CONNECTING DESIGN ITERATIONS TO PERFORMANCE IN ENGINEERING DESIGN

Vrolijk, Ademir-Paolo;
Deng, Yuanzhe;
Olechowski, Alison

University of Toronto

ABSTRACT

No matter a system's size, complexity, or domain, iterations are fundamental to its design process. However, there is a duality: iterations are both signs of usefully exploring the system's design space and failure to find an appropriate solution. This ambiguity means that we have not been able to connect teams' iterating behavior to their design's performance, potentially obscuring a way to influence the design process.

As such, our exploratory study unpacks the relationship between design iterations and performance. We observed 88 teams in the 2020 Robots to the Rescue Competition in rich detail. Using logs of 7,956 iterations on a Computer-Aided Design platform, we analyzed how high- and low-performing teams revised their submissions, searching for consistent differences in their behavior. We found significant differences in the iterations' number, scale, and cadence between these groups of teams. These findings emphasized the correlation between certain iteration patterns and the success of a design: the best teams will likely revise differently than the worst ones. It also showed the importance of a fine-grained, time-dependent view of the design process to resolve open questions in the literature.

Keywords: Design process, Design Revision, Conceptual design, Computer Aided Design (CAD), Multi-user CAD

Contact:
Vrolijk, Ademir-Paolo
University of Toronto
Canada
a.vrolijk@utoronto.ca

Cite this article: Vrolijk, A.-P., Deng, Y., Olechowski, A. (2023) ‘Connecting Design Iterations to Performance in Engineering Design’, in *Proceedings of the International Conference on Engineering Design (ICED23),* Bordeaux, France, 24-28 July 2023. DOI:10.1017/pds.2023.107

1 INTRODUCTION

Iteration is fundamental to the design process (Wynn and Eckert, 2017; Maier and Störrle, 2011). Imagine this scenario: a team designing a new system discovers a problem in their design. If they continue down the path they are on, the system will perform poorly or fail. According to the team's current understanding, something must be done to correct this course. Here, the team diverts some of their effort to solve this problem: reverting and altering the design to, hopefully, change its forecast. With the changes made, the team continues their development down a new path. They progress their design until they encounter another problem, starting the cycle once more. Scenarios like this—a team wrestling with micro-level iterations to impact design outcomes—occur no matter the system's size, complexity, or domain. In design, "iteration is a fact of life" (Wynn and Eckert, 2017, p. 153).

However, there is a duality here: iterations are signs of both failure avoidance and unfruitful searching. On the one hand, designers can creatively cycle their design to usefully explore their system's design space (Baldwin and Clark, 2000). The team arrives at a high-performing system by ideating, assessing, and discarding design configurations that lead to poor ones—for example, selecting among different propulsion mechanisms for skiing uphill (Chusilp and Jin, 2006). The result is a map of what works and what does not. A team making many revisions to their design might mean they are spanning the design space and eliminating bad options. On the other hand, iterations could also imply a failure to find an appropriate solution (Yassine and Braha, 2003)—also referred to as churn (Yassine et al., 2003). These iterations are misguided searches of the design space producing no meaningful increase in performance (Ballard, 2000; Lottaz et al., 1999): for example, the inability of automotive stylists and engineers to quickly converge on a feasible interior design (Yassine et al., 2003). In these cases, the team might "go round and round in circles, unable to produce a form that is thoroughly right" (Alexander, 1964, p. 117).

This ambiguity means we do not understand the relationship between a team's iterating behavior and their design's performance. While all teams iterate their designs during their development, their behavior can paint both a positive and negative picture of the quality of their (intended) design. Empirical studies bear out this difficulty of interpretation. Iterations can mean different things to different actors (cf. Adams and Atman, 2000; Smith and Tjandra, 1998) and might involve the same actions—whether they are expected, add value, or neither (Pektaş and Pultar, 2006; Haller et al., 2015). In the face of this ambiguity, the academic and practitioner guidance has been rather limited: avoid late-stage changes to a design project, as these are especially expensive in terms of effort and/or cost (e.g., Love, 2002; Chua and Hossain, 2012; Kossiakoff et al., 2011). Scholars recommend a similar approach across a portfolio of projects (Khanna et al., 2016). However, while this heuristic lowers the cost of a design by front-loading any changes, it obscures any potential relationship between iterations, their characteristics, and the design's performance.

