

LEADERSHIP IN THE DIGITAL AGE

RENAISSANCE OF THE RENAISSANCE MAN



Niklas Hageback

Leadership in The Digital Age

Leadership in The Digital Age

*Renaissance of
The Renaissance Man*

Niklas Hageback

Leadership in The Digital Age: Renaissance of The Renaissance Man
Copyright © Business Expert Press, LLC, 2021.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopy, recording, or any other except for brief quotations, not to exceed 250 words, without the prior permission of the publisher.

First published in 2021 by
Business Expert Press, LLC
222 East 46th Street, New York, NY 10017
www.businessexpertpress.com

ISBN-13: 978-1-95253-862-9 (paperback)
ISBN-13: 978-1-95253-863-6 (e-book)

Business Expert Press Collaborative Intelligence Collection

Collection ISSN: 2691-1779 (print)
Collection ISSN: 2691-1795 (electronic)

Cover image licensed by Ingram Image, StockPhotoSecrets.com
Cover and interior design by S4Carlisle Publishing Services Private Ltd.,
Chennai, India

First edition: 2021

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America.

Dedication

To Aarik and Bernard

Abstract

This is a book for anyone intrigued by the complexities of digital leadership that require a capability to constantly balance the routines of everyday business with the ability to innovate. Finding the appropriate mix between the dichotomy stability and flexibility has been a delicate task that few, if any, corporations have properly managed to overcome. Why is that? This conundrum becomes acute as businesses embark on digital transformations, an often painful venture highlighting the deficiencies of traditional management styles but also agile methodologies. They deliver results that are far below initial expectations, provide half-baked digital solutions where potential commercial gains are poorly captured and leveraged, and, far too often, are not even identified. Mismatches between technologies, the man-machine (dis)connect, or organizational dysfunctionality are typically identified as root causes, but beneath them lurks a more scathing problem: an inadequate leadership.

And it is generally not because of a lack of technological expertise or poor people management that these skills are often sufficiently resourced. The leadership problem is, instead, of a more imperative nature, this as the digital transformation comes to challenge the business to its core: How do we provide customer value, and how do we organize ourselves in the most (cost-)efficient way?

In essence, it takes aim at a corporation's *raison d'être* and, as such, goes far beyond technical solutions, corporate structure, or indeed even an in-depth understanding of products and markets, this as the digital transformation explicitly (and implicitly) transcends all of these perspectives in its aspiration toward value maximization. By acknowledging these cojoined complexities, elements of what it is that the digital leadership requires can be formulated, namely, a holistic approach capable of incorporating them, facilitated by a cognitive capacity that can craft commercially viable products and services.

But in the age of the specialists that are drilling down in ever more granular minutiae, do there even exist generalists equipped with this holistic mindset, and, if so, what do they look like?

To start with, there are role models that provide inspiration and are worth looking at, but these bring us back a few hundred years in time.

Enter the Renaissance Man.

Keywords

agile; change management; collective intelligence; creativity; digital leadership; digital transformations; innovation; project management; renaissance man

Contents

<i>Acknowledgments</i>	<i>xi</i>
<i>Introduction</i>	<i>xiii</i>
Chapter 1 The Accelerating Pace of Change.....	1
Chapter 2 Agile and Then Some More Agile.....	15
Chapter 3 The Innovating Organization	27
Chapter 4 The Renaissance Man.....	45
Chapter 5 A Beautiful New World?	59
<i>Bibliography</i>	75
<i>About the Author</i>	81
<i>Index</i>	83

Acknowledgments

I would like to express my warmest gratitude to Scott Isenberg and his teams at Business Expert Press and S4Carlisle, respectively, for helping to turn a raw manuscript into a book. I would also like to thank Jim Spohrer, Cognitive OpenTech IBM, for his many valuable comments and insights, which were of great help in my work.

Introduction

Technology alone is not enough. It's technology married with the liberal arts, married with the humanities, that yields us the results that make our hearts sing.

—Steve Jobs, iconic digital leader (1955 to 2011)

The world is in the midst of a digital transformation that is transcending all industries. It is highlighted by two emerging trends that businesses are forced to adhere to:

- All firms are now *technology* firms, whether they acknowledge it or not. This is because upgrades and advances in business models rest on digital solutions, mainly software applications, either developed in-house or third-party tools, customized for bespoke value-added products and services.
- There is a reduced reliance on *quantity* of employees in favor of finding, grooming, and retaining *quality* employees. These typically come with a high level of technological competence, in addition to specific domain expertise. Deploying large numbers of employees to deal with a business challenge is no longer a viable solution, but rather seeking out the extraordinary *man-machine* connect. This is an effect of machines now starting to outmatch large segments of the work force. To fully benefit from these developments, a drastically altered organizational arrangement with a different type of leadership is required.

