# Design Recommendations for Self-Monitoring in the Workplace: Studies in Software Development

Meyer, André; Murphy, Gail C; Zimmermann, Thomas; Fritz, Thomas

Abstract: One way to improve the productivity of knowledge workers is to increase their self-awareness about productivity at work through self-monitoring. Yet, little is known about expectations of, the experience with, and the impact of self-monitoring in the workplace. To address this gap, we studied software developers, as one community of knowledge workers. We used an iterative, user-feedback-driven development approach (N=20) and a survey (N=413) to infer design elements for workplace self-monitoring, which we then implemented as a technology probe called WorkAnalytics. We field-tested these design elements during a three-week study with software development professionals (N=43). Based on the results of the field study, we present design recommendations for self-monitoring in the workplace, such as using experience sampling to increase the awareness about work and to create richer insights, the need for a large variety of different metrics to retrospect about work, and that actionable insights, enriched with benchmarking data from co-workers, are likely needed to foster productive behavior change and improve collaboration at work. Our work can serve as a starting point for researchers and practitioners to build self-monitoring tools for the workplace.

DOI: https://doi.org/10.1145/3134714



Originally published at:
Meyer, André; Murphy, Gail C; Zimmermann, Thomas; Fritz, Thomas (2017). Design Recommendations for Self-Monitoring in the Workplace: Studies in Software Development. PACM on Human-Computer Interaction, 1(CSCW):1-24.
DOI: https://doi.org/10.1145/3134714

PRE-PRINT

# Design Recommendations for Self-Monitoring in the Workplace: Studies in Software Development

ANDRE N. MEYER, University of Zurich
GAIL C. MURPHY, University of British Columbia
THOMAS ZIMMERMANN, Microsoft Research
THOMAS FRITZ, University of Zurich and University of British Columbia

One way to improve the productivity of knowledge workers is to increase their self-awareness about productivity at work through self-monitoring. Yet, little is known about expectations of, the experience with, and the impact of self-monitoring in the workplace. To address this gap, we studied software developers, as one community of knowledge workers. We used an iterative, user-feedback-driven development approach (N=20) and a survey (N=413) to infer design elements for workplace self-monitoring, which we then implemented as a technology probe called *WorkAnalytics*. We field-tested these design elements during a three-week study with software development professionals (N=43). Based on the results of the field study, we present design recommendations for self-monitoring in the workplace, such as using experience sampling to increase the awareness about work and to create richer insights, the need for a large variety of different metrics to retrospect about work, and that actionable insights, enriched with benchmarking data from co-workers, are likely needed to foster productive behavior change and improve collaboration at work. Our work can serve as a starting point for researchers and practitioners to build self-monitoring tools for the workplace.

CCS Concepts: • **Human-centered computing** → **User studies**; *Field studies*; • **Software and its engineering** → Software creation and management;

Additional Key Words and Phrases: Quantified Workplace, Self-Monitoring, Productivity Tracking, Personal Analytics, Workplace Awareness

**ACM Reference format:**
Andre N. Meyer, Gail C. Murphy, Thomas Zimmermann, and Thomas Fritz. 2017. Design Recommendations for Self-Monitoring in the Workplace: Studies in Software Development. *Proc. ACM Hum.-Comput. Interact.* 1, 2, Article 79 (November 2017), 24 pages.
https://doi.org/10.1145/3134714

## 1 INTRODUCTION

The collective behavior of knowledge workers at their workplace impacts an organization's culture [12], success [31] and productivity [50]. Since it is a common goal to foster productive behavior at work, researchers have investigated a variety of factors and their influence on knowledge workers'

This work was funded in part by Microsoft, NSERC and SNF. Authors' addresses: A. N. Meyer, T. Fritz, Department of Informatics, University of Zurich, Zurich, Switzerland, email: ameyer@ifi.uzh.ch, fritz@ifi.uzh.ch; G. C. Murphy, T. Fritz, Department of Computer Science, University of British Columbia, Vancouver, Canada, emails: murphy@cs.ubc.ca, fritz@cs.ubc.ca; T. Zimmermann, Empirical Software Engineering Group, Microsoft Research, Redmond, US, email: tzimmer@microsoft.com.

2573-0142/2017/11-ART79
https://doi.org/10.1145/3134714

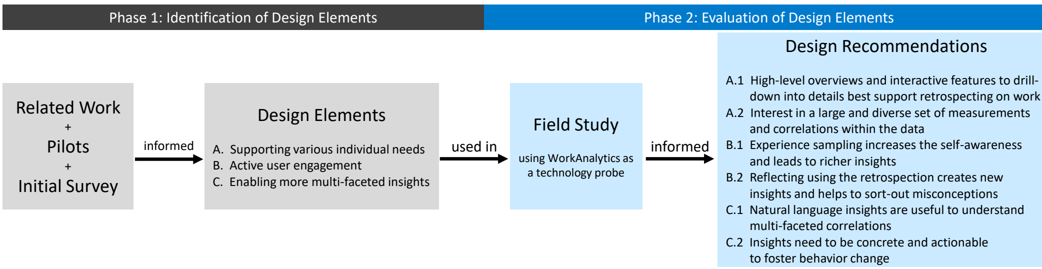


Fig. 1. Summary of the Two-Phase Study Describing the Process.

behavior and productivity, including the infrastructure and office environment [12, 22], the interruptions from co-workers [15, 21], and the teams' communication behaviors [49, 54]. Yet, knowledge workers are often not aware of how their actions contribute to these factors and how they impact both their own productivity at work and the work of others [56].

One way to improve knowledge workers' awareness of their own behavior and foster productive behavior is to provide them with the means to self-monitor and to reflect about their actions, for example through visualizations [57]. This type of self-monitoring approach has been shown to foster behavior change in other areas of life, such as physical activity (e.g., [18, 26]), health (e.g., [10, 19]) and nutrition (e.g., [28]). Existing efforts to map the success of these self-monitoring approaches to the workplace have largely focused on tracking and visualizing data about computer use [40, 58, 68, 71]. Although research has shown that self-monitoring at work can be valuable in increasing the awareness about a certain aspect of work, such as time spent in applications [59, 71] or distracting activities [40], little is known about knowledge workers' expectations of and experience with these tools [59, 68]. The lack of research about what knowledge workers' need from these tools may be one reason why many existing solutions have a low engagement and only short-term use overall [17, 40]. Furthermore, most of these approaches did not consider collaborative aspects of work, such as instant messaging, email or meetings.

We address these gaps by aiming to better understand what information and features knowledge workers expect in workplace self-monitoring tools. To make our investigations tractable, we focus on one community of knowledge workers, software developers, before generalizing to a broader range of knowledge workers in the future. We study software developers due to their extensive use of computers to support both their individual and collaborative work, including the use of issue trackers for collaborative planning [62, 63], code review systems for shared feedback gathering [6], and version control systems for co-editing artefacts [69]. Software developers are also an attractive target given the frequent interest of this community to continuously improve their work and productivity [35, 45]. Furthermore, software developers pursue a variety of different activities at work [5, 21, 30] that vary considerably across work days and individuals [53]. For our investigations, this combination of diversity in activity, similarity in domain and extensive use of a computers yields an ideal combination for considering self-monitoring in the workplace.

To determine a set of design recommendations for building workplace self-monitoring tools, we followed a mixed-method approach, which is summarized in Figure 1. Phase 1 of our approach started with an investigation of software developers' expectations of and requirements for measures to self-monitor their work. A review of related work indicated barriers that have been identified towards the adoption of self-tracking technologies at the workplace, including not fully understanding users' needs [43], not knowing in what measures users are interested in [54, 59, 67], and not providing users with a holistic understanding of their work behavior [2, 7, 27, 33]. To overcome barriers associated with appropriate measures, we analyzed previous work on measures of software development productivity (e.g., [54]) and designed and developed a prototype, called *WorkAnalytics*$_{pilot}$, that captures software development measures, allows software developers to self-monitor their work patterns and provides a retrospective view to a developer of their work day and work week.

We received feedback on the prototype through a pilot study with 20 participants and 5 iterations. Based on what we learned from the pilots, we conducted a study to learn about the design elements, including measures, needed in a self-monitoring tool for software development. We received input from 413 software development professionals for the survey. An analysis of the pilot and survey data indicated three design elements needed to build soft-monitoring tools for a workplace: A) supporting various individual needs for data collection and representation, B) enabling active user engagement, and C) enabling more insights on the multi-faceted nature of work.

In phase 2, we then refined the prototype to accommodate these design elements and conducted a field study involving 43 professional software developers using the refined prototype for three weeks. The refined prototype, which we refer to as *WorkAnalytics*, captures information from various individual aspects of software development work, including application use, documents accessed, development projects worked on, websites visited, as well as collaborative behaviors from attending meetings, and using email, instant messaging and code review tools. In addition, *WorkAnalytics* prompts a user to reflect on their work periodically and to-self report their productivity based on their individual definition. To enable more multi-faceted insights, the captured data is visualized in a daily retrospection (see Figure 2), which provides a higher-level overview in a weekly summary, and allows users to relate various data with each other.

From the field study, we derived six design recommendations, summarized in Figure 1. For instance, we learned that a combination of self-reflection on productivity using self-reports, and observations made from studying the insights in the retrospection enhances participants' awareness about the time spent on various activities at work, about their collaboration with others, and about the fragmentation of their work. In this paper, we report on these six design recommendations and further requests made by participants for features to help them turn retrospective information into action. For instance, participants requested recommendation tools to help them better plan their work, improve their team-work and coordination with others, block out interruptions, and increase their productivity.

This paper provides the following main contributions:

- It demonstrates that self-monitoring at work can provide novel insights and can help to sort out misconceptions about work activities, but also highlights the need for information presented to be concrete and actionable, rather than simply descriptive.
- It demonstrates the value of brief and periodic self-reports to increase awareness of work and productivity for software developers.
- It presents a set of measurements specific to software development that professional software developers report to provide the most value to increase awareness of their work, ranging from the time spent doing code reviews to the number of emails received in a work day.

This paper is structured as follows: We first discuss related work before we present how we identified design elements for self-monitoring in the workplace, and how we incorporated and evaluated them using *WorkAnalytics* as a technology probe. Subsequently, the findings and distilled design recommendations are presented. Finally, we discuss our findings with respect to long-term user engagement, potential impact on individuals and the collaboration with their teams, and the generalizability of our results.