In this exploratory study, we connect a team's iterating behavior to their design's performance. Our setting was the 2020 Robots to the Rescue (RttR) Competition, a large-scale robotics contest where teams submitted problems and solutions. Here, we observed, in rich detail, the design actions of 88 teams on a Computer-Aided Design (CAD) platform over time. Defining an iteration as edits or deletions to their virtual design, we searched for consistent differences in these actions between high- and low-performing teams. Notably, we found significant differences in the number, scale, and cadence of iterations between these groups. These findings emphasized the correlation between certain (micro-level) iterations and the success of a design: the best teams will likely iterate differently than the worst ones. It also showed the importance of a fine-grained, time-dependent view of the design process to resolve open questions in the literature.

We structured the paper as follows: we first describe the ambiguity of iteration as a predictor of performance. Next, we present our research setting, the data collected, the approach, and why our study addresses the gap. We then present our findings. We close with a discussion about these findings and the limits of our study.

https://doi.org/10.1017/pds.2023.107 Published online by Cambridge University Press

2 BACKGROUND

We iterate a design in the hopes of better performance. However, the resulting change(s) could worsen it instead. A team's actions to fix the design might lead to other problems that require attention—rapidly increasing the number of problems instead of converging on a good design. This churn wastes project money and time, which can cause a project to fail (Yassine et al., 2003; Loch et al., 2003; Taylor and Ford, 2006). As such, while iterations have been widely studied (see Wynn and Eckert 2017 for a review), few have tried to connect these to how well a design will perform (e.g., Chusilp and Jin, 2006; Boudouh et al., 2006; Atman et al., 1999).

The pitfalls of iteration are clearest in the design of complex systems: the characteristics of these systems make iterations more likely and more impactful. Complex systems often have many parts that interact in non-simple ways through at least as many interfaces (Simon, 1962). Designing these systems requires a breadth and depth of specialized knowledge which often surpasses the problem-solving capabilities of a single solver (Cyert and March, 1963). Gathering the required knowledge and coordinating the problem-solving effort among the various contributors is a major task (Baldwin and Clark, 2000); a task which can be especially costly if the design team needs to find external contributors (von Hippel, 1994). In these contexts, identifying the "right" design option is rarely straightforward. The design team often cannot foresee how their choice will impact the design's performance until the change is made. A poor design could result from pursuing a design choice based on preliminary information (Loch and Terwiesch, 2005). A team might then try to backtrack and correct their decision, only to end up in a "vicious cycle of continuing revisions" (Huberman and Wilkinson, 2005, p. 308). Worse still, because subsequent choices depend on previous ones, better-performing design families might be out of reach once a decision is made.

In their paper, Loch et al. (2003) recount an anecdote to describe what these micro-level iterations look like. A car engine development manager described to the authors how their company pursued development paths that would not result in the performance improvements they hoped to see. In revising the design, the manager's team would "run into trouble" with their chosen design direction (Loch et al., 2003, p. 187). Facing a lower performance than they started with, they would have to revert their changes—some that might have progressed significantly—and revisit the original decision. This cycle would happen numerous times for the same decision. In short, iterations may not result in a better design even "after a lot of work" (Mihm et al., 2003, p. 734).

Despite the potential for this churn, practitioners will often highlight the positive aspects of what these changes could mean for the design: iterations represent progress. The change aims to increase the design's performance and has a non-zero probability to do so. Many organizations embrace this probability and have adopted iteration as part of their culture. In this view, changing direction after acknowledging something is wrong is synonymous with progress and innovation. For example, iteration is policy at SpaceX, a high-profile government contractor in the space sector. Placing his company in contrast to his government partner, SpaceX's CEO was quoted in 2005 as saying: "Failure is an option here. If things are not failing, you are not innovating enough" (Reingold, 2005, para. 18).

Some organizations have even tried to positively influence the design by directly shaping the iteration process. At Pixar in the 1990s, failure was seen as a "manifestation of learning and exploration" (Catmull and Wallace, 2014, p. 109). To them, new endeavors were going to be full of failures. Promptly changing course when their staff encountered these failures, thus revising their designs, was crucial to find the right solution. Accordingly, their mantra for discovering better performing designs was "fail early and fail fast" (Catmull and Wallace, 2014, p. 109)—a sentiment appearing in other sectors as well (McGrath, 2011).

