TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM School of Management

Behavioral Aspects in Project Management

—

A Comparison of Agile and Waterfall Project Management

Tobias Lieberum

Vollständiger Abdruck der von der TUM School of Management der Technischen Universität München zur Erlangung eines

Doktors der Wirtschafts- und Sozialwissenschaften
(Dr. rer. pol.)

genehmigten Dissertation.

Vorsitz:		Prof. Dr. Martin Grunow
Prüfer der Dissertation:	1.	Prof. Dr. Rainer Kolisch
	2.	Prof. Dr. Michael Kurschilgen

Die Dissertation wurde am 24.10.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Management am 15.06.2023 angenommen.

Behavioral Aspects in Project Management

A Comparison of Agile and Waterfall Project Management

Tobias Lieberum

Acknowledgments

I would like to thank my advisors and co-authors Prof. Dr. Rainer Kolisch, Dr. habil. Sebastian Schiffels, and Prof. Evgeny Kagan, Ph.D., as well as my colleagues at the Chair of Operations Management at the Technical University of Munich.

My research was generously supported by the Hanns Seidel Foundation with funds from the German Federal Ministry of Education and Research.

I am very grateful for the support of my fiancée Lisa da Silva Joao, my parents Dr. Victoria and Dr. Dr. Reinhard Lieberum, my siblings Anne, Dr. Judith-Lisa, and Jan Nikolaus Lieberum, as well as my friends.

Thank you.

Munich, October 2022

Abstract

This dissertation presents two studies that experimentally compare behavioral effects of agile project management and traditional waterfall project management on project performance and execution. Agile development practices are designed to foster innovation and performance. Despite their appeal to practitioners, there is little research into their effectiveness and applicability to different work environments. We develop new experimental approaches to studying human behavior under agile and traditional project management. The experimental results suggest ambiguous effects. The first study shows quantitatively higher project performance from agile project management. In a 2×2 experimental design, we compare the effects of agile sprints—short-term project phases characterized by time-boxed progression from one sprint to the next and self-imposed, phase-specific output goals—with those of traditional project management. Our laboratory results provide strong evidence of higher quantitative project performance when working in agile sprints. They can mitigate a newly described behavioral effect present in traditional project management: When people are free to progress at their pace, they spend too much time on early tasks at the expense of later ones. We refer to this effect as "Progression Fallacy". The second study shows qualitatively ambiguous performance effects from agile project management. We study human behavior in two different work environments, product innovation represented by a creative design task and business model innovation represented by a more structured search task. Our results suggest that agile development practices improve performance in the creative design task, but harm performance in the more analytical search task compared to traditional project management. The effects of Agile on performance are not uniform but depend on the setting and performance measure. Agile project management helps to achieve minimum viable solutions early on. However, it can reduce top performance and lead to more incremental rather than radical innovation. Together, the two studies caution against uniform adoption of the agile approach. The choice of the approach should depend on the nature of the project and the organization's desired risk-return profile. Not only do the behavioral insights of this dissertation apply to project management, but they are also relevant in the broader context of task completion.

Contents

List of Tables

List of Figures

1. Introduction

Worldwide project management is estimated to burn USD 1 million every 20 seconds through poor project performance (Project Management Institute 2018). Such poor performance often manifests itself in cost overruns, underachievement of project goals, and delayed project completion. A major cause of project delays is time-elasticity of work due to behavioral effects among the project members (Gutierrez and Kouvelis 1991). These can occur at all stages of a project. Traditional project management comprises the five stages initiation, planning, execution, monitoring, and closure (Project Management Institute 2017a). Substantial bodies of work have improved the understanding of behavioral effects impacting four of these five: For example, when initiating a project, procrastination, a perception of higher costs of immediate effort compared to the costs of future effort (Wu et al. 2014), needs to be managed. In planning projects, humans tend to succumb to an optimism bias in predictions of how long the completion of a task will take, an effect known as the planning fallacy (Kahneman and Tversky 1979) or Hofstadter's law (Hofstadter 1979). In monitoring projects, an escalation of commitment can result in the continuation of underperforming initiatives aiming to recover losses (Bendoly et al. 2010). The closure of a project often "expands so as to fill the time available for its completion", an effect known as Parkinson's law (Parkinson 1957, p. 2). All of these behavioral effects can result in project delays, if not appropriately managed.

Relatively little is known about behavioral effects during the stage of project execution. A thorough review of the operations management literature finds to the best of my knowledge less research focus on the process of actual project execution once the preparatory steps initiation and planning are completed and the project closure is yet to come. Thus, it is so far less clear, in which way behavioral effects with potentially negative impact on project performance come into play at this stage. This dissertation aims to contribute a behavioral perspective on project execution, linking the behavioral operations literature with the project management literature.

As a way to improve project performance, execution, and innovation, agile project management is currently being discussed in many organizations, but only in few academic contributions. This creates a considerable discrepancy between practical relevance and scientific understanding. In practice, agile development has attracted considerable attention in recent years, particularly in software development, where Agile has reached almost uniform adoption. It is used to improve innovation and performance both in new products and services but also in established processes (Rigby et al. 2016). Indeed, most organizations today report the adoption of at least some agile practices, a development that holds true across industries and also across organizational functions, including research and development, marketing and sales, and even human resources and finance (Panditi 2018).

Despite their appeal to practitioners, there is little research into the effectiveness of agile practices. As a result, the implications of Agile for different types of projects are unclear, making it difficult for managers to decide not only whether it is worth adopting Agile (or not), but also which components of Agile will be effective in their organizations. Conboy (2009, p. 340) defines agility as "the continual readiness [...] to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment." In practice, Agile is a bundle of implementation methods, including the Dynamic Systems Development Method (Stapleton 1998), Feature Driven Development (Coad et al. 1999), eXtreme Programming (Beck 2003), Crystal (Cockburn 2005), Kanban (Anderson 2010), Lean Software Development (Poppendieck and Poppendieck 2010), Scrum (Schwaber and Sutherland 2017), and combinations of these, such as Agile Modeling (Ambler 2002) and Scrumban (Ladas 2009). Most of these share common principles, ranging from working in sprints to early prototyping, assigning decision-making authority to the project teams, visualizing workflows, moving product validation and testing to earlier stages of development, and more.

The by far most widely used approach is Scrum (Scrum Alliance 2015), which thus guides the selection of agile principles in this dissertation. Scrum is centered around three core elements (Schwaber and Sutherland 2017, pp. 3, 4, 9): First of all, “the heart of Scrum”, short-term and time-boxed project phases called sprints, during which self-contained, usable, and potentially releasable project increments are to be created. Secondly, “the essence of Scrum”, a small, highly flexible, and adaptive team of people. Thirdly, iterative, continuous improvement of product, team, and working environment.

This dissertation experimentally compares human behavior in project execution under agile project management with that under traditional waterfall project management. To the best of my knowledge, it is one of the first academic investigations of the strengths and weaknesses of Agile in a controlled laboratory setting. The advantage of the laboratory approach is “to cleanly establish causality [...] compared to other empirical methods. In the laboratory, causality is established by directly manipulating treatment variables at desired levels, and randomly assigning participants to treatments. Random assignment ensures that treatment effects can be attributed to the treatment variables and not be confounded by any other, possibly unobservable, variables. Other empirical methods rely on existing field data, so neither random assignment nor direct manipulation of treatment conditions is possible, so causality cannot be directly established” (Katok 2018, pp. 2–3). Laboratory data can thus support or reject—sometimes only anecdotal—evidence observed in the field and be used to investigate underlying mechanisms in a controlled and replicable manner.

The internal validity of the controlled experimental design helps academia to develop, test, and advance theory and models, while helping practitioners to make informed decisions. This interlink between operations management research and managerial practice is essential to facilitate the implementation of scientific insights in business organizations (Loch and Wu 2007). “When it comes to implementation, the success of operations management tools and techniques, and the accuracy of its theories, relies heavily on our understanding of human behavior” (Bendoly et al. 2006, p. 737). In this dissertation, I therefore focus on human behavior and decision making in project execution under agile and traditional project management schemes. I leave aside the team component of project work as “by focusing on what individuals are actually doing [...] we can understand these processes better” (Grushka-Cockayne et al. 2018, p. 17). Note that team composition in general has been well researched outside the context of Agile; for an overview, see Fu et al. (2016).

Project execution can be stylized by modeling a project P as a matrix with m components, each consisting of up to n features:

$$P = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \dots & \dots & \dots & \dots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix}$$

Traditional waterfall project management and agile project management differ in the sequence of completing components and features. Whereas with Waterfall, the components are sequentially developed in project phases of various lengths, with Agile, features of different components are developed in parallel in sprints of equal length. Each sprint is supposed to end with an iterative increment of the project components (Schwaber and Sutherland 2017), whereas with Waterfall, work on the last component is only started in the last project phase (Royce 1987). This dissertation investigates the behavioral effects of these differences in two experimental studies.

The first study, titled "Should We All Work in Sprints? How Agile Project Management Improves Performance", shows that agile project management can lead to quantitatively better work performance than traditional project management. It is based on an article by Lieberum et al. (2022) written in conjunction with this dissertation and published in *Manufacturing & Service Operations Management*. The second study, titled "One Size Does Not Fit All: Strengths and Weaknesses of the Agile Approach", shows that agile project management has qualitatively ambiguous effects and can—depending on the setting—lead to qualitatively worse work performance than traditional project management. It is based on a working paper by Kagan et al. (2022) written in conjunction with this dissertation (*Reject & Resubmit* at *Management Science* at the time of submission of this dissertation, preprint online at *SSRN*).

In the first study, we present what to our knowledge is the first experimental study on the effects of agile sprints on project performance and execution. We contribute an operationalization of project execution as a stylized real-effort task along with three main findings to the field of project management. First of all, we show how in traditional project management without forced progression from one project phase to the next there is a risk of delayed progression, as project agents spend too much time on early project phases at the expense of later ones. We refer to this newly described effect as “Progression Fallacy”. Secondly, time-boxed progression in agile sprints mitigates the Progression Fallacy and improves the overall performance. Thirdly, we provide evidence that self-imposed, phase-specific output goals with flexible progression, as are common in traditional project management, can amplify progression delay and distort effort. This can be avoided by combining self-imposed, phase-specific output goals with time-boxed progression, as is common in agile project management.

These results have direct managerial implications. Our findings suggest that managers should not only monitor well-known biases in project management, such as planning fallacy, procrastination, or Parkinson’s law. If projects are executed with flexible progression through the project phases, as is common in traditional project management, particular managerial attention is also needed on the optimal time allocation across project phases, in particular when ambitious goals are to be achieved. Otherwise, project members succumbing to overdelivering started tasks can be a driver of underperformance and delays. Working in agile sprints with time-boxed progression and phase-specific output goals can facilitate on-time task completion and improve performance. However, here particular managerial attention is needed for setting sufficiently motivational goals.

In the second study, we experimentally examine how agile project management techniques affect innovation performance. Among some practitioners, agile project management is currently propagated as a panacea for all innovation problems. Others criticize that agile project management might improve quantitative project success but might also inhibit the qualitative degree of innovation as project members only ever work on short-term solutions (Gwosdz 2020). To investigate whether such a generalization is permissible, we use two different stylized fields of innovation work in our experimental design: On the one hand, a creative design task resembling the Scrabble game, which represents product development, and on the other hand, a modified version of the search task Lemonade Stand (Ederer and Manso 2013), which represents business model innovation. Our results suggest that agile project management improves performance in the creative design task, but harms performance in the search task compared to traditional waterfall project management. The effects of Agile on performance are not uniform, but depend on the setting and performance measure. Agile helps to achieve minimum viable solutions early on. However, it can reduce top performance and lead to more incremental rather than radical innovation. We argue that the time pressure from working in short agile sprints can lead to the avoidance of larger, and therefore riskier, developments in favor of incremental improvements. Finally, whereas project members may be more satisfied with having more autonomy under agile project management, we find that it does not lead to significantly higher performance (nor does it cause harm). Instead, most of the difference in performance is due to the iterative nature of Agile, rather than project members having control over time allocation.

Managers need to be cognizant of both the benefits and drawbacks of agile project management. An advantage of Agile is the increased sense of urgency resulting from dividing work into sprints. Especially in creative development, this can help project members overcome initial roadblocks and immerse themselves in the task at hand. However, for more analytical tasks, the sense of urgency can backfire as it leads to insufficient exploration of available solutions and an early commitment to a potentially suboptimal strategy. Whereas such short-termism effects are anecdotally known to some agile practitioners, we are not aware of any studies that rigorously document them. The managerial implications are that the choice of the project management approach should depend on the nature of the project and the organization's desired risk-return profile.

This dissertation is structured as follows: The manuscripts of the two studies are reprinted in Chapters 2 and 3. Chapter 4 summarizes the findings of both studies and discusses the divergent effects of agile project management on project performance and execution. Compared to traditional waterfall project management, our data suggest that agile project management can foster quantitative performance in project execution, but in some cases can harm qualitative performance.

2. Experimental Study 1: Should We All Work in Sprints? How Agile Project Management Improves Performance

Co-authors: Sebastian Schiffels, Rainer Kolisch

Manuscript published in *Manufacturing & Service Operations Management* (Volume 24, Issue 4, July–August 2022, Pages 2293–2309, DOI: 10.1287/msom.2022.1091).

Abstract. ***Problem definition*:** Agile project management, in particular Scrum, is enjoying increased use in practice, despite only scant scientific validation. This article explores how agile project management impacts project performance and execution. We compare the effects of agile sprints—short-term project phases characterized by time-boxed progression from one sprint to the next and self-imposed, phase-specific output goals—with those of traditional project management. ***Methodology / results*:** We decompose the two sprint elements of time-boxed progression and self-imposed, phase-specific output goals as factors in a 2×2 experimental design. We then conceptualize project execution as a simple real-effort task and conduct a controlled laboratory study. For a given duration, participants perform better with time-boxed progression, as without it, i.e. with flexible progression, they spend too much time on early project phases at the expense of later ones. We refer to this effect as "Progression Fallacy" and show how it differs from well-known behavioral effects that cause project delays. Introducing self-imposed, phase-specific output goals in combination with time-boxed progression, as proposed by Scrum, does not significantly improve performance when compared with time-boxed progression alone. However, the combination of self-imposed, phase-specific output goals and flexible progression, as is common in traditional project management, amplifies the Progression Fallacy, with the result that goal-setting has a negative performance effect. In two control treatments, we show that the Progression Fallacy is robust to planning and progression prompts, despite some mitigation. ***Managerial implications*:** This study contributes evidence of higher project performance when working in agile sprints, which mitigate behavioral flaws present in traditional project management. Not only do these behavioral insights apply to project management, they are also relevant in the broader context of task completion.

2.1. Introduction

Agile project management is accorded considerable attention in practice, with 88% of the business organizations surveyed by the Project Management Institute (2017b) making at least some use of agile approaches. Despite its relevance to practitioners and repeated calls by the research community for comparative evaluations with traditional project management approaches (see Hall 2016), to the best of our knowledge and based on a review of the operations management literature, there are no major contributions on the operational effects of agile project management, with the sole exception of Kettunen and Lejeune (2020), who provide theoretical evidence of earlier attainment of return targets when using agile project management than with traditional project management. More than twenty years later, Ettlie's conclusion (1998, p. 4) that "other than case histories [. . .] no systematic evidence demonstrates how agility works" is thus still widely valid for project management, leaving it to practitioners to judge whether and why to favor agile over traditional project management.

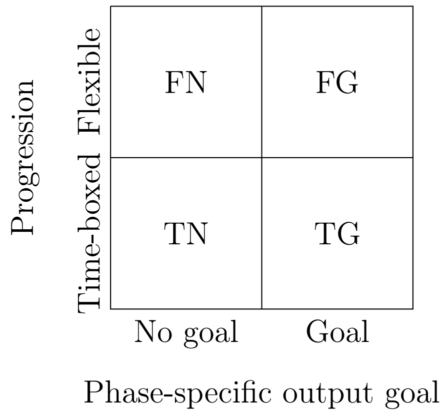
Of the various agile approaches, Scrum has developed into by far the most widely used (Scrum Alliance 2015). At the "heart of Scrum" are short project phases called sprints (Schwaber and Sutherland 2017, p. 9). Sprints are characterized by time-boxed progressions, i.e. project phases of equal length, which cannot be extended, irrespective of what achievements are made in a phase, and by self-imposed, phase-specific output goals, i.e. self-contained increments to be created during a sprint. We investigate how agile sprints impact project performance and execution as compared with traditional project management, where performance is the value generated towards the project objectives and execution is the workload completed in sequential project phases. To enable consistent comparison, we exclude further elements of Scrum, such as the use of small autonomous teams and continuous improvement, which have been researched outside the context of Scrum (for an overview, see Fu et al. (2016) regarding team composition and Zangwill and Kantor (1998) regarding continuous improvement). This focus on the individual is guided by the behavioral operations management literature as "projects are executed, to a certain extent at least, by individuals. Therefore, how individuals operate when working on a project, often in a temporary setting with limitations on time and resources, will impact the success of the project"

(Grushka-Cockayne et al. 2018, p. 382).

The design of agile sprints has two implications: First of all, project execution in sequential time-boxes differentiates agile from more continuous traditional project management. Traditional project management often follows a flexible waterfall approach (Royce 1987), with phases of various lengths and no strictly enforced progression through the project phases. By altering the initial project plans, project agents decide flexibly when to proceed from one project phase to the next. In agile project management, this progression autonomy is reduced through time-boxing. Secondly, because in the goal-setting equation of *output per time* the denominator time is fixed in agile sprints, goal-setting and completion are more standardized than in traditional settings where the time horizon varies for each goal and may be extended if a goal is not met. These differences to traditional project management are up to this point motivated more by practical wisdom than by academic evidence. We decompose the two elements of agile sprints, time-boxed progression and self-imposed, phase-specific output goals, as axes in a 2×2 factorial design in order to enable an experimental investigation of their effects on project performance and execution in both isolation and interaction. Four combinations result from this design: time-boxed with no goals (TN), flexible with no goals (FN), time-boxed with goals (TG), and flexible with goals (FG), as displayed in Figure 2.1. Time-boxed with goals (TG) stylized represents agile project management, whereas flexible with goals (FG) represents traditional project management.

Based on a simple experimental real-effort task, in which participants face a highly transparent time allocation trade-off between stylized, sequential project phases, we explore three core findings. First of all, flexible progression in the absence of goal-setting (the combination FN) results in delayed progression from early to later project phases, in other words, participants spend too much time on early project phases at the expense of later ones. We refer to this effect as "Progression Fallacy" and discuss its differences to other well-researched behavioral effects that also cause project delays. In two control treatments, we show that the effect is robust to planning and progression prompts, despite some mitigation. Secondly, if progression is enforced by time-boxes (the combination TN), the time spent per project phase is by design balanced. This results in a higher overall performance compared to flexible progression. Thirdly, whereas the small

Figure 2.1. 2×2 Decomposition of Progression and Phase-Specific Output Goals



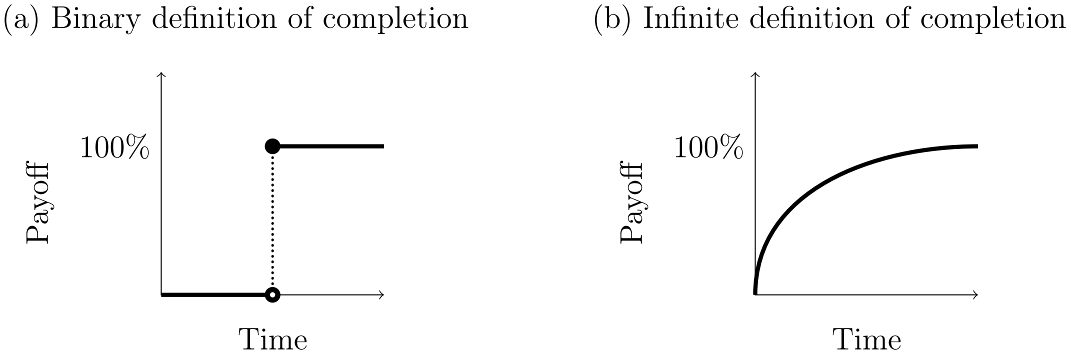
positive performance effect arising from combining time-boxed progression with self-imposed, phase-specific goal-setting (the combination *TG*), as proposed by Scrum, is not significant compared to time-boxed progression alone, we demonstrate a negative interaction effect on performance from combining flexible progression with self-imposed, phase-specific goal-setting (the combination *FG*), as is common in traditional project management. Whereas in the time-boxed setting, participants self-impose their goals against a constant benchmark of their past performance and progress irrespective of the degree of goal achievement, in the flexible setting, they set them against a past performance biased by the Progression Fallacy. This results in overly ambitious goals and amplifies the Progression Fallacy, as participants spend even more time on early project phases pursuing ambitious goals.

Our study provides the first experimental analysis that we are aware of on how agile project management impacts project performance and execution when compared to traditional project management. We make two contributions: Firstly, we introduce an operationalization of project execution as a stylized real-effort task, which enables investigation of the project agents' time allocation and effort choice across the project phases under experimental control. Secondly, we provide evidence of higher project performance from working in agile sprints, which mitigate behavioral flaws present in traditional project management. We present a newly described effect of delayed progression in traditional project execution, which is robust to planning and progression prompts and amplified by goal-setting. This implies that managers should not only monitor well-known biases in project management, such as planning fallacy, procrastination, or Parkinson's law. If projects are executed with flexible progression, as is common in traditional project management, particular managerial attention is also needed regarding the optimal time allocation across project phases, especially when the goals to be achieved are ambitious. One way to achieve this is to combine time-boxed progression and self-imposed, phase-specific output goals in agile sprints, which facilitates on-time task completion, improves the exerted effort, and results in higher overall performance. These insights are not only relevant to project management, but they are also transferable to the broader context of task completion.

2.2. Project Management Setup and Hypotheses

Projects are characterized by the presence of defined start and end dates (Kerzner 2013). Achieving the project objectives, i.e. the target performance, within this time frame is a core aim of project management (Goh and Hall 2013). For many projects, the achievement of objectives can be thought of as an additive function of the payoff from sequential project phases, during which different aspects of the product or service, such as the overall concept, prototype, final form, market entry strategy, etc. are developed. If the project objectives can be precisely defined, they can be broken down into a clear, binary definition of the completion state of the project increments, developed in sequential project phases. An illustrative example is the construction of a road, for which an entire list of predefined steps needs to be concluded (e.g., dispersal of normed layers of bitumen) for the completion state to be reached. If the project is halted halfway through, the completion state cannot be achieved. Equally, further work once the list of predefined steps has been completed is not defined. Thus, project phases with a binary definition of completion have a stepwise payoff function (Figure 2.2(a)). However, if the project objectives cannot be precisely defined, it is often not possible to create a binary definition of the completion state of the project increments. An innovation project is a typical example. Here, even the greatest progress can be further improved at every development stage. To take this to its extreme, the completion state of every project phase is infinite. For a given ability of the project agent

Figure 2.2. Extremes of Project Phases


(a) Binary definition of completion

(b) Infinite definition of completion

and a given random noise (e.g. creativity during ideation or unknown customer taste), the payoff function $p_i(x_i, e_i)$ of these project phases $i \in \{1, \ldots, n\}$ for an invested time x_i and effort e_i has a positive slope with marginally decreasing growth, converging towards an unattainable maximum, i.e. $\lim_{x_i \to \infty} p(x_i, e_i) = 100\%$ (Figure 2.2(b)). Further time spent on a project phase that is already well advanced will still result in a positive incremental payoff. However, at some point the marginal value add will no longer justify the corresponding time allocation.

These two project phase types form the extremes on a continuum. Although projects might include project phases of both types as well as hybrids, this paper focuses on the infinite type. Because there is no clear definition of completion, project phases of this type require a decision by the project agent to terminate one phase and progress to the next one once the extent to which the originally planned project increment has been delivered is deemed sufficient. Such progression decisions can be exposed to behavioral biases, such as a preference for the most satisfying task (Boudreau et al. 2003). We model such potential bias by a subjective weighting w_i of the time allocated to each project phase. For a fixed total duration t of a project, the sum of the duration x_i of all project phases is limited. The time spent per project phase is then an optimization trade-off between the incremental return of time invested in current and future project phases and depends on subjective weighting and effort. It is solved dynamically for each sequential project phase:

$$\max\left\{\sum_{i=1}^{n} w_i\cdot p_i(x_i,e_i) \mid \sum_{i=1}^{n} x_i \le t, x_i \ge 0, \forall i=1,\ldots,n\right\}$$

Traditional and agile project management differ in the degree of freedom that the project agent has in making these sequential decisions through different progression and goal-setting regimes. In this paper, we compare their effects on project performance and execution in a stylized setting, in which the payoff function with marginally decreasing growth is the same in all project phases. We first focus on the difference between time-boxed and flexible progression in isolation, followed by the goal-setting component given a time-boxed or flexible progression.

2.2.1. Role of Time-Boxed Progression

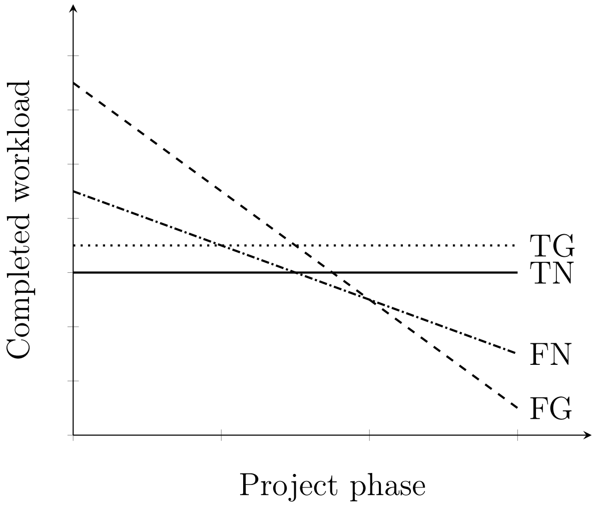
In traditional project management, project agents flexibly progress through the project phases, potentially adjusting initial project plans as they proceed. This autonomy in altering and timing initial operating strategies can create value in the context of R&D uncertainty (Huchzermeier and Loch 2001) and improve worker motivation (Pasmore 1988). However, such autonomy can also expose project agents to behavioral flaws in their time and effort allocation. Hints of such flaws can be found in both field and experimental studies. Thummadi et al. (2012) found that the focus of project agents in a waterfall project in the field was more on the early design phase and less on the later development phase. Consistently with this, Kagan et al. (2018) observed that participants in an experiment spent too much time on the first (design) phase and subsequently transitioned to the second (development) phase with a delay. In their two-stage experiment on the transition from ideation to execution in product development, participants could first experiment and explore several designs without compensation. Once participants transitioned to the development phase, they implemented their preferred design for which they were compensated. When given progression autonomy, participants transitioned with delay, resulting in reduced performance. However, it is thus far not well understood what underlying mechanisms drove these observations. We argue that they are symptoms of a human tendency to delay progressing from early to later tasks.

This conjecture builds on behavioral effects of completion, status quo and present bias. The first bias describes a quasi-need to complete a task once it has been started. For tasks without an objective state of completion, it depends on the subjective perception of sufficiency and increases with the time invested (Zeigarnik 1938). According to this early work, such indefinite and thus uncompleted processes create large tensions. Therefore, the completion bias might induce project agents to overweight any started (subscript s) project phase (i.e. $w_s \uparrow$), resulting in a large time allocation to the respective project phase. The second bias is the human tendency to maintain the status quo, as the perceived advantages of exiting are smaller than the perceived disadvantages, even more so as the number of alternatives, which in our case would be project phases, increases (Samuelson and Zeckhauser 1988, Kahneman et al. 1991). This corresponds to an underweighting of all subsequent project phases (i.e. $w_{s+j} \downarrow, \forall j = 1,\ldots,n-s$). The third bias describes the preference for the earlier moment in trade-off decisions between two future moments (O'Donoghue and Rabin 1999). Whereas people tend to underweight the perceived value of tasks that are further away (Pezzo et al. 2006) and are confident of their performance in these tasks, they perceive more accountability for tasks that are closer in time (Gilovich et al. 1993). Thus, people tend to overweight the present and underweight the future ($w_s > w_{s+j}, \forall j = 1,\ldots,n-s$).

These biases might come into play recurrently in a sequential project management setting, where we expect project agents to overweight earlier project phases and underweight later ones. If the progression is flexible but the total duration of the project is not extended, spending excessive time on the initial project phase will require succeeding project phases to be shortened. In our stylized setting involving identical payoff functions with marginally decreasing growth for each project phase and assuming constant effort, a flexible project agent with biased weights $w_1 > w_2 > \ldots > w_n$ would spend a decreasing amount of time on each successive project phase $x_1^F > x_2^F > \ldots > x_n^F$ as the project progresses. The expected completed workload resulting from this then decreases across project phases, as represented by FN in Figure 2.3.

Against this, we consider the role of time-boxed progression in agile sprints. Every time-box is of equal duration, i.e. $x_i^T = x_{i+1}^T$, $\forall i = 1,\ldots,n-1$, with fixed completion deadlines, after which the project automatically progresses to the

Figure 2.3. Stylized Expected Behavior by Progression and Goal-Setting Regime



next sprint. Because sprints are never extended (Schwaber and Sutherland 2017), the progression flexibility and thus the time allocation trade-off between project phases are abrogated during project execution. Assuming constant effort, the expected completed workload resulting from this in our stylized setting is balanced across project phases, shown as *TN* in Figure 2.3. As the time spent per project phase is balanced, ceteris paribus, it follows that $\sum_{i=1}^{n} p(x_i^T, e) > \sum_{i=1}^{n} p(x_i^F, e)$. The total payoff is greater than in the flexible progression case, where agents spend too much time on early project phases at the expense of later ones. We therefore state the following hypothesis:

Hypothesis 1 (Time-Boxed vs. Flexible Progression). *Time-boxed progression results in a higher overall performance than flexible progression through the project phases.*

2.2.2. Role of Phase-Specific Output Goals

For every Scrum sprint, project agents set themselves a goal of what is to be achieved (Schwaber and Sutherland 2017). Major bodies of research on goal-setting theory find strong support that specific, ambitious, yet realistic goals positively impact performance (Wood et al. 1987, Locke and Latham 2002). Although studies on self-imposed goals in the broader field of economics are still rare, existing work suggests that people set themselves ambitious goals as a source of internal motivation (Goerg and Kube 2012, Hsiaw 2013). Thus, if project agents set themselves a goal, this should increase their motivation and thus their exerted effort, because of their desire to achieve that particular goal.

