Manuscript in preparation for submission to *Management Science*

One Size Does Not Fit All: Strengths and Weaknesses of the Agile Approach

Evgeny Kagan
Carey Business School, Johns Hopkins University, ekagan@jhu.edu

Tobias Lieberum
School of Management, Technical University Munich, and McKinsey & Company, tobias.lieberum@tum.de

Sebastian Schiffels
Lancaster University Management School, Campus Leipzig, s.schiffels@lancaster.ac.uk

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.

Key words: project management; agile; innovation; behavioral operations

1. Introduction

"Agile" is a suite of workflow management techniques aimed at improving innovation performance (Cooper and Sommer 2018, Fernandez and Fernandez 2008). 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

[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`.

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 (Fiore et al. 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, Yoo et al. 2021, Allon 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

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

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 vs 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 pre-defined (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 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 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 incremenetalism 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.

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]

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 pre-set 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

[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.

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 vs 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 (Kagan et al. 2018, Erat and Gneezy 2016, 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 as 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.

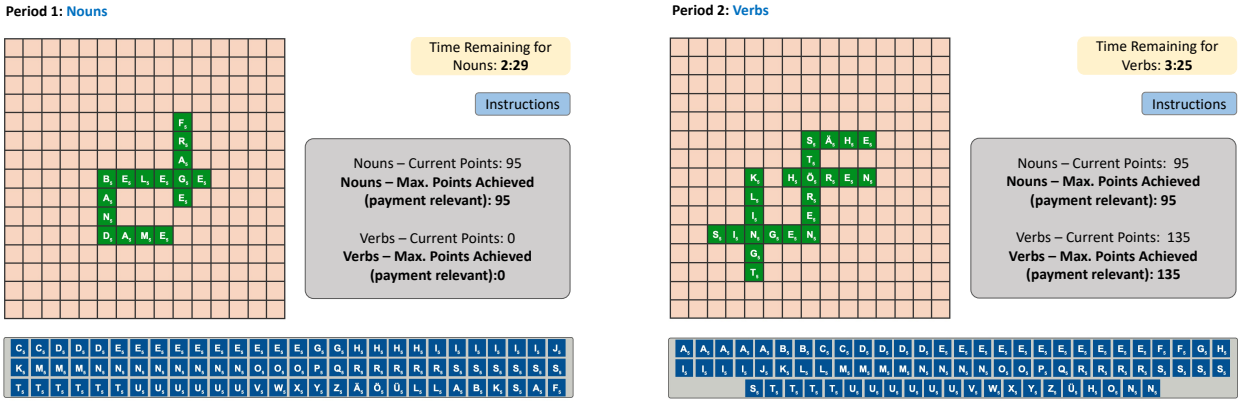
Our second experimental task leverages the approach (more common among economists and business disciplines) of representing innovation as a search process (Levinthal and March 1981, Levinthal 1997). Here, the key objective of the worker is to identify the best solution among a very large number of potential 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. 2021), 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 different incentive systems and find that search strategies are more effective when short-term failure is tolerated and

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

Figure 1 Design Task: Screenshots

(a) Component 1: Nouns (b) Component 2: Verbs



Note: 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 scores after the last valid word, which are for payment (because all words are valid in this example, the scores coincide).

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. Experimental Design

In this Section we present our experimental design. We begin by introducing two real-effort tasks which represent two different innovation settings (Section 3.1). We then present our experimental treatments, and discuss what treatment effects we anticipate given the extant theory (Section 3.2). Finally, we present the details of the performed measurements and protocols (Section 3.3).

3.1. Tasks (Within-Subject)

Our experiments were organized into a 2 (tasks) $\times$ 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 2. We refer to the two tasks as the "Design task" and the "Search task".

[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).

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 its 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 two components is shown in Figure 1.[6] For further details see Section EC.1.

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 1, the noun component (panel a) has four words with 5, 6, 4 and 4 letters respectively. Subjects receive 5 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 1, 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]

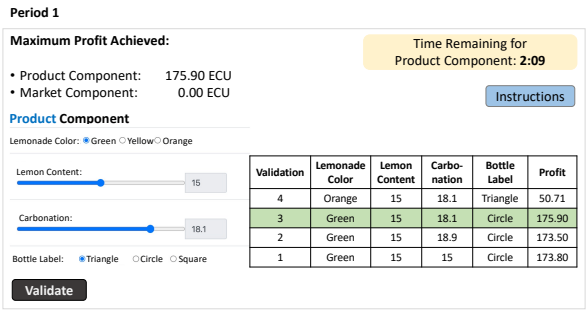
The Scrabble task reproduces several key behavioral dynamics of product development. It is a problem solving task that requires both creative ("divergent", see Sawyer 2011) and analytic (convergent) thinking. Participants begin with an open solution space and limitless creative possibilities. There are no pre-defined 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.