## 2 RELATED WORK

This section provides background on the rise of approaches for self-monitoring various aspects of life and work, and barriers towards the adoption of these self-tracking technologies.

### 2.1 Self-Monitoring to Quantify Our Lives

In the health domain, wearable self-monitoring devices have proliferated in recent years thanks to their miniaturization [20], and are used for tracking physical activity [18, 19, 26, 46], emotional states [52], stress [49, 50], sleep [24, 38] and diets [28]. This self-monitoring and reflection leads to increased self-awareness, which helps to realize bad habits in behavior [14], and often promotes deliberate or unconscious behavior changes [13, 25, 32], so called reactivity effects [55]. For example, physical activity trackers, such as the Fitbit [24], motivate users to a more active and healthy life-style [26, 46]. The Transtheoretical Model (TTM) [57], a well-established theory of behavior change processes, describes behavior change as a sequence of stages which are run through until a behavior change happens and can be maintained. Self-awareness is one of the processes that lets people advance between stages. In particular, it helps people to move from being unaware of the problem behavior (precontemplation stage) to acknowledging that the behavior is a problem and the intention to improve it (contemplation stage). Self-monitoring tools have been shown to help create an understanding of the underlying causes of problematic behavior, to point to a path towards changing the behavior to a more positive one, and to help maintain and monitor the behavior change (e.g. [10, 46]). Researchers also evaluated the social aspects of self-monitoring systems and found that the sharing of data with acquaintances or strangers can be a powerful and durable motivator, but raises privacy concerns due to the sensitivity of the shared data [26, 66].

With our work, we aim to investigate how we can map the success of these approaches to software developers' work, and learn more about their expectations of and experience with self-monitoring tools for the workplace and the impact they may have on productivity and behavior.

### 2.2 Designing and Evaluating Self-Monitoring Tools for Work

In addition to work on quantifying many aspects of a person's life, there is a growing body of HCI research that focuses on quantifying aspects of work and promoting more productive work behaviors with self-monitoring techniques. Many of these approaches focus on the time spent in computer applications [34, 40, 48, 58, 61, 71], the active time on the computer [59], or work rhythms [8]. Some approaches specifically target the activities of software developers in integrated development environments (e.g., Codealike [16], WatchDog [9] and Wakatime [70]). Few of these tools have been evaluated (e.g., [33, 40, 59, 71]), limiting our knowledge of the overall value of these tools to users, particularly limiting our knowledge of which information is of value to users and if the approaches can affect the behaviour of users. As described by Klasnja et al. [41], it is often feasible to evaluate the efficacy of a self-monitoring tool in a qualitative way to identify serious design issues early, while still seeing trends in how behaviour might change in the long-term. In this paper, we follow this recommendation, focusing on facilitating the reasoning and reflection process of a knowledge worker by increasing self-awareness about the monitored aspect of work [33, 40, 57]. We leave an assessment of whether the design recommendations we provide can be embodied in a tool to change user behaviour to future work.

To provide a starting point for building self-monitoring tools targeting software developers at work and evaluate their potential impact on behaviors at work, we conducted a three-week user study to investigate the efficacy of the design elements that we identified from related work, five pilots, and a survey, using *WorkAnalytics* as a technology probe. To our knowledge, this is also the first approach that focuses to raise developers' awareness about their collaborative activities, such as gaining insights about emailing, meeting, and code reviewing.

Previous research has also discovered that users rarely engage with the captured data, resulting in a low awareness and reducing chances for a positive behavior change when using a self-monitoring tool [17, 33, 40]. We compiled and categorized a list of barriers related work has identified towards the adoption of self-monitoring technologies at the workplace:

**Not understanding user needs:** Research has shown that knowledge workers' needs for monitoring their computer use vary and that little is actually known about the measures they are interested in [44, 54, 59, 67]. Users sometimes also have too little time for a proper reflection of the data, or an insufficient motivation to use the tool, which is likely one reason they often stop using it after some time [43]. This emphasizes the importance of understanding users' needs and expectations about how self-monitoring tools should work and what measures they should track, to increase the chance people are trying such a tool and using it over extended periods.

**Lack of data context:** Most tools we found miss the opportunity to provide the user with a more holistic understanding and context of the multi-faceted nature of work, as they only collect data about a single aspect, e.g., the programs used on the computer [14, 23]. This makes it difficult for users to find correlations between data sets and, thus, limits the insights they can get. Behavior change cannot be modelled based on just a few variables, as the broader context of the situation is necessary to better understand the various aspects influencing work behavior and productivity [7, 33]. To overcome this, Huang et al. [33] propose to integrate these self-monitoring approaches into existing processes or tools and place them into an already existing and well-known context which makes it easier for users to engage in an ongoing tool use. Choe et al. [14] further suggest to track many things when users first start a self-monitoring initiative, and then let them decide which measures are necessary for their context to reflect and improve their behavior.

**Difficulties in interpreting the data:** Choe et al. [14] and Huang et al. [33] argue how difficulties in making sense of, organizing or interpreting the data result in a lower adoption of self-monitoring approaches, as users will stop using them. For example, Galesic and Garcia-Retamero [27] found that more than 40% of Americans and Germans lack the ability to understand simple graphs, such as bar or pie charts, which could be a problem for self-monitoring tools as they often visualize the data. To overcome this issue, Bentley and colleagues [10] propose to provide insights from statistically significant correlations between different data types in natural language, which helped participants in the study to better understand the data. Another problem to efficiently interpret data in personal informatics systems is information overload, as described by Jones and Kelly [37]. They found that users generally have a higher interest in multi-faceted correlations (correlations between two distinct data categories), rather than uni-faceted correlations, that reveal "surprising" and "useful" information. Hence, this could help to reduce information overload and provide more relevant insights to users.

**Privacy Concerns:** Another potential pitfall of self-monitoring tools is data privacy, as many users are afraid the data might have a negative influence on their life, such as fearing their managers may know how well they sleep, or that their insurance agency can track their activity. Most privacy concerns can be reduced by letting users decide what and how they want to share their data, by obfuscating sensitive data when it is being shared, by abstracting visualizations, and by letting users opt-out of applications when they think the gained benefits do not outweigh the privacy risks [8, 51].

Besides learning more about software developers' expectations of and experience with a self-monitoring tool for work and productivity, we used our iterative, feedback-driven development process and a survey to investigate how these barriers could be tackled. Based on the findings, we incorporated the identified design elements into our self-monitoring approach *WorkAnalytics* and then used it to evaluate how the design elements affect developers' awareness on work and productivity. Subsequently, we distilled design recommendations for building self-monitoring tools for developers' work.

| Phase 1: Identification of Design Elements for Self-Monitoring at Work (iterative, feedback-driven development of WorkAnalytics) | | | | | |
|---|---|---|---|---|---|
| Method | # Partic. | Company ID | # Developers | Location | Duration |
| **Pilots** | **20** | | | | **2-4 work weeks** |
| Pilot 1 | 6 | A | ca. 3000 | Canada | 2 work weeks |
| Pilot 2 | 2 | B | ca. 150 | Canada | 2 work weeks |
| Pilot 3 | 3 | C | 4 | Switzerland | 2 work weeks |
| Pilot 4 | 5 | D | ca. 50000 | USA | 4 work weeks |
| Pilot 5 | 4 | A | ca. 3000 | Canada | 3 work weeks |
| **Initial Survey** | **413** | **D** | **ca. 50000** | **USA** | **sent out 1600 invitations** |

| Phase 2: Evaluation of the Design Elements for Self-Monitoring at Work (using WorkAnalytics as a technology probe) | | | | | |
|---|---|---|---|---|---|
| Method | # Partic. | Company ID | # Developers | Location | Duration/Timing |
| **Field Study** | **43** | **D** | **ca. 50000** | **USA** | **3 work weeks** |
| Email Feedback | 34 | | | | arbitrarily during the study |
| Intermed. Feedback Survey | 26 | | | | after the first week |
| Data Upload | 33 | | | | at the end of the study |
| Final Survey | 32 | | | | following the data upload |

Table 1. Overview of the Two-Phase Study Describing the Method, Participants, their Employer and Study-Durations.

## 3 PHASE 1 METHOD: IDENTIFYING DESIGN ELEMENTS

To identify design elements for building personalized awareness tools for self-monitoring software developers' work, we defined the following research question:

**RQ1:** *What information do software developers expect and need to be aware of and how should this information be presented?*

To answer this research question, we first reviewed literature of design practices applied in existing self-monitoring tools and of measures that software developers are interested in. We also studied the barriers related work has identified towards the adoption of self-tracking technologies at the workplace, as described in the previous section. Based on our review, we defined design elements and incorporated them into our own self-monitoring prototype for work, called $WorkAnalytics_{pilot}$. We then studied software developers' use of and experience with $WorkAnalytics_{pilot}$ at work, and refined the design elements and tool based on feedback we received through five pilots and a survey.

In what follows, we describe the goals, method and participants of this first phase. Table 1 shows an overview of the pilots and survey that we conducted and situates them within the whole study procedure. The supplementary material [65] contains a list of questions for all surveys and interviews that we conducted as well as screenshots of how $WorkAnalytics_{pilot}$ looked like at various stages until the final version.

### 3.1 Pilots

To examine the features and measurements software developers are interested in and engage with for self-monitoring their work from using them in practice, rather than from doing this hypothetically through an interview or survey, we conducted a set of pilots. Our method has strong similarities to the *Design Based Research* process, where the focus is an iterative analysis, design and implementation, based on a collaboration between practitioners and researchers in a real-world setting that leads to design principles in the educational sector [11]. First, we implemented a self-monitoring prototype, $WorkAnalytics_{pilot}$, incorporating visualizations of work-related measures that we identified to be of interest to software developers in previous research from running a survey with 379 participants [54]. We then conducted a total of five pilots at four companies (see Phase 1 in Table 1 for more details). For each pilot, we had a small set of software developers use $WorkAnalytics_{pilot}$ *in situ*, gather their feedback, and use it to refine and improve the prototype before running the next pilot. Each pilot study ran between 2-4 work weeks. To gather feedback, we conducted interviews with each participant at the end of the pilot period. These interviews were semi-structured, lasted approximately 15 minutes, and focused on what participants would like to change in the application and what they learnt from the retrospection. To find and address problems early during the development, we also conducted daily 5 minute interviews with each participant during the first three pilots. In these short interviews, we gathered feedback on problems as well as changes they would like to be made to the application, and feedback on the visualizations, their representativeness of participants' work and their accuracy. Throughout this phase, we rolled out application updates with bug-fixes, updated visualizations and new features every few days. We prioritized user requests based on the feasibility of implementation and the amount of requests by participants. After 5 pilots we decided to stop since we did not gather any more new feedback and the application was running stable.

