CHALMERS
UNIVERSITY OF TECHNOLOGY

Continuous Experimentation and Human Factors: An Exploratory Study

Continuous Experimentation and Human Factors
An Exploratory Study

Amna Pir Muhammad[1][0000−0001−8328−4149], Eric Knauss[1][0000−0002−6631−872X], Jonas Bärgman[2][0000−0002−3578−2546], and Alessia Knauss[3][0000−0003−4857−7784]

[1] Dept. of Computer Science and Eng., Chalmers | University of Gothenburg, Gothenburg, Sweden
[2] Dept. of Mechanics and Maritime Sciences, Chalmers University of Technology, Gothenburg, Sweden
[3] Zenseact AB, Gothenburg, Sweden

Abstract. In today's rapidly evolving technological landscape, the success of tools and systems relies heavily on their ability to meet the needs and expectations of users. User-centered design approaches, with a focus on human factors, have gained increasing attention as they prioritize the human element in the development process. With the increasing complexity of software-based systems, companies are adopting agile development methodologies and emphasizing continuous software experimentation. However, there is limited knowledge on how to effectively execute continuous experimentation with respect to human factors within this context. This research paper presents an exploratory qualitative study for integrating human factors in continuous experimentation, aiming to uncover distinctive characteristics of human factors and continuous software experiments, practical challenges for integrating human factors in continuous software experiments, and best practices associated with the management of continuous human factors experimentation.

Keywords: Continuous Experimentation · Human Factors · Human Factors Experiments · Continuous Human Factors Experimentation

1 Introduction

In today's fast-paced software development environments, characterized by competitive and unpredictable markets, there is a need to deliver and improve products rapidly [31]. This urgency is intensified by complex customer requirements and rapid technological advancements. Consequently, many software companies have embraced or are transitioning toward continuous experimentation [25, 35].

Continuous software experimentation [4] involves iteratively gathering user feedback and observing user interactions [6]. With the growing significance of software in complex and automated systems, continuous experimentation has become increasingly prevalent across various industries. These systems require robust and continuously evolving software [19]. Researchers have acknowledged

[4] Key terms of this study are defined in Table 1.

that the design for such systems is inherently complex and that a more comprehensive understanding of the real world can be achieved by actively looking at the system from a human factors perspective and not only a technical perspective [2, 11].

In order to ensure the effectiveness, safety, and reliability of systems, particularly complex software systems, it is desirable to provide more holistic knowledge on human factors in continuous experimentation. Especially for safety-critical systems, a human factors perspective may provide crucial in-depth insights. Therefore, integrating human factors experimentation into the continuous experimentation process promises to be a game changer [16, 30]. Human factors refer to the various aspects of individuals, including their physical, cognitive, social, and emotional elements, all of which can significantly influence their performance and interactions with systems [12]. Human factors experiments prioritize studying user behavior and involve experiments with humans as participants [8]. We acknowledge that the concepts of continuous software experimentation and human factors experiments overlap to some extent (i.e., the latter can be a component of the former, and vice versa), but in this study, we discuss them as separate entities as they come from different domains and are likely to complement each other. However, to understand whether HF experiments fit the continuous software experiment practices, one needs to understand in detail where they differ, where they overlap, and in what they can be integrated.

While the significance of human factors has been widely recognized [24, 32] and continuous software experimentation methodologies are widespread in industry and have received extensive research attention [7, 13], there remains a research gap when it comes to incorporating human factors experiments into the well established continuous software experimentation processes [30]. Consequently, further investigation is required to bridge this gap [22].

This research aims to address differences, associated challenges, and best practices for integration of human factors experiments within the context of continuous experimentation. The following research questions (RQs) are used to guide our research:

RQ1: What are main differences when comparing human factors experiments with continuous software experimentation?
RQ2: What are main practical challenges when managing human factors experiments in continuous software experimentation?
RQ3: What are best practices for managing human factors in continuous experimentation?

The findings for RQ1 reveal that while both human factors and software experimentation emphasize the significance of understanding user behavior and needs, they differ in their approach. RQ2 highlights the challenges in managing human factors experiments, pointing to complexities like GDPR compliance, data collection issues, additional costs, and an industry scarcity of experts. RQ3 focuses on best practices in this domain, emphasizing the need to prioritize research based on product timelines, invest in actionable metrics, maintain robust

experimental infrastructure and documentation, and including or transferring human factors knowledge.

The rest of this paper is structured as follows: We start with an overview of definitions for key terms used in this paper in Section 2, which covers background knowledge and related work as well. Section 3 presents the research methodology, and Section 4 outlines the findings. Section 5 presents the discussion and potential threats to validity. Finally, Section 6 concludes the paper.