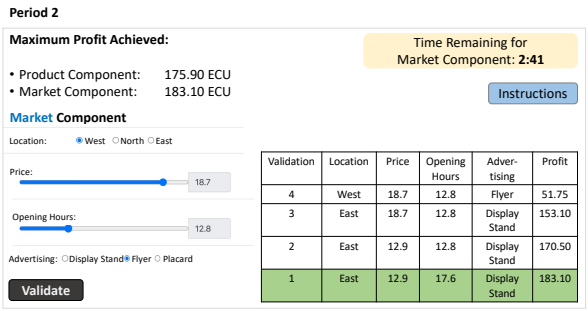
[7] The MIN function ensures that participants work on both components, instead of working on the component they consider easier or more enjoyable. While other payoff functions may be equally suitable to represent complementarities between components, we chose the MIN function mainly to facilitate comprehension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

Figure 2 Search Task: Screenshots

(a) Component 1: Product



(b) Component 2: Market



Note: 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.

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, bottle label. The market component consists of four market attributes: location, price, opening hours, 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 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 3 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 Section EC.1.

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 $92,000^2$

[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.

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 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.

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

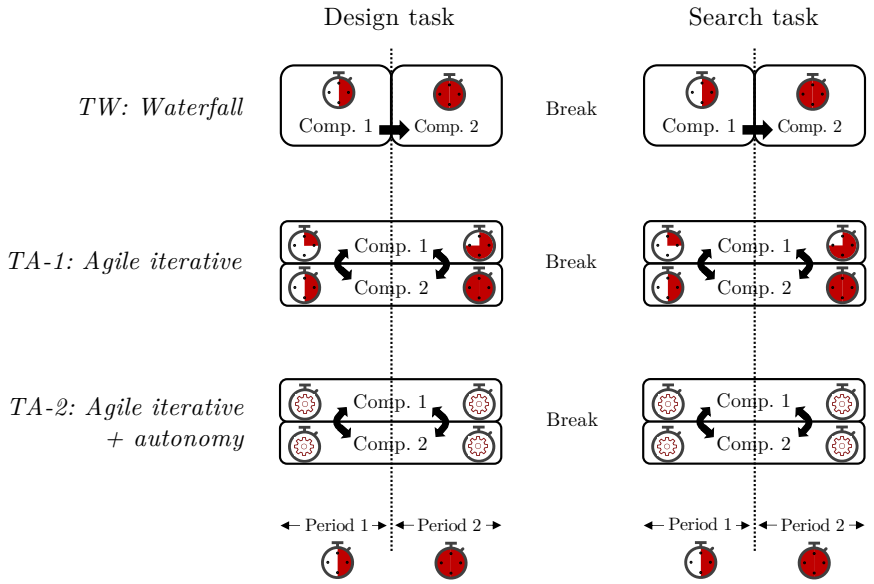
We administer three between-subject treatments. In all three treatments the overall time for each task is fixed, and there are two periods of equal length. However, the workflow, i.e., the sequence of components and the time allocated to each component depends on the treatment. In the *Waterfall* treatment (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 in each component needs to be equal within each period (and therefore in total as well). We label this treatment *Agile iterative* (*TA-1*), because 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.[9]

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).

[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.

Figure 3 Experimental Design



Note. 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.

Taken together, these arguments, and the review of the literature in Section 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 6).

3.3. Parametrization and Experimental Protocol

We conducted pre-tests with 33 participants to calibrate the durations 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 EC.1 and EC.2 for an illustration of the solution space.[10]

The experiment was conducted between December 2020 and April 2021 at a large, public German University. 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 EC.2 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. The total duration of the experiment was 45 minutes.[11]

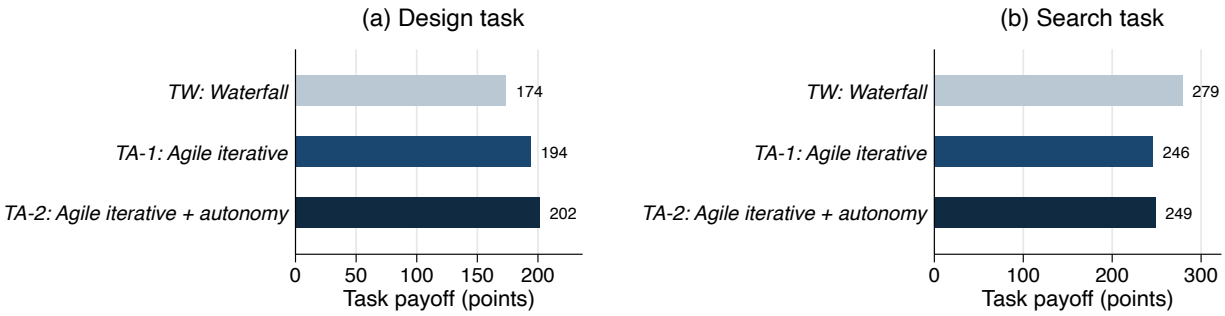
4. Performance Comparisons

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 nonparametric 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 A.1.)

[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 pre-tests. 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.

[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 4 Mean Performance



4.1. Differences in Task Payoff

Figure 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 autonomy are relatively small and not statistically significant (1 to 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 A.1 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 ($p = 0.000$ 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 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 $p > 0.100$).

Table 1 Effects of Agile vs. Waterfall on Performance

	Design task		Search task	
	(1)	(2)	(3)	(4)
Dependent variable:	*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)
No. 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. $^{*}p<0.10$; $^{**}p<0.05$; $^{***}p<0.01$.