Failing fast and early reflects a common heuristic in engineering design: iterations, if any, should happen *early* in the design process (Maier and Rechtin, 2000; Kossiakoff et al., 2011). There are two drivers for this time-dependent thinking. First, learning can drive change. Designers expect to better understand the dynamics of their system as the process unfolds—this is especially true for novel or complex systems.

As such, the first draft of the design (requirements) is broad, underdefined, and not final (Kossiakoff et al., 2011). Second, and relatedly, sunk cost and design solidification make change harder. Early in the process, few(er) parts of the system are locked in, either through detailed design or manufacturing. Revising a design's parts early on will, on average, require less effort than later when they are more detailed. Even churn, though wasteful and unproductive, can be absorbed when the effort required is relatively low. In essence, pushing design teams to iterate early in their process acknowledges the gains that stem from iterations while hedging against their potential drawbacks.

However, in trying to minimize negative outcomes, the heuristic also hinders a deeper understanding. In particular, it obscures any relationship between a team's iterating behavior and the design's performance at that early stage. For example, some teams may explore their design space in a sub-optimal way, systematically leading to poorer designs and related instances of churn. Conversely, other teams may have exploration strategies find valuable performance peaks across the design space, minimizing significant design changes. This relationship is still unclear. We intend to explore iteration beyond the simple existing heuristics, resulting in better early-stage design approaches.

3 RESEARCH APPROACH

3.1 Setting

To understand how design revisions across time affect the quality of the design, we collected highly detailed time-series data on team in a CAD contest. Our setting was the 2020 RttR Competition. This virtual design contest simulated a product development process: teams of high-school students from around the world competed to design a robot to solve a real-world problem over the course of 41 days. It was hosted by PTC Inc., a CAD platform provider, as a replacement for that year's FIRST Robotics Competition[1].

The RttR competition was a success. By the submission deadline, 150 teams from ten countries submitted designs containing over 70,000 unique parts. Submissions were judged by experienced design engineers at PTC. In total, they awarded over $5,000 dollars of prize money to winning teams, along with non-monetary special mentions (like best documentation). These prizes were commensurate with similar competitions on other contest platforms (Szajnfarber et al., 2020; Vrolijk and Szajnfarber, 2023).

PTC's CAD platform, Onshape, tracked all user actions that created the teams' submissions. In Onshape, like other mechanical CAD platforms available, work is divided into different, but connected, design workspaces. Users design and build parts in one, insert and assemble parts in another, and convert parts and/or assemblies into drawings for manufacturing in a third. On that platform, all user actions per team, essentially every click made in their design's history, were automatically timestamped and logged in the backend analytic data. We accessed the timestamped log of all actions, which described each team's design process in rich detail. Additionally, the competition strongly encouraged teams to design their submissions on that platform and required teams to submit their final models in Onshape as well. With data exported as an audit trail, we rebuilt each team's design process in chronological order, and it facilitated our analysis of their iterations.

In sum, several reasons made the 2020 RttR competition a suitable setting to address our gap. First, we accessed numerous, independent design processes tackling comparable problems. Each competing team independently created its solution, resulting in a varied set of submissions and processes. Second, the competition, not the research team, ranked teams' performance. The judges from industry reviewed and scored the teams' robot designs. The scores and associated comments ranked the quality of teams' design and identified who performed best (and worst). Third, we captured the teams' complete process of designing early-stage hardware in CAD (see also Nourimand and Olechowski, 2020). Each team's design process began and ended with the competition. As the platform logged all actions performed by all teams,our data maintained this resolution across teams' entire design process. Thus, we captured how they designed their submissions; logging detailed, timestamped data non-intrusively. The data greatly

[1] The popular hardware-focused robotics contest was cancelled because of COVID-19.

facilitated a time-dependent analysis to explore the relationship we are interested in (see also Zhang et al., 2018; Gopsill et al., 2019).

3.2 Data

We accessed the timestamped log of all CAD actions for 88 teams, which described each team's design process in rich detail. Our analysis excluded teams who did not have timestamped Onshape action logs or had less than two iterations across either the first or second half of the competition[2]. For example, a team's submission would not contain these data if they created (most) of their parts/assemblies outside of Onshape. Thus, we had comparable data across all teams to study their design process in a time-series manner.

3.3 Measures

Our analysis relied on measures describing how well teams did in the contest and the iterations teams made to their designs. We describe these measures below.