As a consequence of this changing landscape, most firms are now finding themselves in the proverbial Darwinian survival mode. It is a situation triggered by many factors incidentally acting in concert: fickle customer demands, increased automation at all levels of the organization to push

down costs, and growing complexity from operating in fragmented markets defined by differing regulations and heightened political risks. It looks quite like a race to the bottom, with the winner only temporarily taking it all, or at least most of it. But with the low barriers to technical innovation, this dominance is often of a fluid character, and so this race is set to recur over and over again. The assumed longevity of premium branding and customer loyalty is quickly becoming obsolete. In this highly dynamic setting, where change in its many aspects overwhelms all commercial activities, and that too at an accelerating rate, what tools do we have at our disposal to handle and tame it to our advantage? Well, *agile* is currently the buzzword that businesses across industries and geographies hang on to. That agile saw its genesis in the software industry was not by chance. The long development cycles, as prescribed by traditional project management methods, led to deliverables that were often already technically antiquated and way off the mark in terms of functionality at the time of sign-off and deployment. Something more flexible was needed in order to be able to incorporate the ongoing demands for change from customers and increased technical complexity and innovation, features reflecting an industry that was in a constant state of flux. To solve this conundrum, the agile philosophy took aim at two key principles:

- Keep bureaucracy at a minimum by, as far as possible, eliminating formal process and protocol, and
- Break down hierarchies and flatten the organization with a focus on concentrated cross-functional teamwork, which can include the end users, empowered with executive decision-making power.

Although straightforward, agile practices keep everyone on their toes. Daily sprints ensure that no slackers need apply and that everyone is held accountable for their productivity. Thus, they are highly demanding yet well endowed if properly performed. Numerous testimonies agree on how products have come to much more closely reflect the customers' actual needs and are delivered in a notably faster and more labor-efficient manner. Agile, in its many variations, is therefore now an acclaimed project management standard. But like previous popular project- and strategic

management concepts that eventually faded into oblivion, is the same fate also awaiting the agile framework?

The author argues that it appears to be different this time. The popularity of agile has given it a longevity that has by far extended that of some of its predecessors, such as Six Sigma, Just in Time, and Balanced Scorecards. Agile, or rather the agile mindset, is here to stay simply because the conditions that brought it forth in the first place, including fast-paced change, have now become the milieu for almost all industries. It is something they will have to adjust to for the foreseeable future, and agile is by far the best tool we have at our disposal to handle it. But agile has limitations, and there are still missing components. The cross-functional teams, with their wide variety of perspectives, are requiring a new type of leadership, namely, a generalist leader, an individual well versed in all relevant knowledge domains, not only technology. It is someone able to comprehend a digital transformation in all its facets and to ensure that potential commercial exploits can be fully capitalized on. Indeed, a very different managerial proposition than the currently preferred specialist manager, who is still being touted in business education. In essence, it is a leadership style with roots reaching back to medieval times, and we are seeing the return of the illustrious *Renaissance Man*.

Agile, given its flexible attributes, is extending beyond project management to include business development, and, more recently, an agile organizational template is seeking to find its form. But will it work? The publicized success stories of triumphant and superior agile organizations have appeared suspiciously few and far between. It certainly does not tally with agile consultants' tall claims of its perceived advantages. Somehow, it appears difficult to fully endorse agile, and a certain amount of bureaucracy and procrastination in decision-making and execution appears inevitable for big corporations. It is possibly the result of an inherent condition in all forms of human collectives that we keep falling back on despite efforts to circumvent these traits. Aspiring to do considerably more with less is an amicable proposition, but by its very nature, it implicitly seeks to eliminate a lot of corporate bureaucracy that is occupied by the middle management cadre. This poses a grave threat to the career man who is considered the backbone of many corporations. It has become a

battle between governing philosophies that is now being fought in many corporate boardrooms, and the short-term outcome is far from certain. One thing appears clear, however: Agile is here to stay, but in what forms and at what organizational layers are questions that remain unanswered. So if agile is unable to go all the way, are there alternatives and should they be considered? Perhaps the concept of collective intelligence can augment initiatives to embrace both innovation and the digital transformation of a corporation.

The vast scale and enormous magnitude of this digital transformation paradigm shift that is now unraveling is not only changing work and management structures in the commercial world but will also affect society as a whole. Can the acclaimed benefits that have been reaped at the corporate and industry levels also materialize at the societal level? Are we standing on the threshold of a beautiful new world enabled by digital technologies, or are there unforeseen consequences that mean that we have inadvertently been nurturing a Frankenstein's monster about to be unleashed?