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 final 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 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 *TA-2*). Thus, the key performance driver in the Design

Table 2 Sum of Scores and Gap between Scores

	Design task		Search task	
	(1)	(2)	(3)	(4)
Dependent variable:	*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**	-6.123
	(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)
No. 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. $^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

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.

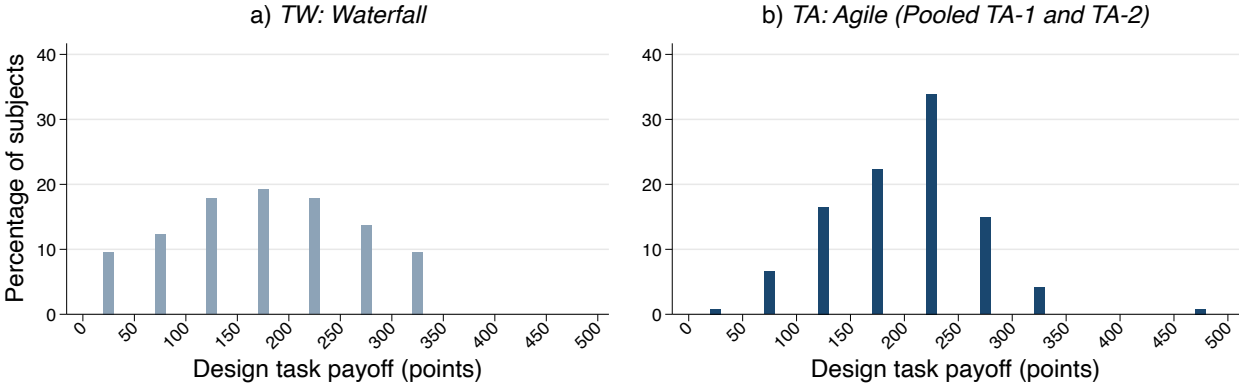
The last two columns of Table 2 focus on the Search task. Column (3) shows that *TW* significantly increases the sum of scores relative to both *TA-1* and *TA-2* ($p = 0.024$ and $p = 0.013$). In contrast, column (4) shows that neither comparison is significant for the gap between the two component scores ($p = 0.817$ 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.

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.

Figure 5 Design Task: Performance Distribution



5.1. Performance Heterogeneity and Learning

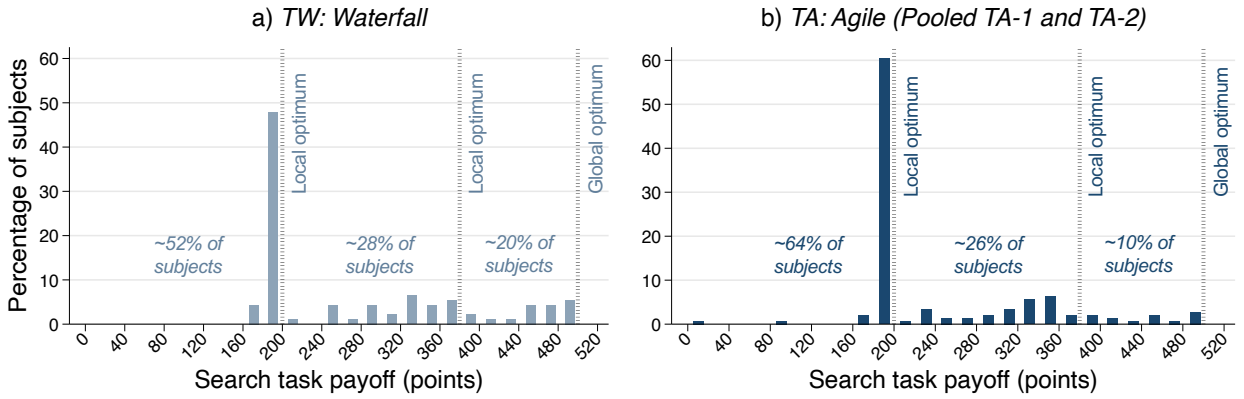
We begin by considering the distributional effects of the Agile approach. Figure 5 shows the distribution of task payoff in the *Waterfall* and the pooled *Agile* treatment in the Design task. The *Waterfall* treatment causes a greater spread in performance with a substantive number of participants (10%) in both the lowest and in the 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 the upper tail of the performance distribution. To examine this 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 to the (pooled) *Agile* treatments (Levene test: $p < 0.001$).[12]

Figure 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]

[12] Treatment-level histograms of task payoff are available in the Appendix (Figure B.1). 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.1.

[13] Treatment-level histograms of task payoff are available in the Appendix (Figure C.1). Quantile regression results that use different quantiles of the performance distribution as the dependent variable are reported in Table C.1.