2 Background and related work

Key terms of this study can be interpreted differently depending on the domain. Hence, for the scope of this study, we use the definitions provided in Table 1.

Table 1: Definitions of key terms used in this study

Term	**Definition**
Continuous (software) experimentation	An approach to support software development, where research and development activities are guided by iteratively conducting experiments, collecting user feedback, and observing the interaction of users with the system or services under development. The goal of continuous software experimentation is to evaluate features, assess risks, and drive evolution [6, 13, 35].
Human factors in development	The field that aims to inform developers by providing fundamental knowledge about human capabilities and limitations throughout the design cycle so that products will meet specific quality objectives. These capabilities and limitations include cognitive, physical, behavioral, psychological, social, effective, and motivational aspects [12, 21].
Human factors experiments	Investigations that focus on how human capabilities and limitations affect specific quality objectives during the interaction between humans and the system, service, or product under development. Thus, humans are part of human factors experiments and their behavior and perception/opinions (of, e.g., the system, service, or product under assessment) can impact the result and consequently the design of the system [9, 28].
Continuous human factors experimentation	An iterative approach in software development that evaluates how human capabilities and limitations impact specific quality objectives during user interactions. It involves ongoing experiments, user feedback, and observations to inform the design process and enhance user experience.

Continuous Software Experimentation. Agile development methodologies have gained widespread popularity in software development due to their iterative and collaborative nature [1]. These methodologies emphasize continuous experimentation, which involves constantly testing and validating hypotheses to make data-driven decisions throughout development [35]. This approach has proven effective in optimizing software products and services.

Continuous experimentation is primarily applied in web-based systems, allowing developers to analyze and deploy changes based on real-world data and user preferences, rather than relying solely on simulations or the opinion of the highest-paid person's opinion (HiPPO) [14]. Leading technology companies like Microsoft, Google, Facebook, and Booking.com utilize online controlled experiments, also known as A/B tests, to evaluate the impact of changes made to their software products and services [5, 7, 13].

Despite the numerous advantages of wide-ranging continuous software experimentation, there are still several challenges that need to be addressed during its implementation. Some of the major hurdles include cultural shifts within development teams, slow development cycles, product instrumentation, and the identification of appropriate metrics for measuring user experience [15, 17]. Rissanen and Münch [26] confirmed these challenges and also found that capturing and transferring user data becomes challenging due to legal agreements.

Human Factors and Experimentation. By including human factors experiments from the outset, it becomes possible to ensure system reliability and evaluate the system considering real-world human constraints [28]. Human factors experiments aim to understand how people interact with technology, products, and systems to optimize usability, user experience, and overall performance [12]. They commonly evaluate aspects such as user interface design, cognitive workload, situation awareness, and user behavior [10, 27, 29, 33].

Continuous Human Factors Experimentation. In terms of testing and experiments, there have been some initial efforts to integrate usability testing and user-centered design practices into agile development, like for example the approach proposed by Nakao et al. [23] to incorporate usability testing throughout the agile development process. Despite these efforts, research has emphasized the need for new processes and tools that empower practitioners of human factors to promote usable and effective products in the agile development environment [30] and the integration of human factors into the well established continuous software experimentation practices used in agile development [22].

Note that our research does not center around the impact of human factors on employees or developers involved in the development processes, as mentioned in [34]. Instead, our focus is primarily on the product itself. By conducting and analyzing a series of semi-structured interviews, we aim to explore the integration of human factors experiments within the context of continuous experimentation in software development.

3 Methodology

Sampling: We conducted interviews with eight professionals (P1-P8). We aimed for a broad sample of expert participants with high experience in human factors,

continuous experimentation, ideally in both fields. This criteria however limits the number of available subjects. Thus, we accepted lower participant numbers than initially planned and focused on interviewing a smaller selection of leading experts in their respective fields for this exploratory study.

We focused on recruiting industry participants from renowned organizations such as Microsoft. Targeting those known for their impactful success stories, to ensure a significant impact and obtain high-quality input. Our academic interviewees have extensive experience collaborating closely with industry, and their credentials include thousands of citations (h-index > 35 in four cases), providing them with a good overview of practices in the field that supports our exploratory study goal.

Table 2 presents each participant's role and experience level.

Data Collection: To gather comprehensive information for our study, we used a qualitative study design inspired by Maxwell [20]. Our data collection involved conducting a series of semi-structured interviews, following a predefined set of open-ended questions while allowing flexibility to include additional follow-up questions when necessary. The interview questions used can be found here.