This book consists of five chapters:

Chapter 1: The Accelerating Pace of Change

The economic circumstances have, over the last century, progressed dramatically—from Henry Ford's assembly line principles with standardized products offered to potential customers, who simply had to accept what was on display, to the current highly individualized expressions of consumerism. This trend is now advancing further through the digital economy. Global brands have come to realize that purchasing preferences are capricious and that relying on consumer loyalty, based on a previous glorious past, is at their own peril. The pace of technological innovations interchangeably interacts with market demands, feeding off each other in symbiosis. Any corporation unwilling to adopt and embrace this reality will soon find itself sharing the destiny of the dinosaurs, a fate that no slick marketing campaign can rescue it from. However, the forces of change from customers as well as a combination of economic, political, regulatory, technical and even cultural trends has created a

mosaic of complexities that are now eclipsing all industries. It obligates them, both explicitly and implicitly, to an *Agilian* approach that only perhaps will enable them to remain competitive. It is beginning to dawn on most executives that the sense of being in control is nowadays merely a hazy illusion.

Chapter 2: Agile and then Some More Agile

That the agile framework was first adopted in the software industry was evolutionary, because frequent change requests had become a way of life for them. Eventually, they had to respond to the volatile environment. Hence, the advent of the Agile Manifesto. Although change was also accelerating in other industries, it was fitting that it took off in software development, because it was a relatively new industry where cultural institutions and folklore had not yet firmly formed and risked stagnating business practices. The positive impacts of applying agile methodologies on project works have been well documented, but are its principles also applicable at the organizational level? With markets in constant change in terms of new goods and services being introduced, and others becoming outdated at an unprecedented pace, the answer is assumed to be an affirmative. However, its success at the micro level has by and large failed to materialize at the macro level. Transforming a whole corporation to live and breathe by the agile ethos has proven difficult, and in most cases too difficult. Although there is no dearth of corporates claiming to be agile throughout, a closer look reveals that many employees, and indeed whole functions, are often only paying lip service. They still do things just the way they have always been doing them. Why has agile proven so hard to implement outside the IT function? Is it a lack of tangible deliverables and measurable metrics that makes it difficult to practically implement? Or is there a clash between the personality types thriving in bureaucracies and the philosophy that underpins agile? A hard-earned insight into many practitioners is that regardless of exquisitely drafted strategies, the success of agile rests on its employees' capabilities and willingness to truly embrace and live by its principles. This is something that has proven easier said than done, because it is commonly acknowledged to be an intense and demanding

form of working, and clearly not suitable for everyone. Most business executives and psychologists do agree that it is more about mindset than about understanding the framework, which itself is fairly uncomplicated. This brings agile's most contentious issue to the forefront: Can this mindset be taught or is there an innate agile personality?

Chapter 3: The Innovating Organization

There is a growing acceptance that agile will work for some part of the organization, but really not at all for others. A broad-brush delineation can be made between bureaucratic functions focused on control and ensuring compliance with various regulations on one side. These are defined by stability and conformity, and a general risk-averse attitude is the preferred personal skill set sought after, something to which agile can do little and can actually be detrimental. On the other side, business units are operating in turbulent environments where embracing calculated risks and developing commercial ventures through innovation are decisive for success. Here, agile has proven to be an important tool to accommodate these creative aspirations and leverage the opportunities that might arise. The ability to innovate has come to be one of the key defining success factors in the digital economy, but can it be facilitated? What organizational form is most suited to optimizing the arrangement of employees' creative abilities and competencies? The innovating organization strikes at the very core of the digital age, and corporations able to incorporate its features in their day-to-day operations will lead rather than follow, an enormous advantage in unpredictable consumer markets.

Chapter 4: The Renaissance Man

As businesses embark on digital transformations, it often becomes apparent that the traditional management styles are not performing satisfactorily, delivering results that are far below initial expectations, providing half-baked digital solutions where potential commercial gains are poorly captured and leveraged and far too often not even identified. Mismatches between technologies, the man-machine (dis-)connect, or organizational dysfunctionality are typically identified as root causes, but beneath them

lurks a more scathing problem: an inadequate leadership. This inadequacy rests on a lack of holistic insights backed by well-rounded skills and sets of knowledge that are required to understand all aspects of a digital transformation, including the stakeholders concerned—employees and customers. What is required to meet these challenges is a modern take of the *Renaissance Man*.

Chapter 5: A Beautiful New World?

The ongoing digital transformations are all-encompassing in that they are disrupting all industries in all geographies, and governmental agencies and bureaucracies are no exceptions. But what is the cumulative impact on society at large? Are we experiencing a bottom-up revolution? Is it a leap into the unknown with unfolding and little known consequences for education, the labor market, and the welfare system; and what about demographics? And can it also extend to politics? Are we set for an unavoidable conflict between top-down authoritarian governing models and the digital transformations that, by design, seek to break down hierarchies?