Figure 6 Search Task: Performance Distribution



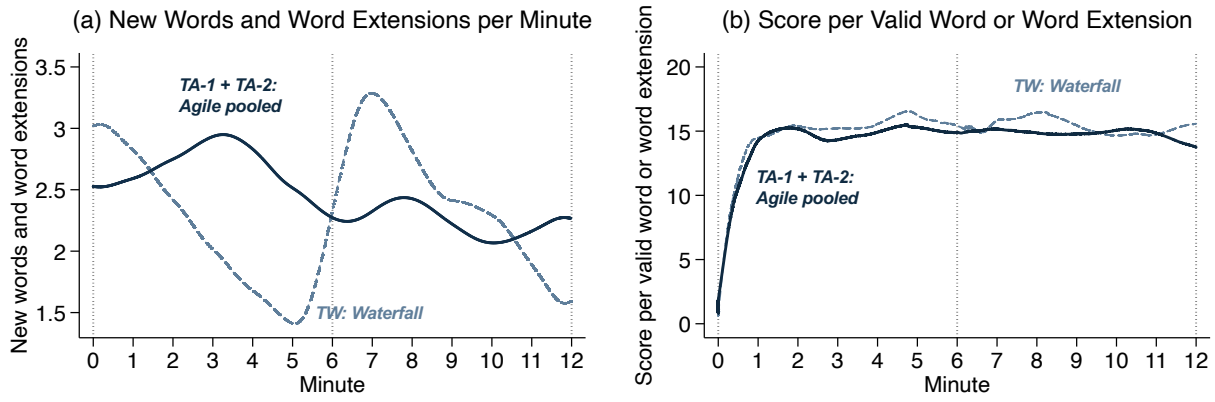
Some of the described variance effects appear to be driven by the differences in the learning patterns. Specifically, in both tasks Component 2 scores are significantly higher than Component 1 scores in the *Waterfall* treatment (paired $t-$tests of final scores, both $p < 0.01$). In contrast, neither score is significantly higher or 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 section 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 6.

5.2. Mechanisms

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. Panel (a) of Figure 7 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 to 1.4 words per minute down in period 1 and from 3.3 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 (nonparametric trend test, both $p = 0.000$). In contrast, the number of words remains quite constant in Agile with 2.1 to 2.9 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 (Panel (b) of Figure 7).

Figure 7 Design Task: Process Variables over Time



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

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* treatment. Only 7% of the participants in *Waterfall* use identical words multiple times, compared to 19% in *Agile*. This difference is statistically 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.2 and B.3 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]

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, vs. exploring a more narrow set of regions with greater thoroughness. This trade-off is captured by the remaining two measures. Specifically, *Explored solution space* measures the breadth of the search, i.e., how much

[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.

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 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) of Figure 8 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 $\times$ 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]

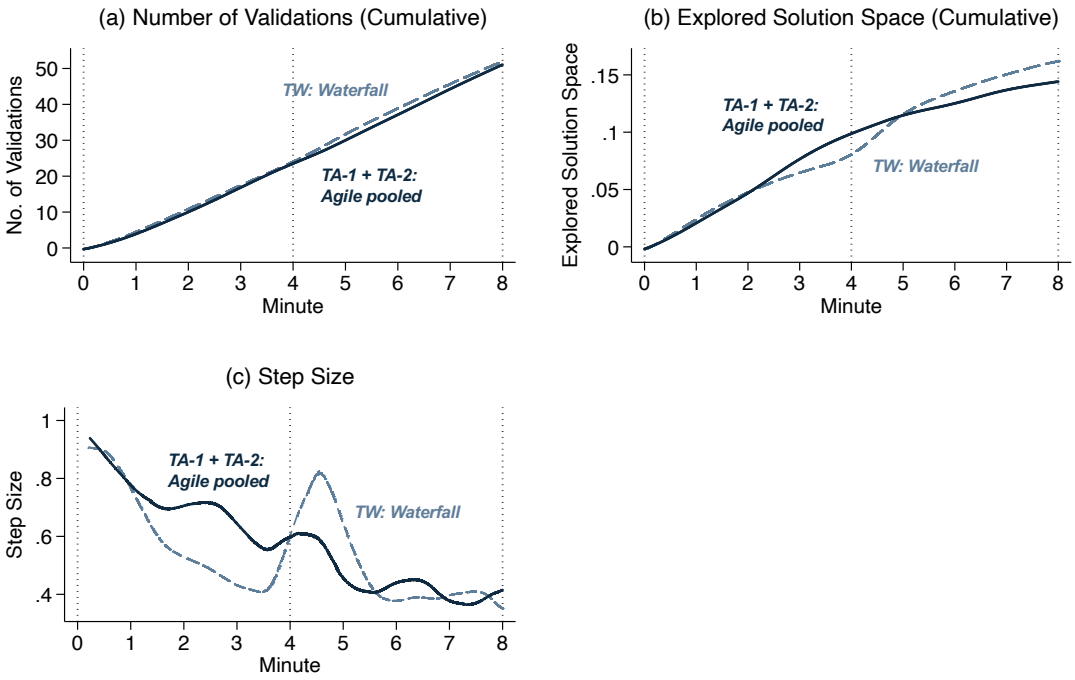
Panel (c) of Figure 8 plots the *Step size*, i.e. the Euclidean distance between the attributes chosen in two subsequent validations. This measure is a proxy for the extent of experimentation and helps understand whether a participant is performing broader exploration (which would manifest in a larger *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 Table C.2 and C.3 in the Appendix we further examine the extent to which each of the three process measures graphed in Figure 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

[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).

Figure 8 Search Task: Process Variables over Time



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

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 treatment differences.