The interviews were conducted online through Zoom, with each session lasting around one hour. We obtained permission from the interviewees to record the sessions, which we later transcribed and anonymized for analysis.

The interview questions were organized into three main categories. The first set aimed to collect demographic information from the interviewees, as well as confirming their experience working with continuous experimentation and human factors. The second set focused on exploring the management of experimentation in both software and human factors contexts. We used these question to get a better understanding of the participants background, how and which experiments they use and generally of the topic under study. Finally, we asked specific questions related to human factors in continuous experimentation. We used the entire data in our analysis and to answer our research questions.

Table 2: Interviewees' roles and relevant work experience (Experience level: Low= 0–5 years, Medium=5–10 years, High= More than 10 years).

ID	**Role**	**Main Domain**	**Continuous experimentat. experience**	**Human Factors experience**
P1	SE Researcher	Academia	High	Low
P2	Human Factors Researcher	Academia	Low	High
P3	Human factors Engineer	Industry	High	High
P4	UX Expert	Industry	High	High
P5	Data Scientist	Industry	High	Low
P6	SE/Human Factors Researcher	Academia	High	High
P7	CS Researcher and IT Consultant	Industry & Academia	High	High
P8	Human Factors Researcher	Academia	Low	High

We initiated each interview by providing a brief overview of the study to establish a shared understanding and create a comfortable environment. We also presented the basic terms and definitions relevant to the study topic, seeking agreement from the interviewees. This approach aimed to establish a common foundation for our discussions, minimize potential confusion, and ensure a consistent standpoint when gathering participants' perspectives. Notably, all participants expressed agreement with our definitions, offering no suggestions for improvement or indicating any discrepancies between their own understanding and our proposed definitions as outlined in Table 1.

Data Analysis: For the qualitative analysis, we employed the thematic analysis approach [4] to identify themes and analyze the content. This approach consists of six key steps. Initially, we comprehensively reviewed all the interview notes and generated research-related memos. To facilitate the process, we employed Nvivo initially and later transitioned to using the Miro board for enhanced visualization. These tools allowed us to assign codes or labels to the text. Through an iterative process, we refined the coding scheme to uncover significant ideas and viewpoints. The codes were then analyzed and grouped together to identify common patterns, thereby defining the themes. Subsequently, we thoroughly reviewed and verified the themes that emerged from the coding process, ensuring clarity, consistency, and addressing any ambiguities, contradictions, or omissions.

4 Findings

We present our findings for each research question with primary themes and their related sub-themes. Figure 1 gives an overview of the main themes.

RQ1: What are main differences when comparing human factors experiments with continuous software experimentation?

F1.1: Contextual Factors

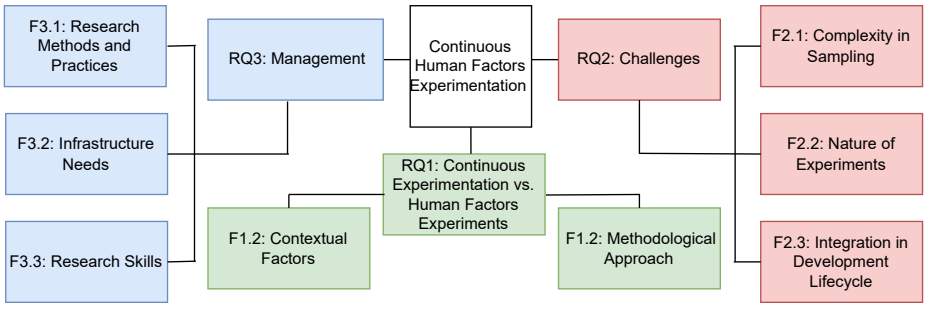

Fig. 1: Overview of key high-level themes identified from the interview analyses.

Human Behavior vs. Technical Aspects: Both software developers and human factors professionals recognize the importance of an intuitive user perspective. They acknowledge that users have varying levels of technical proficiency and may not be inclined to explore complex features. However, human factors experts go a step further by emphasizing the need to understand the underlying reasons for potential user challenges. For example, these challenges could include over-trusting software or avoiding it altogether due to fear or apprehension. To address these concerns, human factors experiments are conducted to gain insights into human behaviors, needs, and experiences. These experiments prioritize the user perspective and strive to optimize user satisfaction and safety. On the other hand, software experiments typically have a more technical development-centric focus. This discrepancy in approach highlights the importance of adopting a human-centric understanding of user behavior and needs, which may differ from the primary focus of developers on technical functionalities.