However, whereas goal-setting theory provides guidance on the general effect of a single goal, the sum of effects of goals in project management has not been examined thoroughly. Doerr and Gue (2013, p. 728) argue that "operations management models may have overlooked goal-setting prescriptions because they are difficult to model and are not as precise and simple as they seem at first glance." In particular, little work has been conducted on the trade-off resulting from sequentially and endogenously setting single goals on multiple variables (Weingarten et al. 2019), as is the case for sequential project phases with a self-imposed output goal each. Whereas some research have found that having multiple goals leads to increased performance (Locke and Latham 1990) and attained goals lead to higher future goals (Bandura 1989), other studies have determined that multiple goals lead to decreasing pleasure from the attainment of success, because feelings regarding one goal are not independent of the others (Weingarten et al. 2019). Consequently, the goal-setting approach of agile sprints consisting of multiple, sequential, self-imposed goals is not yet entirely backed by scientific evidence.

We first consider the impact of phase-specific output goals in combination with time-boxed progression, as proposed by agile sprints (Schwaber and Sutherland 2017). Time-boxed progression prevents too much time being spent on one project phase for achieving overly ambitious goals. At the same time, as the goals are set sequentially and always for the same time horizon, namely the duration of one sprint, agents should be able to set challenging, yet realistic, goals by benchmarking against the past. This continuity should make project agents feel more on track, a feeling which improves the exerted effort (Deci and Ryan 1985). The expected completed workload across the project phases resulting from this is continuously greater than when given time-boxed progression alone, as represented as TG in Figure 2.3. We thus formulate the following hypothesis:

Hypothesis 2 (Goal-Setting Given Time-Boxed Progression). *Combining time-boxed progression with phase-specific output goals results in better per-*

formance than with no goal-setting.

Against this, we consider the impact of phase-specific output goals in combination with flexible progression. This setting is common in traditional project management, where, although goals are usually to be reached within a specified time limit (Locke and Latham 2002), the time limit, i.e. progression deadline, is not strictly enforced. Ambitious goals by definition make goal recipients struggle to achieve them. People usually increase their effort if they are performing below target (Matsui et al. 1983), resulting in prolonged work if they are allowed to control the time they spend on the task (LaPorte and Nath 1976). Compared to the time allocation of agents with flexible progression but with no output goals, the time spent per project phase should be greater for early project phases, in order to enable ambitious goals to be achieved. If the total duration is fixed, but progression is not enforced, more time for early project phases must from some phase q onward result in less time for later project phases, i.e. $x_i^{FG} > x_i^{FN} \forall i = 1, \dots, q \wedge x_i^{FG} < x_i^{FN} \forall i = q+1, \dots, n$.

The expected completed workload is then initially greater than when given flexible progression alone, but decreases faster since less time is available, as shown by *FG* in Figure 2.3. If this conjecture holds, goal-setting in the flexible progression setting results in two opposing effects. On the one hand, ambitious yet realistic goals foster (at least initially) a greater exerted effort in line with goal-setting theory, while on the other hand, they amplify bias in the time allocation across project phases. The former effect results in an (initial) upward shift of the line while the latter effect results in a steeper decrease of the line. Note that depending on the magnitude of these two effects the lines of *FG* and *FN* may intersect or not. While it is impossible to predict which effect dominates, the performance would in any case be worse than when goal-setting is combined with time-boxed progression. We thus state the following hypothesis:

Hypothesis 3 (Goal-Setting Given Flexible Progression). *The effect of phase-specific output goals is less performance-enhancing in interaction with flexible progression than with time-boxed progression.*

2.3. Experimental Design

We establish an experimental environment in which we can compare the effects of agile sprints on project performance and execution with those of traditional project management.

2.3.1. Task Description

We deploy an experiment with a fixed total duration of 15 minutes net working time. During this time, participants are asked to work on five sequential real-effort phases of equal length and complexity, which represent sequential project phases. Each experimental phase consists of one screen with 66 sliders arranged in three scattered columns (see Figure A.1 in the appendix), adapted from Gill and Prowse (2012). Participants move the sliders along bars from zero to desired values using the computer mouse. The mouse wheel and keyboard are deactivated for the task. To reduce learning effects in our repetitive design, we vary the desired value of each slider (within each experimental phase, mean = 50, standard deviation = 30, and values are identical for all participants), whereas in the original design by Gill and Prowse (2012) the desired value is constant at 50. To enable fair comparison, participants can only work through the sliders one after the other starting in the top left corner of the screen and cannot overleap any sliders or otherwise change their sequence within a phase. This is achieved by only displaying the next desired value upon successful positioning of the previous slider. However, not all sliders in a given experimental phase need to be completed.

Participants receive a show-up fee of EUR 4 and are additionally paid for every slider completed during the net working time. To model a decreasing value-add of incremental work within each project phase, the incremental payoff per slider decreases within an experimental phase. Whereas the first correctly completed slider yields a return of 166 experimental currency units, called *reward points*, the last one, i.e. the 66^{th} correctly completed slider within an experimental phase yields a return of only 101 reward points. This payoff structure is identical across all five experimental phases. The reward points are exchanged for EUR at a rate of 5,000 reward points = EUR 1. Participants are expressly informed that all five experimental phases are homogeneous, the marginally decreasing rewards structure is identical across all phases, and not all sliders of all phases need to be completed within the 15 minutes of net working time.

Although Scrum was developed for complex projects (Schwaber and Sutherland 2017), we use the repetitive, simple slider task for several reasons: The repetition of the task avoids random noise from varying tasks and ensures that participants do not find one task more satisfying than the other, eliminating progression bias from intrinsic enjoyment. The simplicity of the task allows us to measure the incremental completed workload (i.e. the number of completed sliders), reduces the random luck present in complex tasks, and makes participants more susceptible to the goal-setting effect than more complex tasks (cf. Locke and Latham 2002). The large number of sliders per experimental phase is chosen to replicate the unlimited amount of effort that can be invested in each project phase with an infinite definition of completion. Based on pretest data, completing 66 sliders takes approximately eight minutes on average. Thus, it is not possible to complete all 66 sliders of all five experimental phases within the 15 minutes of net working time. The optimum time spent on each experimental phase of three minutes, assuming constant effort and ability, can be easily calculated by the participants, given 15 minutes of net working time and five experimental phases. Because of the short completion time for a single slider, participants in the flexible setting can progress to the following phase at any time without set-up costs, whereas longer real-effort tasks might expose participants to a sunk cost bias from leaving a started, but unfinished task.

To create a realistic model of the trade-off decision in project management when allocating time and effort across project phases, a strictly sequential progression through the experimental phases prevents participants from jumping between phases. Although adjustments to previously completed upstream project phases depending on new insights and the requirements of downstream project phases exist in both traditional and agile project management (Kerzner 2013, Schwaber and Sutherland 2017), projects are in the first instance executed sequentially. We adhere to this sequential progression to avoid noise from phase jumping.

2.3.2. Treatment Design

To test the hypotheses, we deploy four treatments in a 2×2 design, varying time-boxed versus flexible progression on the one axis and no goal-setting versus goal-setting on the other (Figure 2.1). All other components of the experiment remain constant across all four treatments. Participants in *FN* proceed through the experimental phases flexibly at their own discretion, i.e. they can work on each phase for as long as they desire, but they have no longer than 15 minutes in total for all five phases. They can only progress to the next experimental phase by actively pressing a button. Participants in *TN* work on each experimental phase for three minutes followed by time-boxed, i.e. automatic progression to the next phase. Participants in both *FN* and *TN* do not set any goals. Treatments *FG* and *TG* are identical in all aspects to *FN* and *TN*, respectively, but with participants additionally setting themselves a goal of how many sliders they wish to complete sequentially after each experimental phase for the next phase. The first experimental phase is completed without a goal, in order to give participants the sense of a realistically achievable output before they set themselves a goal for the second experimental phase. Meeting the goals is not financially incentivized, to avoid a misalignment of incentives between setting easily achievable goals and collecting as many reward points as possible. Given this design, Hypothesis 1 can be tested by comparing *FN* with *TN*, Hypothesis 2 by comparing *TN* with *TG*, and Hypothesis 3 by comparing all four treatments.

We implement the variations between the treatments as follows: Participants pause for 60 seconds after each experimental phase; this does not count towards the total net working time of 15 minutes. During these breaks, participants in goal-setting treatments *FG* and *TG* are informed of how many sliders they completed correctly in the previous phase and set a goal of how many sliders they wish to complete in the next phase. In all treatments, the number of the current experimental phase and the total amount of reward points collected are displayed in the top bar of each participant's computer screen (see Figure A.1 in the appendix). The working time remaining is also displayed. This differs by progression regime. In the treatments with flexible progression, i.e. *FN* and *FG*, the total remaining working time is shown counting down from 15:00 minutes. In the treatments with time-boxed progression, i.e. *TN* and *TG*, the working time remaining for each respective experimental phase is shown counting down from 03:00 minutes. Finally, in the goal-setting treatments *TG* and *FG*, the goal and the correctly completed sliders are shown for each respective phase.

2.3.3. Experimental Procedures and Participants

The hypotheses were tested in a controlled experiment at the experimenTUM laboratory of the Technical University of Munich, Germany using the software z-Tree (Fischbacher 2007). The experiment was conducted in German, the first language of the majority of participants in the lab. Before the experiment was conducted, participants were randomly assigned to a treatment. Each one privately read a paper copy of the treatment-specific experimental instructions (see Section A.2 in the appendix). They were allowed to ask questions for clarification. Afterwards, the participants completed a practice phase on a screen that exactly resembled the five experimental phases of the actual experiment. In this practice phase, participants had to solve three sliders without compensation, in order to familiarize themselves with the task. They then had to do a quiz to test their comprehension of the task, rules, and remuneration (see Section A.3 in the appendix). Participants were informed upfront that if they did not pass the quiz in two trials, they would only receive the show-up fee of EUR 4. This measure was taken to ensure that the results would not be influenced by insufficient understanding of the rules of the experiment. The participants then completed the experiment individually. At the end, they were anonymously asked for their goal-setting behavior and demographic information. They also completed a German test to cross-check their comprehension of the written experiment's rules. Finally, all the participants were privately paid.

A total of 366 participants, mostly business and engineering students, were recruited out of the lab's participant pool. We excluded 24 participants who did not pass the comprehension test. Most of these were non-native speakers and also failed the German test, suggesting a lack of understanding because of language barriers. Of the remaining 342 participants, 84, 90, 79, 89 were in treatments *TN*, *FN*, *TG*, and *FG* respectively. The participants spent approximately 35 minutes in the lab. The 342 participants earned on average EUR 7.63 (median EUR 7.60, minimum EUR 4.90, maximum EUR 9.80), which was above the time-adjusted target compensation of participants at the lab.

Table 2.1. Totals of Collected Reward Points and Completed Sliders by Treatment (Experimental Phases 1–5)

	Reward points		Sliders	
Treatment	Mean	Std. dev.	Mean	Std. dev.
Time-boxed with no goals (TN)	19,961	3,591	130.7	25.4
Flexible with no goals (FN)	18,938	3,316	126.3	21.5
Time-boxed with goals (TG)	20,315	3,512	133.2	25.2
Flexible with goals (FG)	17,822	3,011	118.7	20.5

2.4. Results

The results section begins with Subsection 2.4.1, which compares the overall performance by treatment. This is followed by Subsection 2.4.2, which presents the execution of the workload per experimental phase.

2.4.1. Overall Performance

Overall performance is best measured by the total payoff, i.e. the total reward points collected across the experimental phases. Because the marginal value-add from the invested time and effort decreases within each project phase, the $n^{th}+1$ completed slider of any experimental phase is always less valuable to the overall performance than the n^{th} completed slider of the same phase. Two participants may therefore complete the same workload (number of sliders) overall, but not accrue the same value-add to the project's overall performance (in the form of reward points collected), due to differences in the workload completed per experimental phase.

Table 2.1 presents descriptive statistics of the total reward points collected and total sliders completed by treatment. The mean total reward points collected is highest when time-boxed progression is combined with goal-setting, i.e. in treatment *TG*, which resembles agile project management, and lowest when flexible progression is combined with goal-setting, i.e. in treatment *FG*, which resembles traditional project management. To evaluate how these observations of total payoff relate to our hypotheses, we fit a linear regression model on the overall

Table 2.2. Linear Regression Models on Total Collected Reward Points, Total Completed Sliders (Experimental Phases 1–5), and Seconds per Practice Slider

	Reward points		Sliders		Seconds per practice slider	
	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Intercept	19,961***	22,183***	130.7***	146.0***	14.6***	13.5***
(baseline TN)	(54.50)	(46.70)	(51.73)	(44.48)	(15.11)	(13.18)
Flexible progression	–1,023**	–947**	−4.3	−3.8	0.4	0.3
(simple effect)	(−2.01)	(−2.03)	(−1.24)	(−1.19)	(0.27)	(0.24)
Goal-setting	354	668	2.5	4.5	1.0	0.9
(simple effect)	(0.67)	(1.30)	(0.69)	(1.27)	(0.75)	(0.69)
Flexible progression × goal-	–1,470**	–1,365**	−10.2**	−9.4**	−0.3	−0.5
setting (interaction effect)	(−2.02)	(−2.05)	(−2.03)	(−2.05)	(−0.14)	(−0.28)
Female control		–1,394***		−9.7***		2.7***
		(−4.09)		(−4.10)		(2.77)
Lack of skill control		–124***		−0.8***		
		(−6.48)		(−6.37)		
Private goal control		361		2.1		
		(0.74)		(0.63)		
Number of observations	342	342	342	342	342	342
R^2	0.08	0.23	0.05	0.21	0.00	0.03

Notes. t statistics in parentheses. The number of observations equals the number of participants.
$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

data, with total payoff as the dependent variable and flexible progression and phase-specific goal-setting in isolation and interaction as explanatory variables. We first focus on the effect of time-boxed progression in isolation. Hypothesis 1 postulates that time-boxed progression results in a higher total payoff in treatment *TN* than in *FN*. This simple effect of flexible progression is statistically significant ($p = 0.045$, Table 2.2, Model 1).

Result 1 (Time-Boxed vs. Flexible Progression). *In the absence of goal-setting, time-boxed progression results in a higher overall performance than flexible progression.*

We now focus on the effect of phase-specific goal-setting, given a time-boxed progression. In Hypothesis 2, we state that phase-specific output goals will have a positive effect on the total payoff in settings with time-boxed progression. However, the positive simple effect of phase-specific goal-setting given time-boxed progression is not significant at any conventional level ($p = 0.502$, Table 2.2, Model 1).

Result 2 (Goal-Setting Given Time-Boxed Progression). *The positive effect of phase-specific output goals on the overall performance in settings with time-boxed progression is not significant.*

Against this, we state in Hypothesis 3 that the effect of phase-specific output goals is less performance-enhancing in interaction with flexible progression than with time-boxed progression. The regression analysis indicates a statistically significant, negative interaction effect ($p = 0.044$, Table 2.2, Model 1).

Result 3 (Goal-Setting Given Flexible Progression). *The effect of phase-specific output goals on the overall performance in settings with flexible progression is negative because of the negative interaction effect.*

Our results are robust to three potentially confounding factors: gender, ability, and participants setting themselves private goals. First of all, Gill and Prowse (2019) report a lower performance in the slider task among female participants. We measured the gender effect by introducing a binary *female* dummy, which takes the value 1 for female participants. Overall, female participants indeed performed significantly worse ($p < 0.001$, Table 2.2, Model 2). Secondly, participants' individual ability to correctly position a slider impacts the results. We controlled for the initial ability through the time it took a subject to complete the sliders during the practice phase before the actual experiment, denoted as control variable *lack of skill* in the following. The longer it took, the worse was their performance in the main experiment ($p < 0.001$, Model 2). Note, that there were no statistically significant differences between the treatments in the initial lack of skill, measured by the time needed per practice slider, also when controlling for gender (Models 5 and 6, respectively). Thirdly, in the non-goal-setting treatments *FN* and *TN*, we are interested in the participants' behavior in the absence of goals. After the experiment, we asked the participants in these treatments whether they had set themselves any private goals, to which 36% of the participants gave a positive response (63 out of 174). We introduced a binary *private goal* dummy, which takes the value 1 for privately set goals. Privately set goals among participants in non-goal-setting treatments *FN* and *TN* result in no significant performance effect ($p = 0.458$, Model 2). Thus, controlling for gender, lack of skill, and private goals does not change our results.

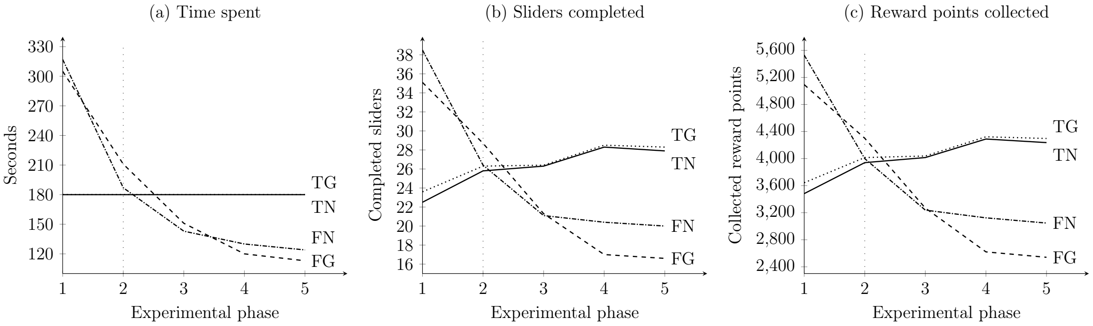
The differences between the treatments in terms of overall performance (reward points collected) result from two components, differences in the total completed workload (sliders completed overall) and in the allocation of the completed workload across the experimental phases (completed sliders per phase). First of all, the number of completed sliders overall is significantly lower when flexible progression and goal-setting interact in treatment *FG* ($p = 0.043$, Table 2.2, Model 3), whereas the negative simple effect of flexible progression and the positive simple effect of goal-setting on the total number of completed sliders are not significant ($p = 0.217$ and $p = 0.492$, respectively, Model 3). We control for the initial lack of skill, gender, and private goals of the participants, and with these controls, the sum of the completed sliders overall is a proxy for the exerted effort (as the total working time is constant for all treatments). Though flexible progression and goal-setting alone do not significantly change the exerted effort ($p = 0.235$ and $p = 0.205$, respectively, Model 4), they significantly diminish the exerted effort if they interact ($p = 0.041$, Model 4). In Subsection 2.4.2, we investigate the second component driving the overall performance, which is the completed workload across the experimental phases.

2.4.2. Execution of the Workload per Experimental Phase

The completed workload across the experimental phases is best evaluated by the number of completed sliders, which is easier to compare than the exogenously predefined, variable payoff per slider measured by the collected reward points. It depends on the time spent on each experimental phase. Figure 2.4 shows the mean of the time spent, sliders completed, and reward points collected per experimental phase by treatment (see underlying data in Table A.1 in the appendix).

In all treatments, participants complete the first experimental phase without any phase-specific output goal, in order to learn what can be deemed a realistic performance level. Beginning with the second phase, subjects in the goal-setting treatments *FG* and *TG* set themselves output goals for each phase. The workload completed across the experimental phases after goals were set resembles our predicted outcome (Figure 2.3). On average, participants in the flexible treatments

Figure 2.4. Means of Time Spent, Sliders Completed, and Reward Points Collected per Experimental Phase by Treatment



Note. Goals are set in treatments TG and FG from the second experimental phase onward.

FN and *FG* complete fewer sliders with each progression, whereas participants in the time-boxed treatments *TN* and *TG* complete slightly more sliders on average over time. This increase in the time-boxed setting is not driven by different learning effects across treatments: For all four treatments, the time needed per slider decreases significantly over time, in line with the learning effects observed by Gill and Prowse (2019), but with no substantial differences between treatment improvement rates across the experimental phases (see Table A.2 in the appendix, Model 1). Thus, the different behaviors in the execution of the experimental phases must be driven by the different progression and goal-setting regimes, which we will investigate in the following.

2.4.2.1. Role of Time-Boxed Progression.

The payoff-maximizing strategy in our experimental design is to complete the same workload, i.e. the same number of sliders, in each experimental phase. However, we observe that subjects in treatment *FN* with flexible progression spend too much time on early phases at the expense of later phases, completing, on average, fewer sliders with each experimental phase (Figure 2.4, (a) and (b)). We fit two regression models on the data of *FN*, where the dependent variable is either the time spent or the amount of sliders completed per experimental phase (Table 2.3, Models 1 and 3–4, respectively) and the independent variable is the experimental phase (denoted as *experimental phase slope effect*). The coefficient estimate shows that subjects spend significantly less time per experimental phase

Table 2.3. Multilevel Mixed-Effects Linear Regression Models on Time Spent and Sliders Completed per Experimental Phase 1–5 for FN and TN

	Time (in seconds)		Sliders			
	FN	TN	FN		TN	
	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Intercept	268.47***	180.04***	33.86***	35.37***	23.49***	27.66***
	(25.04)	(> 99.99)	(26.83)	(22.46)	(41.64)	(22.70)
Experimental phase slope effect (simple effect)	−44.20*** (−8.25)	0.00 (1.53)	−4.30*** (−6.19)	−4.30*** (−6.19)	1.32*** (13.09)	1.32*** (13.09)
Controls	No	No	No	Yes	No	Yes
Number of observations	450	420	450	450	420	420

Notes. Experimental phase 1 set as intercept. Random intercept effects at the subject level included; Models 1, 3 and 4 additionally with random slope effects for better model fit. Models 1 and 3 additionally with covariance unstructured for better model fit. For all models, standard errors clustered at the subject level. z statistics in parentheses. Controls are gender, initial lack of skill, and private goals. Controls not applied to regressions on time, as the total working time is fixed and thus not affected by simple effects of these controls. The number of observations equals the product of the participants and the experimental phases.
$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

with each progression ($p < 0.001$, Model 1). As a consequence, they also complete significantly fewer sliders with each progression ($p < 0.001$, Model 3). The learning effects over time do not compensate for the delayed progression. We refer to this observation as Progression Fallacy.

Result 4 (**Progression Fallacy**). *Flexible progression leads to over-allocation of time to early phases at the expense of later phases.*

As for the overall results in Subsection 2.4.1, the Progression Fallacy is robust to controlling for three potentially confounding factors: gender, lack of skill, and participants setting themselves private goals (Model 4). It also holds for a categorial analysis of the experimental phases (see Table A.3 in the appendix). Most of the participants with particular exposure to the Progression Fallacy in one phase are no more conscientious in the following phase. For example, of the top third of the participants in *FN* who spend the most time on the first experimental phase (i.e. the top 30 participants), 25 participants spend more than a quarter of the total time left for the remaining four phases on the second phase alone. In this second phase, the 25 participants exceeded the mean time left per experimental phase by an average of 155%. The five participants, who spend less than a quarter of the total time left for the remaining four phases on the second phase, followed no clear trend in the later phases, partially levelling their time allocation and partially succumbing to the Progression Fallacy again in later phases.

In contrast, in treatment *TN*, subjects with time-boxed progression by definition spend the same amount of time on each experimental phase (Table 2.3, Model 2). When we fit a regression model on the data of *TN*, we find a significant increase in the number of sliders completed over time ($p < 0.001$, Model 5). Again, our observations are robust to controlling for gender, lack of skill, and private goals (Model 6) and a categorial analysis of the experimental phases (see Table A.3 in the appendix).

2.4.2.2. Role of Phase-Specific Output Goals.

Subjects in the flexible treatments *FN* and *FG* start from a common baseline: In the first experimental phase, for which no goals are set in order to gain experience, the time spent and the number of sliders completed are not significantly different between *FN* and *FG* (rank sum test, $p = 0.702$ and $p = 0.154$, respectively). After phase-specific goals are self-imposed in *FG* from the second phase onward, participants spent on average more time on—and complete more sliders in—the first phases after goal-setting and less time on—and fewer sliders in—the later phases than participants in the flexible treatment *FN* without goals (Figure 2.4, (a) and (b)).

We fit a regression model on the data of *FN* and *FG*, in which the dependent variable is the time spent per experimental phase and the independent variables are the experimental phase and a dummy for the goal-setting as simple effects (denoted as *experimental phase slope* and *goal-setting intercept effects* in Table 2.4), as well as their interaction (denoted as *goal-setting slope effect*). The goal-setting intercept effect indicates that subjects in *FG* spend significantly more time on the first experimental phase after initial goal-setting than subjects in *FN* ($p = 0.001$, Table 2.4, Model 1). However, with each progression, the time spent on an experimental phase in *FG* falls significantly faster than in *FN*, as indicated by the goal-setting slope effect ($p = 0.049$, Model 1), which finally results in less time spent on the last experimental phases in *FG* than in *FN*. As for the overall results in Subsection 2.4.1, these effects are robust to controlling for gender, lack of skill, and private goals (Model 2). Additionally, the results are robust to controlling

Table 2.4. Multilevel Mixed-Effects Linear Regression Models on Time Spent and Sliders Completed per Experimental Phase 2–5 for FG with FN as Baseline

	Time (in seconds)			Sliders		
	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Intercept	176.14^{***}	186.01^{***}	255.24^{***}	24.95^{***}	28.42^{***}	36.60^{***}
(baseline FN)	(35.96)	(31.13)	(41.72)	(28.35)	(24.80)	(32.97)
Experimental phase slope	-20.14^{***}	-20.14^{***}	-20.14^{***}	-1.99^{***}	-1.99^{***}	-1.99^{***}
effect (simple effect)	(−4.94)	(−4.94)	(−4.94)	(−3.27)	(−3.27)	(−3.27)
Goal-setting intercept	21.44^{***}	26.74^{***}	18.57^{**}	2.05	3.56^{**}	2.64^{*}
effect (simple effect)	(2.80)	(3.35)	(1.97)	(1.47)	(2.49)	(1.78)
Goal-setting slope effect	-12.38^{**}	-12.38^{**}	-12.38^{**}	-2.08^{**}	-2.08^{**}	-2.08^{**}
(interaction effect)	(−1.97)	(−1.97)	(−1.97)	(−2.16)	(−2.16)	(−2.16)
Time spent in first			-0.25^{***}			-0.07^{***}
experimental phase			(< −99.99)			(−15.16)
Sliders completed in first			−0.00			0.29^{***}
experimental phase			(−1.24)			(7.90)
Controls	No	Yes	No	No	Yes	Yes
Number of observations	716	716	716	716	716	716

Notes. Experimental phase 2 with first goal-setting set as intercept. Random intercept and slope effects at the subject level included with covariance unstructured. Standard errors clustered at the subject level. z statistics in parentheses. Controls are gender, initial lack of skill, and private goals. Controls not applied to Model 3, as the total working time is fixed with the additional covariate for the time spent in the first experimental phase in this model and thus not affected by simple effects of the controls. The number of observations equals the product of the participants and the experimental phases.

$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Table 2.5. Goal of Completed Sliders per Experimental Phase 2–5 by Treatment

	FG		TG	
Experimental phase	Mean	Std. dev.	Mean	Std. dev.
2	32.0	14.0	27.7	8.2
3	22.8	11.9	27.9	6.7
4	17.0	10.3	26.5	7.3
5	16.2	12.7	27.5	9.0

Note. Goals are set from the second experimental phase onward.

for the time spent on and the sliders completed in the first experimental phase, in which no goals were set (Model 3).

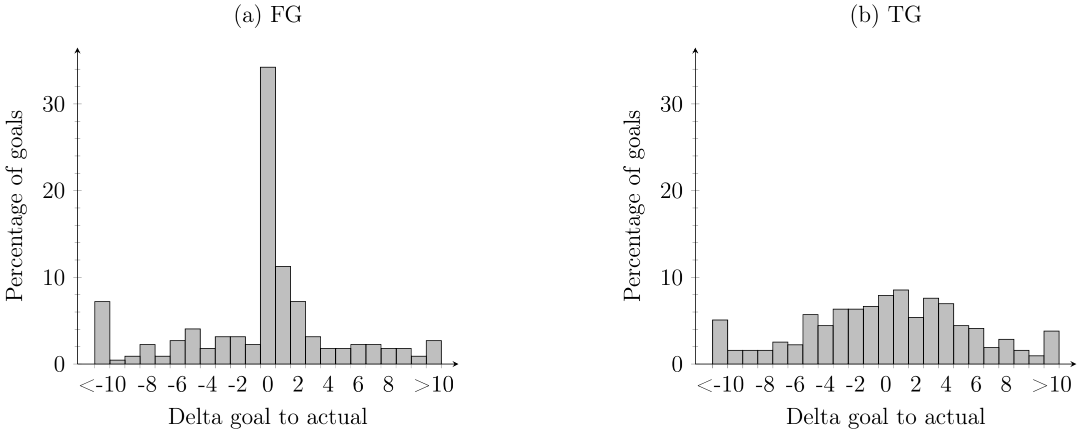
Result 5 (**Progression Fallacy with Goal-Setting**). *Phase-specific goal-setting given flexible progression amplifies the Progression Fallacy.*

This progression delay translates to a less balanced workload per experimental phase, as shown by the equivalent regression models fit on the sliders completed rather than the time spent per experimental phase as the dependent variable, although the goal-setting intercept effect is statistically less pronounced (Table 2.4, Models 4–6, respectively). These results do not fundamentally change with a categorial analysis of the experimental phases (see Table A.4 in the appendix).