CHAPTER 1

The Accelerating Pace of Change

Change before you have to.

—Jack Welch,

American business leader (1935 to 2020)

With Industrialization Came Standardization

For the industrialization phase, commencing in the later part of the 19th century, to at all take off, mass production enabled by standardization was a necessity. It was accompanied by polished marketing campaigns (for the time) that titillated the demand for products that consumers often did not know existed or that they actually had any genuine need for. It was an enormous success, fueling economic growth that provided purchasing power to both a spiraling working class and an emerging middle class, rising from hordes of farm hands and kitchen maids. It was the first time in history really that the *plebeians* of society could so drastically improve their lives, materially at least. The man behind the managerial principles that came to form so much of the industrial era was the American engineer Frederick Winslow Taylor (1856 to 1915), the key tenets outlined in his book *The Principles of Scientific Management* from 1911. The book succinctly summed up the breakthrough insights that would optimize labor productivity using Taylor's scientific method and came to dominate industrial thinking for much of the century. His method allowed for unprecedented effectiveness and efficiency, as it took aim at mass production and quality control organized through hierarchy and specialization. It was the American industrialist Henry Ford (1863 to 1947) who perfected mass

production by applying the assembly-line principle on car manufacturing. Corporations were aptly depicted as mechanical systems, with employees being small cogs in a big wheel. Workers, as much as the consumers, were easily replaceable; hence, the era of industrialism saw numerous clashes between labor unions and employers to establish what minimum worker's rights should include. The assembly-line principles were adapted by most industries, albeit in various degrees, with standardization becoming the zeitgeist extending also into art and architecture. The streamlined Art Deco and Bauhaus replaced the pre-World War I free-wielding school of Art Nouveau. In all, there was little room allowed for individuality; basically it was a "one-size-fits-all" culture, famously highlighted in Ford's car selling slogan "*chose any color you like as long as it is black.*" The murkier aspects of this standardization mania were found in politics and notably both socialism and fascism sought to eliminate all traits of individualism and create the political mass man. In a bizarre manner, it somehow complemented capitalism's commercial mass man. In the end, it came with disastrous consequences for anyone falling outside the norms of political standardization. It found its ultimate sinister expressions in the concentration camps of World War II, as well as Soviet and Chinese communism's *gulags*, genocides which have left deep scars on the human soul that have remained till date. This was truly assembly-line murder factories working in overdrive, attempting to eradicate all tendencies that deviated from the ideal man template, not only physically but also in terms of political inclinations, and implicitly their consumer tastes and preferences. After World War II, with the horrors of collectivism being fully exposed, the consumers in the free world started to prioritize individual choices and preferences ahead of standardized products. Trend-sensitive corporations quickly adjusted themselves by starting to offer a wide variety of goods and services that defined the consumerism frenzy of the 1950s and 1960s. Although the managerial and organizational principles of Taylorism largely remained, over time it would prove incompatible to match the demand of such erratic consumers. The early 1950s saw a new industrial revolution commencing with the birth of a product that would herald the change to come—computers.^{1,2}

The Advent of Agile

The software development life cycle at the early days of the computer industry typically followed rigid project steps that it rarely deviated from. These include the following:

- the collation and gathering of requirements;
- the functional design phase of the software;
- the testing and verification of functionality;
- deployment; and finally
- the maintenance of the software.

Every step of the project life cycle had to be completed and approved before the ensuing phase could commence. If it was discovered that some changes were needed, the entire project had to revert to the initial step, and the cycle had to be redone and reapproved. It goes without saying that this project methodology was exceptionally cumbersome, inflexible, and end heavy in terms of deliverables. Users would typically have to wait for years to see the final product. It came to be labeled the waterfall methodology, reflecting the strict step-by-step approach. It had its origin in engineering and was adequate for its time, as technological innovations and upgrades that triggered change requests were relatively rare. But, from the late 1970s onward, software applications were starting to become more complicated. The first personal computers had become available and introduced to households and workplaces. Developers started to realize the limitations of the waterfall project methodology, and its iterative steps slowed down the software development considerably. More and more change requests were appearing throughout the project cycle, which were driven by the rapid advancement of technological innovations and demand for new functionality from the end-users. By the early 1990s, software development teams started to feel overwhelmed, having trouble keeping up with all the changes coming through. It came to be known as the application delivery lag. Experts in the industry estimated that the average time between a validated business need and the requested