> *"They can develop and test and design and maybe it doesn't need to involve human, then it works fine, as soon as you add human, a whole set of questions & requirements come into place which needs to be considered."* — P8

Human factors experts primarily focus on observing and analyzing human behaviors to collect data using different interaction metrics. Such an environment poses inherent challenges due to numerous uncontrollable variables at play. For instance, humans exhibit a learning effect that can significantly impact the experimental results. Moreover, interpersonal communication and feedback loops among participants may also influence their responses to the experiments.

Conversely, continuous software experimentation primarily focuses on monitoring system behavior rather than directly observing human behavior. Such experiments collect data from performance indicators, system logs, issue reports, or user interactions documented by the software. They are often conducted under controlled conditions, emphasizing variables like reaction time, resource usage, scalability, or software stability. We believe that these differences are brought to a point by the following exemplary quote:

> *"The main difference between human factor and traditional experiments, for instance, is that humans have much more of a learning effect."* — P7

F1.2: Methodological Approach

Diverse Approaches in Experimentation: The methodology for both human factors and continuous software experiments varies depending on the nature and scope of the feature being tested. Various techniques can be employed for both software and human factors experiments.

> *"If it's a very small audience, then product teams can also choose actually to do some surveys and interviews they invite customers in. So it really depends on like what is the scope of the feature that you're testing."* — P5

While some methodologies, such as surveys and interviews, can be utilized for both software and human factors experiments, there are some notable differences in how the results are analyzed and interpreted. We found that while A/B experimentation is a dominant method in continuous software experimentation, it is often only one of many methods used in human factors experiments.

Qualitative and Quantitative: Much like software experiments, human factors involve qualitative and quantitative data analysis. However, the analysis of human factors experiments leans more towards qualitative methods due to the complexity of measuring and interpreting human behavior. Therefore, conducting effective human factors experiments necessitates practitioners with a strong foundation in qualitative methodologies and empirical work involving human participants. Such practitioners are able to capture the rich and nuanced aspects of human behavior and user experience. In contrast, continuous software experiments often adopt a more quantitative approach, aiming to establish causal relationships between independent and dependent variables, allowing for statistical analysis. That said, a substantial part of human factors experiments still involve collecting quantitative data, such as eye-tracking data and performance data (e.g., in the automotive domain in terms of measures of lane keeping, time gaps, etc., or task completion times considering desktop software tools).

> *"If you have a background in quantitative experiments with technical systems, I would think you cannot do [human factors experiments] in a good way. You need some kind of background in doing empirical work with humans."* — P6

RQ2: What are main practical challenges when managing human factors experiments in continuous software experimentation?

F2.1: Complexity in Sampling

Controlled vs. Uncontrolled Variables: One aspect is the presence of a higher amount of uncontrolled variables in human factors experiments. Numerous contextual factors cannot be fully controlled, which poses challenges in ensuring comparability and measuring variables. Lack of control over contextual factors also complicates the analysis, as there may be numerous variables that cannot be fully controlled or accounted for in the experiment.

> *"The other issue is control. I think you will look at situations where there are just a lot of context factors, there is just no way to control everything."* — P1

Statistical Analysis: One challenge lies in the statistical analysis of the data. In certain cases, conducting a rigorous statistical analysis may not be feasible due to the nature of the human factors experiment. For instance, the research goal might involve observing how people react in a particular situation without quantifiable metrics, so conducting a traditional statistical analysis becomes challenging.

> *"It might not be possible to do a proper statistical analysis because you might want to expose people to a certain situation and see what happens."* — P2

Participant Scarcity: Another challenge in human factors experiments is the limited availability of participants. Getting enough people to participate can be difficult, and the scarcity of eligible participants further complicates the process. In contrast, continuous software experiments, especially those conducted online, can be performed on a larger scale. While involving as many participants as possible is generally advised, practical limitations may hinder this goal.

> *"Often these studies are fairly small regarding the number of subjects."* — P6

F2.2: Nature of Experiments

Personal Information and GDPR Issues: When conducting experiments, the collection of personal information can be crucial for understanding human behavior and software performance. In experimental research, collecting personal information is pivotal for understanding both human behavior and software performance. This is particularly evident in human factors experiments, where insights into how individuals from varied backgrounds interact with technology are essential. However, collecting this in-depth personal information presents challenges, mainly due to privacy and ethical issues. The requirements of GDPR regulations amplify these concerns, necessitating meticulous attention. While software experiments might occasionally need such information, the emphasis is much greater in human factors experiments.