Both biased goal-setting and biased goal achievement can affect progression decisions. First of all, when setting goals, participants appear to benchmark their future goals against their performance in previous experimental phases, which in the flexible setting is biased by the overallocation of time to early phases due to the Progression Fallacy. Accordingly, the initial goal in treatment *FG* is on average 15% higher than in treatment *TG* (Table 2.5), whereas less time is available to finish all experimental phases. As a consequence, the average goal in treatment *FG* is adjusted downwards with each experimental phase, whereas it remains relatively stable over time in treatment *TG*.

Secondly, when working towards their goals, participants in the flexible setting appear to avoid missing their goals. We present the distribution of the goal achievement, where participants in *FG* are free to choose their progression point (i.e. excluding the last experimental phase), in Figure 2.5(a). Participants com-

Figure 2.5. Distribution of Goal Achievement by Treatment



Notes. A negative delta indicates a missed goal, a positive delta an exceeded goal in the numbers of sliders. The data for FG excludes the final experimental phase, in which participants no longer make a progression decision.

plete 34% of their goals exactly and exceed them by one slider in 11% of cases. In only 2% of cases do they stop only one slider before their goal. In comparison, Figure 2.5(b) shows the distribution of the goal achievement in treatment *TG*, in which participants cannot choose the progression point. There is no clear spike at zero, but a widespread distribution. We conclude that chasing initially overambitious goals results in more time spent in early phases, which comes at the expense of later phases in *FG*. In contrast, time-boxing requires subjects to progress regardless of their goal achievement.

Result 6 (Time-boxed Progression with Goal-Setting). *Time-boxing prevents delayed progression which is amplified by goal-setting.*

2.5. Robustness Checks of the Progression Fallacy

In this section we present two control treatments, with which we investigate whether the Progression Fallacy observed in the flexible setting is robust to planning and progression prompts, these being two elements of good project management. In the first control treatment, participants plan upfront how much time they wish to spend on each phase. We investigate whether participants are ex-ante able to determine an approximately optimal time allocation and whether rendering the planning more salient mitigates the Progression Fallacy. It is possible that participants in the flexible treatments *FN* and *FG* progressed with delay, as they anticipated learning effects and thus consciously allocated less time to the later phases. They simply might have overestimated their learning rate in line with the planning fallacy, the optimism bias in predictions of how long the completion of a task will take the predictor (Kahneman and Tversky 1979).

In the second control treatment, we additionally introduce a progression prompt, which reminds the participants of their initial progression plan and the track of time during the experimental phases. In the flexible setting, participants must solve the optimization problem of progressing at an optimal pace while simultaneously working on a tedious task. We investigate whether reducing the cognitive load with a progression prompt mitigates the Progression Fallacy. It may be that participants in the flexible treatments *FN* and *FG* progressed with delay, as they became caught up in doing the task and lost track of time.

2.5.1. Design and Implementation of the Control Treatments

The first control treatment is the same as the flexible treatment *FN* on all dimensions, except that prior to the first experimental phase, participants additionally have to enter on the screen how they will allocate the total net working time of 15 minutes across the five experimental phases (see Figure A.2 in the appendix). They are told that it is only for preparation purposes and that adherence to it will not be enforced. We refer to this treatment as *flexible with no goals but with planning (FNP).*

The second control treatment is identical to *FNP* on all dimensions, but with the addition of a progression prompt that informs participants throughout the experimental phases of the phase in which they planned to be working in that moment, including a strong visual warning, should they progress outside their plan (see Figure A.3 in the appendix). We refer to this treatment as *flexible with no goals but with planning and progression prompt (FNPP).*

For an isolated analysis of the effects of planning and progression prompts, these

Table 2.6. Total of Collected Reward Points and Completed Sliders by Control Treatment (Experimental Phases 1–5)

	Reward points		Sliders	
Treatment	Mean	Std. dev.	Mean	Std. dev.
Flexible with no goals (FN, baseline from the main experiment)	18,938	3,316	126.3	21.5
Flexible with no goals but planning (FNP)	18,569	2,502	121.7	17.6
Flexible with no goals but planning and progression prompt (FNPP)	18,833	2,908	124.1	19.9

control treatments are done without goal-setting. We recruited 84 participants out of the same lab's participant pool, but separately after the main experiment. We excluded six participants, who did not pass the comprehension test. Of the remaining 78 participants, there were 38 and 40 participants in treatments *FNP* and *FNPP*, respectively. Due to COVID-19, hygiene measures were in place and the number of participants was reduced.

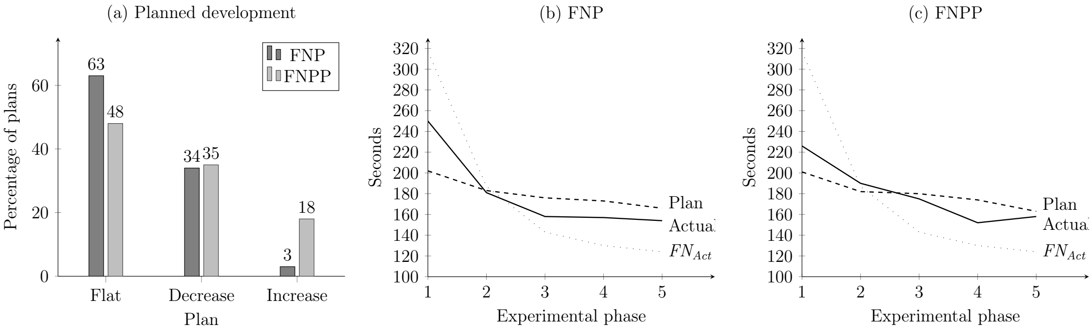
2.5.2. Results of the Control Treatments

We present descriptive statistics by treatment of the total reward points collected and the total sliders completed in Table 2.6. There is no significant difference either between treatments *FN* and *FNP* or between *FN* and *FNPP* neither in the total payoff (rank sum test, $p = 0.518$ and $p = 0.679$, respectively) nor in the amount of completed sliders (rank sum test, $p = 0.279$ and $p = 0.468$, respectively). Planning and progression prompts do not improve the average performance.

For each participant of the control treatments separately, we approximate the slope of the time planned per experimental phase with a linear regression. Based on this, most of the participants planned a flat allocation of the total net working time with three minutes per experimental phase (63% and 48% in *FNP* and *FNPP*, respectively, Figure 2.6(a)), whereas about a third planned a decreasing working time per experimental phase (34% and 35% in *FNP* and *FNPP*, respectively). A minority planned an increasing working time per experimental phase (3% and 18% in *FNP* and *FNPP*, respectively).

In both control treatments, participants on average missed their initial progression plan, spending more time on earlier experimental phases than on later ones, albeit less than in treatment *FN* (Figure 2.6, (b) and (c)). We fit regres-

Figure 2.6. Planned Development of Time Spent (a) and Mean of Time Planned and Actually Spent (b) and (c)



Note. (a) FNPP not adding up to 100% due to rounding.

sion models on these data from *FNP* and *FNPP*, with the time per experimental phase as the dependent variable and the experimental phase as the first independent variable. In order to study the difference between planned and actual time, we set the planned time as the baseline and introduce the two explanatory variables *delta (plan to actual) intercept effect* and *delta (plan to actual) slope effect* (Table 2.7). In both control treatments, participants spend significantly more time on average on the first experimental phase than planned, as indicated by the delta (plan to actual) intercept effect ($p = 0.028$ and $p = 0.044$, Table 2.7, Models "All", respectively). They remain significantly behind their plan in the following phases, as indicated by the delta (plan to actual) slope effect ($p = 0.028$ and $p = 0.045$, Models "All", respectively). This also holds directionally for analysis by type of plan (Models "Flat", "Decrease", "Increase"), although the actual intercept and slope effects are not significant for flat and decreasing plans in FNPP (which might be due to the limited sample sizes in these subgroups). Also, the few participants in FNP who plan a decreasing time per experimental phase adhere to this plan and do not progress later than planned. These results do not fundamentally change after a categorial analysis of the experimental phases (see Table A.5 in the appendix).

We conclude that rendering the plan more salient reduces the Progression Fallacy but does not eliminate it. Most of the participants do not plan less time for the later experimental phases, but they still progress with delay. Additionally introducing a progression prompt to remind participants of their initial progres-

Table 2.7. Multilevel Mixed-Effects Linear Regression Models on Time Planned and Actually Spent per Experimental Phase 1–5 for FNP and FNPP with Planned Time as Baseline (in Seconds)

	FNP				FNPP			
	All	Flat	Decrease	Increase	All	Flat	Decrease	Increase
Intercept	196.42***	180.00***	231.69***	132.00	197.10***	180.00***	244.57***	148.57***
(baseline planned time)	(40.89)	(> 99.99)	(38.48)	(NA)	(25.67)	(> 99.99)	(18.51)	(11.85)
Experimental phase slope effect	−8.21***	0.00	−25.85***	24.00	−8.55**	0.00	−32.29***	15.71**
("plan", simple effect)	(−3.42)	(0.00)	(−8.59)	(NA)	(−2.23)	(0.00)	(−4.89)	(2.51)
Delta (plan to actual) intercept	26.52**	31.02***	−5.96	340.87	17.74**	12.90	29.69	6.97***
effect (simple effect)	(2.20)	(2.76)	(−0.51)	(NA)	(2.01)	(1.35)	(1.36)	(5.76)
Delta (plan to actual) slope	−13.23**	−15.48***	3.01	−170.33	−8.84**	−6.42	−14.81	−3.45***
effect (interaction effect)	(−2.20)	(−2.75)	(0.51)	(NA)	(−2.01)	(−1.34)	(−1.36)	(−5.70)
Number of observations	380	240	130	10	400	190	140	70

Notes. Experimental phase 1 set as intercept. Random intercept effects included at both the subject and the *actual* dummy levels with random slope effects and covariance unstructured. Standard errors clustered at the subject level. z statistics in parentheses. Controls for gender, initial lack of skill, and private goals not applied, as the total working time is fixed and thus not affected by simple effects of these controls. The number of observations equals the product of the participants, the experimental phases, and plan and actual per participant.
$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

sion plan and make them more aware of the track of time, further reduces the Progression Fallacy but, again, does not eliminate it. Participants on average still progress behind schedule. We conclude that the Progression Fallacy is robust to both explicit ex-ante planning and progression prompts, despite some mitigation. Thus, it is neither just the result of a conscious planning decision nor of cognitive overload, but a biased decision to spend too much time on early project phases, which then shortens the later phases.

2.6. General Discussion

We discuss our main findings and their managerial implications in Subsection 2.6.1, followed by the limitations of the study and suggestions for future research in Subsection 2.6.2.

2.6.1. Main Findings and Managerial Implications

Effect of delayed progression. Our experiment suggests that traditional project management can be exposed to delayed progression from early through to later project phases, resulting in poor performance. If participants are free to progress without constraint through the experimental phases, i.e. in the absence of any time-boxed progression (the first core element of agile sprints), they often fail to proceed from early to later phases on time. This observation is robust to high transparency on the optimal, i.e. return-maximizing, strategy, which is to complete the same workload in each experimental phase. Rather, participants complete significantly more work in early experimental phases than in later ones. We refer to this observation as Progression Fallacy, which is robust to ex-ante planning and progression prompts, despite some mitigation. It is different to well-researched behavioral effects that cause project delays, such as planning fallacy, procrastination, and Parkinson's law. Planning fallacy denotes the optimism bias in predictions of how long the completion of a task will take the predictor, and it results in underestimation of the time needed (Kahneman and Tversky 1979). However, the optimism bias occurs for longer durations; the time needed for short, segmented tasks (as in our experiment) tends to be overestimated (Forsyth and Burt 2008). Our control treatments with formal planning confirm that most participants did not plan less time for later phases. These findings indicate that initial project plans are probably not as unrealistic as the planning fallacy suggests, but are partially also not met, because project agents do not progress as planned. Procrastination is a postponement of effort because of a perception that the cost of immediate effort is higher than that of future effort (Wu et al. 2014). Our experimental design leaves little room for procrastination, as it would directly reduce the participants' financial return. Indeed, we do not observe that participants take any significant breaks during the net working time. Finally, Parkinson's law states that work expands to fill the time available (Parkinson 1957). Given our design, which does not allow all sliders of all experimental phases to be solved within the net working time, there is no leeway for expanding work towards the project closure. We conclude that Progression Fallacy is a self-standing behavioral effect that can cause suboptimal project performance, as early project phases are overdelivered at the expense of underdelivered later phases. Rework to mitigate underperformance in later project phases might cause project delays. These findings apply predominantly to traditional project management, in which progression from one phase to the next is not strictly enforced.

Role of time-boxed progression. Our experiment provides evidence that time-boxed progression improves performance compared to flexible progression. The exogenous control of progression by time-boxing eliminates the progression autonomy of the project agents, thus mitigating the Progression Fallacy. The time spent per project phase is by definition smoothened. Consequently, early project phases will receive comparatively little time and sophistication and later project phases more than in flexible progression settings. The performance of each project phase is then more balanced, with less over- and underdelivery than in traditional project management. As long as a complete project increment is created in each sprint, as required by Scrum (Schwaber and Sutherland 2017), an at least minimum viable overall project performance will be achieved on conclusion of all the sprints. During project execution, project agents do not have to ruminate about the time allocation trade-off, but can focus on advancing in the respective project phase, with a strict progression deadline creating time pressure. Thus, time-boxed progression of agile project management is a way of mitigating the Progression Fallacy and improving performance.

Role of phase-specific output goals. Drawing on extensive goal-setting literature for the second core element of sprints, we conjecture that goal-setting will lead to a general increase in performance and exerted effort. The small positive effect of non-binding, self-imposed, phase-specific output goals on performance and exerted effort given time-boxed progression is not significant in our experimental data. In an experimental setting resembling our time-boxed treatments, and using the slider task, Fan and Gómez-Miñambres (2020) show significantly higher performance from non-binding, phase-specific output goals imposed on a team of workers by a manager. This suggests that for time-boxed progression, self-imposed goals might be less effort-enhancing than goals set exogenously in teams.

Against this, we find that goal-setting in combination with flexible progression can deteriorate performance and exerted effort. Participants appear to benchmark their goals against their past performance, which is biased by the Progression Fallacy. This results in overly ambitious goals and amplifies the Progression Fallacy, as participants spend even more time chasing ambitious goals in early experimental phases. We conclude that every goal needs to be associated with a clear and enforced achievement deadline to avoid progression delays. With time-boxing, projects automatically progress, even if a goal is not met. Consequently, goal-setting cannot tempt project agents to spend too much time on a single goal. The most direct implication of our goal-setting findings is that in traditional project management, where goals are set for flexible project phases, particular managerial attention is required on a timely progression through early project phases to avoid the Progression Fallacy and effort distortion. Agile sprints are an effective way of achieving this. However, they require particular managerial attention on self-imposing sufficiently motivational goals, otherwise goal-setting will have little impact.

2.6.2. Limitations and Future Research

This research has limitations, both on conceptual and contextual levels, that invite future research. *Conceptually*, we rely on a highly stylized experimental design. Our findings build on a task that can be broken down to a comparable workload per experimental phase without any interdependence between the experimental phases. While the internal validity of our behavioral findings is stronger given the high transparency of the profit maximizing strategy in a simple, sequential task compared to the noise of a more complex task, it will be interesting to test the boundary conditions of our results in more complex settings involving heterogeneous and interdependent tasks. Project work often involves cognitive and creative tasks, which are particularly inviting for future exploration. Despite our effort in reducing the learning effects in the slider task by varying the desired value, the participants' learning energy overcomes their fatigue. This is the result of a trade-off in the experimental design between sequential, fully comparable, equally enjoyable, precisely measurable real-effort tasks and learning effects. It would be misleading to conclude that in the case of time-boxed progression, even more balanced workload completion could be achieved across project phases by extending earlier and shortening later project phases, as the learning effects in real projects with heterogeneous tasks in every project phase would presumably be lower. Therefore, as experimental artifacts, the learning effects do not limit the validity of our results. Nevertheless, it will be interesting to test the effect of more complex, heterogeneous, and creative real-effort tasks on this dimension too. Additionally, in many projects, the payoff function is different in each project phase, it can be negative (such as in the case of 0% completion of a project phase), and is interdependent with the performance of the other project phases. Thus, various payoff schemes also invite future research. Furthermore, follow-up studies should particularly focus on individual behavior of project workers with regard to the Progression Fallacy. The external validity of our study should be tested in the field with multi-month/year projects with different sizes and team roles. Although there are numerous empirical observations that can be explained by the Progression Fallacy, ranging from anecdotal evidence, such as students in exams spending too much time on early tasks at the expense of later ones, to academic studies, for instance by Thummadi et al. (2012), who observed that the focus of project agents was more on the early design phase and less on the later development phase, no dedicated field study has as yet been made to investigate this effect.

Contextually, we focus on agile sprints as the "heart of Scrum", leaving other elements aside. Whereas this approach allows for a crystallized investigation of sprints, their interplay with other Scrum elements, e.g. small autonomous teams and continuous improvement, invites future research. Follow-up studies should investigate whether the autonomy of the project team in agile project management is indeed the best approach, e.g. with regard to team motivation, on-time progression, and scalability to larger organizations. In Scrum, project teams autonomously set and commit to self-imposed goals (Schwaber and Sutherland 2017). Our results suggest that such endogenous goal-setting is not effective in raising the level of effort exerted. Evidently, the effect of goal-setting on the exerted effort is more nuanced than we initially expected. Besides the mere setting of goals, additional dimensions (e.g., whether they are self- or exogenously imposed) have to be considered to better understand the interplay of goal-setting and exerted effort in this project management context. Future research should investigate whether project agents in agile teams can be nudged to set more effective goals or whether exogenous goal-setting would be more effective. Finally, the continuous improvement philosophy of agile project management allows greater flexibility to make project changes than does traditional project management. This might improve an organization's agility in the literal sense, but it also creates questions regarding the effect of incremental iteration on the degree of creativity and innovation. Thus, extending the study to include the Scrum element of continuous improvement is another promising direction for future research.

3. Experimental Study 2: One Size Does Not Fit All: Strengths and Weaknesses of the Agile Approach

Co-authors: Evgeny Kagan, Sebastian Schiffels

Working paper (*Reject & Resubmit* at *Management Science* at the time of submission of this dissertation; preprint online at *SSRN*, DOI: 10.2139/ssrn.4105914).

Abstract. Agile project management techniques, such as iterative sprints and granting workers task autonomy, have become commonplace in many organizations. We experimentally examine how these techniques affect performance in two innovation settings: (1) a product development setting, represented by a task in which participants build connected word structures using letters of the alphabet, and (2) a business model innovation setting, represented by a task in which participants search for the best combination of business attributes on a multidimensional solution landscape. Our results suggest that the effects of Agile on performance are not uniform and depend on the innovation setting and on the performance measure. Agile improves average performance in the product development setting but lowers average performance in the business model innovation setting. In both settings, Agile techniques lead to more incremental (less radical) strategies, which narrows performance variance. Together, these results caution against uniform adoption of the Agile approach, and suggest that the choice of the approach should depend on the nature of the project and on the desired risk-return profile of the firm.

3.1. Introduction

"Agile" is a suite of workflow management techniques aimed at improving innovation performance (Fernandez and Fernandez 2008, Cooper and Sommer 2018). For example, the Scrum method, a common Agile approach that originated in software development, suggests that people should work in *sprints*—short project phases of equal length, punctuated by Scrum review meetings during which the progress is reviewed, and new tasks are assigned. The common theme of these techniques is that they emphasize iterative design and testing over component-wise development, and worker autonomy over top-down planning.[1]

Consider a web developer building an e-commerce website that has three pages: listings, shopping cart, and payment. The traditional, "Waterfall" approach would prescribe a sequential progression of activities, starting with one component (e.g., listings), and moving on to the next one (e.g., the shopping cart) upon completion. In contrast, the Agile approach would suggest that each development phase should end in a complete iteration of the website. That is, the developer would first create a basic version of listings, cart, and payment pages, potentially using mock-up or demo versions of some of the functionalities. Having built the "bones" of the website, the developer would then add detail and texture in each subsequent iteration.

The original purpose of Agile sprints was to facilitate the integration of user feedback into the development cycle of software applications. More recently, however, the Agile approach has advanced far beyond software, including settings where user feedback is less readily available, and where development is less incremental (Bryar and Carr 2021). For example, at BMW, a German automotive company, designs are kept secret and little user feedback is collected until the official product release. Despite this, BMW's management has recently moved a number of its design teams to the Agile workflow.[2] A similar push towards Agile adoption has been observed in many other settings with little customer involvement, including in pharmaceutical and other science-driven R&D (Di Fiore et al.

[1] More formal definitions of Agile methods, as well as the differences among them, can be found in the "Agile Practice Guide"—a practitioner handbook for Agile implementation, see www.pmi.org/pmbok-guide-standards/practice-guides/agile.

[2] We have personally witnessed this trend in several student projects co-advised with BMW's management.

2019), and even in the business-to-government sector (Roy et al. 2022).

Some features of Agile, for example, customer responsiveness (Srinivasan et al. 1997, Allon et al. 2021, Yoo et al. 2021) and team and communication processes (Wageman 2001, Hoda et al. 2012) have been studied in the academic literature. Other, more operational features of Agile, related to task scheduling, time allocation and worker productivity have received little attention and are still not well understood. Does Agile make workers more productive relative to Waterfall? Does it lead to more creative and diverse solutions? What are the key behaviors causing these differences?

We focus on two operational features of the Agile approach. First, the Agile approach is iterative. That is, in each sprint workers are asked to complete an integrated version of the product (sometimes referred to as a "minimum viable product" during early development), while fine polishing is delayed until later. To achieve this, workers need to split their time between multiple product components. The resulting workflow is quite different from the Waterfall approach which prescribes sequential completion of each component and thus allows the developer to focus on one component at a time. Second, Agile teams are expected to be self-organizing. That is, they are granted the autonomy to decide what to work on, in what sequence, and for how long. The proximity to the development and production process is meant to give workers an informational advantage to decide on the most value-adding use of their time, and also a motivational push, by giving them process ownership (Hackman and Oldham 1976, Raveendran et al. 2022).

The main criticism of the Agile approach is that it may stifle radical innovation. By splitting the work into ever smaller increments and by focusing on rapid product releases, the team may lose sight of the big picture (Petersen and Wohlin 2009). The presence of multiple tasks simultaneously competing for the worker's attention, and the frequent re-assessment of priorities, may further exacerbate this issue by shifting the worker's focus towards intermediate milestones and away from the final deliverable (Bryar and Carr 2021). Thus, both the iterative and incremental nature of Agile sprints, and the delegation of decision control to the worker, may detract from performance.

We use lab experiments to study how the Agile workflow affects worker behaviors, and how these behaviors affect performance. The lab setting is useful to study these questions because it provides a window into the creative processes and the types of activities engaged by people working under Agile versus Waterfall regimes. These insights complement the higher-level findings from the field studies of Agile (MacCormack et al. 2001, Allon et al. 2021, Roy et al. 2022), suggest some causal pathways that drive these findings, and help develop a better understanding of *when* Agile practices may work, and *why* (or why not).

To identify the strengths and the weaknesses of the Agile method we take a broad look across different innovation settings, and the experimental tasks that may represent them. We draw on the rich psychology literature that often uses open-ended design tasks to study creative processes (Sawyer 2011, and references therein), and on the economics-based approach of representing the creative process as search, often on complex multi-dimensional landscapes (Ederer and Manso 2013, Billinger et al. 2014, Sommer et al. 2020). The premise of our study is in recognizing that real-world innovation projects can have activities that are better represented by the more open-ended design tasks, and activities that are closer to the search approach.

Our "Design task" is a open-ended creative task with a material constraint. Participants are given a set of letters of the alphabet and are asked to build words into connected structures, similar to Scrabble. This task has an open solution space allowing boundless creative strategies (within the confines of the given materials). To do well participants need to engage in the types of activities involved in product development, i.e., the creative identification of opportunities (ideation), the choice of the most promising opportunities (selection), and the implementation of these choices into a final, connected design (execution).

Our "Search task" is a more structured task with a predefined (but very large) solution space. In this task participants search for the best combination of business attributes on a multidimensional solution landscape. This task has a finite (but complex) solution space, and is more reflective of business model innovation (Girotra and Netessine 2014), i.e., the systematic search and identification of key business decisions leading to successful new business models. Other examples of search-driven innovation include early-stage R&D activities in the pharmaceutical industry, a startup trying to position a new brand, or algorithm development. More generally, this task represents innovation settings where execution is straightforward once a good idea has been identified.

Our experiments are organized into a 2 (tasks) × 3 (workflows) experiment design. The two tasks (Design task and Search task) are administered within-subject, while the workflow is varied between-subject. The three between-subject treatments vary how participants split their time among problem components. Specifically, in each task there are two components that need to be completed. In the first treatment participants are restricted to work on components in a preassigned order, first completing one component and then moving on to the second one. This is similar to standard Waterfall practice, which prescribes a sequential workflow. In the second treatment participants work on *both* components during each sprint, thus completing a full iteration of the task by the end of the first sprint. However, the amount of time they spend in each component is still fixed. The third treatment is similar to the second one with the additional feature that the amount of time spent on each component is determined endogenously by the worker, rather than being fixed exogenously by the experimenter. Together these treatments allow us to separately identify the effects of the iterative workflow and of the increased autonomy of the Agile approach.

Our experimental results are as follows. First, the performance effects of Agile depend on the innovation setting. Agile significantly outperforms Waterfall in the Design task, and vice versa in the Search task. The size of these treatment effects is substantial, with average performance improvements of 12% to 16% (and up to 27% after controlling for the individual differences). Interestingly, autonomy does not significantly affect performance. That is, the bulk of the performance difference comes from the iterative nature of Agile development, and not from the worker having control over the time allocation.

Second, in addition to the differences in mean performance there are variance effects at the subject pool level. Specifically, in both tasks Waterfall leads to a greater performance variance than Agile. Thus, a firm that follows a high-risk high-reward strategy for its projects may choose a different approach than a firm that wants to improve performance on average.

Third, the performance effects are explained by more incremental (and less radical) behaviors in Agile regimes. In the Design task incrementalism manifests itself in the usage of similar words multiple times. Reusing the words helps maintain steady production pace and continue building and improving upon the existing product in a time and cost-efficient manner, but results in less creative solutions. In the Search task incrementalism manifests itself in the participants fine-tuning their solutions too early, instead of exploring a larger portion of the solution space. Here, the sequential nature of the Waterfall approach helps workers explore a larger number of possibilities, before committing and fine-tuning an already discovered solution. Survey questions further reveal that these behaviors are related to increased urgency and perceived time pressure in Agile regimes.

Taken together, our results caution against a "One size fits all" approach in project and innovation management. An approach that works well for one type of projects may lead to failure in another. Firms that have readily embraced the Agile paradigm may need to reevaluate how they manage workflow—especially for projects where experimentation is relatively cheap Agile may detract from performance. Organizations that manage a variety of different innovation projects should resist the urge to standardize their management approach and should instead tailor the approach to the nature of the project and to the desired risk-return profile of their portfolio.

3.2. Literature

While the Agile approach has attracted significant attention and debate among practitioners (Bazigos et al. 2015, Laufer et al. 2015, Rigby et al. 2016), academic research into its performance effects remains scarce. Nonetheless, we can draw on a large body of literature in organizational theory, psychology, experimental economics, and operations management, that studies broader questions related to innovation processes (Krishnan and Ulrich 2001). We next discuss two streams of literature that inform our experiment design, and that our study contributes to: the literature on Agile development and related process management techniques, as well as innovation experiments that use real-effort tasks. We note that our review focuses on the operational, i.e., process-related aspects of the Agile approach and omits other, team and communication related aspects.[3]

[3] The interested reader is referred to Tuckman (1965), Markham and Markham (1995), and Marks et al. (2001) for key references on self-organizing teams.

Agile Research

The first operational aspect of Agile is its iterative nature (Kettunen and Lejeune 2020). The deliverable for each iterative sprint is typically a demo version of the product that has all its basic functionalities, even during the early review cycles. The initial releases of the product may include rough drafts and mock-ups; the goal of these releases is not to be commercially viable, but rather to learn about the technological feasibility and to collect customer and management feedback (Yoo et al. 2021). After each review, the team can respond to the feedback by focusing on the most value-adding components.

The second operational aspect of Agile is the autonomy granted to developers when deciding how to allocate development time (Maruping et al. 2009, Hodgson and Briand 2013). MacCormack et al. (2001) find, using survey data in the software development sector, that a flexible approach where the team is granted some control over the progression of development activities leads to better results than a more stringent approach that allows teams to proceed from one development activity to another only after satisfying some preset requirements. More recently, Allon et al. (2021) use mobile app store data to show that app developers that are more agile (where agility is measured as the rate of changes to product version in response to user reviews) perform better. Notably, neither of these two studies can rule out the reverse causal sequence that high-performing organizations may also be more likely to adopt flexible development techniques. Our experiment helps validate the causal pathways suggested in these empirical studies and proposes some mechanisms that may be driving these effects.

The closest experimental studies related to Agile are Kagan et al. (2018) and Lieberum et al. (2022). Kagan et al. (2018) find that designers who decide for themselves how to spend time between creative ideation and execution perform worse than designers with exogenously imposed schedules; however, the effect disappears when autonomy is coupled with a performance-oriented deliverable, as would be the case for the Agile approach. Lieberum et al. (2022) show that time-boxing of work, i.e., imposing fixed time intervals for tasks, can improve performance. They use a pure effort (non-creative) slider task and do not study the role of iterative versus non-iterative task sequences. While both these studies examine regimes that give workers more/less process control, neither looks at multiple product components, or explores multiple innovation settings, both of which are central to a better understanding of the effects of Agile.