6. Discussion

Our first result is that, on average, participants in the *Agile* treatment 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 percent (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 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, while Waterfall may be a better choice for search-driven projects, for example, when developing a new business

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

Table 3 Agile vs Waterfall Framework

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

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.

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 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.

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 pre-clinical 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), while 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.

References

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 Science* Forthcoming:1–19.

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

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. *Journal of Applied Psychology* 91(4):963–970.

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

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

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

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 Journal* 42(2):361–385.

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

Bryar C, Carr B (2021) Have we taken agile too far? *Harvard Business Review* Digital Articles:1–6.

Charness G, Grieco D (2019) Creativity and incentives. *Journal of the European Economic Association* 17(2):454–496.

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

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

Chick SE, Gans N, Yapar Ö (2021) Bayesian sequential learning for clinical trials of multiple correlated medical interventions. *Management Science* Forthcoming:1–20.

Cooper RG, Sommer AF (2018) Agile–stage-gate for manufacturers. *Research-Technology Management* 61(2):17–26.

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

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

Fernandez DJ, Fernandez JD (2008) Agile project management – Agilism versus traditional approaches. *Journal of Computer Information Systems* 49(2):10–17.

Fiore D, West K, Segnalini A (2019) Why science-driven companies should use agile. *Harvard Business Review* .

Ghosh S, Wu A (2021) Iterative coordination and innovation: Prioritizing value over novelty. *Organization Science* Forthcoming:1–25.

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

Girotra K, Netessine S (2014) Four paths to business model innovation. *Harvard Business Review* 92(7):96–103.

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

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

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

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

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

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

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

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

Kettunen J, Lejeune MA (2020) Waterfall and agile product development approaches: Disjunctive stochastic programming formulations. *Operations Research* 68(5):1356–1363.

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

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

Laufer A, Hoffman EJ, Russell JS, Cameron WS (2015) What Successful Project Managers Do. *MIT Sloan Management Review* 43–51.

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

Levinthal DA, March JG (1981) A model of adaptive organizational search. *Journal of Economic Behavior & Organization* 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 Operations Management* Forthcoming:1–17.

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

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

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

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

Maruping LM, Venkatesh V, Agarwal R (2009) A Control Theory Perspective on Agile Methodology Use and Changing User Requirements. *Information Systems Research* 20(3):377–399.

Mendelsohn GA, Griswold BB (1964) Differential use of incidental stimuli in problem solving as a function of creativity. *The Journal of Abnormal and Social Psychology* 68(4):431–436.

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

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

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. *Journal of Systems and Software* 82(9):1479–1490.

Powell WB, Ryzhov IO (2012) *Optimal learning* (John Wiley & Sons).

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

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

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

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

Sawyer RK (2011) *Explaining creativity: The science of human innovation* (Oxford university press).

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

Seale DA, Rapoport A (1997) Sequential decision making with relative ranks: An experimental investigation of the "secretary problem". *Organizational Behavior and 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 Science* 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 Science* 66(3):1395–1420.

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

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

Tuckman BW (1965) Developmental sequence in small groups. *Psychological Bulletin* 63(6):384–399.

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

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

Appendix A: Pairwise Correlations

Table A.1 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.*$p < 0.10$; **$p < 0.05$; ***$p < 0.01$.

Appendix B: Design Task: Additional Analyses

In this Appendix we present supporting analyses for Section 5, focusing on the Design task. Figure B.1 presents the histograms of performance by treatment, showing the similarity of *TA-1* and *TA-2*.

Figure B.1 Design Task: Performance Distribution by Treatment

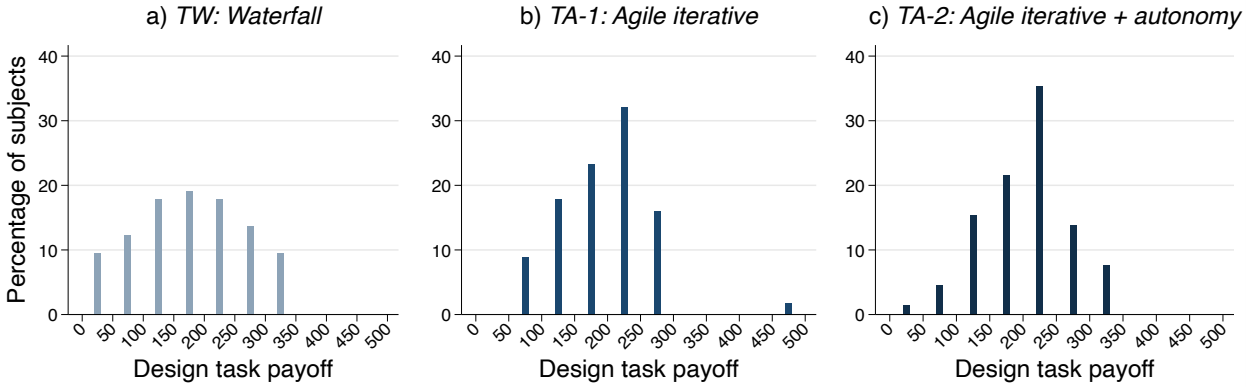


Table B.1 shows the coefficients from quantile regressions of performance on treatments, using the set of covariates used in our main analysis. The analysis shows that, by shrinking the performance variance, *Agile* improves the outcomes mainly in the low range of the performance distribution.

