Report from Dagstuhl Seminar 17102

Rethinking Productivity in Software Engineering

Edited by

Thomas Fritz[1], Gloria Mark[2], Gail C. Murphy[3], and Thomas Zimmermann[4]

1 **Universität Zürich, CH,** `fritz@ifi.uzh.ch`
2 **University of California – Irvine, US,** `gmark@uci.edu`
3 **University of British Columbia – Vancouver, CA,** `murphy@cs.ubc.ca`
4 **Microsoft Corporation – Redmond, US,** `tzimmer@microsoft.com`

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 17102 "Rethinking Productivity in Software Engineering". In the following, we briefly summarize the goals and format of the of the seminar, before we provide insights and an outlook, including a few grand challenges, based on the results and statements collected during the seminar.

Seminar March 5–8, 2017 – http://www.dagstuhl.de/17102
1998 ACM Subject Classification D.2.8 Metrics, D2.9 Software Engineering – Management–Productivity, H.5 Information Interfaces and Presentation, H.1.2 User/Machine Systems – Human Factors, H.3.4 Systems and Software – Current Awareness Systems & Performance evaluation (efficiency and effectiveness), H.4.1 Office Automation – Time management & Workflow Management, K.4.3 Organizational Impacts – Employment
Keywords and phrases productivity, software development, human factors, productivity factors, grand challenges
Digital Object Identifier 10.4230/DagRep.7.3.19
Edited in cooperation with André N. Meyer (Universität Zürich)

1 Executive Summary

Thomas Fritz
Gloria Mark
Gail C. Murphy
Thomas Zimmermann

There is an ever-growing demand of software being built and a shortage of software developers to satisfy this demand, despite the immense growth in the number of professional software developers. To address this demand, industry and research are looking into understanding and improving the productivity of individual software developers as well as teams. A substantial amount of research has examined the meaning of software productivity over the past four decades. Much of this research introduces particular definitions of productivity, considers organizational issues associated with productivity, or is focused on specific tools and approaches for improving productivity. In fact, many of the seminal work on software productivity is from the 80s and 90s (Peopleware, Mythical Man-Month, Personal Software Process).

Rethinking Productivity in Software Engineering, *Dagstuhl Reports*, Vol. 7, Issue 3, pp. 19–26
Editors: Thomas Fritz, Gloria Mark, Gail C. Murphy, and Thomas Zimmermann
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

At the same time, software development has changed significantly over the past decades with the rise of agile development, distributed development, more rapid release cycles and the high fragmentation of today's work. Simultaneously the technology available to software engineers has improved with social coding tools like GitHub[1] and StackOverflow[2] and better IDEs. Furthermore, research communities, in particular the HCI and CSCW communities, have made significant advances in supporting knowledge workers to become more productive that one might be able to also transfer to software engineers.

The goal of this seminar was to rethink, discuss, and address open issues of productivity in software development and how to measure and foster productive behavior of software developers. Specifically, we focused on the following questions:

- What does productivity mean for an individual and teams/organizations and how is it measured?
- What are the dimensions and factors of productivity?
- What are the purposes and implications of measuring productivity?
- What are the grand challenges in research on productivity?

[1] http://www.github.com
[2] http://www.stackoverflow.com

2 Table of Contents

Executive Summary
Thomas Fritz, Gloria Mark, Gail C. Murphy, and Thomas Zimmermann 19

Introduction

Seminar Format 22

Productivity – Insights and Outlook 22

Follow-up Work 23

Overview of the Main Talks

Dark side of Global Agile: Challenging Productivity as a Positive
Pernille Bjørn 24

Programming Productivity Primer
Andrew J. Ko 24

"Stop trying to do what you're trying to do": Developers' Perceptions of Measuring Productivity
Christoph Treude 24

Quantifying mind-wandering in laboratory studies
Marieke van Vugt 25

What is Productivity? Terminology and Influencing Factors
Stefan Wagner 25

Participants 26

3 Introduction

3.1 Seminar Format

In this seminar, we brought together researchers and practitioners with backgrounds in Software Engineering, Human Computer Interaction, and Computer-Supported Collaborative Work who are interested and working on topics related to the productivity of software developers as well as more general knowledge workers. Before the seminar, we conducted a small survey to collect relevant further questions to be addressed in the seminar and the break out groups.

At the seminar, we used a combination of methods to (a) foster vibrant discussions, (b) to address relevant questions on developer productivity as well as to (c) foster interaction and collaborations between attendees. In particular we used a speed dating technique as a way to get to know attendees, short three minute presentations by each attendee to get an overview of everyone's interests and research, fifteen minute talks by a few attendees with various backgrounds to get deeper insights into some of the work in the various areas, and group discussion as well as breakout sessions to enable deeper conversations in smaller groups with the results being reported back to the whole group.