Taken together, the existing literature offers mixed predictions for the effects of Agile techniques on performance. Several observational studies suggest that iterative, more flexible workflow may improve performance. At the same time, experimental studies question some of the benefits of Agile, specifically the effects of autonomy on performance.

Real-Effort Innovation Tasks

Real-effort tasks have become quite common in the experimental literature studying questions related to worker productivity, including incentive design and worker compensation (Charness and Kuhn 2007, Greiner et al. 2011), server behavior in queues and assembly lines (Schultz et al. 1998, Shunko et al. 2018), and innovation (Erat and Gneezy 2016, Kagan et al. 2018, Rosokha and Younge 2020, Lieberum et al. 2022). The challenge for the design of innovation experiments like ours is to choose tasks that reproduce the creative environment, i.e., require a creative generation of new (rather than the use of existing) recipes for success, while at the same time allowing the researcher to maintain experimental control. Further, we are interested in tasks that would allow us to observe not only how well different people perform, but also what behaviors and strategies are driving performance. Fortunately, prior experimental literature has identified several classes of experimental tasks that achieve these goals.

Our first experimental task builds on the long tradition in the psychology literature of using verbal tasks to study creative behaviors (Sawyer 2011). The advantage of verbal tasks is that they do not require specialized training, and that performance can often be assessed using objective metrics. Within verbal tasks, there are some important distinctions. Some researchers use verbal tasks that are based on puzzles or riddles, for example solving a “rebus” (Kachelmeier et al. 2008, Kachelmeier and Williamson 2010, Erat and Gneezy 2016) or deciphering an anagram (scrambled list of letters) to form a word (Mendelsohn and Griswold 1964, Gino and Wiltermuth 2014). In these tasks, performance is measured by the number of puzzles solved, i.e., each puzzle is essentially treated as a new challenge. Such tasks are more reflective of brainstorming/ideation parts of the innovation process, where the objective is to produce as many ideas as possible, and less reflective of the product development setting, which includes ideation, selection, and implementation of ideas. Other verbal tasks are more unstructured, for example, writing an essay (Charness and Grieco 2019). Such tasks rely on subjective performance assessment and are more reflective of fashion or artistic settings. Our version of the verbal task is based on Scrabble; it has both the creative ideation/insight element of building new words, and a more analytic element of integrating the words into a final product that maximizes an objective performance metric. Thus, our task leverages the creative open-endedness of verbal tasks, while also requiring the creative energy to be directed towards a more pragmatic, performance-oriented goal.

Our second experimental task leverages the approach (more common among economists and business disciplines) of representing innovation as a search process (Levinthal and March 1981, Levinthal 1997). Here, the key objective of the worker is to identify the best solution among a very large number of potential solutions. Search models, especially search on complex landscapes, are a natural abstraction for many innovation processes, for example pharmaceutical trials (Powell and Ryzhov 2012, Chick et al. 2020), and other settings where the path to implementation is clear, once a good solution or strategy has been identified. To achieve good performance the developer needs to develop an understanding of the mapping between combinations of product attributes and the resulting performance. Rugged landscape models have been designed specifically to study such complex, multidimensional search processes (Levinthal 1997, Mihm et al. 2003, Sommer and Loch 2004).

While the theoretical/computational literature on complex solution landscapes is quite exhaustive (in particular for NK models; see Baumann et al. 2019, for a recent review), the number of experiments examining human search strategies on a landscape is relatively small. In these experiments the landscape is often represented by a lemonade stand where the decision-maker chooses the lemonade color, sugar content, location and other attributes, which interact in some complex ways (unknown to the participant).[4] Ederer and Manso (2013) examine

[4] Some experimental researchers prefer to use a context-free version of rugged landscape models, see for example, Billinger et al. (2014, 2021). These studies focus mainly on the ability of human decision-makers to calibrate how much to explore versus to exploit. Because we

different incentive systems and find that search strategies are more effective when short-term failure is tolerated and long-term success is rewarded. Sommer et al. (2020) examine whether groups perform better than individuals and find that the number of explored solutions is less predictive of success than the breadth of search. Overall, the experimental rugged landscape literature finds that both incentives and group dynamics matter, but do not delve deeper into questions related to task sequencing or time allocation within the search process. Our study contributes to this literature by examining the effects of workflow management techniques, such as Agile, on search performance.[5]

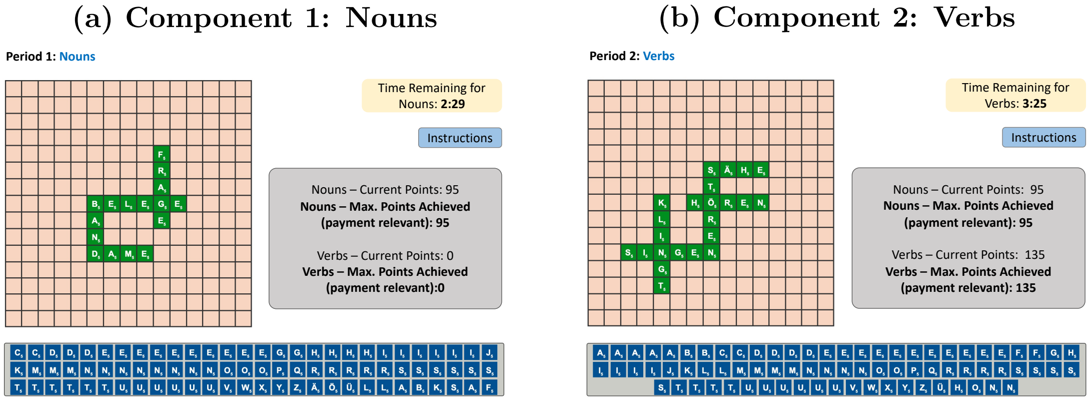
3.3. Experimental Design

In this section we present our experimental design. We begin by introducing two real-effort tasks which represent two different innovation settings (Subsection 3.3.1). We then present our experimental treatments, and discuss what treatment effects we anticipate given the extant theory (Subsection 3.3.2). Finally, we present the details of the performed measurements and protocols (Subsection 3.3.3).

3.3.1. Tasks (Within-Subject)

Our experiments were organized into a 2 (tasks) × 3 (treatments) experiment design. The tasks (administered within-subject, in random order) build on prior experimental work in the innovation and creativity literature discussed in Section 3.2. We refer to the two tasks as the “Design task” and the “Search task”.

Figure 3.1. Design Task: Screenshots



(a) Component 1: Nouns

Period 1: Nouns

Time Remaining for Nouns: **2:29**

Nouns – Current Points: 95
Nouns – Max. Points Achieved (payment relevant): 95

Verbs – Current Points: 0
Verbs – Max. Points Achieved (payment relevant):0

Instructions
FRAGE
BELEGE
BAND
DAME
C C D D D E E E E E E E E E E G G H H H H I I I I I I J
K M M M N N N N N N N N O O O P Q R R R R R S S S S S S
T T T T T T U U U U U U V W X Y Z Ä Ö Ü L L A B K S A F

(b) Component 2: Verbs

Period 2: Verbs

Time Remaining for Verbs: **3:25**

Nouns – Current Points: 95
Nouns – Max. Points Achieved (payment relevant): 95

Verbs – Current Points: 135
Verbs – Max. Points Achieved (payment relevant): 135

Instructions
SÄHE
STÖRE
HÖREN
KLINGT
SINGEN
A A A A A B B C C D D D D E E E E E E E E E E E F F G H
I I I I J K L L M M M M N N N N O O P Q R R R R S S S S
S T T T T U U U U U U V W X Y Z Ü H O N N

Notes: Design task sample screenshots (translated from German) for *Waterfall* treatment. The yellow boxes show the time remaining. The blue buttons are links to the instructions. The gray boxes show both the current scores for each component and the highest scores ever achieved, which are for payment (because all words are valid in this example, the scores coincide).

3.3.1.1. Design Task.

Our Design task is a variation on the Scrabble game. Subjects receive a set of tiles with letters on them, which can be used to form words. The words are then connected in crossword fashion, and must read left to right or top to bottom. Deviating from the classic version of the game, there are two separate boards that represent two product components. On one board subjects may only put nouns, and on the other board they may only put verbs. Each board has 15 × 15 fields. For each board, subjects receive 100 letters with no refill. The list of letters is the same for each participant. The first letter needs to be placed on the field in the middle of the board. Additional words must have at least one of their tiles horizontally or vertically adjacent to an already placed word. Words cannot be formed diagonally. An example of a subject working on each of the

study innovation-related behaviors, we use a contextualized version of the task with the lemonade stand business as the focal context.

[5] Another type of search experiments in the experimental literature are secretary problem experiments, see for example, Seale and Rapoport (1997), Bearden et al. (2006), Palley and Kremer (2014). These experiments represent single-dimensional search, more reflective of consumer or job market search (as opposed to the combinatorial search on multidimensional landscapes).

two components is shown in Figure 3.1.[6] For further details see Subsection B.1.1 in the appendix.

The overall performance, used to determine participant compensation, is computed as follows. First, the number of letters used is counted separately for each component. For overlapping words, the overlap letter is counted twice. For example, in Figure 3.1(a), the noun component has four words with 5, 6, 4, and 4 letters respectively. Subjects receive five points for each letter, yielding $(5 + 6 + 4 + 4) \times 5 = 95$ points in this example. The score is computed analogously in the verb component. For example, in Figure 3.1(b), the subject has five verbs with 4, 5, 6, 6, and 6 letters respectively, yielding the total score of $(4+5+6+6+6) \times 5 = 135$. At the end of the task, the smaller of the two component scores becomes the final payoff. This is to represent that a product has multiple components, and each of the components needs to be done well before the product can be taken to market. In this example the participant would earn $MIN\{95, 135\} = 95$ points.[7]

The Design task reproduces several key behavioral dynamics of product development. It is a problem solving task that requires both creative ("divergent", see Sawyer 2011) and analytic (convergent) thinking. Participants begin with an open solution space and limitless creative possibilities. There are no predefined strategies one can rely on, or decision alternatives to choose from. As in real projects, there are path dependencies: removing and rebuilding words can be costly, requiring more analytic, performance-oriented thinking, especially as the deadline nears. The final deliverable needs to be a product that integrates all the best ideas. The overall design thus requires a holistic approach that includes ideation, selection, and execution.

[6] The validity of each word placed on the board is instantly checked against the online dictionary wiktionary.org and highlighted in green color if valid. Placed words can be modified or deleted during the current period. However, words placed during the first period cannot be deleted during the second period.

[7] The MIN function ensures that participants work on both components, instead of working on the component they consider easier or more enjoyable. While other payoff functions may be equally suitable to represent complementarities between components, we chose the MIN function mainly to facilitate comprehension and easy calculation of profits for participants.

3.3.1.2. Search Task.

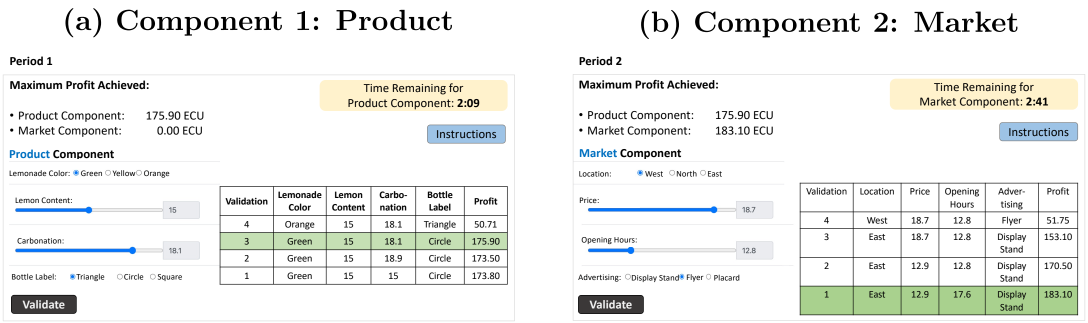
In our Search task subjects search for the profit-maximizing combination of business attributes on a multi-attribute solution landscape. As is common in the experimental literature (see, for example, Ederer and Manso 2013, Sommer et al. 2020) we use the naturalistic framing of the "Lemonade stand" to represent the solution landscape. In this framing, the participant is asked to identify an effective business strategy by repeatedly choosing the values of several business attributes and learning about the payoff resulting from each attribute combination. Deviating from the classic version of the task, we introduce two separate components of the lemonade stand: the product component and the market component. The product component consists of four product attributes: lemonade color, lemon content, carbonation, and bottle label. The market component consists of four market attributes: location, price, opening hours, and advertising. For each component two of the attributes are discrete, while the other two are continuous. Within a component, the payoff is a function of all four attributes.[8]

Figure 3.2 illustrates the decision screens for each of the two components. Participants can modify the attributes as often as they like, however, each time they do so, there is a three second delay until they see the resulting profit. This is to encourage thoughtful choices and to discourage random clicking. As in the Design task, the overall performance used to determine participant compensation is computed by taking the lower of the two component scores, where each component score is the best discovered solution. For further details see Subsection B.1.2 in the appendix.

Similar to the Design task, the Search task is a problem-solving task that involves both ideation and selection. Participants begin with a large, unexplored solution space (with a total of $92,000^2$ combinations). To do well, participants need to be able to effectively explore the space, then narrow down to a good solution region and fine-tune it. Importantly, the Search task does not have an implementation stage and is therefore more reflective of innovation settings

[8] Specifically, lemonade color, bottle label, location and advertising are discrete attributes. The remaining attributes are continuous. The continuous attributes allow inputs in the $[10, 20]$ range, with the choices limited to one digit after the decimal point, yielding a total of 101 possible choices each. Thus, the solution space in each component has $3 \times 3 \times 101 \times 101 = 92,000$ unique combinations. If we consider both components, the overall solution space has $92,000^2$ combinations.

Figure 3.2. Search Task: Screenshots



(a) Component 1: Product

Period 1

Maximum Profit Achieved:

- Product Component: 175.90 ECU
- Market Component: 0.00 ECU

Time Remaining for Product Component: **2:09**

Product Component

Lemonade Color: ● Green ○ Yellow ○ Orange

Lemon Content: 15

Carbonation: 18.1

Bottle Label: ● Triangle ○ Circle ○ Square

Validation	Lemonade Color	Lemon Content	Carbo-nation	Bottle Label	Profit
4	Orange	15	18.1	Triangle	50.71
3	Green	15	18.1	Circle	175.90
2	Green	15	18.9	Circle	173.50
1	Green	15	15	Circle	173.80

(b) Component 2: Market

Period 2

Maximum Profit Achieved:

- Product Component: 175.90 ECU
- Market Component: 183.10 ECU

Time Remaining for Market Component: **2:41**

Market Component

Location: ● West ○ North ○ East

Price: 18.7

Opening Hours: 12.8

Advertising: ○ Display Stand ● Flyer ○ Placard

Validation	Location	Price	Opening Hours	Adver-tising	Profit
4	West	18.7	12.8	Flyer	51.75
3	East	18.7	12.8	Display Stand	153.10
2	East	12.9	12.8	Display Stand	170.50
1	East	12.9	17.6	Display Stand	183.10

Notes: Search task sample screenshots (translated from German) for *Waterfall* treatment. The yellow boxes show the time remaining. The blue buttons are the links to the instructions. In each component participants can adjust each of the four attributes (radio buttons for the two discrete attributes and sliders for the two continuous attributes). The tables show each examined combination, with the best discovered combination highlighted in green.

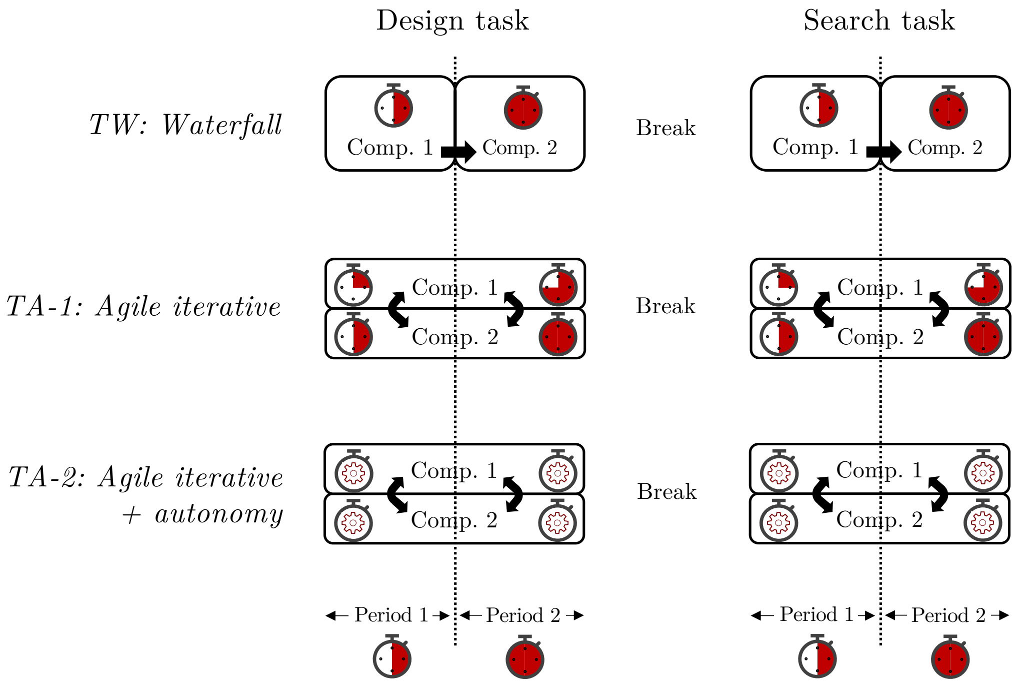
where execution is secondary once a good solution has been identified. This is typically the case in business model innovation, as well as other settings where the identification of the best solution under time constraints is key to successful performance.

3.3.2. Treatments (Between-Subject) and Anticipated Treatment Effects

We administer three between-subject treatments. In all three treatments the overall time for each task is fixed, and there are two periods of equal length. However, the workflow, i.e., the sequence of components and the time allocated to each component, depends on the treatment. In the *Waterfall* treatment (*TW*) participants complete the task sequentially, with exactly half of the total time allocated to each component. That is, in each period participants are restricted to working on one component. The sequence of the components is assigned at random.

In both Agile treatments participants are allowed to switch back and forth between the two components throughout the task. In the first Agile treatment, the total time spent on each component needs to be equal within each period (and therefore in total as well). We label this treatment *Agile iterative* (*TA-1*), because

Figure 3.3. Experimental Design



Notes. Treatment (*TW*, *TA-1*, or *TA-2*), sequence of tasks (Design task → Search task or Search task → Design task), and the first displayed component (Comp. 1 or Comp. 2) are assigned at random at the beginning of the experiment. In *TW* no modifications to the first displayed component can be made after the transition to period 2. In *TA-1* and *TA-2*, in both periods participants are allowed to switch between components as frequently as they see fit.

participants work on both components in each period, and thus complete a full iteration of the task in each period. The second Agile treatment is similar in that participants (can) work on both components in each period. In addition, the 50-50 time split constraint is removed. We label this treatment *Agile iterative + autonomy* (*TA-2*), because participants are given the autonomy to decide how to spend their time in the most productive way. The three treatments are administered between-subject and are summarized in Figure 3.3.[9]

[9] In all treatments, participants cannot go back and alter their choices once a period is completed. This is to reflect path dependencies caused by the choices made early on in the project. In the Design task, this is achieved by freezing the locations of the tiles placed in the first period. In the Search task, this is achieved by freezing half of the search attributes to the values that achieved the highest profit after the first period.

What treatment effects do we anticipate? The standard economic argument is that a relaxation of constraints would help a decision-maker allocate time to the more value-adding component; thus, the added flexibility of the Agile approach should improve performance. This is especially true for the *TA-2* treatment, in which participants can essentially replicate both the *TW* and the *TA-1* conditions. Agility has been shown to improve performance in the software and app development industries (MacCormack et al. 2001, Allon et al. 2021). At the same time, constraints have been shown to be helpful in many complex tasks because they allow the worker to focus on the (creative) task at hand (Sawyer 2011, Kagan et al. 2018, Long et al. 2020). This speaks for a more planned, sequential completion of components, as would be the case in Waterfall (*TW*).

Taken together, these arguments, and the review of the literature in Section 3.2 suggest that no theory or stream of literature offer uniform support for or against Agile. Given the limited theoretical and empirical investigation into the performance of Agile systems, we adopt an inductive, exploratory research approach. That is, rather than forming ex ante hypotheses based on extant theory we first examine behavior and performance in all three treatments (*TW*, *TA-1*, and *TA-2*) and then derive implications for what a more complete theoretical framework of creative behavior in operational systems may look like (Section 3.6).

3.3.3. Parametrization and Experimental Protocol

We conducted pretests with 33 participants to calibrate the duration of each task, the materials (number of letters in the Design task), and the payoff landscape (mapping between attributes and profit in the Search task). Task durations and the number of letters available were chosen to ensure that both time and material constraints were binding for most participants, yet sufficient for some to achieve top performance. We found that these goals were achieved with 100 letters and 6 minutes per period in the Design task, and 4 minutes per period in the Search task.

In the Search task, the created landscapes for each component had two local optima and one global optimum resulting in a solution landscape of moderate complexity. The global optimum for each component was set at 500 points, and the local optima were set at 380 and 200 points, respectively, ensuring that there was an incentive to continue searching once a local optimum was identified. Our parametrization is similar to the one used in Ederer and Manso (2013) and similar to the medium complexity scenario in Sommer et al. (2020). As in the standard implementation of the task, participants were not informed about the structure of the solution space, or the number of the optima. See Figures B.1 and B.2 in the appendix for an illustration of the solution space.[10]

The experiment was conducted at the Technical University of Munich, Germany between December 2020 and April 2021. The experimental interface was programmed in o-Tree (Chen et al. 2016). The experiment was conducted in German, the first language of most of the participants. Participants had to pass a German test to be admitted to the experiment. Participants were then randomly assigned to a treatment (see Section B.2 in the appendix for the full protocol and instructions). For each participant, the treatment (*TW, TA-1, TA-2*) was kept the same for both tasks. The sequence of the tasks and the sequence of components within each task were randomized. Participants were only allowed to proceed to each task after completing a comprehension quiz.

A total of 269 participants were recruited. A total of 13 participants were not admitted to the experiment because they were unable to pass the German test. A total of 62 participants were not admitted to the Design task because they did not pass the quiz. A total of 20 participants were not admitted to the Search task because they did not pass the quiz. The resulting number of valid observations was 194 for the Design task and 236 for the Search task. Participants were paid a fixed show-up fee of EUR 5 and a variable payment based on their performance in each of the two real-effort tasks. The average total payment was EUR 11.33.

[10] In the experiment we used two different parametrizations (each parametrization is a set of realizations of the attributes on the landscape). One of the two parametrizations was then selected at random at the beginning of each session. This was to ensure that behaviors would not be driven by a particular set of parameter realizations. Further, to ensure that no treatment would perform better simply because of the allowable decisions in each period we also conducted computational experiments using Monte Carlo simulations. Specifically, we generated 10,000 instances for each treatment using different search strategies, e.g., choose at random, modify one attribute at random, two attributes, etc. Within each instance we generated 40 validations (attempts), 20 for each component, consistent with the average number of validations (attempts) observed in pretests. We then conducted pairwise treatment comparisons drawing 60 samples for each treatment at random from the 10,000 instances and then compared treatment means using rank sum tests. No treatment was found to be systematically superior in these simulations.

The total duration of the experiment was 45 minutes.[11]

3.4. Performance Comparisons

In this section, we report the results of our analysis focusing on the effects of workflow (*Waterfall* vs. *Agile*) on performance. Our analysis relies on non-parametric tests and OLS regressions and uses two-sided p-values for the relevant statistical comparisons. We first use the subjects' payoff (the lower of the two component scores) as the dependent variable, and then examine several alternative performance measures. Pairwise correlations of the key measurements used in our analysis are found in the appendix (Table B.3).

3.4.1. Differences in Task Payoff

Figure 3.4 shows mean payoff by task and treatment. Several observations are in order. First, both *Agile* workflows improve performance in the Design task, but decrease performance in the Search task, relative to the *Waterfall* treatment. The differences are economically meaningful, ranging between 12% and 16% improvement from switching to the better workflow. Rank sum tests further reveal that the differences between the *Waterfall* treatment and each *Agile (TA-1* and *TA-2)* treatment are at least marginally significant in three of four comparisons (Design task: $p = 0.316$ and $p = 0.063$, Search task: $p = 0.037$, $p = 0.057$). Within *Agile*, if we compare *TA-1* and *TA-2* treatments, the effects of auton-

[11] This resulted in average hourly earnings of EUR 15.11, which is close to the targeted EUR 14/hour rate common for this subject pool.

Figure 3.4. Mean Performance

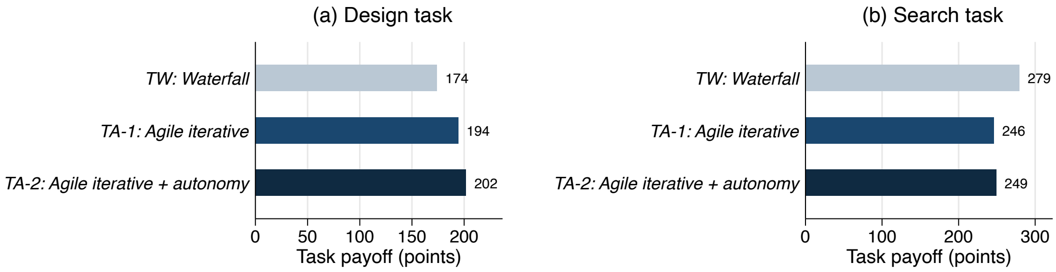


Table 3.1. Effects of Agile vs. Waterfall on Performance

	Design task		Search task	
	(1)	(2)	(3)	(4)
	Task payoff	Task payoff	Task payoff	Task payoff
TW: Waterfall	(Baseline)	(Baseline)	(Baseline)	(Baseline)
TA-1: Agile iterative	20.29	44.63^{***}	-33.45^{**}	-32.03^{**}
	(13.51)	(14.21)	(14.95)	(15.36)
TA-2: Agile iterative + autonomy	27.71^{**}	39.64^{***}	-30.20^{**}	-29.89^{*}
	(12.97)	(12.53)	(15.06)	(15.28)
Controls	No	Yes	No	Yes
Constant	173.90^{***}	165.91^{***}	279.29^{***}	300.70^{***}
	(8.90)	(42.65)	(9.94)	(52.31)
Number of observations	194	194	236	236
R^2	0.03	0.21	0.03	0.04

Notes. OLS regressions with standard errors in parentheses. Controls are age, gender, German native speaker, education, task sequence, component sequence, loss of internet connection, experience with Scrabble (in the Design task), and parameter version (in the Search task). The number of observations equals the number of participants who completed the task.
$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

omy are relatively small and not statistically significant (1%–4% improvement, $p = 0.335$ in the Design task and $p = 0.848$ in the Search task).

Before estimating linear regression models it is worth taking a brief look at some of the correlations between the measurements (Table B.3 in the appendix). First, performance (task payoff) is correlated between the two tasks at $\rho = 0.14$ ($p = 0.066$). The modest size of the correlation coefficient and its significance level both suggest that the two tasks are distinct measures of performance (rather than tests of the same underlying ability). Second, subjects' experience with verbal puzzles such as Scrabble, as well as their level of education are positively correlated with their performance on the Design task ($p < 0.001$ and $p = 0.030$). To account for these individual differences, we will control for them in our regression models.

The regression results are summarized in Table 3.1. Without controls, the positive effect of Agile on payoff performance in the Design task is not significant for *TA-1: Agile iterative*, but is significant for *TA-2: Agile iterative + autonomy* ($p = 0.135$ and $p = 0.034$). However, with controls, the positive effect of Agile becomes strongly significant for both Agile treatments ($p = 0.002$ and $p = 0.002$). The negative effect of *Agile* on performance in the Search task is significant for both *Agile* treatments without controls ($p = 0.026$ and $p = 0.046$), and also with controls ($p = 0.038$ and $p = 0.052$). In sum, the regression analysis provides robust evidence for positive performance effects of *Agile* in the Design task and for negative performance effects of *Agile* in the Search task. Lastly, none of the differences between *TA-1* and *TA-2* are statistically significant (Wald tests, all $p > 0.100$).

3.4.2. Alternative Performance Measures

In addition to examining task payoff (measured as the lower of the two component scores), we are also interested in the overall productivity, and in the difference between component scores. Were the treatment effects on task payoff caused by participants being more productive overall, or, were they caused by a better allocation of time and effort between components, leading to a more balanced performance across the components? To answer this question we first define two measures:

$$Sum\ of scores = Component\ 1\ score + Component\ 2\ score$$

$$Gap\ between\ scores = |Component\ 1\ score - Component\ 2\ score|$$

where each component score is the highest score achieved by the participant in a component (verbs or nouns component in the Design task, and product or market component in the Search task).