**Participants.** For the pilots, we used personal contacts and ended up with a total of 20 professional software developers, 1 female and 19 male, from four different companies of varying size and domains (Table 1). 30% reported their role to be a team lead and 70% an individual contributor—an individual who does not manage other employees. Participants had an average of 14.2 years ($\pm$9.6, ranging from 0.5 to 40) of professional software development experience.

### 3.2 Initial Survey

Following the pilot studies, we conducted a survey 1) to examine whether the measures and features that developers are interested in using for self-monitoring within the target company (company D) overlap with what we had implemented, 2) to learn how the *WorkAnalytics*$_{pilot}$ needed to be adapted to fit into the target company's existing technology set-up and infrastructure, as well as 3) to generate interest in participating in our field study. In the survey, we asked software developers about their expectations and the measurements that they would be interested in for self-monitoring their work. We advertised the survey at company D, sending invitation emails to 1600 professional software developers. To incentivize participation, we held a raffle for two 50 US$ Amazon gift certificates. The initial survey questions can be found in the supplementary material [65]. To analyze the survey, we used methods based on Grounded Theory [64] to analyze the textual data that we collected. This included Open Coding to summarize and label the responses, Axial Coding to identify relationships among the codes, and Selective Coding to factor out the overall concepts, related to what measurements and features participants expect and how their work environment looks like.

**Participants.** From the 1600 invitation emails, we received responses from 413 software developers (response rate: 25.8%), 11% female, 89% male. 91.5% of the participants reported their role to be individual contributor, 6.5% team lead, 1 manager (0.2%), and 1.8% stated they are neither. Participants had an average of 9.6 years ($\pm$7.5, ranging from 0.3 to 36) of professional software development experience.

## 4 PHASE 1 RESULTS: IDENTIFIED DESIGN ELEMENTS

To answer our first research question (**RQ1**), we analyzed related work, investigated developers' experience with pilots of *WorkAnalytics*$_{pilot}$ and analyzed the initial survey. The analysis showed that a design for a work self-monitoring approach should: A) support various individual needs, B) foster active user engagement, and C) provide multi-faceted insights into work. We incorporated these three design elements into a technology probe, *WorkAnalytics*.

*WorkAnalytics* was built with Microsoft's Dot.Net framework in C# and can be used on the Windows 7, 8 and 10 operating system. We created *WorkAnalytics* from the ground up and did not reuse an existing, similar application, such as RescueTime [58], as we wanted to freely extend and modify all features and measurements according to our participants' feedback. A screenshot of the

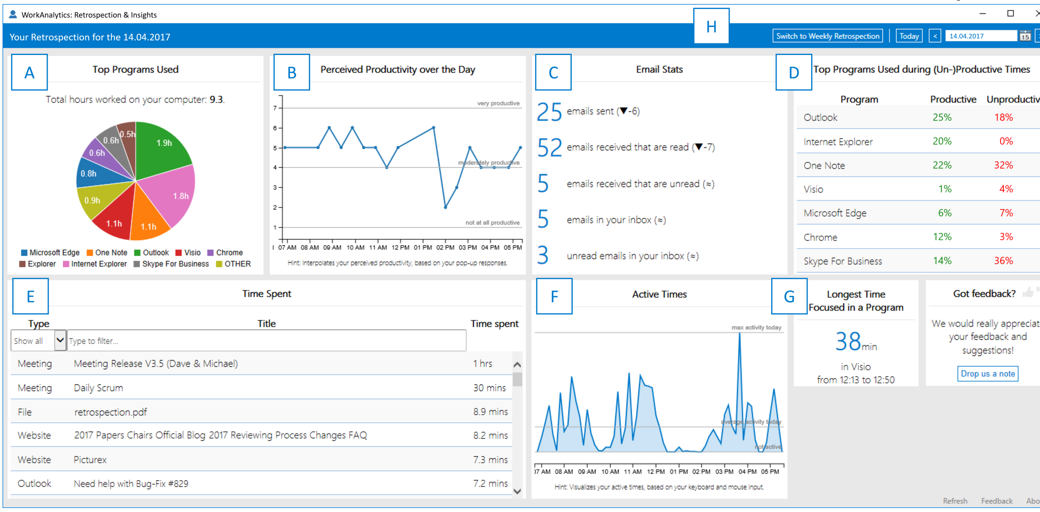


Fig. 2. Screenshot of the Daily Retrospection in *WorkAnalytics*.

main view of the application, the retrospection, is shown in Figure 2. We open-sourced *WorkAnalytics*, opening it up to contributions on GitHub [1].

### 4.1 A: Supporting Various Individual Needs

**Measurement Needs.** The analysis of our initial survey showed that participants are generally interested in a large number of different measures when it comes to the self-monitoring of work. We asked survey participants to rate their interest in a list of 30 work related measures on a five point Likert-scale from 'extremely interesting' to 'not at all interesting'. We chose these measures based on our findings from the pilot phase, on what we were capable to track, and on related work. The list includes measures on time spent in programs, meetings, and specific activities, the amount of code written, commits done, code reviews completed, emails sent and received, and the amount of interruptions experienced and focus at work. Each measure had at least 20% and up to 74% of the participants that rated it as very or extremely interesting. At the same time the combination of measures that each participant was interested in varied greatly across participants. For instance, only 6 of the 30 measures were rated as very or extremely interesting by 60% or more, and 52% of participants were interested in nearly all measures while 25% only wanted very few measures for self-monitoring at work. Overall, the greatly varying interest and the interest in a large number of measures for self-monitoring supports earlier findings by Meyer et al. [54] in the work domain and Choe et al. [14] in the activity and health domain. The complete list of the 30 work related measures, including participants' ratings about their interest in the measures, can be found in the supplementary material [65].

To support these individually varying interests in work measures, we included a wide variety of measures in our application and allowed users to individually select the measures that were tracked and visualized. To capture the relevant data for these measures, *WorkAnalytics* features multiple data trackers: the *Programs Used* tracker that logs the currently active process and window titles every time the user switches between programs or logs 'idle' in case there was no user input for more than 2 minutes; the *User Input* tracker, to collect mouse clicks, movements, scrolling and keystrokes (no key-logging, only time-stamp of any pressed key); and, the *Meetings and Email* trackers, to collect data on calendar meetings and emails received, sent and read, using the Microsoft Graph API of the Office 365 Suite [4].

[1] https://github.com/sealuzh/PersonalAnalytics

The initial version only included the Programs Used tracker, similar to RescueTime [58]. The Programs Used tracker allows the extraction of a multitude of measurements participants wished, including the time spent in specific programs and activities, such as development related activities (e.g. coding, testing, debugging, version control, and development projects worked on) and researching the web, as well as specific code files and documents worked on and websites visited. After the first two pilots, the User Input tracker was added, since 3 of the first 8 participants were interested in knowing when they were producing (e.g. typing on the keyboard) and consuming (e.g. scrolling through text with the mouse) data. Running the initial survey highlighted participants' interest in knowing more concrete details about their collaborative activities, such as planned and unplanned meetings (41%), reading and writing emails (44%), and doing code reviews (47%), which is the reason they were added to the final version of *WorkAnalytics* before running the field study.

**Privacy Needs.** A re-occurring theme during the pilots and initial survey was participants' need to keep sensitive workplace data private. Participants feared that sharing data with their managers or team members could have severe consequences on their employment or increase pressure at work. To account for privacy needs at work, *WorkAnalytics* stores all logged data only locally on the user's machine in a local database, rather than having a centralized collection on a server. This enables users to remain in control of the captured data. To further support the individual needs, the application provides actions to enable and disable data trackers manually, pause the data collection and access (and alter) the raw dataset, which was done by two participants during the field study.

### 4.2 B: Active User Engagement

To be able to generate deeper insights on a user's work and productivity and encourage users to actively reflect upon their work periodically, we decided to include a self-reporting component. Several participants of our initial survey stated interest in self-reporting some data about work that cannot be tracked automatically, in particular more high-level measures on productivity. Furthermore, related work found that users rarely engage by themselves with data captured in a self-monitoring tool, which reduces awareness and chances of positive change [17, 33, 40]. To address this point, we added a pop-up to our application that appeared periodically, by default once per hour[2], and prompted users to self-report their perceived productivity, the tasks they worked on, the difficulty of these tasks and a few other measures. During the first two pilots of our iterative development phase, we found that while the self-reporting might be valuable, it took participants several minutes to answer, and 45% of our participants reported it to be too intrusive, interrupting their work, and decreasing their productivity. As a result, many participants regularly postponed the pop-up or disabled it, which then resulted in less meaningful observations to be presented in the visualization and a smaller satisfaction by participants.

To minimize intrusiveness, yet still encourage periodic self-reflection, we reduced the number of questions in the pop-up to a single question that asks participants to rate their perceived productivity on a 7 point Likert-scale (1: not at all productive, 7: very productive) once per hour. Participants were able to answer the question with a single click or keystroke. See Figure 3 for a screenshot of the pop-up. In case the pop-up appeared at an inopportune moment, participants were able to postpone it for a few minutes, an hour or a whole work day. To further adapt the self-reports to individual preferences, each participant was able to alter the interval at which pop-ups appeared or disable/enable it.

[2]This interval was chosen as a way to balance intrusiveness. While the first two pilots had an interval of 90 minutes which made it harder for participants to remember what exactly happened in that period, most participants preferred to reflect on their productivity once an hour.

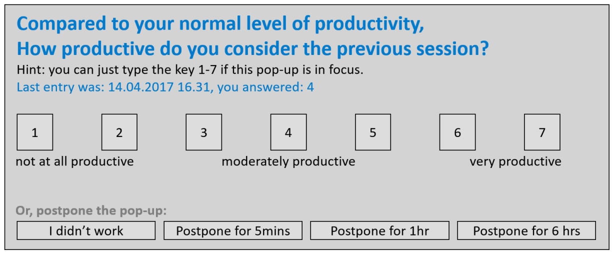


Fig. 3. Screenshot of the Self-Reporting Pop-Up to Collect Perceived Productivity Data and Engage Users.