3.2 Productivity – Insights and Outlook

In the following we will present a short summary that we compiled from the discussions, the presentations and the learnings from the attendees. We will thereby focus on the topic of software developer productivity in general, factors of productivity and the grand challenges that lie ahead of us.

Software Developer Productivity

Productivity is a concept that is difficult to define and measure due to its complexity, its multi-facetted nature and the rather broad concept that the term 'productivity' denotes. Depending on the context, other terms such as 'time well spent', 'software quality', 'velocity', or 'satisfaction' might be better suited. Overall it is important to understand and specify the context in which productivity is being measured to determine how to best measure developer productivity. For instance, measuring productivity for the purpose of providing a developer a retrospection of her work and a sense of achievement is very different to measuring productivity for the purpose of evaluating the implementation of a new development process in an organization.

There is a broad range of dimensions that affect the definition and measurement of productivity, such as the specific purpose (e.g. self-assessment, resource allocation, evaluating the success of interventions such as tools and practices, identifying problems, job satisfaction, quality of output), the unit of analysis (e.g. individual, team, organization, inter-organization), the target audience (e.g. personal, manager, customer, shareholder), the time horizon/period (e.g. immediate feedback, ten years later). A more specific definition of the context for measuring productivity will allow you to determine more meaningful measures of productivity.

Another aspect to consider when measuring productivity is the presentation of the measures and the effect of collecting them. Visualizations of productivity measures might be interpreted differently by different people based on background, culture or other reasons and in addition, the sheer collection of certain work related measures might affect the behavior that is being measured or harm the overall outcome (e.g. developers might try to game the system to achieve a better performance rating).

Factors of Productivity

There is a variety of factors of productivity for knowledge workers, such as the skills of the worker, the time of the day, the nature of the task, the attention fatigue, the breaks taken, the work fragmentation, the goals (tangible & intangible), the coordination and deadlines, the team and social factors, or the rewards. Human factors, also known as soft factors, appear to have the biggest effects on overall productivity, yet they are a lot harder to measure. While some of these factors have already been studied in more depth either in the context of general knowledge workers or specifically for software developers, there are still a lot of open challenges and questions.

Open Questions and Grand Challenges

Below is a list of some of the stated open questions and grand challenges by attendees.

- Develop a theoretical framework for productivity.
- Develop an approach that allows to track "everything" at every moment, including detailed data across a company, biometric data from individuals and data on aspects such as satisfaction, mood, fatigue and motivation. Use the data to profile development work and productivity.
- Design and create a company that implements human values and culture of Zappos and compare with other companies to study the effect of these factors on productivity and outcome.
- Examine the difference of software development to all other kinds of knowledge workers and learn what is unique about software development and what is not.
- Define laws or rules of productivity similar to the laws of software evolution, e.g. a happier developer is a more productive developers, a participatory culture in a team is more productive.
- Develop a mapping from questions on productivity to methodology of studying it.
- Conduct a multitude of comparative studies on productivity at different companies or just on different interventions.
- Collect examples of where measuring productivity was done well and had good outcomes, and distill the insights and guidelines from this collection.
- Understand how to support and facilitate productivity?
- Understand how a people's view of productivity affects their productivity and whether changing the motivation from self-improvement to altruism (shifting away from productivity) may increase it (relating productivity to meaning).

3.3 Follow-up Work

Multiple paths forward to continue the work on productivity have been discussed, especially due to the interdisciplinary interest in the topic that can benefit from researchers from various domains, ranging from the organization of another seminar or workshop to the writing of a book and a collaborative grant proposal. At this point in time, we have put together a web site with related work resources and two participants started to organize the co-writing of a book on topics of productivity.

4 Overview of the Main Talks

4.1 Dark side of Global Agile: Challenging Productivity as a Positive

Pernille Bjørn (University of Copenhagen, DK)

URL http://itu.dk/~pbra/pmwiki/pmwiki.php

The talk explore the dark side of global agile – and how introducing agile scrum processes into a global outsourcing setup creates special conditions for software engineering work, which risk taking away empowerment and work/life balance for software engineers in the global south. Being a global software developer working out of the global south is different than working out of the global north. However, such presentation would not be so much about state-of-art and challenges – but rather about posing a question about productivity in software engineering.