> *"It is a bit hard. Like with the GDPR and everything. How to store stuff actually? It makes it a bit more complicated."* — P4

Prototype vs. Real Environment: Our interviewees mentioned that, although experiments are typically carried out using prototypes or simulators, human factors experts also advocate for conducting experiments in the actual environment where the product will be finally be used. Experiments conducted in real environments offer a more realistic and authentic representation of how participants interact with the product or system in their natural settings. Unlike prototype experiments, where external factors can be tightly controlled, real environment experiments expose participants to multiple variables and contextual factors that can significantly impact human performance and behavior.

> *"Having design prototypes is one approach so that people get the vision behind. But testing in real cars, it makes it so difficult, which is, but also important, to go in that direction or to get more research done."* — P3

Expensive: Human factors experiments are often perceived as more costly compared to continuous software experiments. This perception stems from the direct involvement of real humans participating in real-time scenarios. For instance, experts in human factors often need to recruit participants for their studies, compensating them for their time and effort, which can be a significant expense. On the other hand, many continuous software experiments can gather data online, reducing the need for physical presence and direct human interaction, and direct

payment. While continuous software experiments do have associated costs—such as development, deployment, and server infrastructure—these expenses are generally lower than those of human factors experiments.

> *"We have to pay for this for facilities, we have to pay participants because we get people from the real world, and the preparations is quite prolonged."* — P8

F2.3: Integration in Development Lifecycle

Execution Time: Managing and executing human factors experiments in agile development can be challenging due to their inherent time-consuming nature. Unlike continuous software experiments that typically run for at least a week, human factors experiments often require more time to obtain meaningful results. The duration of such experiments is influenced by the desired change in a metric being measured. Obtaining timely results from human factors, that can be integrated into ongoing projects without significant delays can be difficult, especially in agile, short sprint-based, work flows.

> *"You do a sprint and then you need results to run it and assume you need these kind of results quickly. So not in three months. And that's, I would say that's the problem for integrating these kind of things."* — P2

Infrastructure Needs: One challenge involves obtaining the necessary tools and setup to conduct the desired tests. Ensuring that the basic infrastructure is in place to facilitate the experiments can be a significant hurdle.

> *"If there's getting the right tools and right setup, like the basics in place to even be able to test what you wanna test. That could be a challenge."* — P4

Too Few Human Factors Experts: Many companies struggle with insufficient human factors expertise and limited resources, which can hinder their ability to improve user experience. This deficiency often leads to a few outliers (or even the development team itself) having a disproportionate impact on the final product design. This concern arises from the fact that there are too few human factors experts available, which limits comprehensive evaluations and increases the risk of biased results.

> *"So I think that's what, what other companies are lacking actually: Enough human factors, people doing that kind of work."* — P3

Lack of Motivation: Another challenge is that many individuals with a technical mindset often overlook the importance of understanding human behaviors. This lack of motivation can hinder the collection of relevant data and make it difficult to address the complexities involved in studying human subjects.

> *"How can you influence people? I think that's the number one thing."* — P1

RQ3: What are best practices for managing human factors in continuous experimentation?

F3.1: Research Methods and Practices

Prioritizing Hypotheses/Research Questions: Prioritizing research questions and hypotheses based on the product timetable and development sprints is a crucial aspect in agile development. By identifying the experiments that have the most impact on design decisions and user experience improvement, organizations can allocate resources efficiently and gain valuable insights.

> *"The number of experiments that you can do is basically infinite. So the hardest part in running experiments is how do I prioritize running the most valuable experiments first. And, I think that's where many companies struggle."* — P7

Metrics and Measurement Instrumentation: Based on our interviewees, to enable informed decision-making it is essential to invest in the development of meaningful metrics that align with the desired outcomes. While simple interaction metrics like clicks or selections are useful, it is important to go beyond them and capture success metrics related to user sessions and product features. As one of our interviewees pointed out, the value of experiments ultimately relies on having good metrics and making significant investments in their development. Without such metrics, experiments become less valuable as they fail to provide actionable results for decision-making. It was also emphasised that developing and validating such metrics can (and must be allowed to) take substantial time.

Another critical aspect is the measurement of various metrics that provide insights into different aspects of the product under evaluation. It is worth noting that interviewees stressed the significance of using proper measurement methods to obtain valuable results for making informed decisions. To measure different aspects of the product or system being evaluated, multiple metrics should be computed simultaneously. These metrics should align with the goals of the experiment and help determine what is reasonable to measure and what constitutes a good outcome.