### 4.3 C: Enabling More Multi-Faceted Insights

Related work found that self-monitoring tools often fail to provide sufficient contextual information and a more holistic picture of the monitored behavior that also allows the user to relate the data [7, 14, 33]. Similarly, 35% of pilot study participants asked for weekly summaries to get a more complete picture of the data and a way to compare and relate different work days or weeks with each other. In the initial survey, 41% of the participants wished for a visualization to drill down into the data and learn where exactly they spend their time.

To address this requirement of enabling a more complete picture of the data in our application, we focused on three aspects: providing sufficient contextual information, allowing to get a higher-level overview, and providing ways to relate various data with each other. To provide *sufficient contextual information*, we added several visualizations to the daily retrospection that illustrate how the time of a work day was spent:

- **Top Programs Used:** Pie chart displaying the distribution of time spent in the most used programs of the day (Figure 2A).
- **Perceived Productivity:** Time line illustrating the user's self-reported productivity over the course of the day (Figure 2B).
- **Email Stats:** Table summarizing email related data, such as number of emails sent & received in a work day (Figure 2C).
- **Programs & Productivity:** Table depicting the seven most used programs during the day and the amount of time the user self-reported feeling productive versus unproductive while using them (Figure 2D).
- **Time Spent:** Table showing a detailed break-down of how much time was spent on each information artefact during the work day, including websites visited, files worked on, emails sent/read, meetings in the calendar, as well as code projects and code reviews worked on (Figure 2E).
- **Active Times:** Line chart visualizing the user's keyboard and mouse input over the course of the day. We aggregated the input data by assigning heuristic weights to each input stream that we determined based on our own experience and trials in pilots, e.g. one mouse click has approximately as much weight assigned as three key strokes (Figure 2F).
- **Longest Time Focused:** Minutes that a user spent the longest inside any application without switching (Figure 2G).

For a *higher-level overview*, we added a weekly summary of the data, which shows how often which programs were used on each day of the week, the average self-reported productivity per day, and the productive versus unproductive time spent on the 7 most used programs during the week (same as Figure 2E). The supplementary material [65] contains a screenshot and description of the weekly retrospection.

Finally, to *ease the correlation of data*, as desired by 19% of the participants in the initial survey, we implemented a feature that allows users to pick days or weeks (Figure 2H) and compares them with each other side-by-side and we provide a view that correlates the most used programs during a day with productivity (Figure 2D). In addition to these features, we automatically generated personalized insights. Personalized insights are automatically generated aggregations and correlations within the captured data and presented in natural language. These personalized insights are similar to the correlation and presentation of data that Bentley et al. [10] have shown to increase users' understanding of complex connections in the area of health-monitoring and well-being. To create the personalized insights, we first created a matrix where we correlated each measure with itself (i.e. average per day), with the time of the day (i.e. morning/afternoon), and with the productivity self-reports. To avoid information overload, we just selected insights that might be interesting to users by discarding simple insights from the matrix that were already easily perceptible in the retrospection (e.g. the number of emails sent per day or user input over the day) and removed one insight that we could not produce due to the format of the collected data (number of emails sent/received over the day). For each pair, we created one or more sentences that correlate the items with each other. For example, from the pair 'self-reported productivity' and 'time of day', we created the sentence: "You feel more productive in the [morning/afternoon]" (insight 14). Three of these personalized insights address the participants' focus, which is an abstracted measure for the time spent in a single program before switching to another program. Participants were aware of the definition of focus, as one of the visualizations in the daily retrospection used the same abstraction and included a definition (Figure 2G). We created these personalized insights individually for each user and filtered the ones that were not feasible, e.g. due to participants disabling certain data trackers. Since we wanted to ensure to collect sufficient data before generating these personalized insights and also ensure that they are reasonable, we only included them in the final survey, after users shared their data logs with us. Table 3 presents a list of the 15 personal insights that resulted from this process. The matrix we created to select these insights is available and discussed in the supplementary material [65]. Future versions of *WorkAnalytics* will include the automatic generation of such personalized insights.

## 5 PHASE 2 METHOD: EVALUATING DESIGN ELEMENTS

To evaluate the design elements, and learn how software developers are using and appreciating the identified features and measurements in practice, we formulated a second research question:

**RQ2:** *How do software developers use the measurements and features based on the identified design elements during their work and what is their impact?*

To answer the research question, we conducted a field study with *WorkAnalytics* as a technology probe that implements the previously discussed design elements.

### 5.1 Participants

We recruited participants for this study by contacting the 160 software developers at company D that took our initial survey and indicated their interest in participating. 33 of the 43 participants that signed the consent form were recruited through this follow-up email, and 10 participants were recruited through recommendations from other participants. The only requirements for participating in the study were to be a software developer and to be using a work machine with the Windows operating system. Participants were given two 10 US$ meal cards at the end of the study for compensating their efforts and were promised personalized insights into their work and productivity. All 43 participants are professional software developers working in the same large software company (company D in the pilots), three of them were female and 40 male. The roles, team sizes and projects varied across the participants. 96.7% stated their role to be an individual contributor and 3.3% team lead. Participants had an average of 9.8 years (±6.6, ranging from 0.5 to 30) of professional software development experience. To avoid privacy concerns, we identified participants with a subject id and therefore could not link their responses between the different feedback surveys, emails, and collected data from *WorkAnalytics*. To get feedback on the usefulness of the different design elements from different perspectives, we picked participants with and without previous experience with other self-monitoring tools, such as Fitbit [24] or RescueTime [58].

### 5.2 Procedure

We designed this field study to last three work weeks. At the beginning of the period, we provided participants with detailed information on the study procedure, the data we were going to collect, and the features of *WorkAnalytics*. We then asked participants to install the application on their work machine, continue their regular work day and answer the periodic self-reports when they appeared, by default every 60 minutes. We asked them to contact us via email at any point in time in case they run into an issue, had questions, or suggestions, which 34 participants did once or more. At any point throughout the study, participants were able to change the time period or disable the pop-up completely. Participants could also enable or disable any trackers that logged data for presentation in the retrospection. After the first week, we sent out a short, intermediate feedback survey to collect early feedback on the usefulness, suggestions for improvement, and participants' engagement with *WorkAnalytics*. 26 participants responded. The timing was chosen to make sure participants had used the application for at least 3 to 5 work days, and the tool had captured enough data to show visualizations from various work days.

Shortly before the end of the three work weeks of the study, we asked participants to share the data that *WorkAnalytics* logged on their machine—the reported productivity ratings and the computer interactions—if they were willing to. We also gave each participant the opportunity to obfuscate any sensitive or private information that was logged, such as window titles or meeting subjects, before uploading the data to our secured server. Of the 43 participants, 33 participants shared their data with us, and three of them obfuscated the data before the upload. Due to the sensitivity of the collected data, we did not try to convince participants to share the data and just mentioned the additional insights they would receive when sharing it. We then used the data to automatically generate aggregations and correlations within an individual participant's data, which we will call personalized insights in the following. At the end of the study period, we asked participants to fill out a final survey, independently of whether they uploaded the data or not. The survey contained questions on feature usage and usefulness, possible improvements, potential new measures, and perceived changes in awareness about work and behavior. For participants that shared the collected data with us, the survey also presented the personalized insights, automatically generated for each participant, and questions about them. 32 of the 43 participants participated in the final survey, including 5 that had not previously shared their computer interaction data. The questions from the intermediate survey and final survey can be found in the supplementary material [65].

### 5.3 Data Collection and Analysis

Throughout the field study, we collected qualitative and quantitative data from participants. In particular, the responses to the intermediate feedback survey, final survey, feedback received via email, and the data that *WorkAnalytics* collected. Similar to our approach in the initial survey, we used methods common in Grounded Theory. In this case, the Axial Coding step was also used to identify higher level themes after Open Coding each feedback item separately. Besides creating personalized insights from the collected computer interaction data, we used it to analyze participants' engagement with the retrospection and the answering of the experience sampling productivity pop-up. The computer interaction data span over a period of between 9 and up to 18 work days (mean=13.5, $\pm$2.6). The findings of the analysis of the quantitative and qualitative data from our participants are discussed, and then distilled into design recommendations in the next section.

## 6 PHASE 2 RESULTS: DESIGN RECOMMENDATIONS BASED ON EVALUATING DESIGN ELEMENTS

To answer the second research question (**RQ2**), we focus our analysis of the data collected about the use of *WorkAnalytics*. For each part, we first present the findings before summarizing the design recommendations that we inferred from interpreting the results. The design recommendations are mapped to one of the three design elements (A to C) and are presented in blue boxes to distinguish them from the findings.

### 6.1 Different Granularity of Visualizations

Most participants (70.4%) agreed that the collected data and measures were interesting and relevant to them. Participants valued that the retrospection allowed them to get a high-level overview of the data and also let them drill down into more detail:

> "*Sift through all this information and quickly find what's critical and be able to determine what is furthering one's goals and what [is] not (i.e. is a distraction).*" - F19

Participants used, for instance, the pie chart on the programs executed (Figure 2A) and the active times timeline (Figure 2F) to get an aggregated overview of the past work day, in particular which activities most time was spent on and the most active and inactive times during the day, respectively. When they wanted to further investigate their day and find out more specific details, participants appreciated the availability of other visualizations:

> "*I like that [WorkAnalytics] captures who I am talking with in Skype or Google Hangouts [. . . ]. I like the integration of Outlook in more detail.*" - F42

Several participants (F13, F17, F18) reported having used the time spent table (Figure 2E) regularly to gain deeper insights on with whom they communicate—through email, instant messaging and meetings—and on which artefacts they spent time—document, website, code file, or email.

**Design Recommendation A.1**: For self-monitoring at work users are interested in a quick as well as deep retrospection on their work that are best supported through high-level overviews with interactive features to drill-down into details.