3.3.1 Design performance

We used the competition's team ranking to measure design performance[3]. RttR's judges awarded points according to a common rubric, and the competition awarded prizes based on the points each team received. We used this ranking to separate teams into three groups: high-, medium-, and low-performers. The ranking evaluated 1) the completeness and complexity of a team's design, 2) whether the design adequately addressed the problem a team proposed, 3) the feasibility and manufacturability of a design, and 4) a team's innovative usage of FIRST CAD components. We classified all teams that earned 14 (out of 20) points or more as high-performers—a total of 28 teams. This group included all prize winners except for four honorary mentions. We classified all teams that earned ten or fewer points as low-performers—a total of 27 teams. The remaining 33 teams were classified as medium-performers.

High-performers generally produced the best and most complete submissions. Though several teams stumbled on their design's kinematics and some missing components, the judges complimented their submissions. These teams generally described their solution well. Some received high praise for their designs and their modeling abilities. For their written review of a team's submission, one judge wrote: "Great robot, great kinematics, I'd like to see this [team] at work!" In contrast, many low-performers did not submit a complete CAD design, resulting in poor submissions. Free floating parts, unspecified kinematic constraints, missing descriptions, or generally incomplete models were common problems identified by judges. These issues called the design's feasibility into question. Part of another judge's review on a harvesting robot read as follows: "Incomplete CAD model. No explanation of how robot actually works. Very tough to have robots harvest individual crops with given design considerations."

3.3.2 Iterations

We operationalized iterations as follows. Adapting Deng et al.'s (2022) framework of CAD action types, we counted the *number* of iterations through a team's revising actions: the edits and deletions in a team's action log. An edit was a single, active change made by any team member to a sketch, a part, or an assembly. A deletion was a single removal of any of these from their design. Edits and deletions differ from creation actions, where team members add a sketch, part, or assembly to their design (Deng et al., 2022; Gopsill et al., 2019). Thus, each iteration indicated instances of course correction in a team's design process: team members modified or subtracted from their existing design instead of building on it. To reduce the chance of misinterpreting the team members' actions (e.g., simple mistakes), we counted one iteration when three or more edits or deletions occurred in one minute.

[2] We excluded 62 teams of the total of 150 teams who participated.

[3] RttR's scoring rubric also evaluated how well each team defined the problem they were solving. We omitted this Problem Definition parameter from our analysis as teams' performance was not based on their CAD models.

In an example of one such iteration, a team changed the bottom of the funnel of their pothole-repairing robot to be a single flat surface instead of multiple angled segments. This iteration took one team member one minute to complete and consisted of two deletions and one edit. In total, high-performing teams made 3,589 iterations, medium-performers made 2,826, and low-performers 1,541.

Next, we counted the number of edits and deletions to measure each iteration's *scale*. Scale was a proxy for the breadth and depth of the changes each time the team revised their design. We counted 33,342 edits and deletions for high-performers, 20,583 for medium-performers, and 33,497 for low-performers; the counts informed the scale measure across these groups.

Finally, we characterized how often iterations occurred to measure each team's *cadence* of revision. To ensure a common measurement, we converted the clock time teams spent on their design to a non-dimensional design time. We first identified each iteration's timestamp within a team's design process, then converted these timestamps to percentages of the team's design time[4]. We counted three or more iterations within 1% of the team's accumulated design time as a cluster. To compare all teams fairly, we converted their counts into percentages of their total iterations. We then calculated the average fraction captured by each cluster instead of on their own. As such, cadence measured whether teams iterated their design in concentrated periods of change or steadily along their timeline.

We summarized our operationalizations in Figure 1 below. Here, iterations are depicted as individual red dots. Their scale, the number of their edits and deletions in clock time, is depicted as red lines. Finally, a team's cadence, the number of iterations occurring within clusters, is depicted as a grouping of red dots.

