Build Waiting Time in Continuous Integration – An Initial Interdisciplinary Literature Review

Eero Laukkanen
Department of Computer Science and Engineering
Aalto University
Finland
Email: eero.laukkanen@aalto.fi

Mika V. Mäntylä
Department of Information Processing Science
University of Oulu
Finland
Email: mika.mantyla@oulu.fi

***Abstract*—In this position paper, we present and demonstrate the idea of using an interdisciplinary literature review to accelerate the research on continuous integration practice. A common suggestion has been that build waiting time in continuous integration cycle should be less than 10 minutes. This guideline is based on practitioners' opinion and has not been further investigated. The objective of this study is to understand the effects of build waiting time in software engineering and to get input from waiting time research in other disciplines. The objective is met by performing two literature reviews, first on build waiting time and second on waiting times in the contexts of service operation, web use and computer use. The found effects of build waiting time were categorized into continuous integration specific, cognitive and emotional. Two minute build waiting time was considered optimal, but under 10 minutes was considered acceptable. Insight from other waiting time research suggests that the perceptions of waiting time are important and the perceptions can be lowered by providing feedback and giving developers other activities during the integration.**

I. Introduction

Build waiting time is considered to be an essential aspect in continuous integration (CI) practice. Extreme programming [1] instructs that the build waiting time should be less than 10 minutes. Yet, scientific studies of the topic are scarce and only handful of practitioners reports can be found [2], [3].

One could argue that shorter build waiting time is always better. However, in CI practice, a balance needs to be achieved between a short build time and coverage of testing. Too short build waiting time in CI would mean poor test coverage while too high build waiting time would decrease the developers' process compliance of CI practice.

However, waiting time is not unique to software engineering or CI practice. In fact it has been studied in other contexts, e.g., service operation [4], web use [5] and computer use [6]. Thus, the research of waiting time in other disciplines could provide insight to build waiting time research in software engineering and, thus, accelerate the scientific progress.

The goals of this study are to understand how build waiting time affects developers in the CI process and to provide an optimal build waiting time in such practice. The goals translate directly to the research questions:

1) How build waiting time affects developers in CI practice?
2) What is an optimal waiting time in CI practice?
3) What can be learned from other waiting time research to understand the effects of build waiting time?

By optimal waiting time, we mean the longest build waiting time that does not cause any substantial negative effects for software development. By achieving this optimal build waiting time, one can avoid the negative effects, but at the same time enjoy the benefits of good test coverage. In the terminology of this study, build includes testing and other activities executed in the continuous integration practice.

Two literature reviews are conducted. First, build waiting times are studied in the context of CI. The context for build waiting times is articulated so that it can be compared to other contexts where waiting time is inherent. Second, waiting times are studied in other disciplines. The focus on the second review is on psychological or behavioral effects, so that the results could be applied to the CI practice. Finally, the effects from both studies are compared.

II. Methods

A. Review of build waiting time in continuous integration

Build waiting time is studied in the context of CI. This gives us two topics of interest, build time and continuous integration, that form the search strings used in this study (Table I). Two digital databases were used for our initial searches for this paper. First, Scopus was used to search articles based on titles, abstracts and keywords. The search yielded two articles, both of which were relevant to the study. Second, Google Scholar was used to search the body text of the articles. This time, only the term "build length" was used, because "build time" produced numerous irrelevant results where the term "build time" was used to referring to a concept "during build time" instead of the "duration of the build time". This search resulted in three articles. While the search did not find the already found two articles [2], [7], another relevant article [3] was found.

In addition to the found three, other articles could not be found either by snowballing or by using exploratory search strings. Most of the citations related to build time often give instructions on how to reduce build waiting time, but do not give reasons why the build waiting time should be reduced in the first place. If such why question was asked in an

TABLE I
SEARCH STRINGS AND RESULTS OF BUILD WAITING TIME STUDIES.

Source	Search string	Results	Relevant
Scopus	”continuous integration” AND (”build time” OR ”build length”)	2	2
Google Scholar	”continuous integration” AND ”build length”	3	1
Total		5	3

TABLE II
EXCLUSION STRINGS FOR WAITING TIME STUDIES.