application being tested and launched for deployment was about three years. Within that time frame, functionality, systems, and even entire industries were likely to have undergone changes, sometimes dramatically so. A lot of software were in fact already antiquated by the time of being introduced to their intended users. The verdict was out; the waterfall project methodology could not adequately adapt to the increased pace and the complexity of the change requests. It simply was not designed to deliver applications fast enough or respond effectively to the required changes that arose throughout the projects. But the waterfall methodology was deeply rooted in many practitioners' mentality, living by the devise; that the more time you spent planning, the less time you spend writing code, and the better that code would be. It was a hallmark of the reigning engineering ethos, deliberately process- and documentation heavy, where the emphasis was put on meticulous planning, which made a lot of sense for the construction business. But the challenges that software engineers and programmers now were facing had started to deviate considerably from the requirements of building roads and houses. Software projects no longer had its previous stability in terms of requirements and they had to be concluded a lot quicker than the previous years-long durations. To this came the insight that the requirements of what the software applications were supposed to be doing were never that exact versus the requirements of constructing a bridge or similar. There was always a bit of guesswork involved, and that by default meant that as the project proceeded some change requirements were inevitable. Typically, end-users could approximate the workflows, but it was never a precise model of the real world. The programmers were at times forced to second guess what to automate and how the applications really should work, and of course there were bound to be discrepancies.^{3,4} Hence, the programmer community began exploring new ways to approach software development. In the 1990s, a group of thought leaders developed a project management methodology that was more adapted to quickly react and respond to both change requests and technological upgrades. Various variations dawned, and in the early 2000s, the concept of agile software development was borne to describe the flexible nature of the development cycle. The pressures coming from a steady

stream of technological innovations and the increased pace of change requests were now seen not only in the software business but also in many other industries, such as automotive and aerospace manufacturing. They started to consider this alternative approach, as, for instance, the design time of a new car was cut almost in half in the 1990s, compared to the previous duration of six years; a quicker way of getting things done was necessary. In 2001, at a meeting in the resort town of Snowbird, Utah, some leading software practitioners published the *Manifesto for Agile Software Development*.⁵ The gist of the agile methodology is to enable project teams to quickly build working software and get it into the hands of end-users. Rapid feedback and a willingness to change have turned out to be key success factors. Little is set in stone at the beginning of the project, so if the project team is not entirely sure of the user needs, they first deliver an approximation and then collect feedback, making adjustments accordingly. The pioneers of agile gave some guiding tenets:

- Individuals and interactions precede processes and tools;
- Focus on building software over comprehensive documentation;
- Prioritize customer collaboration over contract negotiation, and;
- Quickly respond to change over following a plan.⁶

Most agile methods break the product development work into small increments that minimize the preplanning phase of the project. The work of the agile teams is assessed in regular meetings called daily sprints. These are accumulated into relatively short time frames, or timeboxes, that typically last from one to four weeks. They involve a cross-functional team covering all key functions: planning, analysis, design, coding, unit testing, and acceptance testing. At the end of each of these sprint periods, a minimum viable product (MVP) is presented to the stakeholders. This approach effectively reduces the overall project risk and allows the product development team to respond quickly to any required changes. A sprint might not add enough functionality to merit a formal release, but the objective is to have the aforementioned MVP available with hopefully few, if any, bugs. Typically, multiple sprints are required to release a software application, or new

features. To ensure that the emphasis is on getting things done, the delivery of functioning software is the primary performance metric. It has proved to work particularly well for complex software as well as product development projects that hold dynamic and nondeterministic characteristics.^{7,8}

So, what are then the main differences between the agile project methodology and the waterfall project methodology? Basically, the agile approach breaks down the project by creating several smaller deliverables that come to form the envisaged product, and which are likely to differ, sometimes considerably so, than what was initially expected. In the waterfall model, the deliverables are exactly planned at the onset of the project with the expectation of few deviations. The testing of the software is a separate step that follows the development step, whereas in agile, the coding and testing are completed in the same sprint. Therefore, the project setup has to adjust to the product rather than vice versa.⁹ Summarizing all these insights, The Agile Manifesto included 12 principles:

1. Deliver customer satisfaction by delivering valuable software continuously.
2. Always accept change of requirements no matter how early or late in the project.
3. Deliver software that works within a shorter timescale.
4. Both developers and business professionals must work closely together daily throughout the duration of the project.
5. Information is best transferred between parties in face-to-face conversations.
6. Motivate people to build a project by creating an environment of appreciation, trust, and empowerment.
7. Working software is the key measure of progress.
8. The agile process promotes sustainable development.
9. Continuous attention to excellence and quality in technical development and design boosts the agility.
10. Simplicity is a vital part of effective agile management.
11. Self-organized teams produce the best architecture, requirements, and design.
12. Teams should reflect through inspection and adaptation to be more effective.¹⁰

However, since the launch of agile, certain criticisms have emerged:

- Despite its intentions, it has often proven to be more developer-centric rather than user-centric.
- The transition from waterfall methods to agile is challenging, as employees can fail to adapt to the new mindset required.
- It focuses more on the (functional) requirements and code development rather than on product design, which sometimes needs a holistic approach. It happens typically when developing large-scale complex products, such as airplanes and spacecraft.
- Agile methodologies are sometimes incompatible in large organizations, as they are being prevented from true deployment due to heavy bureaucracy and vested interests that seek to uphold the status quo not approving of the method's flexible arrangements.¹¹

Agile methodologies will generally fail to work, not due to faulty implementation of the framework itself, which by any standard is rather simplistic, but through a lack of stakeholder commitment. The corporations' collective mindset, including key decision-makers, both formal and informal, may only pay lip service when endorsing the agile philosophy. This as they come to the realization that they are not capable or willing to live by it. A lack of support from executives means that agile methodologies are often implemented bottom-up by true enthusiasts and then confined only to development teams, possibly extending elsewhere in the IT department but rarely beyond that. Herein lies an accountability problem in large organizations, as if only small teams or functions adhere to it, its work practices will often come in conflict with other managerial and organizational models. Given the high intensity and focus that agile sprints demand, team members can then not spend the required time to meet their objectives, by being engaged in various aspects of the corporate bureaucracy. Key insights to a successful agile implementation highlight the need for capable individuals across different disciplines that can quickly act on change requests, and this in a nonhierarchical management style. While tools and processes are important, more important still is to have the right mindset to handle all these changes that can be expected and also proactively generated. Whether there is an agile mindset, perhaps

innate, is a critical question that will be discussed in a later chapter.¹² In practice though and particularly in larger corporations, hybrid models combining elements of agile and waterfall are often deployed, making projects more document- and preplanning-heavy than what the pure agile methodology prescribes.¹³

Scrum

Originating from the overarching agile framework are several subsets, out of which one of the most popular is scrum. The term is borrowed from the sport of rugby, where players with different roles come together in team collaboration to restart a halted game seeking to gain control over the ball. The scrum methodology is backed by a quite extensive body of research that has shown that the ‘best’ results occur when small project teams are allowed to self-organize and operate against objectives to which they have the freedom to decide way forward, this rather than being given specific assignments and micro-managed. Scrum techniques have been calibrated for small teams, typically no more than nine project members. The edifice of scrum rests on three pillars, namely transparency, inspection, and adaptation, which allows it to respond to feedback timely. These three pillars require trust and openness in the team, which the following five values of scrum seek to promote:

1. *Commitment*: Team members individually commit to achieving their team goals, each and every sprint.
2. *Courage*: Team members know they have the courage to work through conflict and challenges together so that they can do the right thing.
3. *Focus*: Team members focus exclusively on their team goals and the sprint backlog; there should be no work done other than through their backlog.
4. *Openness*: Team members and their stakeholders agree to be transparent about their work and any challenges they face.
5. *Respect*: Team members respect each other to be technically capable and to work with good intent.¹⁴

The project work is broken down into sprints, no longer than one month in duration, and its progress is tracked in daily stand-up meetings, called

daily scrums (typically around 15 minutes). During the daily scrum, each team member is expected to be prepared to respond to three questions:

- What did I complete yesterday that contributed to the team meeting our sprint goal?
- What do I plan to complete today to contribute to the team meeting our sprint goal?
- Do I see any impediment that could prevent me or the team from meeting our sprint goal?

Through these ongoing scrums, the project team should be able to identify early on if and when their work starts to deviate and needs to be steered back on path to keep the progression going.^{15,16} One of the key scrum tools is the product backlog, which is a model of the work to be done and includes an ordered list of the product requirements. User stories are deployed, which being only a couple of sentences long aim to capture a description of a software feature from the end-user perspective. This is done by highlighting the type of user, what they want and why, seeking to create a simplified description of a requirement.¹⁷ There are three key roles in a scrum project that differ from the traditional project management setup: product owner, scrum master, and team members. The product owner oversees all the business requirements to ensure the right product functionalities are being built and executed correctly. The product owner is required to be able to prioritize and negotiate change requests, maintain good working relations with the project team and stakeholders, and ultimately have both the authority and integrity to make decisions about the project. The role is critical as it represents the business, typically also the customers, and the functionality they seek, something that requires intense interaction with the development team. But in real life, it often happens that the product owners cannot dedicate enough time, as they often are far too busy focusing on their business activities, whether that be sales, business development, or serving clients. Similarly, the team members can be drawn into other engagements, such as having to involve in unplanned RAD (rapid application development) activities or software support and maintenance. This lack of allocating sufficient time is one of the main sources of failure in scrum projects.

