Mashing Up Software Issue Management, Development and Usage Data

Making Continuous Delivery Visible

Anna-Liisa Mattila, Timo Lehtonen, Henri Terho, Tommi Mikkonen and Kari Systä
Idea

Several tools are used in software project which stores data from the project requirements, specification, version history, etc…

Could we follow the life of a feature from specification to usage?

- How long features are in development?
- How long they are waiting to be deployed?
- How often they are used?
The case

lupapiste.fi

• a service which can be used by citizens and companies to apply for permissions related to the built environment in Finland
• developed by Solita Inc.
• Data considering development of 65 features were collected during 2.5 months
The Data Sources

Issue management system (JIRA)
- Specification data

Version control system (Mercurial)
- Development data

Monitoring platform (Splunk)
- Usage data
**IMS Data**

- IMS ticket is created
- Comment to the ticket is uploaded
- The feature specification starts
- The specification is updated
VCS Data

Feature branch created

Development started

Feature branch is merged to release branch

Feature is done

Release branch is closed

Software is deployed
Usage Data

Monitoring platform gives information on feature usage

- The function called
- The time stamp of function call
- Could not be automatically linked to VCS or IMS data
Data Collection

The data collection workflow relies on certain project conventions:

- the feature branches are used in VCS
- the links to IMS tickets are present in feature branch names

Linking data manually is time consuming and error prone:

- Thus usage data was collected only for 5 features
Data Unification

Raw data collected from the different systems is heterogeneous

In order to visualize the data, it need to be harmonized
Visualization

Feature lifespan timeline

- each feature has its own row which displays the lifespan of a feature
- Different events e.g. commits, specification uploads and usage are shown in different kind of markers in the lane

Stacked bar chart

- Amount of features under development
- Amount of features waiting for deployment
So What?

- Shows how well continuous deployment is realized
  - Can help team to keep up with the CD mindset

- Shows abnormalities in development of a feature
  - Can point out possible problems in the development process

- Usage data combined to development data
  - Feature prioritizing based on usefulness and development costs
Questions?