> *"But at the end of the day about experiments, it all boils down to metrics. If you don't have good metrics and you don't invest significantly into metrics, your experiments will not be valuable."* — P5

Results and Lessons Learned: When determining whether to reuse or evolve experiments, the organization may take several factors into consideration. These factors include the importance of the findings, potential influencing factors, and information indicating changes in the validity of previous results. The relevance of the results and their impact on decision-making are carefully evaluated when planning subsequent experiments. It was also mentioned that the decision to reuse experiments is often driven by the interest and initiative of individuals involved in the projects, rather than being a formalized process.

> *"There are sometimes factors that are influencing what's factors that may confound the outcomes from one experiment such that we need to rerun it in order to make sure that the thing is still true."* — P7

F3.2: Infrastructure Needs

Experimental Setup: The infrastructure should support the setup and integration of different components required for the experiment. This includes ensuring that the necessary tools and setups are in place to conduct the experiments effectively. It may involve creating prototypes, simulating scenarios, or integrating various hardware and software components to enable the desired testing environment. Careful planning of the experiment is crucial.

> *"If there's getting the right tools and getting the right setup or the right HMI, like the basics in place even to be able to test what you wanna test."* — P4

Traceability and Documentation: Maintaining traceability and documentation throughout the experimental process is important. This includes preserving initial design proposals that led to the ideas being tested. Having a clear traceability trail helps in understanding the decision-making process during the experiment and provides valuable insights for product teams. Utilizing an experimentation platform that incorporates this traceability is essential.

> *"So having some traceability on the decisions that led to what is being tested would be very helpful, I think, for product teams. And that should be part of the experimentation platform."* — P5

Collaboration and Management Support: Our interviewees highlighted that infrastructure should facilitate collaboration among different teams involved in the experiments. It should provide a platform for coordinating activities, managing participants, and ensuring the smooth execution of the experiments. Additionally, management buy-in, support, and drive are also important factors to overcome obstacles and successfully implement the infrastructure needed for human factors experiments.

> *"Main obstacle is kind of like management, high management buy-in, and support and then like knowledge on how to design and collect it. So, to me, infrastructure would be something they [practitioners] would know how to solve that."* — P6

F3.3: Research skills

Roles and Responsibilities: Our findings indicate that experiment management becomes a collaborative effort within cross-functional teams in an agile environment. These teams typically include data scientists, engineers, product managers, program managers, and user researchers. Our findings also highlight the pivotal role of data scientists in continuous software experiments and the need for technical support from engineers in human factors experiments. Moreover, considering a single role for responsibility, product managers are crucial in deciding which experiments to run and ensuring that relevant metrics are effectively measured. We learned that while the responsibility for managing continuous

human factors experiments can be shared within a team or primarily held by the manager, it is crucial to recognize that specialized knowledge and expertise are often necessary. Having human factors specialists in human factors experimentation can greatly benefit the planning and management of human factors experiments. Human factors specialists bring specialized knowledge and expertise in research methodology, data analysis, and experimental design to guide the team and ensure precise and accurate experiments.

> *"I really think that it should be less of a single responsibility and more of a team responsibility."* — P7

Knowledge and Training: A solid foundation of knowledge, theory, and models is essential to design and evolve effective human factors experiments. Furthermore, establishing an infrastructure to disseminate this knowledge and provide comprehensive training to researchers and teams is crucial. Agile teams can conduct human factors experiments with appropriate training and methodologies.

> *"A bit with training. If you follow a specific procedure, then I think it's not a problem."* — P4

The training should cover experimental design, research methodology, human factors principles, biases, usability evaluation methods, and research methods. Although individuals inherently possess some understanding of human behavior, training will help broaden their perspective.

5 Discussion

Continuous experimentation for web-based systems has received extensive research attention [7, 13], however, the human factors aspect remains relatively underexplored. This study explores the idea to bridge this gap by discussing the integration of human factors experiments with continuous experimentation. This promises to enable continuous experimentation even in the domain of safety critical systems to a larger extent. Integrating human factors experiments into continuous experimentation presents both benefits and challenges [18]. For instance, these experiments can shed light on usability, user experience, and decision-making [28]. Yet, they also pose challenges, such as the need to execute experiments in real environments with real human participants [3].

We confirm challenges highlighted by previous studies [15, 17, 26] that have investigated challenges in continuous experimentation in general (e.g., cultural shifts and appropriate identification of metrics) also for the integration of human factors into continuous experimentation. On top of that, our findings introduce additional complexities when human factors are integrated into the mix.

