in this way.

The book already presented was reviewed in December 2020 based on the Scrum Guide version 2020 (published November 2020) and adapted where necessary.

FOREWORD

It is difficult to write a book about an agile framework, because agility thrives on the fact that not everything is worked through according to a certain recipe and not every question can be looked up in a kind of manual. Agility lives from experiences, insights, experiments and their evaluation and the measures derived from them. Scrum calls this the three pillars of Scrum: transparency, review and adaptation. You could also call it the agile control loop of Scrum.

When I decided to write this book, it was driven by a customer request. A customer stated that there would probably be no German language book to prepare for the Professional Agile Leadership™ (PAL E) certification exam of Scrum.org and that the source material on the Scrum.org website was so broad that preparing for the exam would mean a lot of effort.

So, I decided to write a presentation of the exam topics for people who are in the same position as my client and prefer to learn on their own rather than have a trainer guide them through the material for two days. However, it was important to me that I did not simply prepare for an existing pool of questions, but also put the knowledge taught into a context that would allow the reader to put what they have learned into practice, to learn more about the values and thinking behind agile transformation and development, and to be able to apply it in their own practice.

However, please keep in mind: You and your team will not become agile by reading a book, but by conducting experiments and the resulting findings and measures. What I can contribute are a few approaches and thought patterns, which you are welcome to use for the implementation. But always keep in mind: It is not about implementing the Scrum of the author, but the Scrum of your team(s), based on your own insights, supported by your own development process.

I wish you and your team much success on this exciting journey through an agile world and hope that it brings you as much joy and insight as it does me.

The author

UNDERSTAND AND APPLY THE SCRUM FRAMEWORK

Before we look at the question of what Scrum is and what agile leadership means in the context of Scrum, it seems important to first look at what agility actually means. What are the ideas behind it and what benefits is agile supposed to bring us? To explore this, it makes sense to first deal with the agile manifesto, the constitution of agility, so to speak:

The agile manifesto

In February 2001, seventeen people met in a ski lodge in the Wasatch Mountains of the American state of Utah to talk, ski, and relax together. They were all dissatisfied with the way software development was taking place and believed that alternatives to documentation-heavy, heavyweight software development processes were needed.

This group of organizational anarchists, who called themselves "The Agile Alliance", jointly formulated and signed the "Agile Manifesto". It should be noted that the people present later went quite different ways and developed different methods and frameworks based on the common foundation "Agile Manifesto". (Co-) developers of "Extreme Programming", "Scrum", "DSDM", "Adaptive Software Development", "Crystal" and others laid here a common foundation for the further development of software development and in many cases also for issues far beyond software development.

For more information on the history of its creation, please visit:
http://agilemanifesto.org/history.html

Manifesto for Agile Software Development

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.

Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Businesspeople and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

In short words, the manifesto thus describes the fundamentals of agile thinking.

Individuals and interactions over processes and tools

Following processes and using tools correctly is undoubtedly a key success factor for carrying out successful development processes. Nevertheless, the agile manifesto rightly places "individuals and interactions" in front of them. Only individuals are in a position to permanently evolve and thus achieve a constant improvement of the development process and the developed products. The interaction between individuals offers additional potential for the work of a team to become more than the sum of the individual performances. On the other hand, a team can only work optimally if each team member is also perceived as an individual with strengths, weaknesses and his or her own personality and is included as such.

Working software over comprehensive documentation

Stipulating requirements in advance and documenting the implemented results are a basis for understanding product content and the foundation for successful maintenance and further development. However, if it is not possible to create a functioning software product, its usefulness is very limited.

Customer collaboration over contract negotiation

Contracts and agreements between customers and manufacturers are of great importance. They form an important basis for ensuring that all parties understand an order in the same way. Nevertheless, contracts cannot depict every possible scenario in detail and anticipate every possible event. They are created based on the state of knowledge at a given point in time. In the course of a project, however, both the customer and the supplier evolve. The available knowledge about the joint project increases and the general conditions also change. In order to take this development into account in the project, good and intensive cooperation with the customer is an indispensable prerequisite.

Responding to change over following a plan

Plans are important and the idea that agile approaches, such as Scrum, would do without plans is absurd. One could even say that there is far more planning in agile techniques. The main difference is that planning in agile environments does not primarily take place at (or before) the start of the project, but on an ongoing basis and always with a "just in time" approach. It makes no sense to create detailed plans for months and years to come if you see customer feedback and responding to it as the basis for enabling maximum value-added development.