### 6.2 Interest in Diverse Set of Measurements

Participants had varying interests in the positive, negative or neutral framing of the data. For instance, while some participants (F19, F25) wanted to learn about what went well, such as the tasks they completed and how much they helped their co-workers, others were more interested in understanding what went wrong:

> "*[. . . ] focus more on things that prevent someone from being able to add business value, rather than arbitrary metrics like commit count, bug count, task completion, etc. [. . . ] I would prefer [the application] to track things that I felt got in the way of being productive.*" - F17

This framing effect in self-monitoring tools has recently been explored by Kim et al. [40], where they found out that only participants with a negative framing condition improved their productivity, while positive framing had little to no impact.

Most participants (69%) wanted *WorkAnalytics* to collect even more data on other aspects of their work to further personalize and better fit the retrospection to their individual needs. For instance, they wanted more detailed insights into collaborative and communicative behaviors by integrating data from and sharing data with other team members (6%) and generating insights into the time spent on technical discussions or helping co-workers (6%). Participants were further interested in collecting data from other work devices (13%), capturing even more coding related data (6%), such as tests and commits, or more high-level measures, such as interruptions or progress on tasks (9%). 80% of the participants were also interested in biometric data, such as heart rate or stress levels, 70% were interested in physical activity data, such as sleep or exercise, and 50% were interested in location based data, such as commute times or visited venues; all in combination with the already collected work data. Similarly, roughly one third of the participants suggested to extend the daily and weekly retrospection, by adding additional visualizations and finer-grained aggregations, to better support them in making observations based on correlations and combinations of several measurements:

> "*[The] active times graph would be neat on the weekly retrospection so that I could get a sense of my most active time of the day without having to navigate through each day.*" - F43

These very diverse requests for extending *WorkAnalytics* with further measures and visualizations emphasize the need for personalizing the experience, to increase satisfaction and engagement.

**Design Recommendation A.2**: For self-monitoring one's work, users are interested in a large and diverse set of data, even from outside of work, as well as in correlations within the data.

### 6.3 Increasing Self-Awareness with Experience Sampling

Participants actively engaged in the brief, hourly self-reports on productivity when they were working on their computer. Over the course of the study, participants self-reported their productivity regularly, on average 6.6 times a day ($\pm 3.8$, min = 1, max = 23) and it usually took them just a couple of seconds, without actually interrupting their work. Two (6%) participants even increased the frequency to answer the pop-up every 30 minutes, while 3 (9%) of the 33 participants, from whom we received data, disabled the self-reports. This shows that the experience sampling method we applied was not considered as too intrusive for most participants.

Being asked in the final survey about the value of and experience with self-reporting their productivity, 59.2% of the participants agreed or strongly agreed that the brief self-reports increased their awareness on productivity and work (see Table 2 for more detail). The self-reports helped participants to realize how they have spent their past hour at work and how much progress they have made on the current task:

> "*It makes me more conscious about where I spent my time and how productive I am.*" - F08

Some participants used the pop-up briefly reflect on whether they have used their time efficiently or not, and if they should consider changing something:

> "*The hourly interrupt helps to do a quick triage of whether you are stuck with some task/problem and should consider asking for help or taking a different approach.*" - F11

The fact that *WorkAnalytics* does not automatically measure productivity, but rather lets users self-report their perceptions, was further valued by participants as some do not think an automated measure can accurately capture an individual's productivity, similar to what was previously found [53]:

> "*One thing I like about [WorkAnalytics] a lot is that it lets me judge if my time was productive or not. So just because I was in a browser or VisualStudio doesn't necessarily mean I was being productive or not.*" - F42

> "*I am much more honest about my productivity levels when I have to self-report, [rather] than if the software simply [. . . ] decided whether or not I was productive.*" - F15

These findings suggest that using experience sampling is a feasible method to manually collect data as long as users have a benefit from their self-reporting.

**Design Recommendation B.1**: Experience sampling in the form of brief and periodic self-reports are valuable to users as they increase the awareness of their work and productivity, and lead to richer insights.

### 6.4 Increasing Self-Awareness with a Retrospection

Participants frequently accessed the daily retrospection, yet the patterns of self-monitoring varied greatly across participants. On average, participants opened the retrospection 2.5 times per day (±3.5, min=0, max=24) for a total of 0.85 minutes (±2.95, min=0, max=42.9), but both aspects varied a lot across participants as the standard deviation (±) and the minimum and maximum show. All participants opened the retrospection more often in the afternoon (mean=1.9) than in the morning (mean=0.6). Yet, 34% of participants opened the application less than 5 times over the whole study period, while 28% used the retrospection at least once a day. Also, while 31% of participants mostly focused on the current day, the other 69% looked and compared multiple work days. Many participants also looked at the weekly retrospection, but access to this one was less often than to the daily one.

While these results show that most participants were actively reflecting about their work using the retrospection, we also received feedback from 2 participants (6%) that they sometimes forgot the retrospection was available:

> "*I forgot I could even look at the retrospection! A new pop-up, maybe Friday afternoon or Monday morning prompting me to review the week's data would be really nice.*" - F14

Overall, the retrospection increased the awareness of the participating software developers and provided valuable and novel insights that they were not aware of before. Overall, participants commented on the retrospection providing novel insights on a variety of topics, such as how they spend their time at work collaborating or making progress on tasks, their productivity over the course of the day, or the fragmentation and context switches at work:

> "*Context switches are not the same as program switches, and I do *lots* of program switches. I still do a lot more context switches than I thought, but it doesn't hurt my perceived productivity.*" - F36

> "*[The] tool is awesome! It [. . . ] helped confirm some impression I had about my work and provided some surprising and very valuable insights I wasn't aware of. I am apparently spending most of my time in Outlook.*" - F42

Reflecting about the time spent at work further helped participants to sort out misconceptions they had about their work:

> "*I did not realize I am as productive in the afternoons. I always thought my mornings were more productive but looks like I just think that because I spend more time on email.*" - F14

The survey responses that are presented in Table 2 and are based on a 5-point Likert-scale (5: strongly agree, 1: strongly disagree) further support these findings. 81.5% of all survey participants reported that installing and running the application increased their awareness, and 59.2% agreed or strongly agreed that they learnt something about their work and productivity, while only 11.1% did not. The responses also show that the retrospection helped participants in particular to learn how they spend their time (85.2% agreed or strongly agreed) and about productive and unproductive times (62.9%).

**Design Recommendation B.2**: Reflecting about work using a retrospective view provides novel and valuable insights and helps to sort out misconceptions about activities pursued at work.

### 6.5 Personalized Insights

The personalized insights that we presented to 27 of the 32 participants in the final survey are based on the same measurements as the ones that are visualized in the retrospection. These insights were created based on correlations and aggregations within the collected data and presented as natural language sentences. The specific insights are presented in Table 3 and details on their creation can be found in Section 4.3. To learn more about the value of the visualizations and the natural language

| | Strongly agree | Agree | Neutral | Disagree | Strongly disagree | N/A |
|---|---|---|---|---|---|---|
| The collected and visualized data is relevant to me. | 18.5% | 51.9% | 22.2% | 7.4% | 0.0% | 0.0% |
| I learned something about my own work and perceived productivity by looking at the retrospection and reflecting. | 29.6% | 29.6% | 25.9% | 11.1% | 0.0% | 3.7% |
| Answering the perceived productivity pop-up questions increased my awareness about my work and perceived productivity. | 18.5% | 40.7% | 25.9% | 7.4% | 7.4% | 0.0% |
| Installing and running the tool raised my awareness about my work and perceived productivity. | 22.2% | 59.3% | 11.1% | 3.7% | 3.7% | 0.0% |
| I used the daily retrospection to reflect about my past work day. | 11.1% | 37.0% | 11.1% | 29.6% | 7.4% | 3.7% |
| I used the weekly retrospection to reflect about my past work week. | 11.5% | 30.8% | 23.1% | 23.1% | 7.7% | 3.8% |
| The retrospection helps me to learn how I spend my time. | 29.6% | 55.6% | 0.0% | 11.1% | 0.0% | 3.7% |
| The retrospection helps me to learn more about my perceived productive times. | 25.9% | 33.3% | 25.9% | 7.4% | 3.7% | 3.7% |
| I now know more about why and when I feel I am productive or unproductive. | 22.2% | 40.7% | 14.8% | 18.5% | 3.7% | 0.0% |
| I tried to change some of my habits or patterns based on what I learned from reflecting about my work. | 14.8% | 25.9% | 11.1% | 40.7% | 3.7% | 3.7% |

Table 2. Survey Responses on Awareness Change.

insights, we asked participants to rate the novelty of each personalized insight. Participants' responses were mixed with respect to the novelty of the automatically generated personalized insights that presented correlations and aggregates within the data in natural language. When rated on a scale from 'extremely novel' to 'not novel at all', only 5 of the 15 personalized insights (see personalized insights in Table 3 marked with an asterisk '*') were rated as 'very novel' or 'extremely novel' by more than half of the participants. This means that participants gained knowledge about most insights either before or during the study. The five insights that were rated as 'very novel' or 'extremely novel' by more than half of the participants are all correlations between two distinct data categories, so called multi-faceted correlations [37], rather than simple aggregates, called uni-faceted correlations, which are easier to understand from simple visualizations [10, 27]. One participant also suggested to integrate these novel personalized insights into the retrospection since it was easier to draw connections between two distinct data categories using natural-language statements, similar to what Bentley et al. [10] found. Research by Jones and Kelly [37] has shown that multi-faceted correlations presented by self-monitoring tools are of higher interest to users than uni-faceted correlations. Paired with our findings above, this suggests to use visualizations for presenting uni-faceted correlations and to present more complex multi-faceted correlations using natural language sentences. Future work could further investigate the effectiveness of these personalized insights and their impact on behavior at work.

**Design Recommendation C.1**: Present multi-faceted correlations using natural language, as users often miss them from reflecting with visualizations.

## 6.6 Potential Impact on Behavior at Work

When we explicitly asked participants if they think they actually changed their behavior during the field study based on the insights they received from using the application, 40.7% reported that they have changed some of their habits based on what they learnt from reflecting about their work. Participants mentioned to be trying to better plan their work (6%), e.g. by taking advantage of their more productive afternoons, trying to optimize how they spend their time with emails (13%), or trying to focus better and avoid distractions (19%).