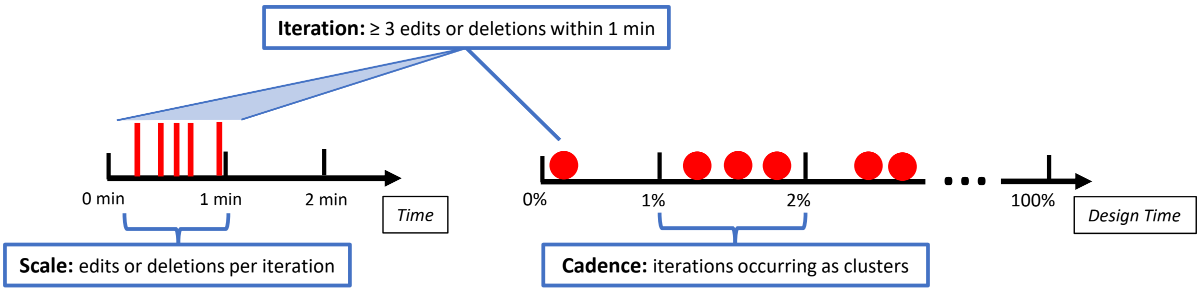

Figure 1. Illustrating the number, scale, and cadence of iterations

4 FINDINGS

We examined the iterating behavior of high- and low-performers to understand how these actions reflected the teams' performance. Our analysis found significant differences in the number, scale, and cadence of iterations between these groups—the means and standard deviations for each characteristic are summarized in Table 1 below. Moreover, a sensitivity analysis that varied the threshold of edits and deletions per iteration (i.e., other than three) revealed the same trends.

First, high- and low-performing teams differed in the *number* of iterations made to their designs. High-performers counted more than twice as many iterations than low-performers. We used a Mann-Whitney U test to verify this difference across the teams; H_0 stated that the average number of iterations between both groups was equal. We rejected H_0 in both the first and second halves of their design process.

Second, the two groups differed in the scale of their iterations. On average, high-performers' iterations were of smaller scale than those of low-performers—the former's iterations counted fewer edits and

[4] For our analysis, we calculated each team's accumulated design time. We removed any idle time (when no actions took place) longer than ten minutes from the team's total time spent on the design. The accumulated design time was then used to turn the respective team's timestamps into non-dimensional values.

https://doi.org/10.1017/pds.2023.107 Published online by Cambridge University Press

Table 1. Means, standard deviations, and comparisons of iteration characteristics

Characteristic	High-performing Teams		Low-performing Teams		Comparison	
	First half	Second half	First half	Second half	First half	Second half
Number	67.9 (63.5)	60.2 (53.1)	28.5 (34.7)	28.6 (35.5)	572.5**	587.5***
Scale	6.19 (3.07)	5.86 (2.28)	8.99 (5.75)	21.9 (69.9)	226.5*	349.5
Cadence	0.031 (0.02)	0.032 (0.02)	0.050 (0.04)	0.041 (0.03)	12508.0***	16432.5

Notes: We compare characteristics of high- and low-performing teams per each half of their design process. The Mann-Whitney U test statistic is reported with p-values: *$p < .05$. ** $p < .01$. ***$p < .001$

deletions. Here, our H_0 tested whether the average number of revising actions between both groups was equal. We rejected H_0 in the first half of the design process: high-performers' scale of iterations differed significantly from low-performers in the first half of the design process.

Lastly, the *cadence* of iteration also differed between high- and low-performing teams. High-performers clustered fewer iterations together, spreading them out across the design process. Here, our H_0 tested whether the fraction of a team's iterations per cluster between both groups was equal. We rejected H_0 in the first half of the design process. In fact, low-performers clustered at least 1.5 times more of their iterations compared to high-performers.

The above results show that high- and low-performing teams iterated differently. High-performers change their design more often than low-performing teams. Their iterations are often shallower than low-performers'. High-performers' also spread their iterations across the design process: they tend to iterate at a relatively steady pace. In contrast, low-performers tend to perform deep iterations in intermittent bursts. In our results, iterating behavior changed with the team's performance. This suggests how a team revises its design might indicate how its design performs.

5 DISCUSSION

5.1 Contributions

Iterations are common in design, yet their implications can be ambiguous. Our work aimed to resolve this ambiguity, relating iterating behavior of the design team to their design's performance. In particular, we focused on a team's micro-level iterations during a single design problem (e.g., Chusilp and Jin, 2006; Smith and Tjandra, 1998; Adams and Atman, 2000). This contrasts with scholars studying iterations across large design projects (e.g. Piccolo et al., 2019). Ultimately, we hope our insights into the relationship will allow us to create more precise heuristics to improve design performance.

Our exploratory study correlated iterating behavior to a team's design performance. We found that iteration patterns among the high-performing teams differed from those among low-performing ones. These results show one potential path to resolving the ambiguity of micro-level iteration: understanding how specific patterns of design actions connect to a design's performance.