Table B.1 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)
Observations	194	194	194	194	194	194	194	194	194

Notes: Table shows quantile regression coefficients. Dependent variable is Design task performance. Each column correspond 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, experience with Scrabble. The no. of observations equals the number of participants who completed the search task. $^{*}p < 0.1$, $^{**}p < 0.05$, $^{***}p < 0.01$

Table B.2 shows the treatment means for the relevant process metrics in the Design task, discussed in the last paragraph of Section 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 "No. of switches between components" ($p < 0.05$) when comparing *TA-1* and *TA-2*. The remaining variables in Table B.2 are not significantly different between *TA-1* and *TA-2*.

Table B.2 Design Task: Process Variable Means

	TW	*TA-1*	*TA-2*
Task metrics			
Share of time spent in Noun component	0.50	0.50	0.49
No. of switches between components	1	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. *$p < 0.1$, **$p < 0.05$, ***$p < 0.01$

Finally, Table B.3 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.3 Design Task: Effects of Process Variables on Performance

Dependent Variable:	(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	165.90***	174.70***	181.70***
	(42.65)	(41.80)	(42.12)
N	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: Table shows quantile regression coefficients. Dependent variable is Design task performance. Controls are age, gender, German native speaker, education, task sequence, component sequence, loss of internet connection, and parameter version. 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. The no. of observations equals the number of participants who completed the task. In column (2) the no. of observations is reduced: 1 participant did not produce any valid words. *$p < 0.1$, **$p < 0.05$, ***$p < 0.01$

Appendix C: Search Task: Additional Analyses

In this Appendix we present supporting analyses for Section 5, focusing on the Search task. Figure C.1 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 C.1 Search Task: Performance Distribution by Treatment

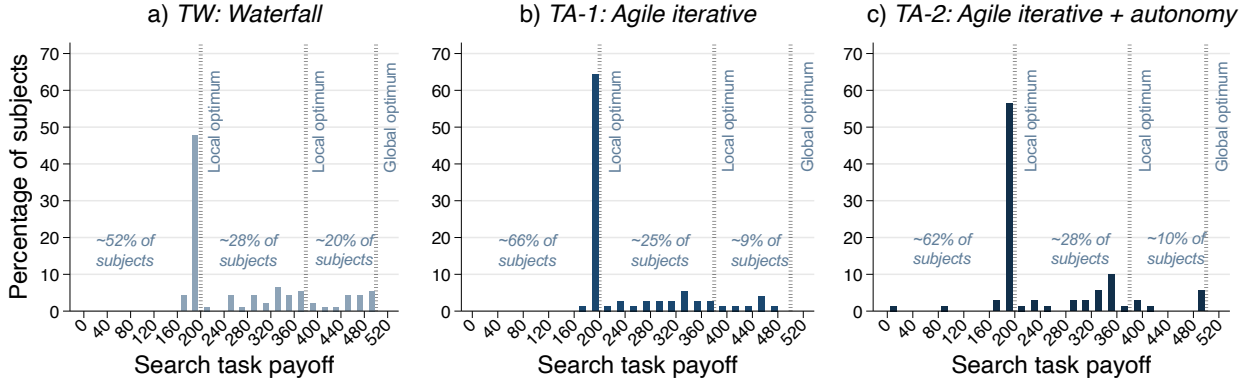


Table C.1 shows the coefficients from quantile regressions of performance on treatments, using the set of covariates used in our main analysis. The analysis presents additional evidence for the result that, by shrinking the performance variance, *Waterfall* improves the outcomes mainly in the mid to top range of the performance distribution (60th and 70th percentile, corresponding to the lowest payoffs in the global optimum region).

Table C.1 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)	(8.384)	(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)	(123.0)
Observations	236	236	236	236	236	236	236	236	236

Notes: Table shows quantile regression coefficients. Dependent variable is Search task performance. Each column correspond 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. $^{*}p < 0.1$, $^{**}p < 0.05$, $^{***}p < 0.01$

Table C.2 shows the treatment means for the relevant process metrics in the Search task, discussed in the last paragraph of Section 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 "Share of time spent in Product component" ($p < 0.05$), and the "No. of switches between components" ($p < 0.01$). The remaining variables in Table C.2 are not significantly different between *TA-1* and *TA-2*.

Table C.2 Search Task: Process Variable Means

	TW	*TA-1*	*TA-2*
Overall			
Share of time spent in Product component	0.50	0.50	0.53**
No. of switches between components	1.00	3.74***	4.77***
Number of validations	51.86	53.40	49.35*
Lanscape coverage	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. *$p < 0.1$, **$p < 0.05$, ***$p < 0.01$

Finally, Table C.3 shows the effects of the three process metrics introduced at the beginning of Section 5.2.2 on performance, after controlling for treatment effects, as well as the share of performance variation explained by these variables.