40.7% of the participants self-reported that they did not change their behavior, either because they did not want to change something (6%) or they were not sure yet what to change (13%). The latter ones mentioned that they needed more time to self-monitor their current behavior and learn more

| | Novelty | | | | Behavior Change | | |
|---|---|---|---|---|---|---|---|
| | extremely | very | somewhat | not | yes | no | dk |
| 1. The program you spend most time is X, followed by Y. | 4.0% | 16.0% | 36.0% | 44.0% | 24.0% | 68.0% | 8.0% |
| 2. The program you switch to the most is X. | 11.5% | 30.8% | 34.6% | 23.1% | 23.1% | 65.4% | 11.5% |
| 3. You spend X% of the time on your computer in program X, Y, and Z. | 0.0% | 32.0% | 32.0% | 36.0% | 28.0% | 60.0% | 12.0% |
| 4. X is the program you focus on the longest. | 17.4% | 21.7% | 26.1% | 34.8% | 17.4% | 73.9% | 8.7% |
| 5. You feel [more/less] productive when you are focused less. * | 23.5% | 29.4% | 17.6% | 29.4% | 52.9% | 23.5% | 23.5% |
| 6. When you feel productive, you spend more time in program X than in Y. | 15.0% | 20.0% | 10.0% | 55.0% | 30.0% | 45.0% | 25.0% |
| 7. When you feel unproductive, you spend more time in program X than in Y. * | 27.8% | 22.2% | 22.2% | 27.8% | 38.9% | 38.9% | 22.2% |
| 8. You spend more time in Outlook in the [morning/afternoon] than [afternoon/morning]. | 4.8% | 28.6% | 33.3% | 33.3% | 23.8% | 66.7% | 9.5% |
| 9. You usually work more focused in the [morning/afternoon]. * | 26.1% | 30.4% | 34.8% | 8.7% | 52.2% | 43.5% | 4.3% |
| 10. On average, you spend X hours on your computer per work day. | 31.8% | 18.2% | 22.7% | 27.3% | 45.5% | 40.9% | 13.6% |
| 11. You feel more productive on days you spend [more/less] time on your computer. * | 23.5% | 35.3% | 11.8% | 29.4% | 35.3% | 64.7% | 0.0% |
| 12. You feel [more/less] productive when you send more emails. | 14.3% | 14.3% | 42.9% | 28.6% | 35.7% | 57.1% | 7.1% |
| 13. You feel [more/less] productive when you have more meetings. | 10.0% | 20.0% | 50.0% | 20.0% | 40.0% | 50.0% | 10.0% |
| 14. You usually feel more productive in the [morning/afternoon]. | 8.7% | 34.8% | 39.1% | 13.0% | 39.1% | 47.8% | 13.0% |
| 15. You usually take X long breaks (15+ minutes) and Y short breaks (2-15 minutes) from your computer per day. * | 21.7% | 52.2% | 17.4% | 8.7% | 43.5% | 47.8% | 8.7% |

Table 3. Participants' Ratings on the Novelty and Potential for Behavior Change of Personalized Insights.

about their habits, and that *WorkAnalytics* does not offer much help yet in incentivizing or motivating them to change their behavior. In particular, participants stated that the visualizations and correlations were not concrete and actionable enough for knowing what or how to change:

> "*While having a retrospection on my time is a great first step, I gained [. . . ] interesting insights and realized some bad assumptions. But ultimately, my behavior didn't change much. Neither of them have much in way of a carrot or a stick.*" - F42

> "*It would be nice if the tool could provide productivity tips - ideally tailored to my specific habits and based on insights about when I'm not productive.*" - F15

Several participants went on to make specific recommendations for more concrete and actionable support to motivate behavior change. These recommendations ranged from pop-ups to encourage more focused work, to recommend a break from work, all the way to intervening and blocking certain applications or web sites for a certain time:

> "*If [the tool] thinks I am having a productive day, it should just leave me alone and not ask any questions. If I am having an unproductive day and [it] can help me overcome it (e.g. go home and get some sleep) the tool should suggest that.*" - F10

> "*Warnings if time on unproductive websites exceeds some amount, and perhaps provide a way for the user to block those sites (though not forced).*" - F29

When we explicitly asked participants to rate whether or not the 15 personalized insights make them think about or plan their work differently, results indicated that most of the 15 personalized insights are again not actionable enough to foster a behavior change (see results on the right sight of Table 3). The five insights with the highest potential (between 40% and 52.9% of participants agreed) are mostly related to work fragmentation and focus on work.

**Design Recommendation C.2**: Self-monitoring insights often need to be very concrete and actionable to foster behavior change at work.

## 7 DISCUSSION

This section discusses implications that emerged from our study with respect to long-term user engagement, awareness about team-work and collaborations and, ultimately, behavior change.

### 7.1 Design for Personalization

One of our goals was to find out whether the expectations of software developers for a self-monitoring approach are similar or if they are diverging. While existing commercial self-monitoring tools to quantify our lives, such as the Fitbit [24], offer only few options for personalization and are still successful at motivating users to live a healthier life [26, 47], our results on self-monitoring at work suggest that personalization is crucial.

In the pilot studies and the field study, participants uniformly expected different measurements to be visualized at different levels of granularity, similar to findings in other areas [42, 52]. These individual expectations might be explained by the very different types of tasks and work that software developers, even with very similar job profiles, have to accomplish [53]. The ability to customize the measurements that are being captured and how they are visualized is one way to support the personalization. This customizability could not only foster interest in long-term usage, as data relevant to the user is available, but could also reduce privacy concerns that software developers might have.

While many participants were initially skeptical about self-monitoring their work, we received no privacy complaints and most participants (33 of 43) even shared their data with us for the analysis. Almost all participants even went one step further: after a few days of using *WorkAnalytics* and becoming certain that their data is treated confidentially, they started to comment about possible extensions and additional measures for their self-monitoring at work. This includes more insights about their collaborative activities with other people, as discussed in more detail later in this section, but also adding even more measurements specific to their job as software developers, such as the commits they make to the version control tool or insights into their patterns of testing and debugging code.

While it might seem surprising that developers requested many development-unrelated measures for self-monitoring their work, this can be explained by the amount of time they spend with development related activities, on average between 9% and 21%, versus other activities, such as collaborating (45%) or browsing the web (17%) [29, 53]. As most study participants (84.6%) were interested to continue using *WorkAnalytics* after the study had ended, we concluded that the initially identified design elements to support various individual needs, actively engage the user, and enable more multi-faceted insights are valuable for self-monitoring at work.

### 7.2 Increased Engagement through Experience Sampling

As noted in previous research, many self-monitoring approaches suffer from an extremely low user engagement with the data [17, 33, 40]. For example, RescueTime, which visualizes the captured data on a dashboard in the browser, was found to be used only a few seconds per day (mean=4.68 $\pm$12.03) [17]. Similar to the reports in our field study, participants' reasons for this low engagement might be that users forget about the availability of the data visualizations. A simple and periodic reminder, e.g., to let users know that there is a new summary on the work week, might increase the engagement with these visualizations and dashboards. Recently, researchers have explored how adding an ambient widget and presenting a summary of the captured data always visible on the user's screen can increase the engagement with the data (e.g., [17, 40, 71]). For example, the ambient widget by Kim et al [40] increased the use of RescueTime to about a minute a day.

In this paper, we assessed another approach, namely a periodic pop-up to self-report productivity. Our findings show that the self-report helped users to quickly reflect on how efficiently they spent their time, which then also resulted in an increased engagement. Our results show that using experience sampling is a feasible method to manually collect data that is difficult to capture automatically and is (mostly) appreciated as long as users have a benefit from self-reporting, e.g. by getting additional or more fine-grained insights. It is up to future work to determine how long the positive effects of self-reporting or ambient widgets lasts, whether users might at some point loose interest after having learnt 'enough' about their work, and whether it might be beneficial to only include these features in certain time periods. More research is required to understand how this can be generalized to other domains.

### 7.3 Actionability for Behavior Change

Most health and sports tracking systems have been shown to foster positive behavior changes due to increased self-awareness. In our study, 40.7% of the participants explicitly stated that the increased self-awareness motivated them to adapt their behavior. While motivating changes in behavior was not a primary goal, the study gave valuable insights into where and how self-monitoring tools at work could support developers in the process. The very diverse set of insights in *WorkAnalytics* that participants wished for, made it more difficult to observe a specific problem behavior and define a concrete, actionable goal for a behavior change, which is a basic requirement for starting a change according to the theory of behavior change process TTM [57]. Rather than just enabling an increased self-awareness, it might also be important to provide users with concrete recommendations for active interventions and alerts when certain thresholds are reached. Participants suggested to block distracting websites after the user spent a certain amount of time on them, or to suggest a break after a long time without one, similar to what was recently suggested [1, 23]. At the same time, not all insights are actionable as developers sometimes have little power to act on an insight, similar to what Mathur et al. found from visualizing noise and air quality at the workplace [51]. As an example, most developers can likely not just stop reading and responding to emails. Another extension to possibly make insights more actionable is to let users formulate concrete behavior change goals based on the insights they make from using the retrospection and experience sampling component. For example, a user could set a goal to regularly take a break to relax or to have an empty email inbox at the end of the day. This goal setting component could leverage experience sampling further and learn when and how users are interested and open to receive recommendations of how to better reach their goal.

Approaches aiming to foster long-term behavior changes need to offer means to actively monitor and maintain a behavior change [57] and help avoiding lapses, a frequent reason for abandoning behavior change goals [1]. In the future we plan to experiment with and evaluate these different forms of how insights could be improved to make them more actionable, and then evaluate the longer-term impact of *WorkAnalytics* on software developers' actual behavior at work.

### 7.4 Benchmarking

A re-occurring feedback by participants was the wish for a way to benchmark their work behavior and achievements with their team or other developers with similar job profiles and to improve their work habits based on the comparisons with others, similar to what was previously described by Wood [72]. Given the privacy concerns at work, adding such a component to the self-monitoring for work could, however, severely increase pressure and stress for users who are performing below average. Also, given our participants' interest in a high variety and large set of work related measures indicates that even within one domain—software developers in our case—users might work on fairly different tasks and that it might be impossible to find a 'fair' set of measures for comparing and benchmarking individuals. More research is needed to examine how and in which contexts such a social feature might be beneficial as well as which aggregated measures might be used for some sort of comparison without privacy concerns. For example, one could anonymously collect the data related to developers' work habits, such as fragmentation, time spent on activities, and achievements, combine them with job profile details and then present personalized insights and comparisons to other developers with a similar job profile. One such insight could be to let the developer know that others spend more time reading development blogs to educate themselves or that they usually have less meetings in a work week. Besides having anonymous comparisons between developers, it could further be beneficial to let users compare their work habits with their previous self, e.g. from one month ago, and enable them to reflect on how their behaviors change over time. Although research has shown that benchmarking features in physical activity trackers foster competition with peers to be more active [26, 60], additional research is needed to determine whether they also lead to a positive behavior change at the workplace.