Table 3.2 reports the results of linear regression models with *Sum of scores* as the dependent variable in columns (1) and (3) and *Gap between scores* as the dependent variable in columns (2) and (4). Consider first the Design Task. Column (1) shows that relative to *TW*, only the *TA-1* treatment significantly increases the overall production of valid words, measured as the sum of scores across the two components ($p = 0.015$). In contrast, increased autonomy in *TA-2* does not significantly improve production relative to *TW* ($p = 0.110$). However, column (2) shows that both *TA-1* and *TA-2* significantly reduce the gap between scores, with a larger effect size for *TA-2* ($p = 0.009$ for *TA-1* and $p < 0.001$ for

Table 3.2. Sum of Scores and Gap between Scores

	Design task		Search task	
	(1)	(2)	(3)	(4)
	Sum of scores	Gap between scores	Sum of scores	Gap between scores
TW: Waterfall	(Baseline)	(Baseline)	(Baseline)	(Baseline)
TA-1: Agile iterative	63.09**	-26.18^{***}	-60.17^{**}	3.89
	(25.61)	(9.98)	(26.51)	(16.81)
TA-2: Agile iterative + autonomy	36.31	-42.98^{***}	-65.91^{**}	-6.12
	(22.59)	(8.80)	(26.38)	(16.73)
Controls	Yes	Yes	Yes	Yes
Constant	408.62***	76.80**	711.29***	109.90*
	(76.86)	(29.95)	(90.28)	(57.27)
Number of observations	194	194	236	236
R^2	0.21	0.16	0.07	0.07

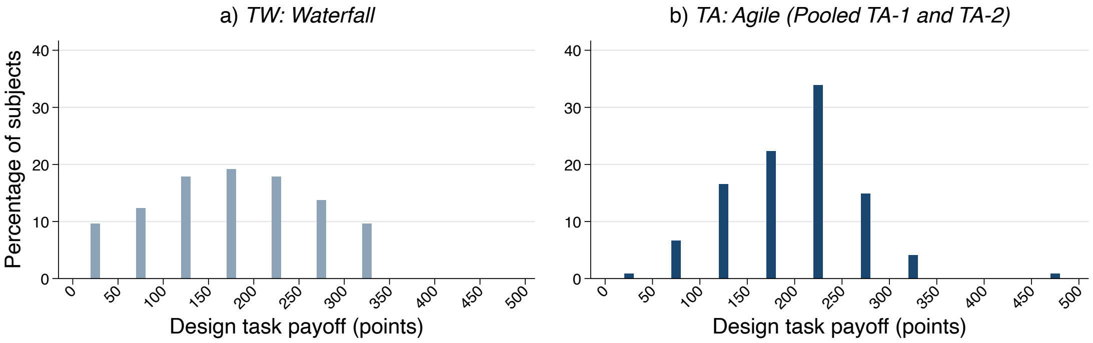
Notes. OLS regressions with standard errors in parentheses. Controls are age, gender, German native speaker, education, task sequence, component sequence, loss of internet connection, experience with Scrabble (in the Design task), and parameter version (in the Search task). The number of observations equals the number of participants who completed the task.
$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

TA-2). Thus, the key performance driver in the Design task appears to depend on the version of the *Agile* treatment: the iterative cycles help improve productivity; further, the increased autonomy helps participants allocate more time to the more value-adding activity.

The last two columns of Table 3.2 focus on the Search task. Column (3) shows that *TW* significantly increases the sum of scores relative to both *TA-1* and *TA-2* ($p = 0.024$ and $p = 0.013$). In contrast, column (4) shows that neither comparison is significant for the gap between the two component scores ($p = 0.817$ and $p = 0.715$). Thus, in the Search task *TW* appears to dominate both *TA-1* and *TA-2* primarily because of greater overall productivity, and not because of a more even allocation of performance between components.

Lastly, three of the four comparisons between *TA-1* and *TA-2* treatments are not statistically significant (Wald tests, $p > 0.267$) and one is marginally significant ($p = 0.074$) suggesting that there are no meaningful differences among the two *Agile* treatments.

Figure 3.5. Design Task: Performance Distribution



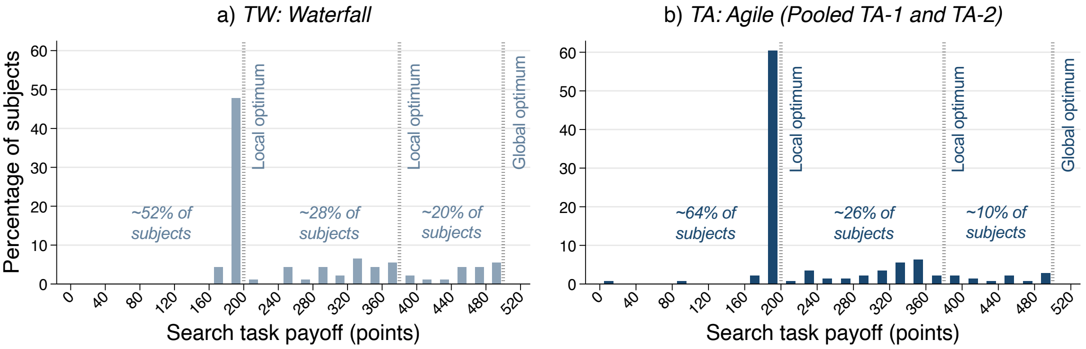
3.5. Performance Heterogeneity, Learning, and Mechanisms

In this section, we increase the level of detail and examine the micro-level dynamics of work under *Waterfall* and *Agile* regimes. We begin by exploring the performance distributions in each task. We then dive deeper into the relevant behaviors in each task and identify the key process indicators driving performance. For the purposes of exposition we pool *TA-1* and *TA-2* treatment data and compare behaviors in the pooled (*TA-1* + *TA-2*) treatment against *TW*. More detailed treatment comparisons are available in the appendix.

3.5.1. Performance Heterogeneity and Learning

We begin by considering the distributional effects of the Agile approach. Figure 3.5 shows the distribution of task payoff in the *Waterfall* and the pooled *Agile* treatment in the Design task. The *Waterfall* treatment causes a greater spread in performance with a substantive number of participants (10%) in both the lowest and highest bins. In contrast, in the *Agile* treatments performance is more tightly clustered around the mean and has a more pronounced fall-off towards both the lower and upper tail of the performance distribution. To examine these variance effects more formally we conduct a test of equality of variance and find the performance variance to be significantly higher in the *Waterfall* relative

Figure 3.6. Search Task: Performance Distribution



to the (pooled) *Agile* treatments (Levene test: $p < 0.001$).[12]

Figure 3.6 shows the performance distribution in the Search task. Here, the advantage of the *Waterfall* treatment is related to the participants' ability to discover multiple local optima. Indeed, a smaller number of subjects in the *Waterfall* treatment are stuck in the bottom local optimum where one can reach at most 200 points (52% vs. 64%, Proportion test: $p = 0.074$). Further, the proportion of subjects able to identify the highest optimum is significantly higher in *Waterfall* (20% vs. 10%, Proportion test: $p = 0.031$). Thus, the *Waterfall* treatment appears to shift the distribution of outcomes away from the bottom local optimum, and towards the global optimum. We will later see that this shift is related to the explore-exploit patterns observed in each treatment. Finally, as in the Design task, the variances between the *Waterfall* and (pooled) *Agile* treatments in the Search task are significantly different (Levene test: $p = 0.026$).[13]

Some of the described variance effects appear to be driven by differences in the learning patterns. Specifically, in both tasks Component 2 scores are significantly higher than Component 1 scores in the *Waterfall* treatment (paired t-tests of highest scores, both $p < 0.01$). In contrast, neither score is significantly higher

[12] Treatment-level histograms of task payoff are available in the appendix (Figure B.3). Quantile regressions that use different quantiles of the performance distribution as the dependent variable confirm that *Agile* mainly improves the outcomes of the low performers and does not significantly affect high performers. Quantile regression coefficients are reported in Table B.4 in the appendix.

[13] Treatment-level histograms of task payoff are available in the appendix (Figure B.4). Quantile regression results that use different quantiles of the performance distribution as the dependent variable are reported in Table B.7 in the appendix.

nor lower in the *Agile* treatments. The treatment differences in learning patterns are intuitive: In the *Waterfall* treatment participants work on a single component within each period; thus the lessons from the first period can be applied when working on the second component. In contrast, in the *Agile* treatments participants can work on both components within a period; hence, the displayed sequence of components has only a small effect on performance. The sequential nature of the Waterfall process may thus lock in the initial differences in ability between low and high performers, leading to a greater spread in final performance.

Taken together, the variance effects identified in this subsection suggest that the choice of the development approach (*Waterfall* vs. *Agile*) may depend on the risk appetite of the decision-maker. The managerial implications of this result will be discussed in Section 3.6.

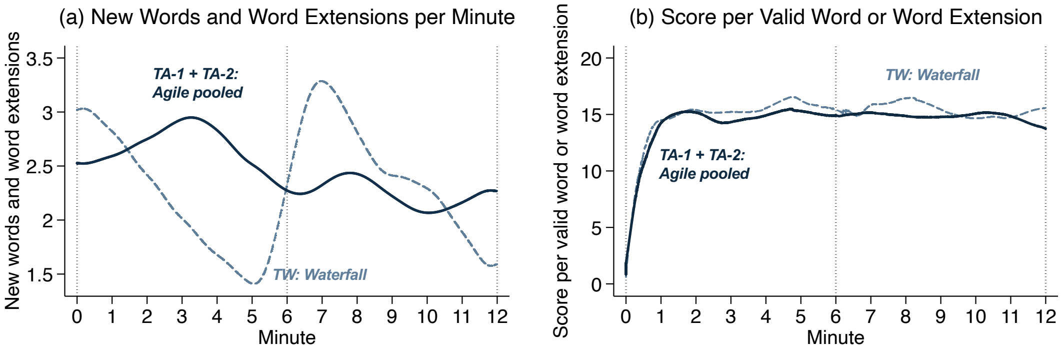
3.5.2. Mechanisms

3.5.2.1. Design Task.

In the Design task, two natural measurements of the creative process are the ability of participants to form words and the length of those words. Figure 3.7(a) shows that the number of words added to the board is quite high at the beginning of each period in the *Waterfall* treatment. However, the number of words drops rapidly over the course of each period, from 3.0 to 1.3 words per minute down in period 1 and from 3.4 to 1.6 words per minute in period 2. Indeed, the decline in the number of words is significant in both periods in the *Waterfall* treatment (non-parametric trend test, both $p < 0.001$). In contrast, the number of words remains quite constant in Agile with 2.1 to 3.0 words per minute throughout the task (trend tests, $p = 0.702$ and $p = 0.359$). Further, the average length of words does not appear to change over time in either treatment (Figure 3.7(b)).

How are participants in *Agile* regimes able to maintain high productivity and better distribute their effort among the two components? One effective strategy appears to be word recycling. That is, rather than looking for entirely new words participants can use the same or similar words multiple times. Our data show that this strategy is associated with increased performance and is used more frequently in the *Agile* treatments. Only 7% of the participants in *Waterfall* use identical words multiple times, compared to 19% in *Agile*. This difference is statistically

Figure 3.7. Design Task: Process Variables over Time



Note. Graphs use locally weighted smoothing (bandwidth: 0.2).

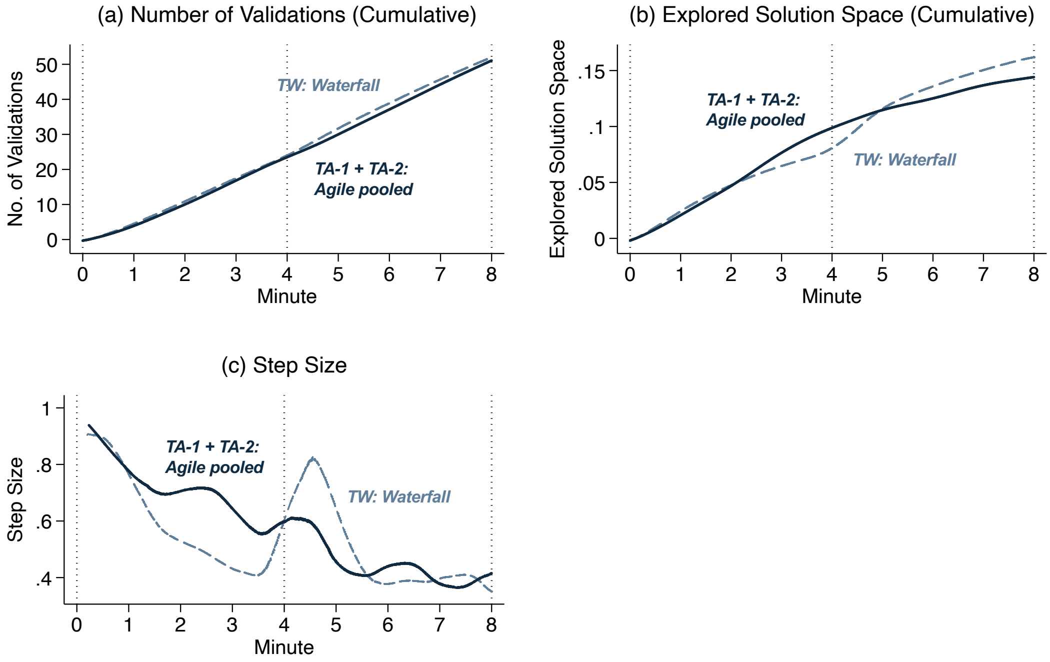
significant (rank sum test, $p = 0.020$). Indeed, we find that this behavior explains between 10% and 23% of the treatment differences in performance (Tables B.5 and B.6 in the appendix). Thus, participants in *Agile* regimes appear to discover more time and cost-efficient (though not necessarily more creative) strategies that help them achieve higher performance.[14]

3.5.2.2. Search Task.

Next, we unpack the drivers of performance in the Search task. To get a sense of the search process in each treatment, we examine three common metrics used to characterize the search process on complex solution landscapes (see, for example, Sommer et al. 2020): the total count of examined solutions (*Number of validations*), the coverage of the solution space (*Explored solution space*), and the magnitude of the differences between two subsequent solutions *(Step size)*. The *Number of validations* is a proxy for the number of ideas. Since only the best solution counts, in the Search task each additional idea should—on average—lead to weakly better results. However, because the time, and thus the number of attempts, is finite, there is a trade-off between exploring a large number of disparate solution regions, versus exploring a more narrow set of regions with greater thor-

[14] We also examine whether participants reuse partial (rather than complete) words and find similar treatment differences and performance effects. Note also that we measure only the direct effects of word recycling on performance; other more indirect contributions may come from new combinations that are enabled by the recycled words and would have been impossible had the recycled word not been added to the board.

Figure 3.8. Search Task: Process Variables over Time



Note. Graphs use locally weighted smoothing (bandwidth: 0.2).

oughness. This trade-off is captured by the remaining two measures. Specifically, *Explored solution space* measures the breadth of the search, i.e., how much of the idea pool has been explored (Kavadias and Sommer 2009, Erat and Gneezy 2016, Kornish and Hutchison-Krupat 2017), while *Step size* measures whether the subject is experimenting with new solutions, or fine-tuning the current one (Billinger et al. 2014).

Figure 3.8 plots each of the three measures. Panel (a) shows that in both *Agile* and *Waterfall* treatments participants explore approximately 50 distinct solutions at a near-constant pace. That is, they validate a new combination of attributes approximately every nine seconds. At the end of the task, there are no significant differences in the number of validations between treatments (rank sum test, $p = 0.447$). That is, the differences in the number of validations are not a significant driver of the performance advantage of *Waterfall* in the Search task.

Panel (b) plots the *Explored solution space* over time. To compute this measure, we first partition each of the four attributes of each component into three buckets, resulting in a solution space of 162 distinct fields (2 components × 3 buckets$^{4\ attributes}$ = 162), and then calculate what proportion of those fields has been explored. In the first period participants explore significantly more of the solution space in *Agile* than in *Waterfall* (10% vs. 8%, rank sum test, $p < 0.001$). However, *Agile* participants do significantly less exploration in the second period, relative to *Waterfall* (4% vs. 8%, rank sum test, $p < 0.001$). At the end of the task, *Waterfall* participants have explored on average 16% of all fields, whereas *Agile* have explored significantly less with only 14% (rank sum test, $p = 0.029$).[15]

Panel (c) plots the *Step size*, i.e. the Euclidean distance between the attributes chosen in two subsequent validations. This measure is a proxy for the extent of experimentation and helps understand whether a participant is performing broader exploration (which would manifest in a larger *Step size*) or exploitation, i.e. fine-tuning (which would manifest in a smaller *Step size*).[16] In *Waterfall* participants explore broadly at the beginning of each period, cutting their *Step size* in half as they end each period. In contrast, in *Agile* there does not appear to be a sharp increase in step size after the first period, but rather two small increases in the middle of each period. Together panels (b) and (c) suggest that participants in *Agile* are under-exploring the solution landscape during the second period and anchor too strongly on the solutions discovered in the first period.

In Tables B.8 and B.9 in the appendix, we further examine the extent to which each of the three process measures graphed in Figure 3.8, (a)–(c) can explain performance differences between *Agile* and *Waterfall.* Notably, two of the three measures, *Number of validations* and *Explored solution space* are positive predictors of performance, even after controlling for the treatment effects. Among these two measures, the more accurate predictor is *Explored solution space*, which explains between 15% and 28% of the difference in performance. In contrast, *Step size* does not explain the treatment differences.

[15] The results in this paragraph are robust to an alternative discretization of the continuous variables into four/five buckets.

[16] We focus on the component that is currently being worked on by the participant (as opposed to looking at the sum of Euclidean distances across both components).

3.6. Discussion

Our first result is that, on average, participants in the *Agile* treatments outperform *Waterfall* in the Design task, but lag behind in the Search task. The differences are not only statistically, but also economically significant: The performance improvement from switching to the better approach, measured by the average marginal effects in regression analysis, ranges from 12% to 16% (and up to 27% after controlling for the individual differences). The main implication of these results in practice is that Agile techniques may work better in some settings and worse in others. Projects that involve creative generation and implementation of new strategies, as would be the case in product design and development, may be able to benefit from Agile techniques. In contrast, projects that require a more analytic search and selection of the best alternative, for example in business model innovation, may not.

In our Design task, being able to switch back and forth between multiple creative subtasks led to increased productivity (measured by the total length of words produced) and to a more even allocation of creative performance among the components (measured by the gap between component scores). Our analysis of the performance and behavior changes over time suggests that being able to switch from one subtask to the other may help prevent creative blocks and achieve high productivity faster. In contrast, with sequential completion of components, especially in the early phase workers appeared to struggle with generating and implementing ideas, which led to reduced productivity and uneven component performance.[17]

Different from the Design task, in the Search task parallel completion of subtasks was shown to reduce performance. This is because in *Agile* regimes workers appeared to cut short their exploration efforts, committing instead to the first acceptable solution. The presence of parallel subtasks appears to put workers under pressure to deliver acceptable performance after the initial sprint, discourages broader exploration of the solution landscape, and leads to quicker convergence to a (local) optimum. In contrast, sequential completion of subtasks encourages a more effective and better calibrated explore-exploit strategy, resulting in

[17] This also suggests that *Waterfall* may perform better if the sequence of tasks is ordered from the easier to the more difficult component.

improved performance.

While worker performance in both *Agile* treatments was significantly different from *Waterfall,* we saw no significant differences *among* the two *Agile* treatments. That is, the main performance differentiator is the ability to work on multiple components within each sprint and not the increased/decreased worker autonomy. This null result is surprising in two ways. First, standard economic theory would predict that a less constrained action set should improve production output. Workers have an informational advantage and should be in a better position to determine the best use of their time and effort—restricting their autonomy adds an extra constraint. Second, autonomy has been shown to have motivational effects in some job design settings (Hackman and Oldham 1976, Raveendran et al. 2022). Our data show no evidence of these effects. We thus add to the growing body of work showing that more autonomy may not always improve performance in complex tasks, such as product design (Kagan et al. 2018), project selection and abandonment (Long et al. 2020), and time and effort allocation (Lieberum et al. 2022), and show that autonomy may indeed be harmful in search tasks.

In addition to the treatment differences in average performance we also saw some differences in variance. Specifically, the *Agile* approach decreased the frequency of low performance outcomes and also decreased the frequency of top performance outcomes, condensing the performance distribution more closely around its mean. This has meaningful implications for firms that manage a portfolio of innovation projects or choose from a pool of submissions (for example, through crowdsourcing contests or other competitive programs). If the objective is to avoid failure or to maximize average performance of the projects, then the choice of the management technique should depend on the type of project: Agile may be a better choice in product design and development, whereas Waterfall may be a better choice for search-driven projects, for example, when developing a new business model. However, if the objective is to maximize the number of top performing projects, then Waterfall may be preferred to Agile in both settings. These risk-return trade-offs are summarized more succinctly in Table 3.3.

Our study does not examine the psychological drivers of the observed behaviors, nor do we measure personality traits that may moderate the observed effects. One plausible psychological explanation, however, appears to be the increase in time pressure perceived by the participants in our *Agile* treatments. Indeed, in

Table 3.3. Agile vs. Waterfall Framework

	Objective:		
Innovation setting:	Avoid failure	Maximize average performance	Maximize top performance
Product development	Agile	Agile	Agile/Waterfall
Business model innovation	Agile/Waterfall	Waterfall	Waterfall

our exit questionnaire participants in the *Agile* treatments reported a significantly higher perceived time pressure relative to *Waterfall* ($p = 0.008$ for TW vs. pooled *TA-1 + TA-2* comparison), despite the total working time being the same in all treatments. Increasing time pressure and urgency may help productivity, especially at the low end of the performance distribution. Meanwhile, time pressure may hinder workers from developing a holistic cognitive approach necessary to solve more analytic problems, like the problem faced by the study participants in our Search task. Indeed, the organizational psychology literature suggests that the time pressure caused by frequent deadlines may be harmful or beneficial, depending on the environment and on the personality of the worker (Amabile et al. 2002, Baer and Oldham 2006) as well as on how success is measured (Ghosh and Wu 2021). A more nuanced understanding of these effects may help firms better leverage the strengths and the weaknesses of the Agile approach.

3.7. Concluding Remarks

As Agile expands beyond software development, it is important for both researchers and practitioners to develop a more systematic understanding of the method's relative strengths and weaknesses. In this paper we have taken a behavioral approach to add to this understanding. We focused on two operational features of the Agile approach (job sequencing and decision control) and examined their effects on human behavior in two distinct real-effort tasks: a design task and a search task. To be able to examine Agile techniques, we adapted classic versions of these tasks (used in prior experimental literature) by splitting each task into two components and by incentivizing workers based on the lower component score.

Taken together our results suggest a contingent framework for the choice of the approach, where the contingencies are (1) the nature of the innovation task and (2) the desired risk-return profile of the project. Our data suggest that design performance benefits from Agile techniques. This is because Agile increases productivity, particularly in the early phases of work, and allows cross-component learning. Agile appears to especially benefit low performers. In contrast, search performance suffers from Agile techniques. This is because significant portions of the solution space remain unexplored in Agile regimes, and as a result, many workers are unable to identify and climb the "hill" with the global optimum. In contrast, Waterfall facilitates more effective and better calibrated explore-exploit behaviors that lead to superior results.

Our findings have several meaningful implications for practice. First, our results caution against a "One size fits all" approach when choosing a workflow management approach. An approach that works well for one type of projects may lead to failure for a different type. Even within a project there may be phases that are more search-driven and phases that are more design-driven. For example, in preclinical vaccine development, an R&D team would first explore a large number of alternatives to identify the most effective combination of an antigen and an adjuvant co-injected with the antigen (a search task). Having identified the compounds the team would then proceed to vaccine formulation (a design task). Second, the right approach may also depend on the objectives of the manager. Agile reduces performance variance and may therefore be preferred when there are few fallback options if the project fails. In contrast, Waterfall may be preferred when risks can be afforded, for example when managing a more speculative project in the firm's portfolio, or when managing a large number of teams trying to solve the same problem.

As one of the first experimental tests of the Agile/Waterfall dichotomy our study takes a broad look across structurally distinct innovation tasks. Future work may be able to generate more textured insights by examining search or design activities in more detail. For example, the experimental rugged landscape literature suggests that humans adopt different strategies and perform differently in different parametrizations of the rugged landscape. Indeed, Billinger et al. (2014) show that humans may underexplore or overexplore depending on landscape complexity. Similarly, it is possible that the performance effects of Agile may depend on the availability and cost of resources in design tasks. A natural next step would thus be to examine how task complexity interacts with the effectiveness of the Agile approach. Second, given our preliminary findings related to the role of time pressure, it may be interesting to explore the moderating effects of personality. Do Agile techniques work better/worse for different personality types? Finally, an interesting extension of our study would be to examine hybrid regimes that leverage the benefits of each approach. Such hybrid approaches are becoming increasingly common (Roy et al. 2022). In a hybrid approach, Agile may help in the initial sprints (to help jump-start creative production), whereas Waterfall may be more beneficial in the later sprints to organize and finesse the solutions. An adaptation of our experiment to study these questions would be both straightforward and informative.

4. Conclusion

Project management is to a large extent a people business. The reaction of project members and other stakeholders to different project management schemes heavily influences project success. As the global economy increasingly evolves from product to service offerings, and thus an ever-increasing proportion of the workforce is engaged in service delivery, the importance of successful project management will only increase. To address this evolution, the agile development paradigm is gaining acceptance across a wide range of industries and business functions, yet little is known about whether and when agile project management leads to good results. With billions of U.S. dollars lost globally each year due to poor project performance, this discrepancy between practical relevance and academic rigor in understanding Agile leaves us with an open question: Does Agile make project performance indeed less (or even more) fragile? This dissertation brings us closer to understanding, and thus answering, this question by comparing agile project management with traditional waterfall project management in the light of human behavior. It consists of two separate experimental studies that examine the topic from different, complementary perspectives.

In the first study, we present what to our knowledge is the first experimental publication on the effects of agile sprints on quantitative project performance and execution. We contribute an operationalization of project execution as a stylized real-effort task along with three main findings to the field of project management. First of all, we show how in traditional waterfall project management without forced progression through the project phases, there is a risk of delayed progression as project agents spend too much time on early project phases at the expense of later ones. We refer to this newly described effect as “Progression Fallacy”. Secondly, time-boxed progression in agile sprints mitigates the Progression Fallacy and improves the overall performance. Thirdly, we provide evidence that self-imposed, phase-specific output goals with flexible progression, as are common in traditional project management, can amplify progression delay and distort effort. This can be avoided by combining self-imposed, phase-specific output goals with time-boxed progression, as is common in agile project management.

These findings have direct managerial implications. Our results suggest that managers should not only monitor well-known biases in project management, such as planning fallacy, procrastination, or Parkinson's law. If projects are executed with flexible progression, as is common in traditional project management, particular managerial attention is also needed on the optimal time allocation across project phases, in particular when ambitious goals are to be achieved. Otherwise, project members succumbing to overdelivering started tasks can be a driver of underperformance and delays. Working in agile sprints with time-boxed progression and phase-specific output goals can facilitate on-time task completion and improve performance. However, here particular managerial attention is needed on setting sufficiently motivational goals; otherwise, goal-setting has little impact.

In the second study, we experimentally examine how agile project management techniques affect innovation performance. We contribute a performance comparison between agile project management and traditional waterfall project management in the two common innovation archetypes of creative design and search to the field of project management. Our results suggest that agile project management improves performance in the creative design task, but harms performance in the search task compared to traditional waterfall project management. The effects of Agile on project performance are not uniform, but depend on the setting and performance measure. Agile helps to achieve minimum viable solutions early on. In both tasks, it reduces performance variance. In particular, low performers seem to benefit from the urgency created by agile sprints, which helps mainly in the design task. However, Agile can reduce top performance and lead to more incremental innovation. We argue that the time pressure from working in short agile sprints can lead to the avoidance of larger, and therefore riskier, developments in favor of incremental improvements. Finally, whereas project members may be more satisfied with having more autonomy under agile project management, we find that it does not lead to significantly higher performance (nor does it cause harm). Instead, most of the difference in performance is due to the iterative nature of Agile, rather than project members having control over time allocation.

These findings provide an evidence base for a debate that is actively being conducted among practitioners: Does Agile promote or hinder innovation performance? The answer is: It depends. On the one hand, the increased urgency that comes from dividing work into sprints can help project agents overcome initial roadblocks and immerse themselves in the task at hand—especially in creative development. On the other hand, iterative creation of project increments can incentivize exploitation strategies that refine existing solutions rather than creating radical innovations—especially in more analytical tasks. Here, the sense of urgency can backfire as it leads to insufficient exploration of available solutions and a premature commitment to a potentially suboptimal strategy. Whereas such short-termism effects are anecdotally known to some agile practitioners, we are not aware of any studies that rigorously document them. The managerial implications are that the choice of the project management approach should depend on the nature of the project and the organization's desired risk-return profile.

Merging the insights from both studies, the effects of agile project management are ambiguous. On the one hand, it can help mitigate behavioral flaws in traditional waterfall project management and improve quantitative performance. In particular, it can help to achieve a more balanced performance across the different components of a project, as early project phases receive less of the overproportional time and attention they receive under traditional project management. This leaves more resources available for later phases. It can also help to get started and deliver a minimum viable solution quickly—especially for creative tasks and low performers. On the other hand, it can distort qualitative performance by promoting mediocre solutions. Stellar innovation seems to benefit from the greater degrees of freedom offered by traditional project management. Taken together, these findings suggest that organizations should think carefully when deciding whether and when to adopt Agile. Instead of a uniform adoption, the appropriate project management approach should depend on the nature of the project and the organization's desired risk-return profile. To achieve this, managers need to be familiar with both the advantages and disadvantages of agile and traditional project management. Due to the heterogeneity of tasks within a project, best-of-breed hybrid versions of agile and traditional waterfall project management can leverage the strengths of both approaches.

The findings from this dissertation apply not only to project management, but are also relevant in the broader context of task completion and extend to many areas of life: Regardless of what you are working on, be it a commercial project, an academic paper, or charitable volunteering, you should be aware of the advantages and disadvantages of a time-boxed versus a more flexible, as well as an iterative versus a sequential progression regime—especially if you are setting ambitious goals. Thus, this dissertation lays the foundation for future research in project management as well as in related social science fields, both on the content and methodological level: On the content level, we introduce the newly described "Progression Fallacy" as well as ambiguous effects of different progression regimes and goal-setting as core findings. On the methodological level, we introduce new operationalizations of project execution as stylized experimental tasks, which enable investigation of the project agents' behavior across project phases under laboratory control.