The scrum master's role is more of a coach rather than project manager, helping the project team work effectively together in accordance with the scrum principles. It is very much a hands-on role where the scrum master helps the team by removing anything that can impair performance, facilitates meetings, keeps tab on progress, and is in charge of handling general problem-solving but typically does not have people management responsibilities. In short, the scrum master is expected to be an expert on the scrum framework and the various tools deployed. The scrum master ultimately acts as a buffer between the team and any distracting influences. It is a role that can coexist side by side with the project manager who is in charge of project scope, cost, personnel, risk management, and other typical project manager responsibilities.

One of the most important roles beyond the educational aspect, and one of the well-documented key success factors of the scrum methodology, is the promotion of self-organization and cross-functionality within the project team. The team members themselves are expected to be empowered to lead this effort, with the scrum master's role solely being of an advisory nature. It is also one of the most sensitive points, as it challenges, and sometimes confronts, existing career hierarchies. So, finding a workable synergy between product owner, scrum master, and team members is a decisive factor that comes to determine much of a scrum project's success.^{18,19}

Some of the limitations of the scrum methodology have been found to be as follows:

- Problems can arise when the team cannot physically sit together and/or are only involved part-time, having to simultaneously manage competing tasks. Scrum advocates close an ongoing interaction, as that enables the perceived “sum is more than the components” effect.
- If team members lack well-rounded skills and are merely highly specialized in single-domain expertise areas, the cross-functional self-organizing effect might not arise as disparate jargons and narratives hinder connectivity.
- Products with many external dependencies. The scrum methodology rests on dividing product developments into short sprints, out of which some might come with plenty of external dependencies,

such as user acceptance testing or coordination with other teams. These can lead to delays and the failure of individual sprints, putting the whole project at risk.

- Products that are mature, or legacy, needing an upgrade, or have required regulated quality controls are less suitable, as in scrum, product increments should be fully developed and tested in a single sprint. Products that need large amounts of safety testing or similar, such as medical devices, components for airplanes, or nuclear power plants, for each release are therefore better suited for the waterfall methodology.²⁰

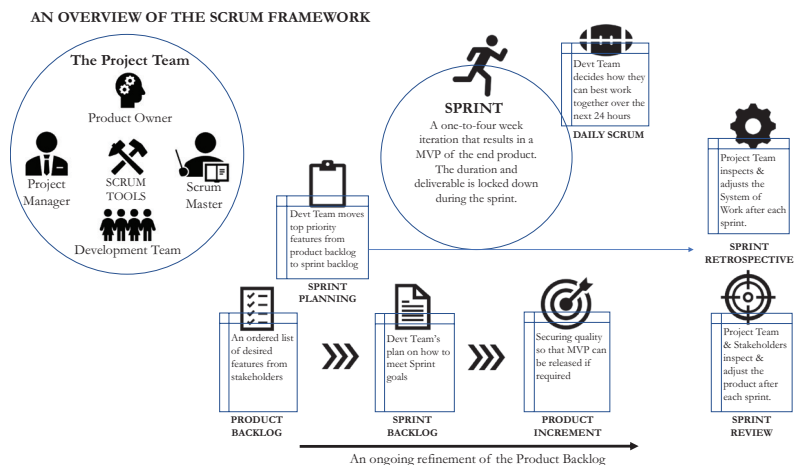


Figure 1.1 A graphical depiction of the scrum framework with its workflow and key tools

How Change Became the Only Constant

The old platitude that change is the only constant has now become a worrisome fact that most businesses have to adjust to. Resting on old laurels and expecting customer loyalty have come to prove to be a proverbial death trap. Companies such as Blockbuster and Kodak are stark reminders of the fate that awaits corporates that are dominated by inertia and ignorance. It is becoming evident to most executives that they must learn to live with this uncertainty and find a framework that allows their businesses to embrace and thrive on it, or at least not be succumbed by it.

But How Did We Get to This Point?

We have come a long way since the industrial era and standardization of products offered with a “take it or leave it” approach. Consumerism in the digital economy is instead focused on individualization of goods and services to an extent never seen before, all underpinned by an accelerating pace of technical innovation. This is putting corporations under an enormous pressure to at all times make sure that their offerings are alluring and competitively priced. Marketing campaigns can now only do so much to rescue a dying brand whose products have become outdated, being a thing of the past with potentially only nostalgic value. And it is quickly moving beyond just manufacturing; the high-end service sector that up till recently had been little impacted by automation and digitalization efforts is now being hit with full force. What all this is adding up to are that corporations and employees alike will have to learn how to live in volatile markets. They must, in the true sense of the word, become agile to survive so that they quickly can respond and adapt to an evolving business environment. It means, to always be ready to deliver greater value in every area of the organization and hopefully reap the commercial benefits rather than face extinction. If customers cannot get the products and services they expect, they will simply go somewhere else; it is only a click away. Customers rightfully demand and expect that businesses will be able to rapidly change in order to meet their needs. Being big is no longer beautiful in many industries, in fact smaller players and start-ups often benefit from not having built up costly bureaucracies. This as scaling up not so much any longer depends on expanding manpower, as digitalization efforts by-and-large can easily produce more volume.