This work also added nuance to the current understanding of micro-level iterations. Previously, a handful of studies had shown that the number of revisions—or, in some cases, transitions between problem-solving steps—is correlated to design quality (Atman et al., 1999; Adams and Atman, 2000). Our results concur with theirs: we showed that the number of iterations strongly differed between high- and low-performers. However, our study is the first to show the importance of the scale and cadence of these iterations. Though previous studies might have had similar variables in their models (e.g., Chusilp and Jin, 2006; Boudouh et al., 2006), they were not deemed as significant predictors of success. In our study, they are.

Third, we revealed this correlation through a highly detailed, time-series view of the design process. Iterations emphasize that design is a process. They remind us that a process- and/or time-dependent-view is crucial to understanding the patterns through which a design succeeds. Without a process perspective,

https://doi.org/10.1017/pds.2023.107 Published online by Cambridge University Press

we must rely on stochastic models to explain behavior (Huberman and Wilkinson, 2005), which overlooks the links between these constructs. Similarly, discarding information on the timing and sequencing of design actions overlooks characteristics that help explain *why* and *when* a design goes wrong. As such, a CAD platform provides the ideal research instrument through its built-in audit trail of the design process.

Finally, this exploratory study opens the door to work we intend to pursue. Specifically, further characterizing the (CAD) design process differences that signal performance and grounding them in the search and problem-solving literature (e.g., Cyert and March, 1963; Gavetti and Levinthal, 2000) To accomplish this work, future research will (collect and) map team deliberations before, during, and after a design iteration in addition to the time-series data described here.

5.2 Limitations

Our exploratory study also had its limits. First, teams in our context could not represent all designers, as they consisted of high school students. As such, though our findings could inform how we educate future engineers, we hesitate to generalize them broadly. Specifically, the relative differences might not generalize outside our context: for example, the best design teams might not always revise more than twice as much as the worst ones. However, scholars have shown that some design behaviors lead to successful performance while others do not (Lifshitz-Assaf et al., 2021). Likewise, we expect that high-performing teams will, generally, *exhibit different iteration patterns* than low-performing ones. However, the specifics of these patterns are not yet understood.

Second, teams in our context did not construct their designs, further narrowing our generalizations. Prototypes are a necessary component of the engineering design process; when developing hardware, designers use them to validate their (thus far virtual) design choices against the intended performance (Kossiakoff et al., 2011). Designers often iterate the design based on these outcomes. Our setting did not encompass validation or prototyping of the robots. As such, we could only describe the relationship between iteration and performance of the virtual designs. We intend to address this limitation in future studies.

Lastly, our data did not capture the teams' decision-making processes, preventing us from making causal links. Our data lacked the "why" behind each iteration. This limits our insight into what drove the characteristics and timing of these patterns: e.g., inexperience with the CAD platform, poor planning, combating churn, or issues with task allocation. Thus, our results are limited to linking iterating behavior to performance, not the decisions that drove that behavior. We also intend to address this limitation with additional data in future studies.

REFERENCES

Adams, R.S. and Atman, C. (2000), "Characterizing Engineering Student Design Processes An Illustration Of Iteration", in: *2000 Annual Conference*, pp. 5.143.1–5.143.11.

Alexander, C. (1964), *Notes on the Synthesis of Form*, Harvard University Press.

Atman, C.J., Chimka, J.R., Bursic, K.M. and Nachtmann, H.L. (1999), "A comparison of freshman and senior engineering design processes", *Design Studies*, Vol. 20 No. 2, pp. 131–152, http://doi.org/10.1016/S0142-694X(98)00031-3.

Baldwin, C.Y. and Clark, K.B. (2000), *Design Rules Vol. 1: The Power of Modularity.*, MIT Press, Cambridge.

Ballard, G. (2000), "Positive vs negative iteration in design", in: *8th Annual Conference of the International Group for Lean Construction*, Brighton, UK.

Boudouh, T., Anghel, D.C. and Garro, O. (2006), "Design Iterations in a Geographically Distributed Design Process", in: H.A. ElMaraghy and W.H. ElMaraghy (Editors), *Advances in Design*, Springer Series in Advanced Manufacturing, Springer, London, pp. 377–385, http://doi.org/10.1007/1-84628-210-1_31.

Catmull, E. and Wallace, A. (2014), *Creativity, Inc.: Overcoming the Unseen Forces That Stand in the Way of True Inspiration*, Random House, New York, first edition.