Given the breadth of the research subject, further work is needed to advance our understanding. In terms of content, this dissertation cannot cover all the differences between agile and traditional waterfall project management. In particular, core elements such as planning, feedback, and team composition should be further investigated in the context of our findings. Methodologically, the laboratory findings of this dissertation should be verified in the field. In particular, insights into long-term differences between agile and traditional waterfall project management are of interest. Do the strengths and weaknesses of both approaches persist over time, or do they converge, e.g., because one leads to greater exhaustion of the project team? Reinforced by future advancements, the results of this dissertation can contribute to better project performance.

Bibliography

Allon G, Askalidis G, Berry R, Immorlica N, Moon K, Singh A (2021) When to be agile: Ratings and version updates in mobile apps. *Management Sci.* 68(6):3975–4753.

Amabile TM, Hadley CN, Kramer SJ (2002) Creativity under the gun. *Harvard Bus. Rev.* 80(8):52–63.

Ambler SW (2002) *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process* (John Wiley & Sons, New York).

Anderson DJ (2010) *Kanban: Successful Evolutionary Change for Your Technology Business* (Blue Hole Press, Sequim, WA).

Baer M, Oldham GR (2006) The curvilinear relation between experienced creative time pressure and creativity: Moderating effects of openness to experience and support for creativity. *J. Appl. Psych.* 91(4):963–970.

Bandura A (1989) Self-regulation of motivation and action through internal standards and goal systems. Pervin LA, ed. *Goal Concepts in Personality and Social Psychology* (Lawrence Erlbaum Associates, Hillsdale, NJ), 19–85.

Baumann O, Schmidt J, Stieglitz N (2019) Effective search in rugged performance landscapes: A review and outlook. *J. Management* 45(1):285–318.

Bazigos M, Smet AD, Gagnon C (2015) Why agility pays. *McKinsey Quart.* (4):28–35.

Bearden JN, Rapoport A, Murphy RO (2006) Sequential observation and selection with rank-dependent payoffs: An experimental study. *Management Sci.* 52(9):1437–1449.

Beck K (2003) *Extreme Programming Explained: Embrace Change*, 8th ed. (Addison-Wesley, Boston).

Bendoly E, Croson R, Goncalves P, Schultz K (2010) Bodies of knowledge for research in behavioral operations. *Production Oper. Management* 19(4):434–452.

Bendoly E, Donohue K, Schultz KL (2006) Behavior in operations management: Assessing recent findings and revisiting old assumptions. *J. Oper. Management* 24(6):737–752.

Billinger S, Srikanth K, Stieglitz N, Schumacher TR (2021) Exploration and exploitation in complex search tasks: How feedback influences whether and where human agents search. *Strategic Management J.* 42(2):361–385.

Billinger S, Stieglitz N, Schumacher TR (2014) Search on rugged landscapes: An experimental study. *Organ. Sci.* 25(1):93–108.

Boudreau J, Hopp W, McClain JO, Thomas LJ (2003) On the interface between operations and human resources management. *Manufacturing Service Oper. Management* 5(3):179–202.

Bryar C, Carr B (2021) Have we taken agile too far? *Harvard Bus. Rev.*, Digital Article (April 9), https://hbr.org/2021/04/have-we-taken-agile-too-far.

Charness G, Grieco D (2019) Creativity and incentives. *J. Eur. Econom. Assoc.* 17(2):454–496.

Charness G, Kuhn P (2007) Does pay inequality affect worker effort? Experimental evidence. *J. Labor Econom.* 25(4):693–723.

Chen DL, Schonger M, Wickens C (2016) oTree—An open-source platform for laboratory, online, and field experiments. *J. Behav. Experiment. Finance* 9:88–97.

Chick SE, Gans N, Yapar O (2021) Bayesian sequential learning for clinical trials of multiple correlated medical interventions. *Management Sci.* 68(7):4919–4938.

Coad P, Lefebvre E, DeLuca J (1999) *Java Modeling in Color with UML: Enterprise Components and Process* (Prentice-Hall, Upper Saddle River, NJ).

Cockburn A (2005) *Crystal Clear: A Human-Powered Methodology for Small Teams* (Addison-Wesley, Boston).

Conboy K (2009) Agility from first principles: Reconstructing the concept of agility in information systems development. *Inform. Systems Res.* 20(3):329–354.

Cooper RG, Sommer AF (2018) Agile—Stage-gate for manufacturers. *Res.-Tech. Management* 61(2):17–26.

Deci EL, Ryan RM (1985) *Intrinsic Motivation and Self-Determination in Human Behavior* (Springer, Boston).

Di Fiore A, West K, Segnalini A (2019) Why science-driven companies should use agile. *Harvard Bus. Rev.*, Digital Article (November 4), https://hbr.org/2019/11/why-science-driven-companies-should-use-agile.

Doerr KH, Gue KR (2013) A performance metric and goal-setting procedure for deadline-oriented processes. *Production Oper. Management* 22(3):726–738.

Ederer F, Manso G (2013) Is pay for performance detrimental to innovation? *Management Sci.* 59(7):1496–1513.

Erat S, Gneezy U (2016) Incentives for creativity. *Experiment. Econom.* 19(2):269–280.

Ettlie JE (1998) R&D and global manufacturing performance. *Management Sci.* 44(1):1–11.

Fan J, Gómez-Miñambres J (2020) Nonbinding goals in teams: A real effort coordination experiment. *Manufacturing Service Oper. Management* 22(5):1026–1044.

Fernandez DJ, Fernandez JD (2008) Agile project management—Agilism versus traditional approaches. *J. Comput. Inform. Systems* 49(2):10–17.

Fischbacher U (2007) z-Tree: Zurich toolbox for ready-made economic experiments. *Experiment. Econom.* 10(2):171–178.

Forsyth DK, Burt CDB (2008) Allocating time to future tasks: The effect of task segmentation on planning fallacy bias. *Mem. Cogn.* 36(4):791–798.

Fu R, Subramanian A, Venkateswaran A (2016) Project characteristics, incentives, and team production. *Management Sci.* 62(3):785–801.

Ghosh S, Wu A (2021) Iterative coordination and innovation: Prioritizing value over novelty. *Organ. Sci.*, ePub ahead of print October 11, https://doi.org/10.1287/orsc.2021.1499.

Gill D, Prowse V (2012) A structural analysis of disappointment aversion in a real effort competition. *Amer. Econom. Rev.* 102(1):469–503.

Gill D, Prowse V (2019) Measuring costly effort using the slider task. *J. Behav. Experiment. Finance* 21:1–9.

Gilovich T, Kerr M, Medvec VH (1993) Effect of temporal perspective on subjective confidence. *J. Personality Soc. Psych.* 64(4):552–560.

Gino F, Wiltermuth SS (2014) Evil genius? How dishonesty can lead to greater creativity. *Psych. Sci.* 25(4):973–981.

Girotra K, Netessine S (2014) Four paths to business model innovation. *Harvard Bus. Rev.* 92(7/8):96–103.

Goerg SJ, Kube S (2012) Goals (th)at work: Goals, monetary incentives, and workers' performance. Discussion Paper Series of the Max Planck Institute for Research on Collective Goods, Bonn, Germany.

Goh J, Hall NG (2013) Total cost control in project management via satisficing. *Management Sci.* 59(6):1354–1372.

Greiner B, Ockenfels A, Werner P (2011) Wage transparency and performance: A real-effort experiment. *Econom. Lett.* 111(3):236–238.

Grushka-Cockayne Y, Erat S, Wooten J (2018) New product development and project management decisions. Donohue K, Katok E, Leider S, eds. *The Handbook of Behavioral Operations* (John Wiley & Sons, Hoboken, NJ), 367–392.

Gutierrez GJ, Kouvelis P (1991) Parkinson's law and its implications for project management. *Management Sci.* 37(8):990–1001.

Gwosdz MM (2020) Does Scrum ruin great engineers or are you doing it wrong? *The Overflow* (June 29), https://stackoverflow.blog/2020/06/29/does-scrum-ruin-great-engineers-or-are-you-doing-it-wrong/.

Hackman JR, Oldham GR (1976) Motivation through the design of work: Test of a theory. *Organ. Behav. Human Performance* 16(2):250–279.

Hall NG (2016) Research and teaching opportunities in project management. Gupta A, Capponi A, eds. *Optimization Challenges in Complex, Networked and Risky Systems*, TutORials in Operations Research (INFORMS, Catonsville, MD), 329–388.

Hoda R, Noble J, Marshall S (2012) Self-organizing roles on agile software development teams. *IEEE Trans. Software Engrg.* 39(3):422–444.

Hodgson D, Briand L (2013) Controlling the uncontrollable: 'Agile' teams and illusions of autonomy in creative work. *Work, Employment, Soc.* 27(2):308–325.

Hofstadter DR (1979) *Gödel, Escher, Bach: An Eternal Golden Braid* (Basic Books, New York).

Hsiaw A (2013) Goal-setting and self-control. *J. Econom. Theory* 148(2):601–626.

Huchzermeier A, Loch CH (2001) Project management under risk: Using the real options approach to evaluate flexibility in R&D. *Management Sci.* 47(1):85–101.

Kachelmeier SJ, Reichert BE, Williamson MG (2008) Measuring and motivating quantity, creativity, or both. *J. Accounting Res.* 46(2):341–373.

Kachelmeier SJ, Williamson MG (2010) Attracting creativity: The initial and aggregate effects of contract selection on creativity-weighted productivity. *Accounting Rev.* 85(5):1669–1691.

Kagan E, Leider S, Lovejoy WS (2018) Ideation–execution transition in product development: An experimental analysis. *Management Sci.* 64(5):2238–2262.

Kagan E, Lieberum T, Schiffels S (2022) One size does not fit all: Strengths and weaknesses of the agile approach. Preprint, submitted May 23, https://dx.doi.org/10.2139/ssrn.4105914.

Kahneman D, Knetsch JL, Thaler RH (1991) Anomalies: The endowment effect, loss aversion, and status quo bias. *J. Econom. Perspect.* 5(1):193–206.

Kahneman D, Tversky A (1979) Intuitive prediction: Biases and corrective procedures. *TIMS Stud. Management Sci.* 12:313–327.

Katok E (2018) Designing and conducting laboratory experiments. Donohue K, Katok E, Leider S, eds. *The Handbook of Behavioral Operations* (John Wiley & Sons, Hoboken, NJ), 1–33.

Kavadias S, Sommer SC (2009) The effects of problem structure and team diversity on brainstorming effectiveness. *Management Sci.* 55(12):1899–1913.

Kerzner H (2013) *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, 11th ed. (John Wiley & Sons, Hoboken, NJ).

Kettunen J, Lejeune M (2020) Technical note—Waterfall and agile product development approaches: Disjunctive stochastic programming formulations. *Oper. Res.* 68(5):1356–1363.

Kornish LJ, Hutchison-Krupat J (2017) Research on idea generation and selection: Implications for management of technology. *Production Oper. Management* 26(4):633–651.

Krishnan V, Ulrich KT (2001) Product development decisions: A review of the literature. *Management Sci.* 47(1):1–21.

Ladas C (2009) *Scrumban: Essays on Kanban Systems for Lean Software Development* (Modus Cooperandi Press, Seattle).

LaPorte RE, Nath R (1976) Role of performance goals in prose learning. *J. Ed. Psych.* 68(3):260–264.

Laufer A, Hoffman EJ, Russell JS, Cameron WS (2015) What successful project managers do. *MIT Sloan Management Rev.* 43–51.

Levinthal DA (1997) Adaptation on rugged landscapes. *Management Sci.* 43(7):934–950.

Levinthal DA, March JG (1981) A model of adaptive organizational search. *J. Econom. Behav. Organ.* 2(4):307–333.

Lieberum T, Schiffels S, Kolisch R (2022) Should we all work in sprints? How agile project management improves performance. *Manufacturing Service Oper. Management* 24(4):2293–2309.

Loch CH, Wu Y (2007) Behavioral operations management. *Foundations Trends® Tech., Inform. Oper. Management* 1(3):121–232.

Locke EA, Latham GP (1990) *A Theory of Goal Setting & Task Performance* (Prentice-Hall, Englewood Cliffs, NJ).

Locke EA, Latham GP (2002) Building a practically useful theory of goal setting and task motivation: A 35-year odyssey. *Amer. Psych.* 57(9):705–717.

Long X, Nasiry J, Wu Y (2020) A behavioral study on abandonment decisions in multistage projects. *Management Sci.* 66(5):1999–2016.

MacCormack A, Verganti R, Iansiti M (2001) Developing products on "internet time": The anatomy of a flexible development process. *Management Sci.* 47(1):133–150.

Markham SE, Markham IS (1995) Self-management and self-leadership reexamined: A levels-of-analysis perspective. *Leadership Quart.* 6(3):343–359.

Marks MA, Mathieu JE, Zaccaro SJ (2001) A temporally based framework and taxonomy of team processes. *Acad. Management Rev.* 26(3):356–376.

Maruping LM, Venkatesh V, Agarwal R (2009) A control theory perspective on agile methodology use and changing user requirements. *Inform. Systems Res.* 20(3):377–399.

Matsui T, Okada A, Inoshita O (1983) Mechanism of feedback affecting task performance. *Organ. Behav. Human Perform.* 31(1):114–122.

Mendelsohn GA, Griswold BB (1964) Differential use of incidental stimuli in problem solving as a function of creativity. *J. Abnormal Soc. Psych.* 68(4):431.

Mihm J, Loch C, Huchzermeier A (2003) Problem-solving oscillations in complex engineering projects. *Management Sci.* 49(6):733–750.

O'Donoghue T, Rabin M (1999) Doing it now or later. *Amer. Econom. Rev.* 89(1):103–124.

Palley AB, Kremer M (2014) Sequential search and learning from rank feedback: Theory and experimental evidence. *Management Sci.* 60(10):2525–2542.

Panditi S (2018) Survey data shows that many companies are still not truly agile. *Harvard Bus. Rev.*, Sponsor Content (March 22), https://hbr.org/sponsored/2018/03/survey-data-shows-that-many-companies-are-still-not-truly-agile.

Parkinson CN (1957) *Parkinson's Law and Other Studies in Administration* (Houghton Mifflin, Boston).

Pasmore WA (1988) *Designing Effective Organizations: The Sociotechnical Systems Perspective* (John Wiley & Sons, New York).

Petersen K, Wohlin C (2009) A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *J. Systems Software* 82(9):1479–1490.

Pezzo SP, Pezzo MV, Stone ER (2006) The social implications of planning: How public predictions bias future plans. *J. Experiment. Soc. Psych.* 42(2):221–227.

Poppendieck M, Poppendieck T (2010) *Lean Software Development: An Agile Toolkit* (Addison-Wesley, Boston).

Powell WB, Ryzhov IO (2012) *Optimal Learning* (John Wiley & Sons, Hoboken, NJ).

Project Management Institute (2017a) *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*, 6th ed. (Newtown Square, PA).

Project Management Institute (2017b) Pulse of the profession: Success rates rise—Transforming the high cost of low performance. Report, Newtown Square, PA.

Project Management Institute (2018) Pulse of the profession: Success in disruptive times—Expanding the value delivery landscape to address the high cost of low performance. Report, Newtown Square, PA.

Raveendran M, Puranam P, Warglien M (2022) Division of labor through self-selection. *Organ. Sci.* 33(2):810–830.

Rigby DK, Sutherland J, Takeuchi H (2016) Embracing agile: How to master the process that's transforming management. *Harvard Bus. Rev.* 94(5):40–50.

Rosokha Y, Younge K (2020) Motivating innovation: The effect of loss aversion on the willingness to persist. *Rev. Econom. Statist.* 102(3):569–582.

Roy D, Mishra A, Sinha KK (2022) Taxing the taxpayers: An empirical investigation of the drivers of baseline changes in U.S. federal government technology programs. *Manufacturing Service Oper. Management* 24(1):370–391.

Royce WW (1987) Managing the development of large software systems: Concepts and techniques. Riddle WE, ed. *Proc. 9th Internat. Conf. Software Engrg.* (IEEE Computer Society Press, Washington, DC), 328–338.

Samuelson W, Zeckhauser R (1988) Status quo bias in decision making. *J. Risk Uncertainty* 1(1):7–59.

Sawyer RK (2011) *Explaining Creativity: The Science of Human Innovation*, 2nd ed. (Oxford University Press, New York).

Schultz KL, Juran DC, Boudreau JW, McClain JO, Thomas LJ (1998) Modeling and worker motivation in JIT production systems. *Management Sci.* 44(12-part-1):1595–1607.

Schwaber K, Sutherland J (2017) The Scrum Guide™: The definitive guide to Scrum: The rules of the game. Report. https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf.

Scrum Alliance (2015) The 2015 state of Scrum report. Report, Westminster, CO.

Seale DA, Rapoport A (1997) Sequential decision making with relative ranks: An experimental investigation of the "secretary problem". *Organ. Behav. Human Decision Processes* 69(3):221–236.

Shunko M, Niederhoff J, Rosokha Y (2018) Humans are not machines: The behavioral impact of queueing design on service time. *Management Sci.* 64(1):453–473.

Sommer SC, Bendoly E, Kavadias S (2020) How do you search for the best alternative? Experimental evidence on search strategies to solve complex problems. *Management Sci.* 66(3):1395–1420.

Sommer SC, Loch CH (2004) Selectionism and learning in projects with complexity and unforeseeable uncertainty. *Management Sci.* 50(10):1334–1347.

Srinivasan V, Lovejoy WS, Beach D (1997) Integrated product design for marketability and manufacturing. *J. Marketing Res.* 34(1):154–163.

Stapleton J (1998) *DSDM: Dynamic Systems Development Method* (Addison-Wesley, Harlow, England).

Thummadi V, Lyytinen K, Berente N (2012) Iterations in software development processes: A comparison of agile and waterfall software development projects. AIS Special Interest Group on IT Project Management, ed. *Proc. 7th Internat. Res. Workshop IT Project Management* (Association for Information Systems, Atlanta), 5–15.

Tuckman BW (1965) Developmental sequence in small groups. *Psych. Bull.* 63(6):384.

Wageman R (2001) How leaders foster self-managing team effectiveness: Design choices versus hands-on coaching. *Organ. Sci.* 12(5):559–577.

Weingarten E, Bhatia S, Mellers B (2019) Multiple goals as reference points: One failure makes everything else feel worse. *Management Sci.* 65(7):3337–3352.

Wood RE, Mento AJ, Locke EA (1987) Task complexity as a moderator of goal effects: A meta-analysis. *J. Appl. Psych.* 72(3):416–425.

Wu Y, Ramachandran K, Krishnan V (2014) Managing cost salience and procrastination in projects: Compensation and team composition. *Production Oper. Management* 23(8):1299–1311.

Yoo OS, Huang T, Arifoğlu K (2021) A theoretical analysis of the lean start-up method. *Marketing Sci.* 40(3):395–412.

Zangwill WI, Kantor PB (1998) Toward a theory of continuous improvement and the learning curve. *Management Sci.* 44(7):910–920.

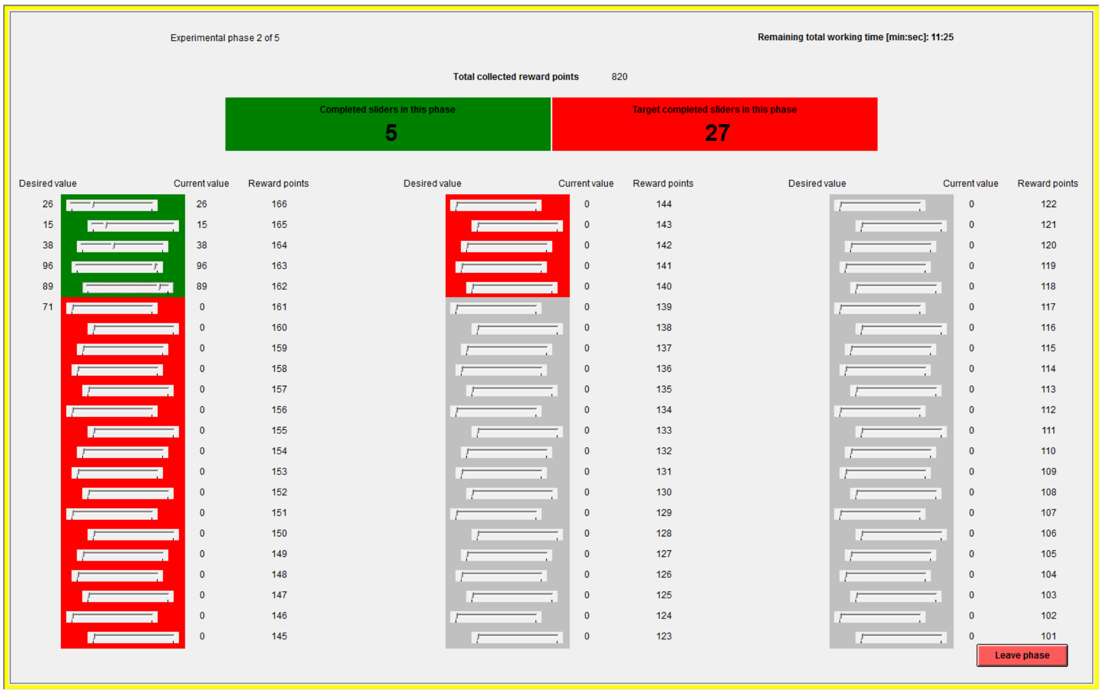
Zeigarnik B (1938) On finished and unfinished tasks. Ellis WD, ed. *A Source Book of Gestalt Psychology* (Kegan Paul, Trench, Trubner & Company, London), 300–314.

A. Appendix Experimental Study 1: Screens, Experiment Instructions, Comprehension Test, and Additional Analyses

The following screens, experiment instructions, and comprehension test are translated from German. Differences between the treatments are highlighted in square brackets.

A.1. Main Screens of the Experiment

Figure A.1. Screen for Each Experimental Phase



Experimental phase 2 of 5

Remaining total working time [min:sec]: 11:25

Total collected reward points 820

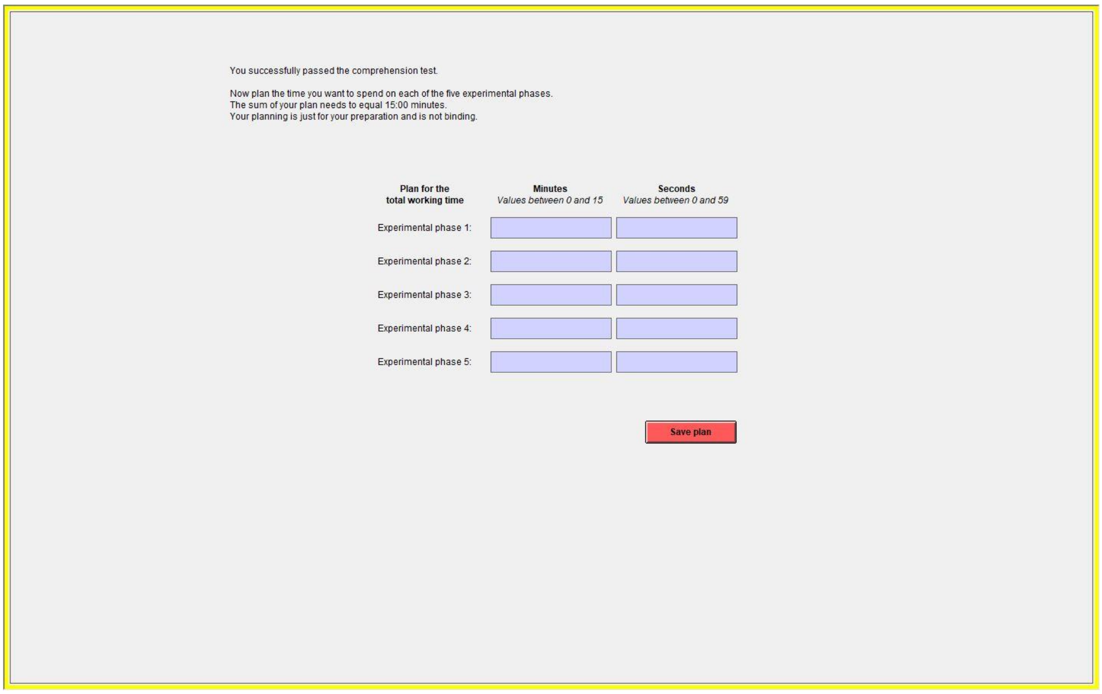
Completed sliders in this phase **5**

Target completed sliders in this phase **27**

Desired value	Current value	Reward points	Desired value	Current value	Reward points	Desired value	Current value	Reward points
26	26	166		0	144		0	122
15	15	165		0	143		0	121
38	38	164		0	142		0	120
96	96	163		0	141		0	119
89	89	162		0	140		0	118
71	0	161		0	139		0	117
	0	160		0	138		0	116
	0	159		0	137		0	115
	0	158		0	136		0	114
	0	157		0	135		0	113
	0	156		0	134		0	112
	0	155		0	133		0	111
	0	154		0	132		0	110
	0	153		0	131		0	109
	0	152		0	130		0	108
	0	151		0	129		0	107
	0	150		0	128		0	106
	0	149		0	127		0	105
	0	148		0	126		0	104
	0	147		0	125		0	103
	0	146		0	124		0	102
	0	145		0	123		0	101

Note. Screens differ slightly due to differences in the treatments (i.e. the displayed time and "Leave phase" button differ according to the progression regime and the displayed goal differs according to the goal-setting regime); the example shown is for FG with a goal of 27 completed sliders; five sliders are currently completed.

Figure A.2. Planning Screen for Control Treatments FNP and FNPP

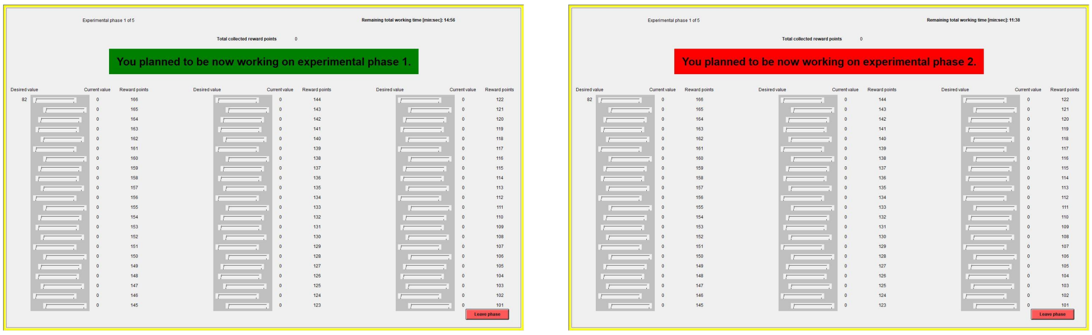


You successfully passed the comprehension test.

Now plan the time you want to spend on each of the five experimental phases.
The sum of your plan needs to equal 15:00 minutes.
Your planning is just for your preparation and is not binding.

Plan for the total working time	Minutes *Values between 0 and 15*	Seconds *Values between 0 and 59*
Experimental phase 1:		
Experimental phase 2:		
Experimental phase 3:		
Experimental phase 4:		
Experimental phase 5:		

Figure A.3. Progression Prompt for Control Treatment FNPP



(a) Progression on plan

Experimental phase 1 of 5

Remaining total working time [min:sec]: 14:56

Total collected reward points 0

You planned to be now working on experimental phase 1.

Desired value	Current value	Reward points	Desired value	Current value	Reward points	Desired value	Current value	Reward points
82	0	166		0	144		0	122
	0	165		0	143		0	121
	0	164		0	142		0	120
	0	163		0	141		0	119
	0	162		0	140		0	118
	0	161		0	139		0	117
	0	160		0	138		0	116
	0	159		0	137		0	115
	0	158		0	136		0	114
	0	157		0	135		0	113
	0	156		0	134		0	112
	0	155		0	133		0	111
	0	154		0	132		0	110
	0	153		0	131		0	109
	0	152		0	130		0	108
	0	151		0	129		0	107
	0	150		0	128		0	106
	0	149		0	127		0	105
	0	148		0	126		0	104
	0	147		0	125		0	103
	0	146		0	124		0	102
	0	145		0	123		0	101

(b) Progression out of plan

Experimental phase 1 of 5

Remaining total working time [min:sec]: 11:38

Total collected reward points 0

You planned to be now working on experimental phase 2.

Desired value	Current value	Reward points	Desired value	Current value	Reward points	Desired value	Current value	Reward points
82	0	166		0	144		0	122
	0	165		0	143		0	121
	0	164		0	142		0	120
	0	163		0	141		0	119
	0	162		0	140		0	118
	0	161		0	139		0	117
	0	160		0	138		0	116
	0	159		0	137		0	115
	0	158		0	136		0	114
	0	157		0	135		0	113
	0	156		0	134		0	112
	0	155		0	133		0	111
	0	154		0	132		0	110
	0	153		0	131		0	109
	0	152		0	130		0	108
	0	151		0	129		0	107
	0	150		0	128		0	106
	0	149		0	127		0	105
	0	148		0	126		0	104
	0	147		0	125		0	103
	0	146		0	124		0	102
	0	145		0	123		0	101

Note. The example shown is based on a plan to work on experimental phase 1 for three minutes and then progress to experimental phase 2.