Subject	Exclusion search strings
Medicine	”renal transplant”, ”mortality”, ”HIV”, ”cancer”
Math	”waiting time distribution”, ”single machine problem”, ”waiting time problem”

existing study, the study should have been found by our search strings. In addition, our other, more broad and systematic but unpublished, literature review about CI did not find other related articles to this study. Thus, it seems that existing knowledge is limited to the collected three articles.

Three kinds of data were extracted from the found articles: effects of build waiting time, decision making context related to build waiting time and suggested optimal waiting time.

B. Review of waiting time in general

Waiting time is a broad topic: simple search string ”waiting time” yields 469 000 results from Google Scholar. Through initial unsystematic searches with Google Scholar, we identified some relevant contexts for our study that would allow us to test whether work in other disciplines can help the build time research in software engineering. Specific exclusion strings were used for rough exclusion of the studies from medicine and math as those contexts were too distant to our topic, see Table II.

After initial searching, three relevant waiting time contexts were identified: *service operation, web use* and *computer use*. These contexts were searched further by forward snowballing the articles that were initially selected as significant. The goal was to find literature reviews to allow us effectively extract the main findings from these sciences. The study by Baker and Cameron [4] was selected initially as the source for service operation, and the study by Nah [5] was selected to cover web and computer use. Both articles were forward snowballed to find more recent literature review studies. The articles remained to be primary sources for service operation and web use, but for computer use, a more recent article was found by Dabrowski and Munson [6].

From the selected articles, effects of waiting time, decision making context and suggested optimal waiting time were extracted. These could be compared to the results of the first literature review.

III. RESULTS

A. Build waiting time

1) Brooks: Brooks [2] lists multiple effects of build waiting time by comparing his experiences in two software projects (2 and 20 minute build waiting times):

- Long builds → Seldom code check-ins → Large commits → Troublesome merges
- Long builds → Avoid small commits → Unwillingness to refactor or do other small changes
- Long builds are complex and hard to fix when broken
- Long builds → Interrupt development flow → Start to do something else or interruption later to fix the build
- Long builds → Reduced developer satisfaction

Based on these effects, build time affects four variables: *commit size and frequency, build down time, development flow* and *developer satisfaction*.

Brooks does not state any explicit recommendation of an optimal build waiting time. However, he provides developers' perceptions indicating that in this case two minutes was an optimal build time.

2) Rogers: Rogers [3] introduces strategies to implement CI in larger and more complex projects based on his experiences. Regarding build waiting time, he states:

- Long builds → Infrequent integration frequency
- Short builds (less than 2 min) → Integration at will
- Medium builds (5–10 min) → Integration several times a day
- Long builds (30–40 min) → Integration once a week → Allocation a day of work for the integration
- Long builds & High integration effort → Integrations become an interruption → Integration are scheduled
- Developers do not like spending time integrating the code

As a summary, according to Rogers build waiting time affects *commit size and frequency, integration effort* and *development flow*.

Rogers argues that optimal build time depends on the project goals. Projects should establish a maximum build length and keep on it. However, he states that under two minute build, developers can integrate as many times they want. Also suboptimal build time of 5 to 10 minutes still allows developers to integrate *several times a day*, which is a common definition for CI. For build times longer than that, the integration frequency exceeds the definition of continuity.

3) Rasmusson: Rasmusson [7] provides reasons to shorten build waiting time based on his experiences that

- Short builds → Fast feedback
- Long builds → Slow feedback → ”pain”
- Infrequent integration → Increase integration difficulties
- Long builds → Distract flow → Doing other things during build time is bad
- Long builds → Increase short-cuts
- Long builds → Lessen team morale

The effects described by Rasmusson can be described as *feedback delay, integration effort, development flow* and *team morale*.

TABLE III
EFFECTS OF WAITING TIME IN OTHER CONTEXTS.

Context	Effects
Web waiting time [5]	flow of thought (1 s), short-term memory (2 s), attention/focus on dialogue (10 s), satisfaction (12 s), performance and frustration (30 s)
System response time [6]	errors, productivity, user adjustment, psychology: anger, frustration, annoyance, physiology: stress, anxiety

Rasmusson recommends that developer builds are under 10 minutes: "Ten minutes is about all I can take before I begin to feel like I am unproductive." He also mentions that in the early stages of project, the build will be much shorter and thus team members are more productive.