Chua, D.K.H. and Hossain, M.A. (2012), "Predicting Change Propagation and Impact on Design Schedule Due to External Changes", *IEEE Transactions on Engineering Management*, Vol. 59 No. 3, pp. 483–493, http://doi.org/10.1109/TEM.2011.2164082.

https://doi.org/10.1017/pds.2023.107 Published online by Cambridge University Press

Chusilp, P. and Jin, Y. (2006), “Impact of Mental Iteration on Concept Generation”, *Journal of Mechanical Design*, Vol. 128 No. 1, pp. 14–25, http://doi.org/10.1115/1.2118707.
Cyert, R.M. and March, J.G. (1963), “A behavioral theory of the firm”, in: *Organizational Behavior 2: Essential Theories of Process and Structure*, Vol. 2, M.E. Sharpe, Inc., Englewood Cliffs, NJ, fourth edition, pp. 169–187.
Deng, Y., Mueller, M., Rogers, C. and Olechowski, A. (2022), “The multi-user computer-aided design collaborative learning framework”, *Advanced Engineering Informatics*, Vol. 51, p. 101446, http://doi.org/10.1016/j.aei.2021.101446.
Gavetti, G. and Levinthal, D. (2000), “Looking Forward and Looking Backward: Cognitive and Experiential Search”, *Administrative Science Quarterly*, Vol. 45 No. 1, pp. 113–137, http://doi.org/10.2307/2666981.
Gopsill, J.A., Snider, C. and Hicks, B.J. (2019), “The emergent structures in digital engineering work: What can we learn from dynamic DSMs of near-identical systems design projects?”, *Design Science*, Vol. 5 No. e28, pp. 1–29, http://doi.org/10.1017/dsj.2019.20.
Haller, M., Lu, W., Stehn, L. and Jansson, G. (2015), “An indicator for superfluous iteration in offsite building design processes”, *Architectural Engineering and Design Management*, Vol. 11 No. 5, pp. 360–375, http://doi.org/10.1080/17452007.2014.937793.
Huberman, B.A. and Wilkinson, D.M. (2005), “Performance Variability and Project Dynamics”, *Computational & Mathematical Organization Theory*, Vol. 11 No. 4, pp. 307–332, http://doi.org/10.1007/s10588-005-5587-5.
Khanna, R., Guler, I. and Nerkar, A. (2016), “Fail Often, Fail Big, and Fail Fast? Learning from Small Failures and R&d Performance in the Pharmaceutical Industry”, *Academy of Management Journal*, Vol. 59 No. 2, pp. 436–459, http://doi.org/10.5465/amj.2013.1109.
Kossiakoff, A., Sweet, W.N., Seymour, S. and Biemer, S.M. (2011), *Systems Engineering Principles and Practice*, Wiley-Interscience, Hoboken, N.J, 2 edition edition.
Lifshitz-Assaf, H., Lebovitz, S. and Zalmanson, L. (2021), “Minimal and Adaptive Coordination: How Hackathons’ Projects Accelerate Innovation without Killing it”, *Academy of Management Journal*, Vol. 64 No. 3, pp. 684–715, http://doi.org/10.5465/amj.2017.0712.
Loch, C., Mihm, J. and Huchzermeier, A. (2003), “Concurrent Engineering and Design Oscillations in Complex Engineering Projects”, *Concurrent Engineering*, Vol. 11 No. 3, pp. 187–199, http://doi.org/10.1177/106329303038030.
Loch, C.H. and Terwiesch, C. (2005), “Rush and Be Wrong or Wait and Be Late? A Model of Information in Collaborative Processes”, *Production and Operations Management*, Vol. 14 No. 3, pp. 331–343, http://doi.org/10.1111/j.1937-5956.2005.tb00028.x.
Lottaz, C., Clément, D.E., Faltings, B.V. and Smith, I.F.C. (1999), “Constraint-Based Support for Collaboration in Design and Construction”, *Journal of Computing in Civil Engineering*, Vol. 13 No. 1, pp. 23–35, http://doi.org/10.1061/(ASCE)0887-3801(1999)13:1(23).
Love, P.E.D. (2002), “Influence of Project Type and Procurement Method on Rework Costs in Building Construction Projects”, *Journal of Construction Engineering and Management*, Vol. 128 No. 1, pp. 18–29, http://doi.org/10.1061/(ASCE)0733-9364(2002)128:1(18).
Maier, A.M. and Störrle, H. (2011), “What are the Characteristics of Engineering Design Processes?”, in: *Proceedings of the 18th International Conference on Engineering Design*, Copenhagen, Denmark, pp. 188–198.
Maier, M. and Rechtin, E. (2000), *The Art of Systems Architecting*, CRC Press, Boca Raton, FL, second edition.
McGrath, R.G. (2011), “Failing by design”, *Harvard Business Review*, Vol. 89 No. 4, pp. 76–83.
Mihm, J., Loch, C. and Huchzermeier, A. (2003), “Problem–Solving Oscillations in Complex Engineering Projects”, *Management Science*, Vol. 49 No. 6, pp. 733–750, http://doi.org/10.1287/mnsc.49.6.733.16021.
Nourimand, A. and Olechowski, A. (2020), “Prominence of Conceptual Design with Computer-Aided Design Tools for Junior and Senior Product Designers”, in: *2020 ASEE Virtual Annual Conference Content Access*, Virtual Online, http://doi.org/10.18260/1-2--35101.
Pektaş, Ş.T. and Pultar, M. (2006), “Modelling detailed information flows in building design with the parameter-based design structure matrix”, *Design Studies*, Vol. 27 No. 1, pp. 99–122, http://doi.org/10.1016/j.destud.2005.07.004.
Piccolo, S.A., Maier, A.M., Lehmann, S. and McMahon, C.A. (2019), “Iterations as the result of social and technical factors: Empirical evidence from a large-scale design project”, *Research in Engineering Design*, Vol. 30 No. 2, pp. 251–270, http://doi.org/10.1007/s00163-018-0301-z.
Reingold, J. (2005), “Hondas in Space”, *Fast Company*.
Simon, H.A. (1962), “The Architecture of Complexity”, *Proceedings of the American Philosophical Society*, Vol. 106 No. 6, pp. 467–482.
Smith, R.P. and Tjandra, P. (1998), “Experimental observation of iteration in engineering design”, *Research in Engineering Design*, Vol. 10 No. 2, pp. 107–117, http://doi.org/10.1007/BF01616691.