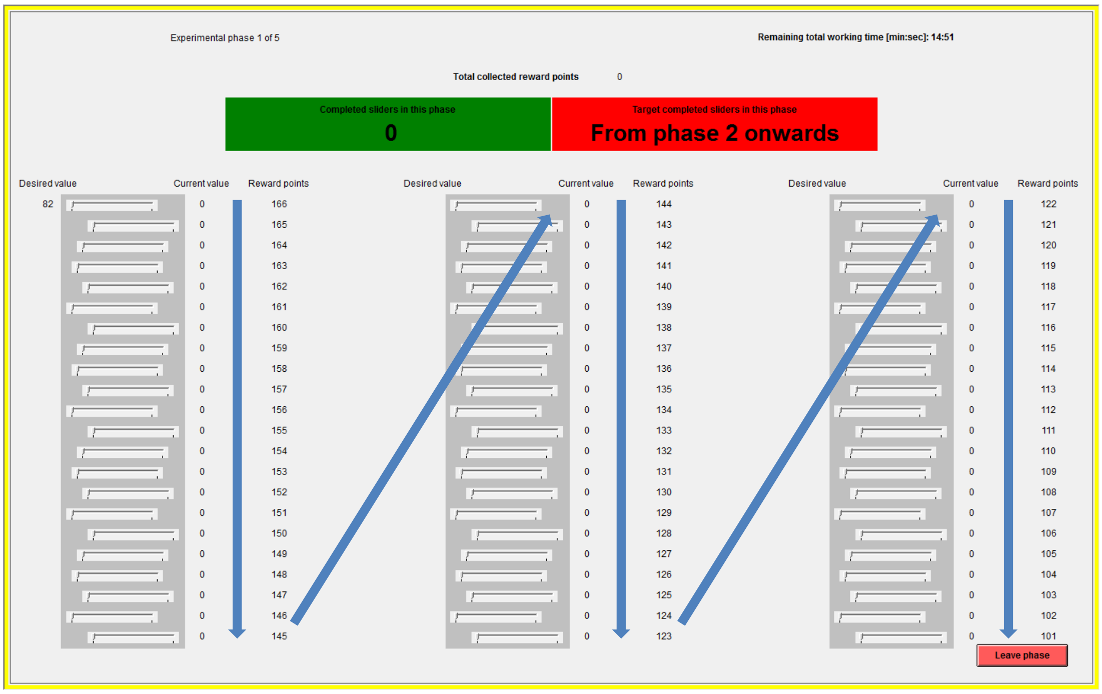
A.2. Instructions

For this experiment, you have a total of 15 minutes to sequentially process 5 tasks of the same workload and difficulty. Each task consists of 66 sliders. Each slider has a current value and a desired value. You can earn money by using the computer mouse to move the slider from the current to the desired value. The keyboard and mouse wheel are deactivated for positioning the sliders. All slider tasks are completed beginning from the upper left-hand corner of the screen moving downwards and then through each successive column to the right. For each correctly positioned slider, you will be credited the number of reward points specified beside it and you will obtain the desired value of the next slider. The number of reward points per slider decreases within each task. The first slider in the upper left-hand corner of the screen always yields 166 reward points, whereas the next slider BELOW always yields 165, and so on. This distribution of the reward points is the same across all five tasks. Below you can see the screen on which you will be working [Figure A.4, the screen differs slightly according to the differences in the treatments; the example shown is for FG]. *The arrows indicate the workflow.*
[Treatment TN:] *The current task, your total score, and the working time remaining for the current task are displayed in the upper bar of the screen. You do not have to solve all sliders and you have 3 minutes to work on each of the five tasks. Each task ends automatically. Note that you cannot return to previous tasks.*
It takes 60 seconds to load each subsequent task. This time does NOT count towards the total working time of 15 minutes.

[Treatment FN and control treatments FNP and FNPP:] *The current task, your total score, and the total working time remaining are displayed in the upper bar of the screen. You do not have to solve all sliders and you can work on each of the five tasks for as long as you wish, but you may not spend more than 15 minutes in total. Click on the "Leave phase" button in the lower right-hand corner to proceed to the next task. Note that you cannot return to previous tasks.*
It takes 60 seconds to load each subsequent task. This time does NOT count towards the total working time of 15 minutes.

[Treatment TG:] *The current task, your total score, your target, your completed*

Figure A.4. Slider Screen for Each Experimental Phase



Experimental phase 1 of 5

Remaining total working time [min:sec]: 14:51

Total collected reward points 0

Completed sliders in this phase
0

Target completed sliders in this phase
From phase 2 onwards

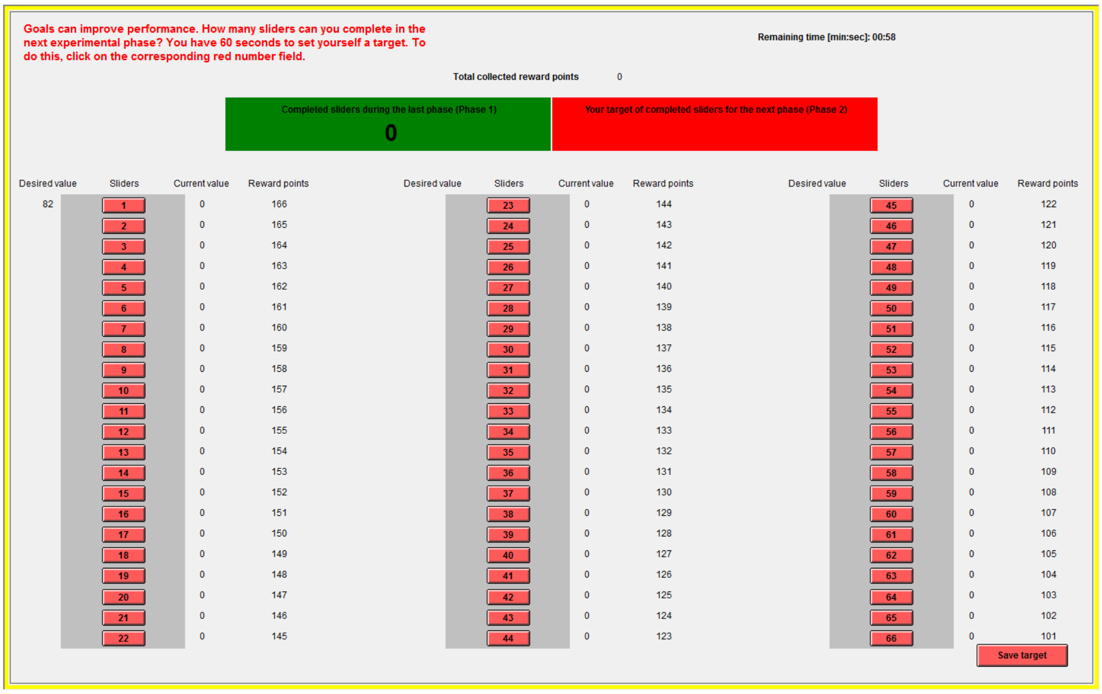
Desired value	Current value	Reward points	Desired value	Current value	Reward points	Desired value	Current value	Reward points
82	0	166		0	144		0	122
	0	165		0	143		0	121
	0	164		0	142		0	120
	0	163		0	141		0	119
	0	162		0	140		0	118
	0	161		0	139		0	117
	0	160		0	138		0	116
	0	159		0	137		0	115
	0	158		0	136		0	114
	0	157		0	135		0	113
	0	156		0	134		0	112
	0	155		0	133		0	111
	0	154		0	132		0	110
	0	153		0	131		0	109
	0	152		0	130		0	108
	0	151		0	129		0	107
	0	150		0	128		0	106
	0	149		0	127		0	105
	0	148		0	126		0	104
	0	147		0	125		0	103
	0	146		0	124		0	102
	0	145		0	123		0	101

Note. Screen differs slightly according to differences in the treatments (i.e. the displayed time and "Leave phase" button differ by progression regime and the displayed goal differs by goal-setting regime); the example shown is for FG.

sliders, and the working time remaining for the current task are displayed in the upper bar of the screen. You do not have to solve all sliders and you have 3 minutes to work on each of the five tasks. Each task ends automatically. Note that you cannot return to previous tasks.
After each task, you have 60 seconds to set yourself a target of how many sliders you wish to complete in the next task, based on your previous performance. This time does NOT count towards the total working time of 15 minutes. Below you can see the target screen [Figure A.5]. *Click on one of the red number fields to set a target and then click on the "Save target" button at the bottom right.*

[Treatment FG:] *The current task, your total score, your target and completed sliders for the current task, and the total working time remaining are displayed in the upper bar of the screen. You do not have to solve all sliders and you can work on each of the five tasks for as long as you wish, but you may not spend more than*

Figure A.5. Goal-Setting Screen for Treatments TG and FG



Goals can improve performance. How many sliders can you complete in the next experimental phase? You have 60 seconds to set yourself a target. To do this, click on the corresponding red number field.

Remaining time [min:sec]: 00:58

Total collected reward points 0

Completed sliders during the last phase (Phase 1)
0

Your target of completed sliders for the next phase (Phase 2)

Desired value	Sliders	Current value	Reward points	Desired value	Sliders	Current value	Reward points	Desired value	Sliders	Current value	Reward points
82	1	0	166		23	0	144		45	0	122
	2	0	165		24	0	143		46	0	121
	3	0	164		25	0	142		47	0	120
	4	0	163		26	0	141		48	0	119
	5	0	162		27	0	140		49	0	118
	6	0	161		28	0	139		50	0	117
	7	0	160		29	0	138		51	0	116
	8	0	159		30	0	137		52	0	115
	9	0	158		31	0	136		53	0	114
	10	0	157		32	0	135		54	0	113
	11	0	156		33	0	134		55	0	112
	12	0	155		34	0	133		56	0	111
	13	0	154		35	0	132		57	0	110
	14	0	153		36	0	131		58	0	109
	15	0	152		37	0	130		59	0	108
	16	0	151		38	0	129		60	0	107
	17	0	150		39	0	128		61	0	106
	18	0	149		40	0	127		62	0	105
	19	0	148		41	0	126		63	0	104
	20	0	147		42	0	125		64	0	103
	21	0	146		43	0	124		65	0	102
	22	0	145		44	0	123		66	0	101

15 minutes in total. Click on the "Leave phase" button in the lower right-hand corner to proceed to the next task. Note that you cannot return to previous tasks. After each task, you have 60 seconds to set yourself a target of how many sliders you wish to complete in the next task, based on your previous performance. This time does NOT count towards the total working time of 15 minutes. Below you can see the target screen [this is the same screen as for TG, Figure A.5]. *Click on one of the red number fields to set a target and then click on the "Save target" button at the bottom right.*

[All treatments:] *The experiment will end automatically after 15 minutes of total working time. For taking part in the experiment, you will receive a show-up fee of EUR 4.00. In addition, your reward points accumulated from all five tasks will be paid out at the end of the experiment, at an exchange rate of EUR 1 for every 5,000 reward points.*

You must pass the comprehension test given before the start of the experiment, otherwise you will only receive the show-up fee.

[Control treatments FNP and FNPP:] *You also need to plan in advance how you*

wish to distribute your 15 minutes of total working time among the five tasks. This planning is only relevant for your preparation and is not binding on the task.
[Control treatment FNPP:] *You will be alerted during the experiment if you deviate from your plan.*
[All treatments:] *To begin the experiment, enter the access code:* [treatment specific code]

A.3. Comprehension Test

Our experimental design requires all participants to pass the comprehension test below in a maximum of two attempts, in order to be included in the results of the experiment. The correct answer is highlighted in bold.

1. *Do you need to complete all sliders in all tasks? (Yes/**No**)*
2. *Do all sliders yield the same amount of reward points? (Yes/**No**)*
3. *Can you return to previous tasks? (Yes/**No**)*
4. *Can you decide when to start the next task? (**Yes/No** depending on the treatment)*
5. *In which direction does the number of reward points per slider decrease? (**First down, then to the right**/First to the right, then down)*
6. [Only for treatments TN and FN as well as control treatments FNP and FNPP:] *It takes 60 seconds to load the next task. This time does not count towards your total working time. (**True**/False)*
7. [Only for treatments TG and FG:] *After each task, you have 60 seconds in which to set yourself a target of how many sliders you wish to complete in the next task. This time does not count towards your total working time. (**True**/False)*
8. [Only for treatments FN and FG and control treatments FNP and FNPP respectively:] *Can you proceed to the next task even if you have not yet completed the 66th slider of a task? (**Yes**/No)*
9. [Only for treatments TN and TG] *Will the next task start even if you have not yet completed the 66th slider of a task? (**Yes**/No)*

A.4. Additional Analyses

Table A.1. Mean of Time Spent, Sliders Completed, and Reward Points Collected per Experimental Phase 1–5 by Treatment

	Time spent (in seconds)					Sliders completed					Reward points collected				
Treatment	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Time-boxed with no goals (TN)	180	180	180	180	180	22.5	25.8	26.3	28.3	27.9	3,482	3,940	4,015	4,289	4,236
	(0)	(0)	(0)	(0)	(0)	(5.2)	(5.7)	(6.1)	(5.7)	(5.6)	(743)	(807)	(858)	(791)	(783)
Flexible with no goals (FN)	317	187	143	130	124	38.5	26.4	21.1	20.4	20.0	5,529	3,999	3,238	3,124	3,047
	(147)	(57)	(67)	(64)	(77)	(16.6)	(9.5)	(10.6)	(11.1)	(12.8)	(2,034)	(1,293)	(1,554)	(1,652)	(1,898)
Time-boxed with goals (TG)	180	180	180	180	180	23.6	26.3	26.4	28.5	28.3	3,643	4,015	4,038	4,322	4,297
	(0)	(0)	(0)	(0)	(0)	(5.1)	(5.9)	(5.5)	(5.8)	(5.8)	(728)	(825)	(765)	(795)	(791)
Flexible with goals (FG)	305	211	151	120	113	35.1	28.7	21.3	17.0	16.6	5,093	4,300	3,268	2,621	2,541
	(141)	(67)	(67)	(77)	(86)	(16.2)	(11.7)	(10.5)	(11.0)	(13.1)	(2,018)	(1,543)	(1,531)	(1,664)	(1,965)
Flexible with no goals but planning (FNP)	250	181	158	157	154	30.3	24.5	22.1	22.1	22.7	4,513	3,749	3,405	3,414	3,488
	(87)	(41)	(40)	(40)	(52)	(12.4)	(8.3)	(6.9)	(6.2)	(8.0)	(1,554)	(1,142)	(1,013)	(918)	(1,189)
Flexible with no goals but planning and progression prompt (FNPP)	226	190	175	152	158	26.3	25.8	24.9	23.4	23.9	3,935	3,925	3,783	3,561	3,629
	(101)	(51)	(63)	(59)	(72)	(13.6)	(8.4)	(9.7)	(10.6)	(10.9)	(1,755)	(1,103)	(1,360)	(1,555)	(1,586)

Notes. Standard deviations in parentheses. Goals are set in treatments TG and FG from the second experimental phase onward.

Table A.2. Multilevel Mixed-Effects Linear Regression Models on Seconds per Slider by Experimental Phase 1–5

	Model 1	Model 2	Model 3	Model 4
Intercept				
TN (Baseline)	8.152***	7.198***	8.467***	7.513***
	(36.22)	(29.34)	(35.84)	(29.30)
FN	−0.150	−0.179	−0.187	−0.216
	(−0.56)	(−0.71)	(−0.67)	(−0.81)
TG	−0.356	−0.456*	−0.458	−0.558*
	(−1.18)	(−1.68)	(−1.45)	(−1.92)
FG	0.390	0.225	0.503	0.338
	(1.28)	(0.80)	(1.59)	(1.14)
Experimental phase slope effect				
TN (Baseline)				
Average per experimental phase	−0.406***	−0.406***		
	(−11.75)	(−11.75)		
Phase 2			−1.060***	−1.060***
			(−6.61)	(−6.61)
Phase 3			−1.143***	−1.143***
			(−6.47)	(−6.47)
Phase 4			−1.739***	−1.739***
			(−10.67)	(−10.67)
Phase 5			−1.688***	−1.688***
			(−10.96)	(−10.96)
FN				
Average per experimental phase	0.094	0.094		
	(1.33)	(1.33)		
Phase 2			0.140	0.140
			(0.68)	(0.68)
Phase 3			0.172	0.172
			(0.70)	(0.70)
Phase 4			0.543**	0.543**
			(2.14)	(2.14)
Phase 5			0.268	0.268
			(0.91)	(0.91)
TG				
Average per experimental phase	0.066	0.066		
	(1.34)	(1.34)		
Phase 2			0.286	0.286
			(1.34)	(1.34)
Phase 3			0.250	0.250
			(1.06)	(1.06)
Phase 4			0.323	0.323
			(1.47)	(1.47)
Phase 5			0.313	0.313
			(1.45)	(1.45)
FG				
Average per experimental phase	0.030	0.030		
	(0.24)	(0.24)		
Phase 2			−0.137	−0.137
			(−0.61)	(−0.61)
Phase 3			−0.322	−0.322
			(−1.15)	(−1.15)
Phase 4			0.228	0.228
			(0.58)	(0.58)
Phase 5			−0.030	−0.030
			(−0.06)	(−0.06)
Controls				
Female		0.490***		0.490***
		(2.96)		(2.96)
Lack of skill		0.054***		0.054***
		(5.19)		(5.19)
Private goal		−0.070		−0.070
		(−0.27)		(−0.27)
Number of observations	1,710	1,710	1,710	1,710

Notes. Experimental phase 1 set as intercept, as overall learning is evaluated. Random intercept and slope effects at the subject level included with covariance unstructured. Standard errors clustered at the subject level. z statistics in parentheses. Interpolated for missing values. The number of observations equals the product of the participants and the experimental phases. $^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Table A.3. Multilevel Mixed-Effects Linear Regression Models on Time Spent and Sliders Completed per Experimental Phase 1–5 for FN and TN—Categorial Analysis

	Time (in seconds)		Sliders			
	FN	TN	FN		TN	
	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Intercept	316.60***	180.04***	38.47***	40.43***	22.51***	26.68***
	(20.42)	(> 99.99)	(22.02)	(20.51)	(39.92)	(22.09)
Experimental phase slope effect (simple effect)						
Phase 2	−129.85***	0.01	−12.09***	−12.09***	3.24***	3.24***
	(−8.68)	(1.23)	(−6.52)	(−6.52)	(7.56)	(7.56)
Phase 3	−173.64***	0.00	−17.34***	−17.34***	3.79***	3.79***
	(−8.18)	(0.61)	(−6.55)	(−6.55)	(7.58)	(7.58)
Phase 4	−186.52***	0.01	−18.09***	−18.09***	5.74***	5.74***
	(−8.66)	(1.49)	(−6.73)	(−6.73)	(13.18)	(13.18)
Phase 5	−192.67***	0.01*	−18.48***	−18.48***	5.36***	5.36***
	(−8.60)	(1.76)	(−6.51)	(−6.51)	(11.87)	(11.87)
Controls						
Female				−1.11		−1.70*
				(−1.44)		(−1.69)
Lack of skill				−0.13***		−0.22***
				(−2.86)		(−2.77)
Private goal				1.38*		−0.82
				(1.73)		(−0.73)
Number of observations	450	420	450	450	420	420

Notes. Experimental phase 1 set as intercept. Random intercept effects at the subject level included; Models 1, 3 and 4 additionally with random slope effects with covariance unstructured for better model fit. For all models, standard errors clustered at the subject level. z statistics in parentheses. Controls are not applied to regressions on time, as the total working time is fixed and thus not affected by simple effects of these controls. The number of observations equals the product of the participants and the experimental phases.

$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Table A.4. Multilevel Mixed-Effects Linear Regression Models on Time Spent and Sliders Completed per Experimental Phase 2–5 for FG with FN as Baseline—Categorial Analysis

	Time (in seconds)			Sliders		
	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Intercept	186.75***	196.56***	265.86***	26.38***	29.85***	38.02***
(baseline FN)	(31.14)	(27.15)	(33.25)	(26.43)	(23.93)	(28.56)
Experimental phase slope effect						
(simple effect)						
Phase 3	−43.79***	−43.79***	−43.79***	−5.26***	−5.26***	−5.26***
	(−4.20)	(−4.20)	(−4.20)	(−3.58)	(−3.58)	(−3.58)
Phase 4	−56.67***	−56.67***	−56.67***	−6.00***	−6.00***	−6.00***
	(−5.05)	(−5.05)	(−5.05)	(−3.63)	(−3.63)	(−3.63)
Phase 5	−62.83***	−62.83***	−62.83***	−6.39***	−6.39***	−6.39***
	(−5.00)	(−5.00)	(−5.00)	(−3.39)	(−3.39)	(−3.39)
Goal-setting intercept effect	24.22***	29.48***	21.34*	2.33	3.84**	2.92
(simple effect)	(2.62)	(3.10)	(1.79)	(1.47)	(2.37)	(1.63)
Goal-setting slope effect						
(interaction effect)						
Phase 3	−16.02	−16.02	−16.02	−2.13	−2.13	−2.13
	(−1.06)	(−1.06)	(−1.06)	(−0.98)	(−0.98)	(−0.98)
Phase 4	−34.12*	−34.12*	−34.12*	−5.74**	−5.74**	−5.74**
	(−1.94)	(−1.94)	(−1.94)	(−2.21)	(−2.21)	(−2.21)
Phase 5	−35.23*	−35.23*	−35.23*	−5.72**	−5.72**	−5.72**
	(−1.86)	(−1.86)	(−1.86)	(−1.97)	(−1.97)	(−1.97)
Controls						
Female		−7.37*			−2.51***	−1.19***
		(−1.70)			(−2.89)	(−3.02)
Lack of skill		−0.70***			−0.23***	−0.05**
		(−2.67)			(−5.08)	(−2.19)
Private goal		10.38*			2.75**	0.53
		(1.82)			(2.13)	(0.81)
Time spent in first experimental phase			−0.25***			−0.07***
			(< −99.99)			(−15.16)
Sliders completed in first experimental phase			−0.00			0.29***
			(−1.26)			(7.90)
Number of observations	716	716	716	716	716	716

Notes. Experimental phase 2 with first goal-setting set as intercept. Random intercept and slope effects at the subject level included with covariance unstructured. Standard errors clustered at the subject level. z statistics in parentheses. Controls for gender, initial lack of skill, and private goals not applied to Model 3, as the total working time is fixed with the additional covariate for the time spent in the first experimental phase in this model and thus is not affected by simple effects of the controls. The number of observations equals the product of the participants and the experimental phases.

$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Table A.5. Multilevel Mixed-Effects Linear Regression Models on Time Planned and Actually Spent per Experimental Phase 1–5 for FNP and FNPP with Planned Time as Baseline (in Seconds)—Categorial Analysis

	FNP				FNPP			
	All	Flat	Decrease	Increase	All	Flat	Decrease	Increase
Intercept	202.11***	180.00***	249.23***	120.00	201.38***	180.00	257.14***	147.86***
(baseline planned time)	(31.24)	(21.07)	(29.04)	(NA)	(22.03)	(NA)	(15.26)	(11.18)
Experimental phase slope effect (“plan”)								
(simple effect)								
Phase 2	−19.34***	0.00	−61.15***	60.00	−19.38***	0.00	−62.50***	14.29
	(−3.22)	(0.00)	(−6.75)	(NA)	(−2.70)	(0.00)	(−4.69)	(1.05)
Phase 3	−25.66***	0.00	−79.62***	60.00	−21.50**	0.00	−81.43***	40.00
	(−3.20)	(0.00)	(−5.80)	(NA)	(−2.00)	(NA)	(−3.93)	(1.87)
Phase 4	−29.61***	0.00	−91.15***	60.00	−27.13**	0.00	−98.21***	41.43
	(−3.27)	(0.00)	(−5.91)	(NA)	(−2.10)	(0.00)	(−3.77)	(1.78)
Phase 5	−35.92***	0.00	−114.20***	120.00	−38.88**	0.00	−143.57***	65.00**
	(−3.46)	(0.00)	(−10.98)	(NA)	(−2.43)	(0.00)	(−5.53)	(2.94)
Delta (plan to actual) intercept effect	47.57***	51.02***	10.73	443.64	24.28**	14.74	43.26	12.23*
(simple effect)	(2.99)	(4.22)	(0.56)	(NA)	(2.28)	(1.69)	(1.55)	(2.11)
Delta (plan to actual) slope effect								
(interaction effect)								
Phase 2	−49.23***	−50.95***	−27.77	−286.94	−16.12	−13.62	−18.40	−18.32
	(−3.55)	(−4.89)	(−1.40)	(NA)	(−1.58)	(−1.32)	(−0.71)	(−1.52)
Phase 3	−66.04***	−66.68***	−21.98	−623.41	−29.37*	−1.27	−77.48*	−9.42
	(−3.08)	(−4.87)	(−0.84)	(NA)	(−1.78)	(−0.08)	(−1.92)	(−1.61)
Phase 4	−63.01***	−68.58***	−9.61	−623.41	−46.71***	−39.19**	−73.60*	−13.32
	(−2.82)	(−3.85)	(−0.32)	(NA)	(−2.82)	(−2.29)	(−1.78)	(−1.73)
Phase 5	−59.26**	−68.58***	5.95	−683.41	−28.89	−19.32	−46.44	−19.76**
	(−2.38)	(−3.06)	(0.24)	(NA)	(−1.63)	(−0.86)	(−1.12)	(−3.23)
Number of observations	380	240	130	10	400	190	140	70

Notes. Experimental phase 1 set as intercept. Random intercept effects included at both the subject and the *actual* dummy levels with random slope effects and covariance unstructured. Standard errors clustered at the subject level. z statistics in parentheses. Exceptions for computational reasons and/or better model fit: FNP Flat without clustering of standard errors at the subject level. FNP Increase, FNPP Flat, FNPP Increase with simple linear regression and t statistics in parentheses. FNPP All without random intercept effect at the *actual* dummy levels and without unstructured covariance. Controls for gender, initial lack of skill, and private goals not applied, as the total working time is fixed and thus not affected by simple effects of these controls. The number of observations equals the product of the participants, the experimental phases, and plan and actual per participant.

$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

B. Appendix Experimental Study 2: Experimental Design, Protocol, Instructions, and Additional Analyses

B.1. Experimental Design Details and Parametrization

B.1.1. Design Task

The Design task administered in our experiments is based on the Scrabble game. Following the classic German version of Scrabble, 100 tiles (blank tiles excluded) were made available to the subjects for each (nouns and verbs) component. The tiles were not refilled for the second period. The tiles given to participants at the beginning of the task were as follows (number of tiles with each letter is given in parentheses):

E (15), N (9), S (7), I (6), R (6), T (6), U (6), A (5), D (4), H (4), G (3), L (3), O (3), M (4), B (2), W (1), Z (1), C (2), F (2), K (2), P (1), Ä (1), J (1), Ü (1), V (1), Ö (1), X (1), Q (1), Y (1)

B.1.2. Search Task

The Search task is based on the classic Lemonade Stand game (Ederer and Manso 2013), revised to include two separate components, each with a separate, independent solution landscape. The first component is the Product component, consisting of four attributes (lemonade color, lemon content, carbonation, shape of the bottle label). The second component is the Market component, consisting of four attributes (location, price, opening hours, advertising). For each component two of the attributes are discrete, whereas the other two are continuous. Figures B.1 and B.2 show the landscapes for all combinations of the discrete attributes for the Product component and the Market component, along with the two local maxima and the global maximum.

Subjects were presented with two components (Product and Market), with each component containing four attributes.

Product component:

1. Lemonade color = Green, Yellow, Orange

2. Lemon content = 10, 10.1, 10.2, ..., 19.9, 20

3. Carbonation = 10, 10.1, 10.2, ..., 19.9, 20

Table B.1. Optimal Selections and Maximum Profit by Component and Parameter Version

	Version 1			Version 2		
Product						
Lemonade color	Green	Yellow	Orange	Green	Yellow	Orange
Lemon content	18.5	11.6	17.6	11.5	12.4	18.4
Carbonation	16.9	18.5	12.2	13.1	17.8	11.5
Bottle label	Square	Triangle	Circle	Square	Triangle	Circle
Maximum Profit	200	380	500	380	500	200
Market						
Location	West	North	East	West	North	East
Price	17.1	17.9	10.9	12.9	19.1	12.1
Opening hours	18.5	11.8	17.3	11.5	12.7	18.2
Advertising	Placard	Display stand	Flyer	Placard	Display stand	Flyer
Maximum Profit	380	500	200	200	380	500

4. Bottle label = Square, Triangle, Circle

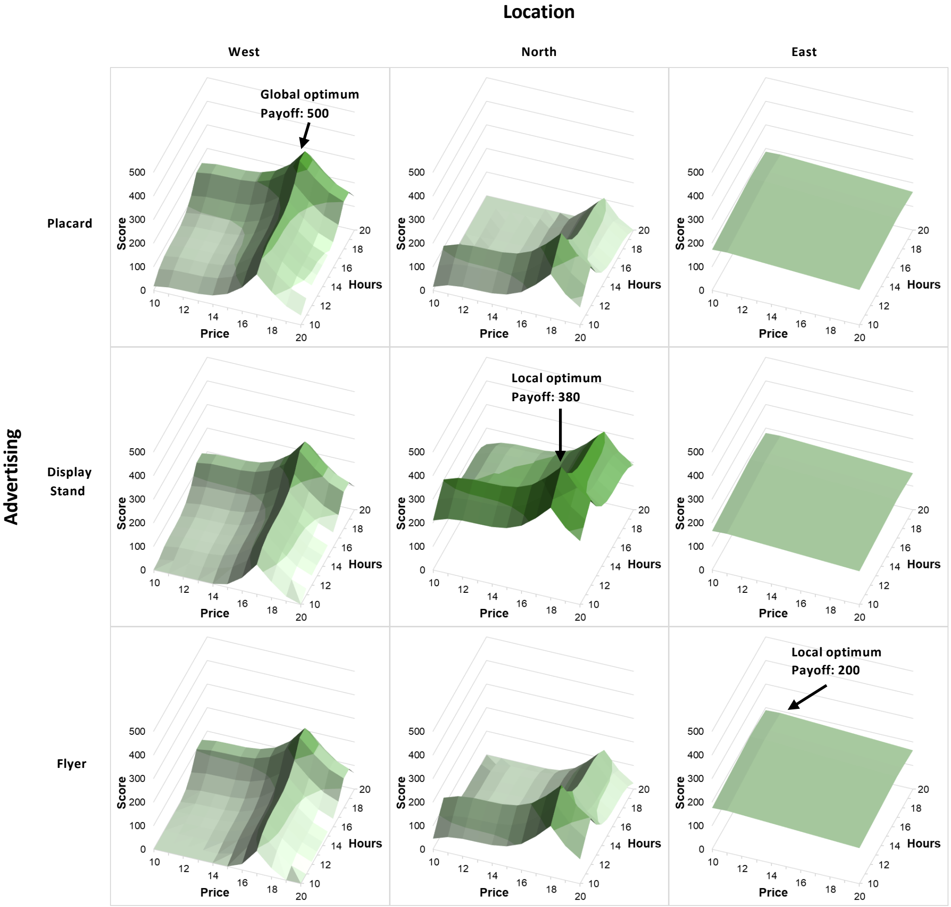
Market component:

1. Location = West, North, East
2. Price = 10, 10.1, 10.2, ..., 19.9, 20
3. Opening hours = 10, 10.1, 10.2, ..., 19.9, 20
4. Advertising = Placard, Display stand, Flyer

For each lemonade color (in the Product component) and location (in the Market component), there is a predefined, optimal selection resulting in a maximum profit. To avoid the possibility that our effects were driven by a single parameter version we used two parameter versions for each component. Table B.1 shows the optimal selections and maximum profits for each component and parameter version.