### 7.5 Team-Awareness

Even though most insights available within *WorkAnalytics* appear to be about each user's own work habits, some insights also reveal details about the individuals' collaboration and communication patterns with their team and other stakeholders. These are, for example, insights about their meeting, email, instant messaging, social networking, and code review behavior. Nonetheless, participants were interested in even more measures, especially with respect to revealing (hidden) collaboration and communication patterns within their teams. Having detailed insights into how the team coordinates and communicates at work could help developers make more balanced adjustments with respect to the impact their behavior change might have on their team. For example, being aware of co-workers' most and least productive times could help to schedule meetings at more optimal times, similar to what Begole suggested for teams distributed across time zones [8]. Related to an approach suggested by Anvik et al. where work items and bug reports were automatically assigned to developers based on previously assigned and resolved work items [3], it could be beneficial for improving the coordination and planning of task assignments by also taking into account each developer's current capacity and workload. Being more aware of the tasks each member of the team is currently working on and how much progress they are making could also be useful for managers or team leads to identify problems early, e.g. a developer who is blocked on a task [36] or uses communication tools inefficiently [63], and take appropriate action. A similar approach, WIPDash, has been shown to improve daily stand-up meetings by providing teams with shared dashboard summaries of work items each developer was assigned to and has completed, as these dashboards increase the awareness about each developer's progress on tasks [36]. Visualizing the current productivity and focus to co-workers could prevent interruptions at inopportune moments, where resuming the interrupted task might be more costly than at a moment of low focus. To streamline inopportune interruptions at work, Züger et al. suggested to visualize the current focus to the team by using a "traffic light like lamp" [73].

As the envisioned additions and extensions to *WorkAnalytics* might increase an individual's productivity, they might negatively affect the overall team productivity or the collaboration within teams. For example, a developer who is stuck on a task cannot ask a co-worker for help that blocks out interruptions. This is why self-monitoring tools for teams at work could not only motivate a collective improvement of the team-productivity, but also help to monitor the success and impact of these changes on other stakeholders. Future work could explore how self-monitoring at work supports team collaboration, by analyzing collaboration practices within development teams and comparing them to other teams. This work could be based on the Model of Regulation, recently introduced by Mendez et al. [5], as it helps to systematically evaluate and understand how teams self-regulate their own tasks and activities, other team-members, and how they create a shared understanding of their project goals.

## 8 GENERALIZABILITY AND LIMITATIONS

We focused our work on one type of knowledge workers, software developers, to gather insights into one work domain before generalizing to a broader range of knowledge workers in the future. Software developers have been referred to as the *knowledge worker prototype* as they are often not only the first ones to use and tweak tools, but also have lower barriers for building and improving tools themselves [39]. While software developers experience extensive interaction and collaboration with co-workers through their computer use, we believe that many of the observations made from building and evaluating *WorkAnalytics* with developers are also helpful for self-monitoring tools in other work domains, especially since the studied features and most tracked measures can be re-used in or easily ported to other domains.

The main threat to the validity and generalizability of our results is the external validity, due to the selection of field study participants that were all from the same company and had limited gender diversity. We tried to mitigate these threats by advertising the study and selecting participants from different teams in the company, at different stages of their project, and with varying amounts of experience. Participants tested *WorkAnalytics* over a duration of several weeks and were studied in their everyday, real-world work environment and not in an experimental exercise. Moreover, the development of the application was designed together with participants from three other companies of varying size, reducing the chance that we built an application that is just useful for software developers at one company. Although our findings shed light on how awareness and engagement can be increased, it is not clear how *WorkAnalytics* affects software developers using it over longer than the three-week period studied. We are aware that there is a certain self-selection bias towards participants who are in general more willing to quantify various aspects of their life, and use the collected data to increase their awareness.

## 9 CONCLUSION

One way to improve the productivity and well-being of knowledge workers is to increase their self-awareness about productivity at work through self-monitoring. Yet, little is known about the expectations of and experience with self-monitoring at the workplace and how it impacts software developers, one community of knowledge workers on which we focused. Based on previous work, an iterative development process with 5 pilot studies and a survey with 413 developers, we factored out design elements that we implemented and refined with *WorkAnalytics* as a technology probe for self-monitoring at work. We then evaluated the effect of these design elements on self-awareness of patterns of work and productivity and their potential impact on behavior change with 43 participants in a field study, resulting in design recommendations.

We found that experience sampling, using minimal-intrusive self-reporting, and the retrospective summary of the data enhances the users' engagement and increases their awareness about work and productivity. Participants reported that by using our self-monitoring approach, they have made detailed observations into how they spend their time at work collaborating or working on tasks, when they usually feel more or less productive, and sort out misconceptions they had about their activities pursued at work, such as spending a surprisingly high amount of time collaborating with others via email. Our work provides a set of design recommendations for building self-monitoring tools for developers' work and possibly other types of knowledge workers. We discuss potential future work to further increase engagement with the data and to enhance the insights' actionability by providing users with recommendations to improve their work, by adding social features to motivate users to compete with their peers, and by increasing the team awareness to help teams reduce interruptions, improve the scheduling of meetings, and the coordination of task assignments.

## 10 ACKNOWLEDGEMENTS

The authors would like to thank the study participants and the anonymous reviewers for their valuable feedback.

## REFERENCES


[1] Elena Agapie, Daniel Avrahami, and Jennifer Marlow. 2016. Staying the Course: System-Driven Lapse Management for Supporting Behavior Change. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*.

[2] Jessica S. Ancker and David Kaufman. 2007. Rethinking health numeracy: A multidisciplinary literature review. *Journal of the American Medical Informatics Association* 14, 6 (2007), 713–721.

[3] John Anvik, Lyndon Hiew, and Gail C. Murphy. 2006. Who Should Fix This Bug?. In *Proceedings of the 28th International Conference on Software Engineering (ICSE '06)*. ACM, 361–370.

[4] Microsoft Graph API. 2017. https://graph.microsoft.io. (2017). Retrieved July 9, 2017.

[5] Maryi Arciniegas-Mendez, Alexey Zagalsky, Margaret-Anne Storey, and Allyson Fiona Hadwin. 2017. Using the Model of Regulation to Understand Software Development Collaboration Practices and Tool Support. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17)*. ACM, 1049–1065.

[6] Alberto Bacchelli and Christian Bird. 2013. Expectations, Outcomes, and Challenges of Modern Code Review. In *Proceedings of the 2013 International Conference on Software Engineering*. 712–721.

[7] Lyn Bartram. 2015. Design Challenges and Opportunities for Eco-Feedback in the Home. *IEEE Computer Graphics and Applications* 35, 4 (2015).

[8] James B. Begole, John C. Tang, Randall B. Smith, and Nicole Yankelovich. 2002. Work Rhythms: Analyzing Visualizations of Awareness Histories of Distributed Groups. 230 (2002).

[9] M. Beller, I. Levaja, A. Panichella, G. Gousios, and A. Zaidman. 2016. How to Catch 'Em All: WatchDog, a Family of IDE Plug-Ins to Assess Testing. In *2016 IEEE/ACM 3rd International Workshop on Software Engineering Research and Industrial Practice (SER IP)*. 53–56.

[10] Frank Bentley, Konrad Tollmar, Peter Stephenson, and Levy Laura. 2013. Health Mashups: Presenting Statistical Patterns between Wellbeing Data and Context in Natural Language to Promote Behavior Change. 20, 5 (2013), 1–27.

[11] Ann Brown. 1992. Design Experiments: Theoretical and Methodological Challenges in Creating Complex Interventions in Classroom Settings. *Journal of the Learning Sciences* 2, 2 (1992), 141–178.

[12] Chloë Brown, Christos Efstratiou, Ilias Leontiadis, Daniele Quercia, and Cecilia Mascolo. 2013. Tracking Serendipitous Interactions: How Individual Cultures Shape the Office. *CoRR* (2013).

[13] Rafael A. Calvo and Dorian Peters. 2014. *Self-Awareness and Self-Compassion*. MIT Press, 304.

[14] Eun Kyoung Choe, Nicole B. Lee, Bongshin Lee, Wanda Pratt, and Julie a. Kientz. 2014. Understanding quantified-selfers' practices in collecting and exploring personal data. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems (CHI '14)* (2014), 1143–1152.

[15] Jan Chong and Rosanne Siino. 2006. Interruptions on software teams: a comparison of paired and solo programmers. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. ACM, 29–38.

[16] Codealike. 2017. http://codealike.com. (2017). Retrieved July 9, 2017.

[17] Emily I. M. Collins, Anna L. Cox, Jon Bird, and Daniel Harrison. 2014. Social Networking Use and RescueTime: The Issue of Engagement. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '14 Adjunct)*. ACM, 687–690.

[18] Sunny Consolvo, Predrag Klasnja, David W. McDonald, Daniel Avrahami, Jon Froehlich, Louis LeGrand, Ryan Libby, Keith Mosher, and James A. Landay. 2008. Flowers or a Robot Army?: Encouraging Awareness & Activity with Personal, Mobile Displays. In *Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp '08)*. ACM, 54–63.

[19] Sunny Consolvo, David W. McDonald, Tammy Toscos, Mike Y Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby, Ian Smith, and James A. Landay. 2008. Activity sensing in the wild: a field trial of ubifit garden. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, 1797–1806.

[20] Kate Crawford, Jessa Lingel, and Tero Karppi. 2015. Our metrics, ourselves: A hundred years of self-tracking from the weight scale to the wrist wearable device. *European Journal of Cultural Studies* 18 (2015), 479–496.

[21] Mary Czerwinski, Eric Horvitz, and Susan Wilhite. 2004. A diary study of task switching and interruptions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 175–182.

[22] Tom DeMarco and Tim Lister. 1985. Programmer performance and the effects of the workplace. In *Proceedings of the 8th international conference on Software engineering*. IEEE Computer Society Press, 268–272.