B. Waiting time in general

1) Service operation: Baker and Cameron [4] formed a conceptual model how service circumstances influence waiting time perception. These circumstances include service environment, queuing perceptions, attributions for delay and filled time with social interaction or with other distraction. The circumstances can be modified to increase user satisfaction. For example, service environment, including lightning, the temperature, music, color and furnishings, can be adapted to the service context. Queue perceptions can be modified by showing the progress of a queue and maintaining adherence to first-in/first-out policy, demonstrating social justice. Attributions for delay can be adjusted with spatial layout and employee visibility. Finally, the waiting time can be filled with distracting furnishings or social interaction.

Baker and Cameron did not propose any optimal waiting times. Instead, they encouraged minimizing the perception of waiting time instead of the actual waiting time.

2) Web use: The literature review by Nah [5] described five effects of waiting time (Table III) that happen on different timescales. Nah's own study suggest that the task given to the web user can affect the tolerable waiting time and that giving feedback increases the tolerable waiting time. Additionally, multi-media content, users' expectations, time pressure and environmental factors can affect the tolerable waiting time.

3) Computer use: Dabrowski and Munson [6] summarized the studies regarding computer system response time. The effects of system response time are summarized in Table III. The *user adjustment* means, that the users adjust their response time and style according to the system response time.

Considering optimal waiting time, Dabrowski and Munson notice that tolerable response time depends on the complexity of the task, the type of the interaction and users' expectations. They divide tasks into two categories: *control tasks*, where the user would like the software to behave like a physical device and *conversational tasks*. For control tasks, delays should not be noticeable, and the article recommends a delay between 50–200 ms. For conversational tasks, there is no such precise optimal waiting time, and research results vary.

TABLE IV
EFFECTS OF BUILD WAITING TIME AND IN THE SOURCES.

Effect	Brooks [2]	Rogers [3]	Rasmusson [7]
CI specific			
Commit size & frequency	✓	✓	–
Build down time	✓	–	–
Integration effort	–	✓	✓
Cognitive			
Development flow	✓	✓	✓
Feedback delay	–	–	✓
Emotional			
Developer satisfaction	✓	–	–
Team morale	–	–	✓

IV. DISCUSSION AND SUMMARY

A. Effects of build waiting time

Seven effects of build waiting time were found in the literature and they are categorized in Table IV into *CI specific, cognitive* and *emotional* effects. Effects of build waiting time can be summarized as follows. Developers make the decision when and how often to commit based on the build waiting time. The need for feedback and task completion encourages developers to commit often. Integration effort and flow disruption reduce the willingness to commit often. If build waiting time is short enough, then no explicit decision need to be done and integration can be performed as often as wanted. Otherwise, development flow is disrupted and developer allocates specific time slot for performing the integration.

While the sources state that build waiting time of 2–10 minutes is considered optimal, we find the quality of evidence for this weak. All the sources were experience reports reported by the same company. Since no scientific investigation on the issue was found, we conclude that there is a lack of empirical quantitative research to address this question.

B. Insights from other waiting time literature

Service operation context suggests that developer satisfaction can be increased by adjusting perceptual waiting time instead of the actual waiting time. However, since the development flow is an important aspect of CI practice, one needs to address the perceptual waiting time by not disrupting the flow. sThe service operation context confirms that waiting time is related to user satisfaction.

Web use context suggests that giving feedback increases tolerable waiting time. Thus, giving feedback about the progress of the build could maintain the flow of development further and make the waiting time feel more satisfactory. The web use context confirms that waiting time is related to memory (listed as feedback delay in Table IV), flow and satisfaction.

Conversational tasks in computer use context proposes that tolerable waiting time fluctuates. As the CI process is an conversational activity, it implies that the CI process could be designed to have multiple options based on the waiting time that is acceptable for the task at hand. Simple tasks such as