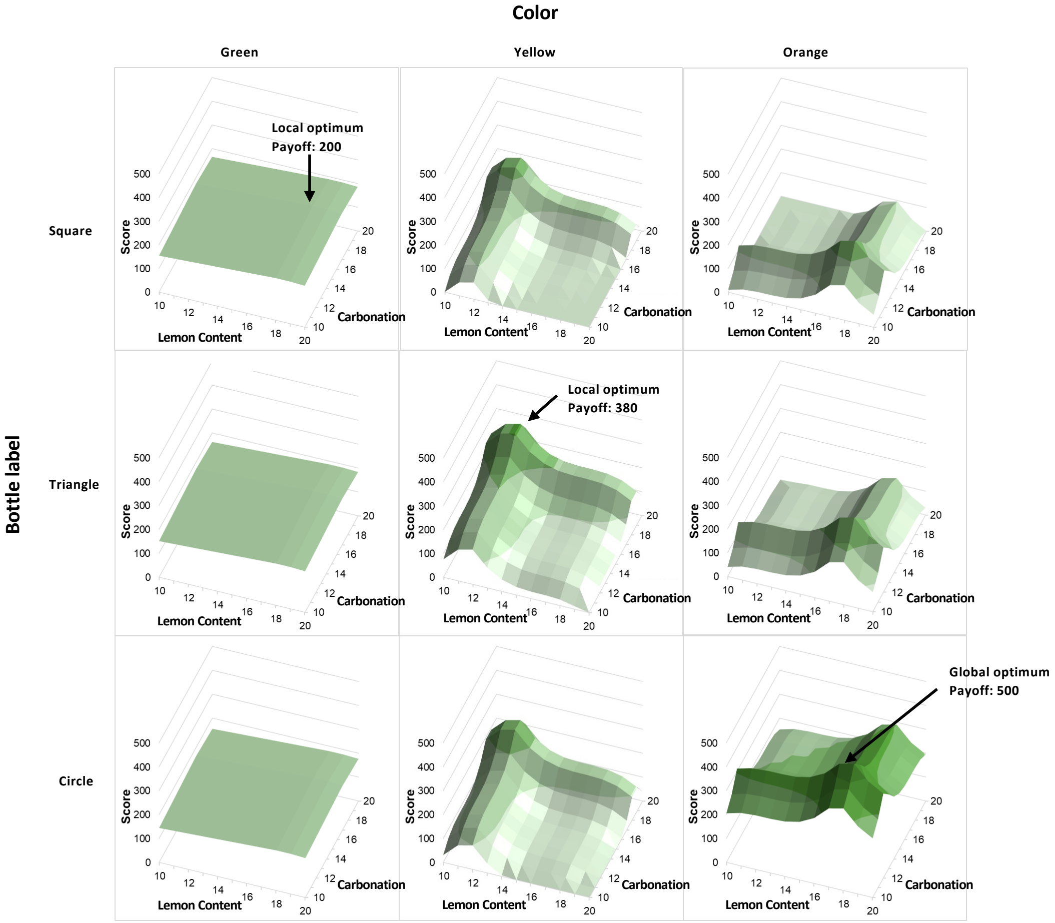
For the market component, Figure B.1 shows the three maxima, each of which corresponds to a combination of location and advertising. For the product component, Figure B.2 shows the three maxima, each of which corresponds to a

Figure B.1. Landscapes for the Market Component



Notes. The graphs show one of the two parametrizations used in the experiment. The second parametrization was similar, but had a different price/hour combination for the location of the local and global maxima.

Figure B.2. Landscapes for the Product Component



Notes. The graphs show one of the two parametrizations used in the experiment. The second parametrization was similar, but had a different lemon content/carbonation combination for the location of the local and global maxima.

Table B.2. Penalties by Component and Parameter Version

		Version 1			Version 2		
Product							
Lemonade color		Green	Yellow	Orange	Green	Yellow	Orange
Bottle label	Square	0	75	195	0	165	10
	Triangle	3	0	165	75	0	3
	Circle	10	45	0	45	195	0
Market							
Location		West	North	East	West	North	East
Advertising	Display stand	45	0	10	10	0	165
	Flyer	75	165	0	3	75	0
	Placard	0	195	3	0	45	195

combination of lemonade color and bottle label. As shown in Table B.1, we set these three maxima to 200, 380, and 500 points, respectively. Note that while the optimal locations of the remaining attributes are unchanged if we move vertically in Figures B.1 and B.2, the locations change if we move horizontally. This corresponds to the medium complexity scenarios used in the prior rugged landscape literature (see, for example, Sommer et al. 2020, and references there). The penalties for the discrete attributes (Lemonade color and Bottle label for the Product component as well as Location and Advertising for the Market component) are given in Table B.2. The penalties for the lowest local maximum (at 200) for the continuous attributes (Lemon content and Carbonation for the Product component as well as Price and Opening hours for the Market component) are linear. They were computed by multiplying each unit of absolute deviation by a constant, i.e. $absolute\ deviation \times 3$. The penalties for the two highest maxima (at 380 and 500 points) both follow an S-shaped curve. They were computed based on exponentiation of the absolute deviation, calibrated by three constants, i.e. $(\frac{absolute\ deviation}{5} - 1)^3 \times 150 - 150$. The penalties for deviations from the maxima were chosen to make finding a good combination of attributes difficult, but achievable (the level of difficulty was found to be appropriate after conducting pilots with 33 participants).

B.2. Experimental Protocol and Instructions

All experiments were conducted using o-Tree (Chen et al. 2016). Due to COVID-19 restrictions, all experiments were conducted online, using Zoom for monitoring the participants. Zoom meetings were set up with the lead experimenter as the host and other experimenters as co-hosts. Participants received Zoom links via email in the morning of the day of the experiment. Upon sign-up, participants were renamed to preserve anonymity. During the experiment participants were able to chat with the experimenter and ask questions. The instructions are summarized below (translated from German):

B.2.1. Introduction

Announcement

[The announcement was read loud.]

"Welcome to today's experiment. The experiment will take about 45 minutes. Participation in the experiment is only possible with the Google Chrome browser and a computer mouse. Participation with another browser as well as with cell phone or tablet is not possible due to technical reasons. If you do not meet this condition, you cannot participate in the experiment. In this case, please leave the Zoom meeting now.

Please leave your camera on for the entire duration of the experiment. This is only to ensure that everything runs smoothly. There will be no recording. By voluntarily participating in this experiment, you expressly consent to this use in accordance with the General Data Protection Regulation. If you do not want to agree to the camera use, you can leave the Zoom meeting now without further consequences. If you lose your internet connection during processing, dial into this Zoom meeting again. We will then explain the further procedure to you.

Do you have any questions? Then write a private message to the lead experimenter via the Zoom chat. There are several comprehension tests. Do not hesitate to write to me if something is unclear.

We will now send you a custom link through Zoom chat. Copy and paste it into your Chrome browser. You can start working on it right away. When you reach the end of the experiment, you can leave this Zoom meeting and close the *experiment.*

Thank you for participating in this scientific study!"

Opening Screens

Welcome to today's experiment! It is good to have you with us!

This is an individual experiment. To ensure scientific validity, the tasks vary between the participants of this experiment. Therefore, please do not attempt to interact with each other or third parties. The use of cell phones, tablets, software, and internet applications other than this experiment is strictly prohibited for the entire duration of the experiment. Violations will result in exclusion from further participation in experiments in the experimenTUM laboratory. Do not press the reload, back, or forward buttons on your browser, or the F5 key, as this will cancel the experiment. Please keep your camera turned on throughout the experiment. If you have any questions, please write us a private message to the experimenter in Zoom Chat.

As announced in the invitation of the experiment, a confident command of the German language is important for this experiment. Therefore, you must first pass a German test.

[Followed by the German test.]

B.2.2. Part 1 of the Experiment

[Note: Part 1 and Part 2 of the experiment were displayed in random order.]

Please read the following instructions carefully and answer the comprehension questions. You will have two attempts to pass the comprehension questions. If you do not successfully pass the comprehension questions, you will not participate in this part of the experiment and will not be compensated for it. If you have any questions about the instructions, please write a PRIVATE message to the experimenter using the Zoom chat function.

Background

In this part of the experiment, you will develop the most profitable business model for a lemonade stand by selecting a product and market strategy from numerous *options. The product component consists of four product characteristics:*

1. *Lemonade color*
2. *Lemon content*
3. *Carbonation*
4. *Bottle label*

The market component consists of four market characteristics:

1. *Location*
2. *Price*
3. *Opening hours*
4. *Advertising*

On the computer screen you can choose different combinations of the product and market characteristics. For this purpose, you can change single, several or all characteristics of a component at the same time. Then click on the "Validate selection" button to see the profit resulting from your selection. This is displayed in the fictitious currency ECU. In a table you can see all your combinations validated so far and their profitability.

Within a component, all characteristics influence the profitability. However, your decisions on the product strategy do not influence the profitability of the market strategy and vice versa.

The most profitable combination in each case has been defined by chance. Therefore, do not try to draw conclusions about the best strategy from your own experience outside the experiment, but explore the respective circumstances without bias. For example, do not let your life experience guide you as to which lemon content or price customers would value most, but test the taste and willingness to pay in the experiment. Please note that product and market components are equally important for the success of your business model, i.e. the maximum achievable profit each from product and market strategy is identical.

Your task

[Note: Product and market component were displayed in random order.]

There are two game phases during which you can develop your strategies. Both phases last four minutes each. In between you have a break of 30 seconds.

[Waterfall:] *During the first phase, you can work exclusively on the product strategy; during the second phase, you can work exclusively on the market strategy. You can change and validate the characteristics as many times as you want within a phase. However, your decisions in the first phase (the four characteristics lemonade color, lemon content, carbonation, and bottle label for product strategy) are set and cannot be changed during the second phase.*

[Agile iterative:] *During both phases, you can spend exactly two minutes working on the product strategy and two minutes working on the market strategy. To do this, you can switch back and forth between the two components, but only until two minutes are reached on one component. You can change and validate the characteristics as many times as you want within each two-minute period. However, four of the eight characteristics (lemonade color and lemon content for product strategy, location and price for market strategy) are set after the first phase based on the highest profit achieved and then cannot be changed during the second phase.*

[Agile iterative + autonomy:] *During both phases, you are free to decide how long you work on the product and market strategy, respectively. To do this, you can switch back and forth between the two components. You can change and validate the characteristics as many times as you want within a phase. However, four of the eight characteristics (lemonade color and lemon content for product strategy, location and price for market strategy) are set after the first phase based on the highest profit achieved and then cannot be changed during the second phase.*

Your compensation

[All treatments:] *Your compensation depends on the profitability of each of your product and market strategies. First, the combination with the highest profit is selected separately for each product and market component from all trials. That is, it is not the last chosen combination that is decisive, but the most profitable one. Second, for your business to be successful, both product and market components* *must convince customers. Thus, you will be paid the LOWER profit from product and market strategy.*

The following example illustrates the payoff (the profit values shown are arbitrarily chosen and not representative). You have tried five combinations for your product strategy and three combinations for your market strategy:

Product strategy	Profit
Combination 1	ECU 20
Combination 2	ECU 10
Combination 3	ECU 60
Combination 4	ECU 30
Combination 5	ECU 20
Market strategy	Profit
Combination 1	ECU 50
Combination 2	ECU 30
Combination 3	ECU 10

First, the combination with the highest profit is determined for product and market strategy individually. In our example, this is Combination 3 for the product strategy and Combination 1 for the market strategy. Second, you are paid the lower profit of the two strategies, i.e. in this case, Combination 1 of the market strategy (ECU 50). The higher profit of the product strategy (ECU 60) is not paid out. The exchange rate is ECU 70 = EUR 1.00.

[Followed by the first comprehension test.]

B.2.3. Part 2 of the Experiment

[Note: Part 1 and Part 2 of the experiment were displayed in random order.]

Please read the following instructions carefully and answer the comprehension questions. You will have two attempts to pass the comprehension questions. If you do not successfully pass the comprehension questions, you will not participate in this part of the experiment and will not be compensated for it. If you have any questions about the instructions, please write a PRIVATE message to the experimenter using the Zoom chat function.

Background

In this part of the experiment, you will form German nouns and verbs (no adjectives, names, brands, cities, etc.) from letters, each on its own game board, similar to Scrabble. Declension and conjugation forms are allowed. There are 100 different letters available for each game board.

You must place the first letter on the orange square in the middle of the game board. Subsequent letters must always be placed directly next to other letters and cannot be placed without this connection.

All letter combinations must make valid words from left to right and top to bottom, but not diagonally. A word is only valid if it is listed at Wiktionary.org (Wiktionary.org is a word collection similar to the Duden). It is then displayed in green.

Your task

[Note: Nouns and verbs component were displayed in random order.]

There are two game phases during which you can form words. Both phases last six minutes each. In between you have a break of 30 seconds.

[Waterfall:] *During the first phase, you can work exclusively on the game board for nouns; during the second phase, you can work exclusively on the game board for verbs.*

[Agile iterative:] *During both phases, you can work for exactly three minutes on the game board for nouns and three minutes on the game board for verbs. To do this, you can switch back and forth between the two game boards, but only until three minutes are reached on one game board.*

[Agile iterative + autonomy:] *During both phases, you are free to decide how long you work on the game board for nouns and the game board for verbs, respectively. To do this, you can switch back and forth between the two game boards.*

[All treatments:] *Letters can be changed and removed only during the phase in which they are placed, i.e. letters that you have placed in the first phase cannot be changed or removed in the second phase. To remove letters, drag them from the edge of the letter field back into the letter pool.*

Your compensation

Your compensation depends on the number of correctly placed letters on both game boards.

*First, the correctly placed letters are counted separately for each of the two game boards. Letters used for two words are counted twice. Each letter is worth 5 points. There are no bonus points, each word is counted only once and each valid letter gives the same score. For example, if there are 2 words with 4 and 6 letters on one board, the score is (4+6) * 5 = 50 points.*

If not all placed letters result in valid words, the game board is invalid and the highest score before the game board became invalid is counted. Therefore, the current score can be lower than the highest score. For example, if you fail to finish a word in the last seconds of the working time, the highest score before you started the invalid word counts.

You will be paid only the LOWER of the scores of both fields. For example, if you have accumulated 50 points for nouns and 60 points for verbs, you will be paid 50 points (these point values are arbitrarily chosen and not representative). The exchange rate is 70 points = EUR 1.00.

[Followed by the second comprehension test.]

B.3. Additional Analyses

B.3.1. Pairwise Correlations

Table B.3. Pairwise Correlations

Variables	*Design task payoff*	*Search task payoff*	*Female (0-1)*	*Age (years)*	*Advanced degree (0-1)*	*Scrabble experience (0-1)*	*German native speaker (0-1)*
Design task payoff	1.00						
Search task payoff	0.14* (0.07)	1.00					
Female (0-1)	0.04 (0.59)	0.04 (0.61)	1.00				
Age (years)	0.08 (0.32)	0.07 (0.38)	0.20*** (0.01)	1.00			
Advanced degree (0-1)	0.16** (0.03)	0.07 (0.38)	0.18** (0.02)	0.53*** (0.00)	1.00		
Scrabble experience (0-1)	0.32*** (0.00)	0.12 (0.11)	0.12 (0.11)	0.09 (0.26)	0.15** (0.04)	1.00	
German native speaker (0-1)	0.00 (0.96)	0.02 (0.82)	0.01 (0.92)	−0.06 (0.43)	0.02 (0.78)	−0.17** (0.03)	1.00

Note: Table shows pairwise Pearson correlation coefficients and significance levels for the 174 participants who completed both tasks.
$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

B.3.2. Design Task: Additional Analyses

In this appendix we present supporting analyses for Section 3.5, focusing on the Design task. Figure B.3 presents the histograms of performance by treatment, showing the similarity of *TA-1* and *TA-2*.

Figure B.3. Design Task: Performance Distribution by Treatment

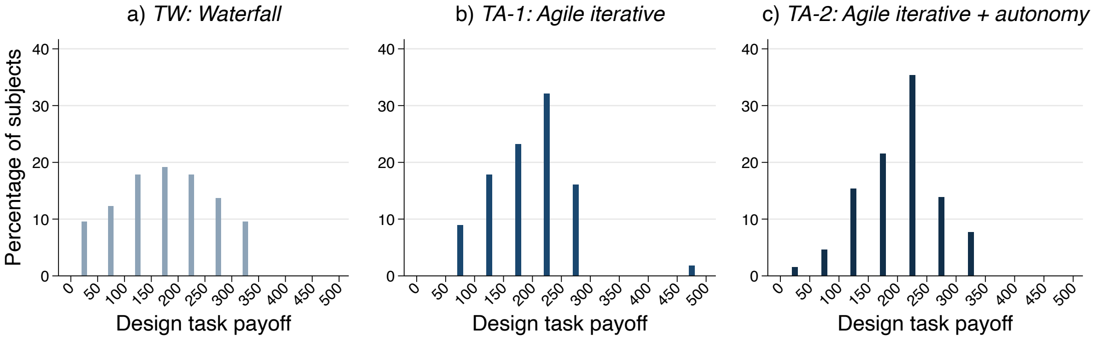


Table B.4 shows the coefficients from quantile regressions of performance on treatments, using the set of covariates used in our main analysis. The analysis shows that, by shrinking the performance variance, *Agile* improves the outcomes mainly in the low range of the performance distribution.

Table B.4. Design Task: Quantile Regressions

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Quantile:	10	20	30	40	50	60	70	80	90
TW: Waterfall	(baseline)	(baseline)	(baseline)	(baseline)	(baseline)	(baseline)	(baseline)	(baseline)	(baseline)
TA-1: Agile iterative	109.50***	49.58**	46.82**	35.95**	42.34***	30.00*	32.50*	29.00	−3.00
	(27.15)	(24.98)	(20.68)	(17.66)	(15.64)	(15.92)	(17.98)	(19.03)	(17.84)
TA-2: Agile iterative	107.30***	58.75***	60.45***	40.68***	36.88***	26.79*	24.17	13.67	−13.50
+ autonomy	(23.99)	(22.08)	(18.27)	(15.61)	(13.82)	(14.06)	(15.89)	(16.81)	(15.76)
Controls	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Constant	73.50	112.90	190.50***	204.90***	204.10***	229.60***	250.00***	269.00***	243.50***
	(82.91)	(76.30)	(63.16)	(53.94)	(47.76)	(48.60)	(54.92)	(58.10)	(54.48)
Number of observations	194	194	194	194	194	194	194	194	194

Notes: Table shows quantile regression coefficients. Dependent variable is Design task performance. Each column corresponds to a quantile, starting from the 10^{th} to the 90^{th} quantile. Controls are age, gender, German native speaker, education, task sequence, component sequence, loss of internet connection, and experience with Scrabble. The number of observations equals the number of participants who completed the task.
$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Table B.5 shows the treatment means for the relevant process metrics in the Design task, discussed in the last paragraph of Subsubsection 3.5.2.1. In addition to the p-values in the table, which denote comparisons between *TW* and *TA-1* as well as between *TW* and *TA-2*, there is also a significant difference for the variable "Number of switches between components" ($p < 0.05$) when comparing *TA-1* and *TA-2*. The remaining variables in Table B.5 are not significantly different between *TA-1* and *TA-2*.

Table B.5. Design Task: Process Variable Means

	TW	TA-1	TA-2
Overall task			
Share of time spent in Noun component	0.50	0.50	0.49
Number of switches between components	1.00	4.61***	6.26***
Share of participants recycling words	0.07	0.25***	0.14
Share of recycled words	0.01	0.04***	0.02
Noun component			
Word length	5.02	4.91	5.14
Number of valid words	8.81	9.25	8.68
Final noun score if nouns first	209.76	214.39	225.00*
Final noun score if nouns second	232.34	234.13	211.03
Verb component			
Word length	5.44	5.71	6.05**
Number of valid words	7.03	7.63	7.42
Final verb score if verbs first	162.65	220.43*	204.85*
Final verb score if verbs second	232.56	210.15	235.48

Notes: Table shows treatment averages for the relevant variables. Significance levels for treatment comparisons are computed using two-sided rank sum tests. Asterisks denote comparisons that use *TW* as the baseline.
$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Finally, Table B.6 shows the effects of word recycling on performance, after controlling for treatment effects, as well as the share of performance variation explained by these variables.

Table B.6. Design Task: Effects of Process Variables on Performance

	(1) Task payoff	(2) Task payoff	(3) Task payoff
TW: Waterfall	(baseline)	(baseline)	(baseline)
TA-1: Agile iterative	44.63***	37.14***	38.39***
	(14.21)	(14.12)	(14.08)
TA-2: Agile iterative + autonomy	39.64***	35.42***	38.79***
	(12.53)	(12.33)	(12.28)
Share of recycled words		221.00**	
		(95.55)	
Ever recycled words? (0-1)			62.06***
			(21.05)
Constant	165.91***	174.70***	181.70***
	(42.65)	(41.80)	(42.12)
Number of observations	194	193	194
R^2	0.207	0.225	0.243
Variation explained			
TW vs. *TA-1*		14.72%	22.54%
TW vs. *TA-2*		6.53%	10.17%

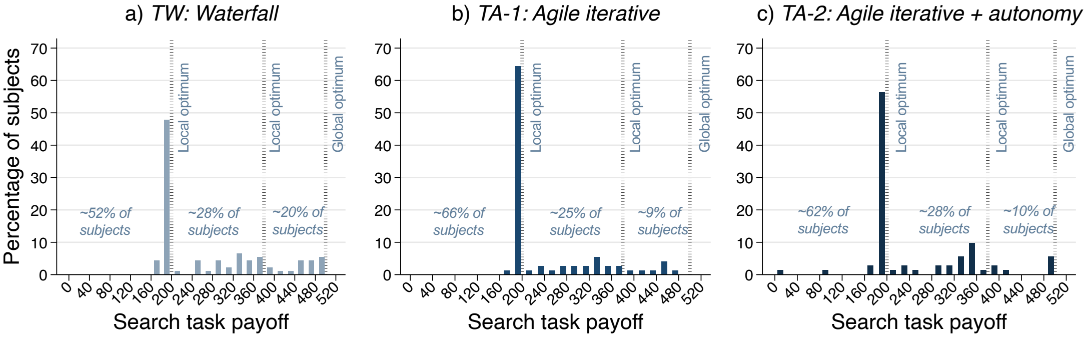
Notes: OLS regressions with standard errors in parentheses. Dependent variable is Design task performance. Controls are age, gender, German native speaker, education, task sequence, component sequence, loss of internet connection, and experience with Scrabble. The number of observations equals the number of participants who completed the task. In column (2) the number of observations is reduced: One participant did not produce any valid words. Variation explained is computed by examining the ratio of the predicted performance difference due to the process variable to the predicted performance difference due to both the process variable and the treatment dummy. See Kagan et al. (2018) for a detailed description of this procedure.

$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

B.3.3. Search Task: Additional Analyses

In this appendix we present supporting analyses for Section 3.5, focusing on the Search task. Figure B.4 presents the histograms of performance by treatment, confirming that in both *TA-1* and *TA-2* a large share of subjects are stuck in the lowest optima and only a small share discovers the global optimum region.

Figure B.4. Search Task: Performance Distribution by Treatment



a) *TW: Waterfall*

b) *TA-1: Agile iterative*

c) *TA-2: Agile iterative + autonomy*

Table B.7 shows the coefficients from quantile regressions of performance on treatments, using the set of covariates used in our main analysis. The analysis presents additional evidence for the result that, by shrinking the performance variance, *Waterfall* improves the outcomes mainly in the mid to top range of the performance distribution (60^{th} and 70^{th} percentile, corresponding to the lowest payoffs in the global optimum region).

Table B.7. Search Task: Quantile Regressions

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Quantile:	10	20	30	40	50	60	70	80	90
TW: Waterfall	(baseline)	(baseline)	(baseline)	(baseline)	(baseline)	(baseline)	(baseline)	(baseline)	(baseline)
TA-1: Agile iterative	0.44	−2.70*	−2.19*	−1.20	−0.73	−71.23*	−85.74**	−42.81	−49.99
	(3.35)	(1.48)	(1.25)	(8.38)	(27.37)	(36.20)	(38.32)	(42.92)	(38.85)
TA-2: Agile iterative	−3.18	−1.20	−1.29	−0.90	−0.87	−66.66*	−78.32**	−28.75	−39.31
+ autonomy	(5.17)	(1.80)	(1.15)	(8.19)	(27.08)	(36.37)	(35.11)	(36.79)	(38.17)
Controls	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Constant	180.50***	193.20***	195.50***	198.50***	197.50***	309.20***	323.20***	404.50***	606.80***
	(14.94)	(5.47)	(3.83)	(9.08)	(44.86)	(76.03)	(88.89)	(99.87)	(123.00)
Number of observations	236	236	236	236	236	236	236	236	236

Notes: Table shows quantile regression coefficients. Dependent variable is Search task performance. Each column corresponds to a quantile, starting from the 10^{th} to the 90^{th} quantile. Controls are age, gender, German native speaker, education, task sequence, component sequence, loss of internet connection, and parameter version. The number of observations equals the number of participants who completed the task.
$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Table B.8 shows the treatment means for the relevant process metrics in the Search task, discussed in the last paragraph of Subsubsection 3.5.2.2. In addition to the p-values in the table, which denote comparisons between *TW* and *TA-1* as well as between *TW* and *TA-2*, there is also a significant difference for the variables “Share of time spent in Product component” ($p < 0.05$) and “Number of switches between components” ($p < 0.01$) when comparing *TA-1* and *TA-2*. The remaining variables in Table B.8 are not significantly different between *TA-1* and *TA-2*.

Table B.8. Search Task: Process Variable Means

	TW	TA-1	TA-2
Overall task			
Share of time spent in Product component	0.50	0.50	0.53**
Number of switches between components	1.00	3.74***	4.77***
Number of validations	51.86	53.40	49.35*
Explored solution space	0.16	0.15*	0.14**
Step size	0.54	0.56	0.55
Market component			
Final market score if market first	319.44	303.45	282.98
Final market score if market second	396.37	316.48***	308.51***
Product component			
Final product score if product first	276.73	317.11	335.91**
Final product score if product second	366.52	296.03***	308.10***

Notes: Table shows treatment averages for the relevant variables. Significance levels for treatment comparisons are computed using two-sided rank sum tests. Asterisks denote comparisons that use *TW* as the baseline.
$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Finally, Table B.9 shows the effects of the three process metrics introduced in Subsubsection 3.5.2.2 on performance, after controlling for treatment effects, as well as the share of performance variation explained by these variables.

Table B.9. Search Task: Effects of Process Variables on Performance

	(1) Task payoff	(2) Task payoff	(3) Task payoff	(4) Task payoff
TW: Waterfall	(baseline)	(baseline)	(baseline)	(baseline)
TA-1: Agile iterative	-32.03^{**}	-33.25^{**}	-26.30^{*}	-31.76^{**}
	(15.36)	(15.09)	(15.07)	(15.20)
TA-2: Agile iterative + autonomy	-29.89^{*}	-26.12^{*}	-18.86	-26.52^{*}
	(15.28)	(15.07)	(15.23)	(15.17)
Number of validations		1.49^{***}		
		(0.49)		
Explored solution space			449.90^{***}	
			(126.70)	
Step size				25.06
				(40.98)
Constant	300.70^{***}	210.70^{***}	222.00^{***}	290.50^{***}
	(52.31)	(59.37)	(55.63)	(57.23)
Number of observations	236	236	236	235
R^2	0.035	0.073	0.086	0.036
Variation explained				
TW vs. TA-1		0.00%	15.43%	0.00%
TW vs. TA-2		12.49%	38.00%	0.00%

Notes: OLS regressions with standard errors in parentheses. Dependent variable is Search task performance. Controls are age, gender, German native speaker, education, task sequence, component sequence, loss of internet connection, and parameter version. The number of observations equals the number of participants who completed the task. In column (4) the number of observations is reduced: One participant only explored one solution so the *Step size* variable could not be computed. Variation explained is computed by examining the ratio of the predicted performance difference due to the process variable to the predicted performance difference due to both the process variable and the treatment dummy. See Kagan et al. (2018) for a detailed description of this procedure.

$^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.