Table C.3 Search Task: Effects of Process Variables on Performance

Dependent Variable:	(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)
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: Table shows quantile regression coefficients. 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 no. of observations equals the number of participants. 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. In column (4) the no. of observations is reduced: 1 participant only explored one solution so the *Step size* variable could not be computed. *$p < 0.1$, **$p < 0.05$, ***$p < 0.01$

Electronic Companion

Appendix EC.1: Experimental Design Details and Parametrization

EC.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 (Noun and Verb) 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)

EC.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 variables (lemonade color, lemon content, carbonation, shape of the bottle label). The second component is the Market component, consisting of four variables (location, price, opening hours, advertising). For each component two of the variables are discrete, while the other two are continuous. Figure EC.1 and Figure EC.2 show the landscapes for all combinations of the discrete variables 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 parameters.

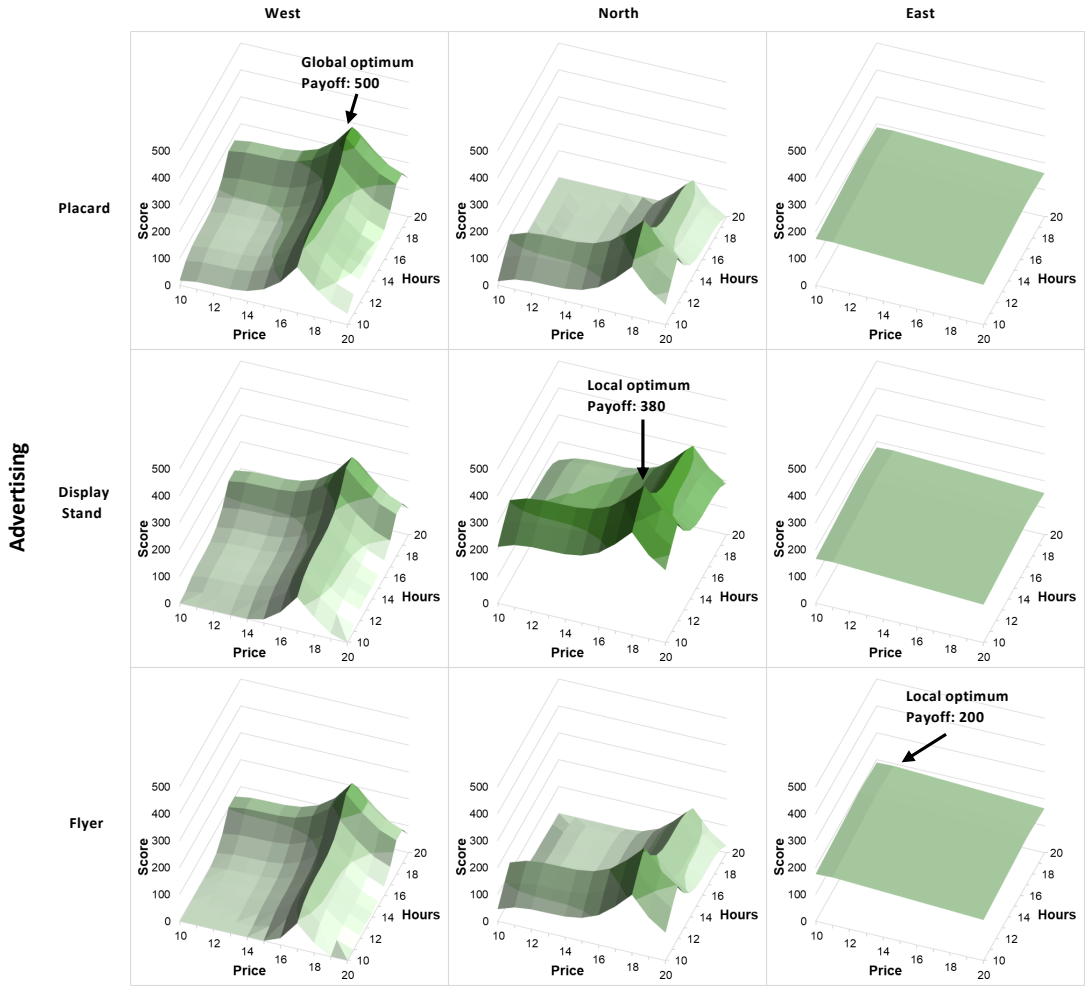
Product component:

1. Color = Green, Yellow, Orange
2. Lemon content = 10, 10.1, 10.2, ..., 19.9, 20
3. Carbon dioxide content = 10, 10.1, 10.2, ..., 19.9, 20
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