Moreover, our findings indicate that the integration of human factors in continuous experimentation is currently lacking. One of the contributing factors to this gap is the shortage of human factors experts available to collaborate with teams engaged in continuous experimentation [21]. While these teams conduct experiments tailored to their specific system components, they often lack input

from human factors specialists. Another factor is the usually higher complexity of human factors experiments. On the fast pace of continuous experimentation, this affects options for data collection and appears to cause human factors experiments leaning towards qualitative data collection in this context.

To effectively integrate human factors experiments into continuous experimentation, companies should consider including human factors experts within teams and raising awareness among developers about the importance of incorporating human factors. The successful execution of human factors experiments by teams requires developers to be skilled in empirical study methods, enabling them to conduct impactful human factors experiments.

Threats to validity: The interdisciplinary nature and vast scope of the fields involved introduces a threat to **Construct Validity** in that various definitions exist for the same terms, such as "human factors". Consequently, different individuals may have different interpretations. We have included clear definitions of the key concepts in interviews and report to mitigate this threat and ensure a common understanding of the fundamental concepts used in this study. Additionally, experienced authors were involved in the study to address the risk of construct validity. Their expertise assisted the first author in developing an interview guide that effectively aligned with the study's research objectives. For **Internal Validity**, we implemented measures to reduce bias and confounding variables, such as having multiple authors conduct each interview to minimize personal bias. Due to the specialized scope and high demands on participant expertise (human factors and continuous experimentation), we had to rely on convenience sampling, taking into account both the profile and availability of potential subjects. Consequently, the low number of participants introduces a threat to **External Validity**. We aimed to mitigate this threat by aiming for covering a wide range of roles, domains, and cultural backgrounds. Finally, to ensure **Reliability,** we implemented various measures. Throughout the interviews, we had multiple researchers present to enhance the reliability of our data. Additionally, we provided used materials and a detailed analysis process, enabling other researchers to replicate our methodology in diverse contexts. Moreover, the authors actively engaged in discussions to maintain consistency in the coding results. However, despite our efforts, we acknowledge the possibility of some subjectivity in our analysis.

6 Conclusion

This qualitative exploratory study investigates the integration of human factors with continuous experimentation. To effectively integrate human factors experiments in continuous experimentation, there's a pressing need for upgraded infrastructure, improved developers' awareness about the importance of human factors, and training developers in empirical study methods essential for effective human-centric experimentation.

By fostering interdisciplinary collaboration and promoting the integration of human factors considerations into continuous experimentation, organizations can enhance the user experience, and improve the quality of software and systems.

Future research should focus on developing frameworks and detailed guidelines for effectively incorporating human factors into continuous experimentation processes, leading to the creation of more user-centric, safe, and acceptable systems.

Acknowledgements: The authors express their gratitude to the interviewees for their valuable time and insights. The project has received funding from the Marie Skłodowska-Curie grant agreement 860410 under the European Union's Horizon 2020 research and innovation program.

References

1. Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J.: Agile software development methods: Review and analysis. arXiv preprint arXiv:1709.08439 (2017)
2. Boy, G.A.: Human-centered design of complex systems: An experience-based approach. Design Science **3**, e8 (2017)
3. Charlton, S.G., O'Brien, T.G.: Handbook of human factors testing and evaluation. CRC Press (2019)
4. Clarke, V., Braun, V., Hayfield, N.: Thematic analysis. Qualitative psychology: A practical guide to research methods **3**, 222–248 (2015)
5. Fabijan, A., Dmitriev, P., Olsson, H.H., Bosch, J.: The evolution of continuous experimentation in software product development: from data to a data-driven organization at scale. In: 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE). pp. 770–780. IEEE (2017)
6. Fagerholm, F., Guinea, A.S., Mäenpää, H., Münch, J.: The right model for continuous experimentation. Journal of Systems and Software **123**, 292–305 (2017)
7. Feitelson, D.G., Frachtenberg, E., Beck, K.L.: Development and deployment at facebook. IEEE Internet Computing **17**(4), 8–17 (2013)
8. Franklin, A.D.: What makes a'good'experiment? The British Journal for the Philosophy of Science **32**(4), 367–374 (1981)
9. Gandevia, S.: A human factor in 'good'experiments. The British Journal for the Philosophy of Science **37**(4), 463–466 (1986)
10. Hancock, P., Caird, J.K.: Experimental evaluation of a model of mental workload. Human factors **35**(3), 413–429 (1993)
11. Hancock, P.A.: Some pitfalls in the promises of automated and autonomous vehicles. Ergonomics **62**(4), 479–495 (2019)
12. Human Factors and Ergonomics Society: Definitions of human factors and ergonomics (2023), `https://www.hfes.org/About-HFES/What-is-Human-Factors-and-Ergonomics`, accessed on 17 Feb, 2023
13. Kevic, K., Murphy, B., Williams, L., Beckmann, J.: Characterizing experimentation in continuous deployment: a case study on bing. In: 39th International Conference on Software Engineering:(ICSE-SEIP). IEEE (2017)
14. Kohavi, R., Henne, R.M., Sommerfield, D.: Practical guide to controlled experiments on the web: listen to your customers not to the hippo. In: Proceedings of the 13th ACM SIGKDD. pp. 959–967 (2007)
15. Kohavi, R., Crook, T., Longbotham, R., Frasca, B., Henne, R., Ferres, J.L., Melamed, T.: Online experimentation at microsoft. Data Mining Case Studies **11**(2009), 39 (2009)
16. Lee, J.D., Seppelt, B.D.: Human factors in automation design. Springer handbook of automation pp. 417–436 (2009)