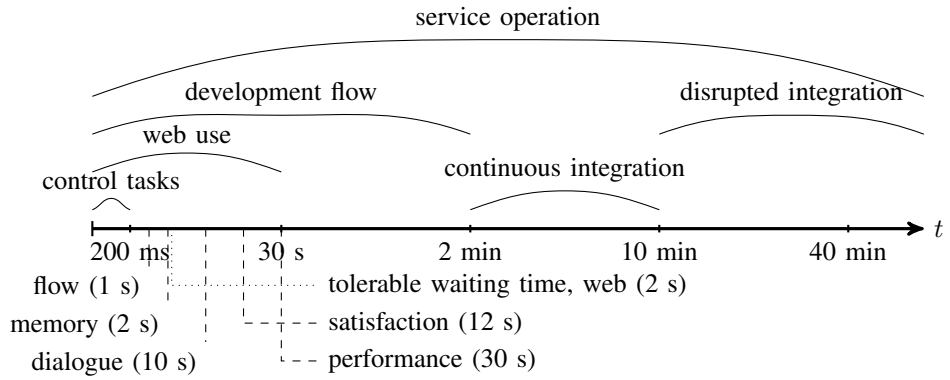

Fig. 1. Comparison of the timescales between build waiting time and other waiting time contexts.

TABLE V
COMPARING BUILD WAITING TIME AND OTHER CONTEXTS.

Context	Similarities	Differences	Guidances
Service operation	Emotional effects	Service user can be distracted	Control perceived waiting time
Web use	Cognitive and emotional effects	Shorter timescale	Give feedback
Computer use	Emotional effects, user adjustment	Shorter timescale	Tolerable waiting time depends on the task

small refactorings could be accepted faster, whereas complex changes could be tested more carefully.

Computer use studies state that too long waiting times cause anger, frustration and annoyance, which matches the decreased developer satisfaction due to long build times. Also the user adjustment effect can be related to the CI context. Developers are said to adjust to the build waiting time by doing commits less often or reserving special time for integration work.

In this study we found similar, yet more elaborately described, effects from other waiting time literature as was found from SE literature (Table V). Perhaps the effects are somewhat universal even when the timescales (Fig. 1), and contexts vary.

V. THREATS TO VALIDITY

All primary articles for build waiting times were published as experience reports. Furthermore, the writers of the articles were from the same company, ThoughtWorks. Thus, the evidence cannot be considered scientific, and there might be bias because of single company source.

Search completeness was not the main goal, as we only wanted to experiment the idea of an interdisciplinary literature review. Perhaps, we would have found more papers with a more comprehensive search, e.g. using synonyms such as build duration or build queue time. Furthermore, searching other databases than Scopus and Google Scholar might have produced more results.

VI. CONCLUSIONS

This study makes five contributions. First, we present and demonstrate the idea of using an interdisciplinary literature review of literature reviews as an approach to accelerate software engineering research. Second, based findings from other disciplines we suggest that CI waiting time should be modified by giving feedback, controlling perceived waiting time and having different waiting times for different tasks. Third, from CI research it was found that build waiting time has CI specific, cognitive and emotional effects. Fourth, while optimal waiting time of 2–10 minutes was provided, we found that there is a lack of empirical research considering this. Fifth, other waiting time research confirms and elaborates the negative cognitive and emotional effects presented in CI research.

The future research of CI should focus on shortening the build time and on the concept of development flow as it was considered important. Determining optimal build waiting time in continuous integration practice requires more research. Finally, a more in-depth interdisciplinary literature review should be performed and the idea of interdisciplinary review could be used in other studies.

REFERENCES

[1] K. Beck, *Extreme programming explained: embrace change*. Addison-Wesley Professional, 2000.

[2] G. Brooks, "Team pace – keeping build times down," in *Proceedings of the Agile 2008*, Washington, DC, USA, 2008, pp. 294–297.

[3] R. O. Rogers, "Scaling continuous integration," *Extreme Programming and Agile Processes in Software Engineering, Proceedings*, vol. 3092, pp. 68–76, 2004.

[4] J. Baker and M. Cameron, "The effects of the service environment on affect and consumer perception of waiting time: an integrative review and research propositions," *Journal of the Academy of Marketing Science*, vol. 24, no. 4, pp. 338–349, 1996.

[5] F. F.-H. Nah, "A study on tolerable waiting time: how long are web users willing to wait?" *Behaviour & Information Technology*, vol. 23, no. 3, pp. 153–163, 2004.

[6] J. Dabrowski and E. V. Munson, "40 years of searching for the best computer system response time," *Interacting with Computers*, vol. 23, no. 5, pp. 555–564, Sep. 2011.

[7] J. Rasmusson, "Long build trouble shooting guide," in *Extreme Programming and Agile Methods - XP/Agile Universe 2004*, 2004, no. 3134, pp. 13–21.