The agile methodology has proven from the project management perspective to provide a capable framework to deal with change. Might it also be applicable at the organizational level?

Endnotes

1. D. Nelson. 1970. *Frederick W. Taylor and the Rise of Scientific Management* (Madison: MIT Press).
2. A.E. Musson and E. Robinson. 1969. *Science and Technology in the Industrial Revolution* (Toronto, ON: University of Toronto Press).

3. Iacocca Institute. 1991. *21st Century Manufacturing Enterprise Strategy: An Industry Led View* (Bethlehem, PA: Iacocca Institute, Lehigh University).
4. G. Lee and W. Xia. 2010. "Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility," *MIS Quarterly* 34, no. 1, pp. 87-114.
5. K. Beck, J. Grenning, R.C. Martin, M. Beedle, J. Highsmith, S. Mellor, A. van Bennekum, A. Hunt, K. Schwaber, A. Cockburn, R. Jeffries, J. Sutherland, W. Cunningham, J. Kern, D. Thomas, M. Fowler, and B. Marick. 2001. *Manifesto for Agile Software Development* (Agile Alliance).
6. K. Beck, J. Grenning, R.C. Martin, M. Beedle, J. Highsmith, S. Mellor, A. van Bennekum, A. Hunt, K. Schwaber, A. Cockburn, R. Jeffries, J. Sutherland, W. Cunningham, J. Kern, D. Thomas, M. Fowler, and B. Marick. 2001. *Principles behind the Agile Manifesto* (Agile Alliance).
7. A. Moran. 2014. *Agile Risk Management* (Cham: Springer Verlag, SpringerBriefs in Computer Science).
8. K. Beck, J. Grenning, R.C. Martin, M. Beedle, J. Highsmith, S. Mellor, A. van Bennekum, A. Hunt, K. Schwaber, A. Cockburn, R. Jeffries, J. Sutherland, W. Cunningham, J. Kern, D. Thomas, M. Fowler, and B. Marick. 2001. *Principles behind the Agile Manifesto* (Agile Alliance).
9. B. Boehm and R. Turner. 2004. *Balancing Agility and Discipline: A Guide for the Perplexed* (Boston, MA: Addison-Wesley).
10. K. Beck, J. Grenning, R.C. Martin, M. Beedle, J. Highsmith, S. Mellor, A. van Bennekum, A. Hunt, K. Schwaber, A. Cockburn, R. Jeffries, J. Sutherland, W. Cunningham, J. Kern, D. Thomas, M. Fowler, and B. Marick. 2001. *Principles behind the Agile Manifesto* (Agile Alliance).
11. K. Beck and C. Andres. 2004. *Extreme Programming Explained: Embrace Change*. 2nd ed., (The XP Series) (Boston, MA: Addison-Wesley).
12. S. Ashmore and K. Runyan. 2014. *Introduction to Agile Methods* (Boston, MA: Addison-Wesley).
13. A. Moran. 2015. *Managing Agile: Strategy, Implementation, Organisation and People* (Cham: Springer Verlag).

14. K. Rubin. 2013. *Essential Scrum. A Practical Guide to the Most Popular Agile Process* (Boston, MA: Addison-Wesley).
15. K. Schwaber. 2004. *Agile Project Management with Scrum* (Washington, DC: Microsoft Press, Developer Best Practices).
16. G. Verheyen. 2013. *Scrum: A Pocket Guide (A Smart Travel Companion)* (Zaltbommel: Van Haren Publishing, Best Practice).
17. P. Deemer, G. Benefield, C. Larman, and B. Vodde. 17 December 2012. "The Scrum Primer: A Lightweight Guide to the Theory and Practice of Scrum," (Version 2.0), InfoQ. https://www.infoq.com/minibooks/Scrum_Primer/ (accessed 1 June 2020).
18. R. Pichler. 2010. *Agile Product Management with Scrum: Creating Products that Customers Love* (Upper Saddle River, NJ: Addison-Wesley).
19. P. Deemer, G. Benefield, C. Larman, and B. Vodde. 17 December 2012. "The Scrum Primer: A Lightweight Guide to the Theory and Practice of Scrum," (Version 2.0), InfoQ. https://www.infoq.com/minibooks/Scrum_Primer/ (accessed 1 June 2020).
20. K. Rubin. 2013. *Essential Scrum. A Practical Guide to the Most Popular Agile Process* (Boston, MA: Addison-Wesley).