4.2 Programming Productivity Primer

Andrew J. Ko (University of Washington – Seattle, US)

URL http://faculty.washington.edu/ajko/

In this talk I present recent evidence about software engineering productivity from multiple levels, including individual, team, organizational, and market perspectives. I discuss important discoveries at each of these levels and pose new questions about the relationships between these levels.

4.3 "Stop trying to do what you're trying to do": Developers' Perceptions of Measuring Productivity

Christoph Treude (University of Adelaide, AU)

URL http://ctreude.ca/

Software developers pursue a wide range of activities as part of their work, and making sense of what they did in a given time frame is far from trivial as evidenced by the large number of awareness and coordination tools that have been developed in recent years. To inform tool design for making sense of the information available about a developer's activity, we conducted a survey with 156 GitHub users to investigate how they would summarize and measure development activity. In addition to proposing several formulas for productivity, participants warned that measuring development activity can be dangerous and that metrics are likely to be gamed. Aspects to consider include the created business value, the quality of work produced, the difficulty of a task, and the context of the work.

4.4 Quantifying mind-wandering in laboratory studies

Marieke van Vugt (University of Groningen, NL)

URL http://www.rug.nl/staff/m.k.van.vugt/

Mind-wandering is a process of task-unrelated thought that could sometimes hinder productivity, but may also be beneficial in other circumstances. We can measure mind-wandering using a triangulation approach, combining first- and third-person measures. Specifically, we give people a boring task and every 30–60 seconds, we ask them whether they were paying attention to the task, or other things. We can then relate objective task performance back to these subjective "thought probes". Studies show that just prior to responding off-task, performance is worse, variability in response time is increased, and event-related potentials have a lower amplitude. It is important to distinguish between mind-wandering that is easy to disengage from, and that is not so disruptive from mind-wandering that is more ruminative in nature and difficult to disengage from. Ruminative mind-wandering can even lower working memory capacity. In short, measures used in the study of mind-wandering may be interesting to include in studies of productivity.

4.5 What is Productivity? Terminology and Influencing Factors

Stefan Wagner (Universität Stuttgart, DE)

URL http://www.iste.uni-stuttgart.de/se/menschen/stefanwagner.html

Increasing productivity is a general goal in software engineering research. Yet, there is a lot of uncertainty about what productivity means in knowledge work and software engineering in particular. I describe a terminology that defines productivity in terms of effectiveness and efficiency which describe the functionality and quality of a software system with respect to its purpose and the effort put into building it. Profitability extends this by including effects of price and cost inflation. Performance furthermore includes marketing or corporate learning.

In Wagner and Ruhe (2008), we collected 51 factors that influence productivity as stated in the literature. There is a huge variety of factors ranging from product and process factors to team and organisational factors. In a recent study (Karimi et al., 2016), we found that also personality (in particular conscientiousness) and programming styles influence productivity in student projects.

Finally, the ProdFLOW approach by Siemens (Ruhe, Wagner, 2008) is an industrial method to first investigate a specific business context by interviews and qualitative analysis to derive the important productivity levers. Only then, we try to measure the improvement of these levers.

Participants

- Christian Bird
Microsoft Corporation – Redmond, US
- Pernille Bjørn
University of Copenhagen, DK
- Marcos Borges
University of Rio de Janeiro, BR
- Duncan Brumby
University College London, GB
- Robert Feldt
Chalmers UT – Göteborg, SE
- Thomas Fritz
Universität Zürich, CH
- James D. Herbsleb
Carnegie Mellon University – Pittsburgh, US
- Christian Janssen
Utrecht University, NL
- Ciera Jaspan
Google Inc. – Mountain View, US
- Katja Kevic
Universität Zürich, CH
- Andrew J. Ko
University of Washington – Seattle, US
- Gloria Mark
University of California – Irvine, US
- André Meyer
Universität Zürich, CH
- Gail C. Murphy
University of British Columbia – Vancouver, CA
- Emerson Murphy-Hill
Google Inc. – Mountain View, US
- Brad A. Myers
Carnegie Mellon University – Pittsburgh, US
- Christopher J. Parnin
North Carolina State University – Raleigh, US
- Martin Robillard
McGill University – Montreal, CA
- Caitlin Sadowski
Google Inc. – Mountain View, US
- Federica Sarro
University College London, GB
- David C. Shepherd
ABB – Raleigh, US
- Janet Siegmund
Universität Passau, DE
- Margaret-Anne Storey
University of Victoria, CA
- Christoph Treude
University of Adelaide, AU
- Marieke van Vugt
University of Groningen, NL
- Stefan Wagner
Universität Stuttgart, DE
- Thomas Zimmermann
Microsoft Corporation – Redmond, US