17. Lindgren, E., Münch, J.: Software development as an experiment system: A qualitative survey on the state of the practice. In: Agile Processes in Software Engineering and Extreme Programming: 16th International Conference, XP. Springer (2015)
18. Madni, A.M.: Integrating humans with and within complex systems. CrossTalk **5** (2011)
19. Maruping, L.M., Matook, S.: The evolution of software development orchestration: current state and an agenda for future research. European Journal of Information Systems **29**(5), 443–457 (2020)
20. Maxwell, J.A.: Qualitative research design: An interactive approach. Sage publications (2012)
21. Muhammad, A.P., Knauss, E., Bärgman, J.: Human factors in developing automated vehicles: A requirements engineering perspective. Journal of Systems and Software p. 111810 (2023)
22. Muhammad, A.P., Knauss, E., Bärgman, J., Knauss, A.: Towards challenges and practices with managing human factors in automated vehicle development. In: 31st IEEE International Requirements Engineering Conference, (RE'23). IEEE (2023)
23. Nakao, Y., Moriguchi, M., Noda, H.: Using agile software development methods to support human-centered design. NEC Technical Journal **8**(3), 37–40 (2014)
24. Norman, D.: Design of everyday things, revised and expanded: Basic books (2013)
25. Olsson, H.H., Alahyari, H., Bosch, J.: Climbing the "stairway to heaven"–a mulitiple-case study exploring barriers in the transition from agile development towards continuous deployment of software. In: 2012 38th euromicro conference on software engineering and advanced applications. pp. 392–399. IEEE (2012)
26. Rissanen, O., Münch, J.: Continuous experimentation in the b2b domain: a case study. In: 2015 IEEE/ACM 2nd International Workshop on Rapid Continuous Software Engineering. pp. 12–18. IEEE (2015)
27. Royer, M., Houser, K., Durmus, D., Esposito, T., Wei, M.: Recommended methods for conducting human factors experiments on the subjective evaluation of colour rendition. Lighting Research & Technology **54**(3), 199–236 (2022)
28. Sætren, G.B., Hogenboom, S., Laumann, K.: A study of a technological development process: Human factors—the forgotten factors? Cognition, Technology & Work **18**, 595–611 (2016)
29. Shneiderman, B.: Human factors experiments in designing interactive systems. Computer **12**(12), 9–19 (1979)
30. Steinberg, R., Grumman, N.: Human factors at the speed of relevance for agile engineering. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting. vol. 66. SAGE Publications Sage CA: Los Angeles, CA (2022)
31. Verhoef, P.C., Broekhuizen, T., Bart, Y., Bhattacharya, A., Dong, J.Q., Fabian, N., Haenlein, M.: Digital transformation: A multidisciplinary reflection and research agenda. Journal of business research **122**, 889–901 (2021)
32. Wickens, C.D., Gordon, S.E., Liu, Y., Lee, J.: An introduction to human factors engineering, vol. 2. Pearson Prentice Hall Upper Saddle River, NJ (2004)
33. Williams, K.W.: Impact of aviation highway-in-the-sky displays on pilot situation awareness. Human Factors **44**(1), 18–27 (2002)
34. Yaman, S.G.: Initiating the transition towards continuous experimentation: empirical studies with software development teams and practitioners. Series of Publications A (2019)
35. Yaman, S.G., Munezero, M., Münch, J., Fagerholm, F., Syd, O., Aaltola, M., Palmu, C., Männistö, T.: Introducing continuous experimentation in large software-intensive product and service organisations. Journal of Systems and Software **133**, 195–211 (2017)