Szajnfarber, Z., Zhang, L., Mukherjee, S., Crusan, J., Hennig, A. and Vrolijk, A. (2020), "Who Is in the Crowd? Characterizing the Capabilities of Prize Competition Competitors", *IEEE Transactions on Engineering Management*, Vol. 69 No. 4, pp. 1–15, http://doi.org/10.1109/TEM.2020.2991370.

Taylor, T. and Ford, D.N. (2006), "Tipping point failure and robustness in single development projects", *System Dynamics Review*, Vol. 22 No. 1, pp. 51–71, http://doi.org/10.1002/sdr.330.

von Hippel, E. (1994), "Sticky Information and the Locus of Problem Solving: Implications for Innovation", *Management Science*, Vol. 40 No. 4, pp. 429–439, http://doi.org/10.1287/mnsc.40.4.429.

Vrolijk, A. and Szajnfarber, Z. (2023), "The Opportunists in Innovation Contests", *Research-Technology Management*, Vol. 66 No. 1, pp. 30–40, http://doi.org/10.1080/08956308.2022.2132771.

Wynn, D.C. and Eckert, C.M. (2017), "Perspectives on iteration in design and development", *Research in Engineering Design*, Vol. 28 No. 2, pp. 153–184, http://doi.org/10.1007/s00163-016-0226-3.

Yassine, A. and Braha, D. (2003), "Complex Concurrent Engineering and the Design Structure Matrix Method", *Concurrent Engineering*, Vol. 11 No. 3, pp. 165–176, http://doi.org/10.1177/106329303034503.

Yassine, A., Joglekar, N., Braha, D., Eppinger, S. and Whitney, D. (2003), "Information hiding in product development: The design churn effect", *Research in Engineering Design*, Vol. 14 No. 3, pp. 145–161, http://doi.org/10.1007/s00163-003-0036-2.

Zhang, H.Z., Xie, C. and Nourian, S. (2018), "Are their designs iterative or fixated? Investigating design patterns from student digital footprints in computer-aided design software", *International Journal of Technology and Design Education*, Vol. 28 No. 3, pp. 819–841, http://doi.org/10.1007/s10798-017-9408-1.