[23] Daniel A Epstein, Daniel Avrahami, and Jacob T Biehl. 2016. Taking 5: Work-Breaks, Productivity, and Opportunities for Personal Informatics for Knowledge Workers. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*.

[24] Fitbit. 2017. http://fitbit.com. (2017). Retrieved July 9, 2017.

[25] B. J. Fogg. 2003. *Persuasive Technology: Using Computers to Change What We Think and Do*. Elsevier Science.

[26] Thomas Fritz, Elaine M. Huang, Gail C. Murphy, and Thomas Zimmermann. 2014. Persuasive Technology in the Real World: A Study of Long-term Use of Activity Sensing Devices for Fitness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, 487–496.

[27] Mirta Galesic and Rocio Garcia-Retamero. 2011. Graph literacy a cross-cultural comparison. *Medical Decision Making* 31, 3 (2011), 444–457.

[28] Roland Gasser, Dominique Brodbeck, Markus Degen, Jürg Luthiger, Remo Wyss, and Serge Reichlin. 2006. Persuasiveness of a mobile lifestyle coaching application using social facilitation. In *International Conference on Persuasive Technology*. Springer, 27–38.

[29] Márcio Kuroki Gonçalves, Leidson de Souza, and Víctor M. González. 2011. Collaboration, Information Seeking and Communication: An Observational Study of Software Developers' Work Practices. *Journal of Universal Computer Science* 17, 14 (2011), 1913–1930.

[30] Victor M. González and Gloria Mark. 2004. Constant, Constant, Multi-tasking Craziness: Managing Multiple Working Spheres. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, 113–120.

[31] G. Hofstede. 1994. *Cultures and Organizations: Software of the Mind : Intercultural Cooperation and Its Importance for Survival*. HarperCollins.

[32] Victoria Hollis, Artie Konrad, and Steve Whittaker. 2015. Change of Heart: Emotion Tracking to Promote Behavior Change. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)* (2015), 2643–2652.

[33] Dandan Huang, Melanie Tory, and Lyn Bartram. 2016. A Field Study of On-Calendar Visualizations. In *Proceedings of Graphics Interface 2016*. 13–20.

[34] Hubstaff. 2017. http://hubstaff.com. (2017). Retrieved July 9, 2017.

[35] Watts S Humphrey. 2000. The Personal Software Process SM (PSP SM). November (2000).

[36] Mikkel R Jakobsen, Roland Fernandez, Mary Czerwinski, Kori Inkpen, Olga Kulyk, and George G. Robertson. 2009. WIPDash: Work Item and People Dashboard for Software Development Teams. In *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part II*. Springer-Verlag, 791–804.

[37] Simon L. Jones and Ryan Kelly. 2017. Dealing With Information Overload in Multifaceted Personal Informatics Systems. *Human Computer Interaction* (2017), 1–48.

[38] Matthew Kay, Eun Kyoung Choe, Jesse Shepherd, Benjamin Greenstein, Nathaniel Watson, Sunny Consolvo, and Julie A. Kientz. 2012. Lullaby: A Capture and Access System for Understanding the Sleep Environment. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12)*. ACM, 226–234.

[39] Allan Kelly. 2008. *Changing Software Development: Learning to Become Agile*. Wiley.

[40] Young-Ho Kim, Jae Ho Jeon, Eun Kyoung Choe, Bongshin Lee, Kwonhyun Kim, and Jinwook Seo. 2016. TimeAware: Leveraging Framing Effects to Enhance Personal Productivity. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. 272–283.

[41] Predrag Klasnja, Sunny Consolvo, and Wanda Pratt. 2011. How to Evaluate Technologies for Health Behavior Change in HCI Research. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3063–3072.

[42] Saskia Koldijk, Mark Van Staalduinen, Stephan Raaijmakers, and Wessel Kraaij. 2011. Activity-Logging for Self-Coaching of Knowledge Workers. 0–3.

[43] Ian Li, Anind Dey, and Jodi Forlizzi. 2010. A stage-based model of personal informatics systems. *Proceedings of the 28th international conference on Human factors in computing systems (CHI '10)* (2010), 557.

[44] Ian Li, Anind Dey, and Jodi Forlizzi. 2011. Understanding my data, myself: supporting self-reflection with Ubicomp technologies. In *Proceedings of the 13th international conference on Ubiquitous computing (UbiComp '11)*. 405.

[45] Paul Luo Li, Andrew J. Ko, and Jiamin Zhu. 2015. What Makes a Great Software Engineer?. In *Proceedings of the 37th International Conference on Software Engineering - Volume 1 (ICSE '15)*. IEEE Press, 700–710.

[46] James Lin, Lena Mamykina, Silvia Lindtner, Gregory Delajoux, and Henry Strub. 2006. Fish'n'Steps: Encouraging Physical Activity with an Interactive Computer Game. In *UbiComp 2006: Ubiquitous Computing*. Lecture Notes in Computer Science, Vol. 4206. Chapter 16, 261–278.

[47] Lena Mamykina, Elizabeth Mynatt, Patricia Davidson, and Daniel Greenblatt. 2008. MAHI: Investigation of Social Scaffolding for Reflective Thinking in Diabetes Management. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, 477–486.

[48] Manictime. 2017. http://manictime.com. (2017). Retrieved July 9, 2017.

[49] Gloria Mark, Shamsi T. Iqbal, Mary Czerwinski, Paul Johns, and Akane Sano. 2016. Email Duration, Batching and Self-interruption: Patterns of Email Use on Productivity and Stress. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*, Vol. 21. 98–109.

[50] Gloria Mark, Shamsi T. Iqbal, Mary Czerwinski, Paul Johns, and Akane Sano. 2016. Neurotics Can't Focus: An in situ Study of Online Multitasking in the Workplace. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1739–1744.

[51] Akhil Mathur, Marc Van Den Broeck, Geert Vanderhulst, Afra Mashhadi, and Fahim Kawsar. 2015. Tiny Habits in the Giant Enterprise: Understanding the Dynamics of a Quantified Workplace. In *Proceedings of the Joint International Conference on Pervasive and Ubiquitous Computing and the International Symposium on Wearable Computers (Ubicomp/ISWC'15)*. 577–588.
[52] Daniel McDuff, Amy Karlson, and Ashish Kapoor. 2012. AffectAura: an Intelligent System for Emotional Memory. ACM.
[53] André N. Meyer, Laura E Barton, Gail C Murphy, Thomas Zimmermann, and Thomas Fritz. 2017. The Work Life of Developers: Activities, Switches and Perceived Productivity. *Transactions of Software Engineering* (2017), 1–15.
[54] André N. Meyer, Thomas Fritz, Gail C. Murphy, and Thomas Zimmermann. 2014. Software Developers' Perceptions of Productivity. In *Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2014)*. ACM, 19–29.
[55] Rosemery O. Nelson and Steven C. Hayes. 1981. Theoretical explanations for reactivity in self-monitoring. *Behavior Modification* 5, 1 (1981), 3–14.
[56] Dewayne E. Perry, Nancy A. Staudenmayer, and Lawrence G. Votta. 1994. People, Organizations, and Process Improvement. *IEEE Software* 11, 4 (1994), 36–45.
[57] James O. Prochaska and Wayne F. Velicer. 1997. The Transtheoretical Change Model of Health Behavior. *American Journal of Health Promotion* 12, 1 (1997), 38–48.
[58] RescueTime. 2017. http://rescuetime.com. (2017). Retrieved July 9, 2017.
[59] John Rooksby, Parvin Asadzadeh, Mattias Rost, Alistair Morrison, and Matthew Chalmers. 2016. Personal Tracking of Screen Time on Digital Devices. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 284–296.
[60] John Rooksby, Mattias Rost, Alistair Morrison, and Matthew Chalmers. 2014. Personal Tracking as Lived Informatics. (2014).
[61] Slife. 2017. http://www.slifelabs.com. (2017). Retrieved July 9, 2017.
[62] Margaret-Anne Storey, Leif Singer, Brendan Cleary, Fernando Figueira Filho, and Alexey Zagalsky. 2014. The (R)Evolution of Social Media in Software Engineering. In *FOSE 2014 Proceedings of the on Future of Software Engineering*. 100–116.
[63] Margaret Anne Storey, Alexey Zagalsky, Fernando Figueira Filho, Leif Singer, and Daniel M. German. 2017. How Social and Communication Channels Shape and Challenge a Participatory Culture in Software Development. *IEEE Transactions on Software Engineering* 43, 2 (2017), 185–204.
[64] Anselm Strauss and Juliet Corbin. 1998. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*.
[65] Link to Supplementary Material. 2017. https://doi.org/10.5281/zenodo.884051. (2017).
[66] Tammy Toscos, Anne Faber, Shunying An, and Mona P Gandhi. 2006. Chick Clique : Persuasive Technology to Motivate Teenage Girls to Exercise. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*. 1873–1878.
[67] Christoph Treude, Fernando Figueira Filho, and Uirá Kulesza. 2015. Summarizing and Measuring Development Activity. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*. 625–636.
[68] Christoph Treude and Margaret-Anne Storey. 2010. Awareness 2.0: Staying Aware of Projects , Developers and Tasks using Dashboards and Feeds. In *2010 ACM/IEEE 32nd International Conference on Software Engineering*. 365–374.
[69] Bogdan Vasilescu, Kelly Blincoe, Qi Xuan, Casey Casalnuovo, Daniela Damian, Premkumar Devanbu, and Vladimir Filkov. 2016. The Sky is Not the Limit: Multitasking on GitHub Projects. 994–1005.
[70] Wakatime. 2017. http://wakatime.com. (2017). Retrieved July 9, 2017.
[71] Steve Whittaker, Victoria Hollis, and Andrew Guydish. 2016. Don't Waste My Time: Use of Time Information Improves Focus. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*.
[72] Joanne V. Wood. 1989. Theory and research concerning social comparisons of personal attributes. *Psychological Bulletin* 106, 2 (1989), 231–248.
[73] Manuela Züger, Christopher Corley, André N. Meyer, Boyang Li, Thomas Fritz, David Shepherd, Vinay Augustine, Patrick Francis, Nicholas Kraft, and Will Snipes. 2017. Reducing Interruptions at Work: A Large-Scale Field Study of FlowLight. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. 61–72.

Received May 2017; revised July 2017; accepted November 2017