Figure EC.1 Landscapes For the Market Component

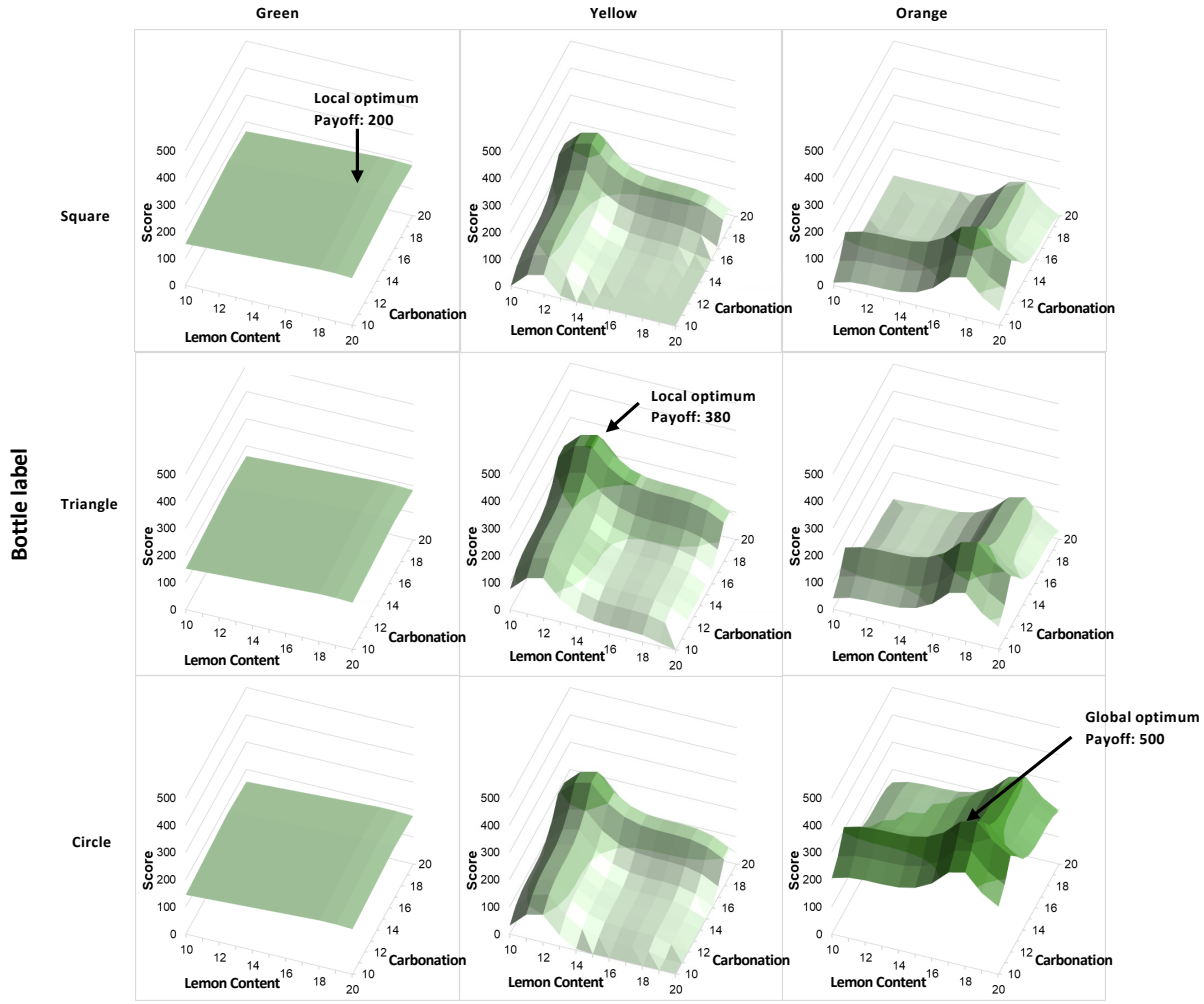


Note. The graphs show one of the two parametrizations used in the experiment. The second parametrization was similar, but had a different price/hour combination for the location of the local and global maxima.

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, generated by randomly choosing a point on each of the two continuous parameters. Table EC.1 shows the optimal selections and maximum profits for each component and parameter version.

For the market component, Figure EC.1 shows the three maxima, each of which corresponds to a combination of location and advertisting. For the product component, Figure EC.2 shows the three maxima, each of which correspond to a combination of lemonade color and bottle label. As shown in Table EC.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

Figure EC.2 Landscapes For the Product Component Color



Note. 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.

in Figures EC.1 and EC.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 EC.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. $\textit{absolute deviation} \times 3$. The penalties for the two highest maxima (at 380 and 500 points) both follow a S-shaped curve. They were computed based on exponentiation of the absolute deviation, calibrated by three constants, i.e.

Table EC.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

$(\frac{absolute\ deviation}{5} - 1)^3 \times 150 + 150$. 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).

Table EC.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

Appendix EC.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. All instructions were read loud. The instructions are summarized below (translated from German):

Introduction

"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's 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 [lab name blinded for review]. 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.]

Part 1 of the Experiment - Instructions and Comprehension Test

[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 experiment director 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. *Color*
2. *Lemon content*
3. *Carbon dioxide content*
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

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 color, lemon content, carbon dioxide content, 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 (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. 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 (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 cannot then 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:

Table EC.3

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.

EC.2.1. Part 2 of the Experiment

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 experiment director 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 playing field, similar to Scrabble. Declension and conjugation forms are allowed. There are 100 different letters available for each game field.

You must place the first letter on the orange square in the middle of the game field. Further letters must always be placed directly on 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

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

[Waterfall:] *During the first phase, you can work exclusively on the playfield for nouns; during the second phase, you can work exclusively on the playfield 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 playfields, but only until three minutes are reached on one playfield.*

[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 playing fields indefinitely.*

[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 playing fields.

*First, the correctly placed letters are counted separately for each of the two game fields. 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 field is invalid and the highest score before the game field became invalid is valid. 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 editing time, the highest score before you started the invalid word counts.

You will be paid only the LOWER of the score 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 are not representative). The exchange rate is 70 points = EUR 1.00.