University of Zurich
Department of Informatics

s.e.a.l.
software evolution & architecture lab

FeedBag Dashboard

Master Project

Master Project
Project period
Severin Siffert
severin.siffert@uzh.ch
Timothy Zemp
timothy.zemp@uzh.ch
Sarah Zurmhle
sarah.zurmuehle2@uzh.ch
Sebastian Proksch

1 Project Description

1.1 Overview

1.2 Scope of the work

1.3 Intended results

2 Background Material

2.1 Why a dashboard?

Meyer et al. [**?**] stated, that there are three important points one has to consider when building a soft-monitoring tool for a workplace:

1. the varied need of the users have to be met in the data collection and representation step
2. it should be possible for the users to actively engage with the system
3. the tool should provide more insights into the users work

To fulfil all these requirements, creating a dashboard which can be adapted freely to the users needs and let the users interact with the tool, seems to be the best solution for our use case.

2.2 Useful Productivity Measures

In 2014 Meyer et al. [**?**] conducted a survey to find out what activities software developers think of as productive. As part of their research, they ask developers which measurements would help them to assess their personal productivity. The following 23 measures were their result:

- Number of closed work items (tasks and bugs)
- Spend time on each work item
- Spend time reviewing code.
- Spend time writing code.
- Number of contributed code reviews
- Number of created work items which were fixed
- Number of created work items
- Spend time in meetings
- Number of signed off code reviews
- Number of written test cases
- Spend time on web browsing for work related information
- Number of attended meetings
- Average time for signing off on code reviews
- Spend time in each code project or package
- Number of written test cases which afterwards failed
- Average time to respond to emails
- Spend time on web browsing during work for personal matters
- Number of learned API methods each day
- Number of commits
- Number of written emails
- Number of changed lines of code per day
- Number of changed code elements for the first time

These measures give us a good overview about what developers think can help them improve their own productivity. However, they are not all applicable for our dashboard. For instance, we cannot measure how much time the developers spend in meetings nor can we measure activities outside of Visual Studio (e.g. the time spend on writing emails). Therefore, we selected the measures of Meyer et al.'s findings [**?**] which were the most relevant for our project:

- Spend time on each work item (coding, testing and debugging)
- Spend time reviewing code.
- Spend time writing code.
- Number of created work items
- Number of signed off code reviews
- Number of written test cases
- Average time for signing off on code reviews
- Spend time in each code project or package
- Number of written test cases which afterwards failed
- Number of commits
- Number of changed lines of code per day

In 2017, Meyer et al. [**?**] designed recommendations for self-monitoring in the workspace. For their study, they ask developers to rate how interesting some metrics would be to reflect on their work day or work week. There were some metrics, which are unsuitable for our study, so we picked the ones which can be implemented in our dashboard:

- Spend time on each file

- Spend time on each Visual Studio project
- Number of looked at or edited code files
- Spend time writing code
- Number of done code reviews
- Spend time on code reviews
- Number of commits
- Commit size
- Spend time on testing and debugging

The blue entries are also appearing in the findings of Meyer et al. [**?**] which gives them even more creditability. According to the combined findings of Meyer et al. [**?**] and Meyer et al. [**?**] we conclude that it would be the most efficient to include time and quantitative measures in our dashboard. Therefore, visualising time intervals, timelines or histograms might be an effective way to visualise our metrics.

2.3 Dashboard Design

Meyer et al. [**?**] defined six tool design recommendations for helping developers increasing their productivity:

1. "High-level overviews and interactive features to drill- down into details best support retrospecting on work
2. Interest in a large and diverse set of measurements and correlations within the data
3. Experience sampling increases the self-awareness and leads to richer insights
4. Reflecting using the retrospection creates new insights and helps to sort-out misconceptions
5. Natural language insights are useful to understand multi-faceted correlations
6. Insights need to be concrete and actionable to foster behavior change" [**?**, p. 2]

For our project, recommendation 2 and 5 are important. It is interesting, that developers prefer correlating data. We implemented this aspect by correlating time with quantity. Furthermore, we were surprised, that visualisations are not as clear to every developer than we thought. We kept this piece of information in mind while designing our graphics.

3 Project Management

3.1 Objectives and priorities

3.2 Criteria for success

3.3 Method of work

3.4 Quality management

Documentation
Validation steps

4 Plan with Milestone

4.1 Project steps

4.2 Tentative schedule

5 Incremental Processing

- 2018: Efficient finer-grained incremental processing with MapReduce for big data

  - Most logging data is immutable in
  - More complicated: Also support modifying data
  - Difficult tradeoff: How big do we make the chunks? Big = lots to update, even with little new data, Small = too hard to search through all chunks
  - Incremental processing can save 85%+ of processing time
  - Overview over existing incremental systems
- 2013: large-scale incremental processing with mapreduce (HadUP)
  - Current approaches are usually only for Big Data
  - Typical for incremental processing: deltas are much smaller than complete data
  - Common problem: small input change affects a lot of output (eg with PageRank)
  - Many approaces use memoization of intermediary results