Software metrics to Predict the health of a project?

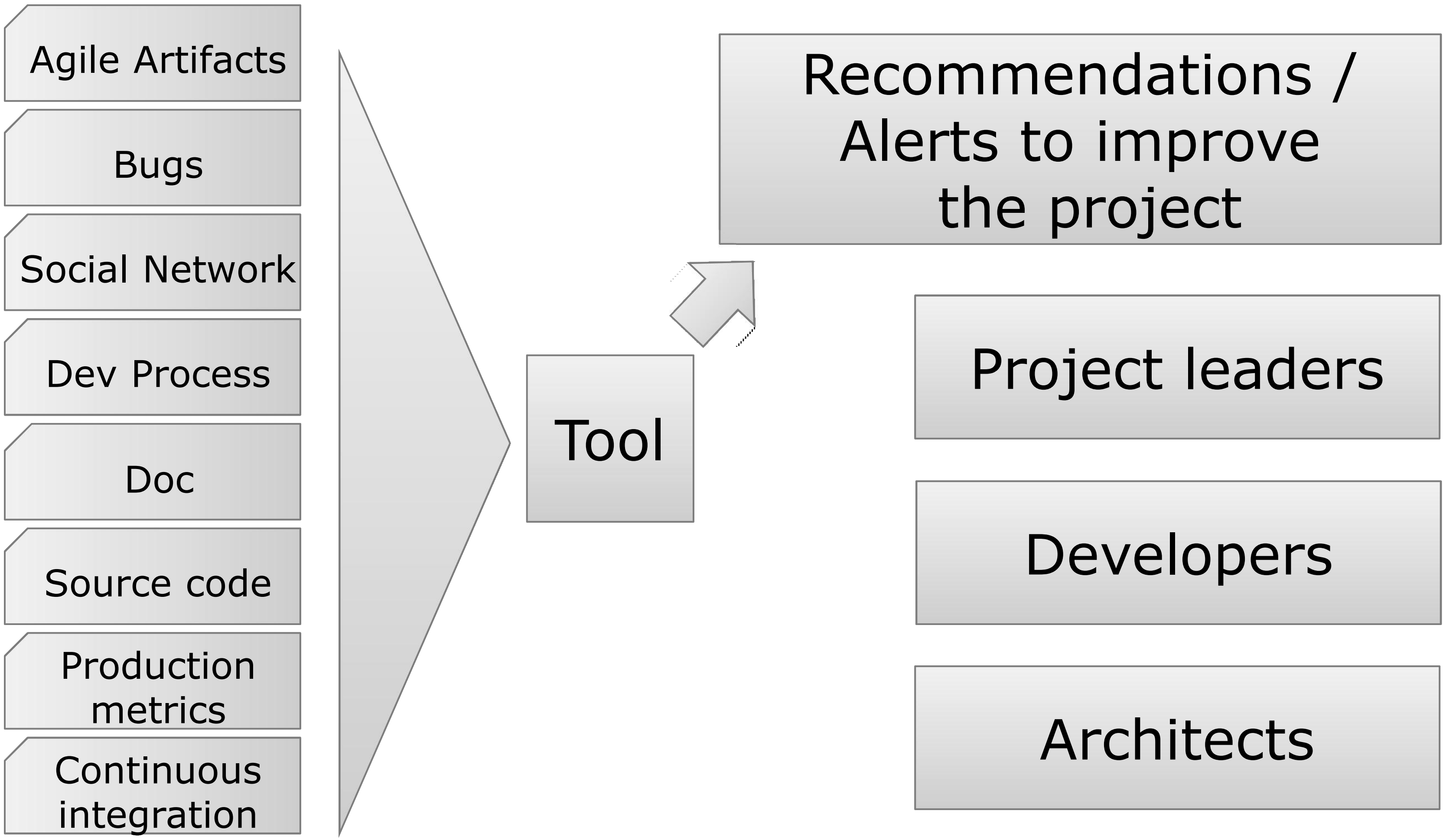
Vincent Blondeau

15 – July – 15

Context

- Industrial PhD in a major international IT company
  - 7 300 employees
  - 17 countries
  - Problems from the field

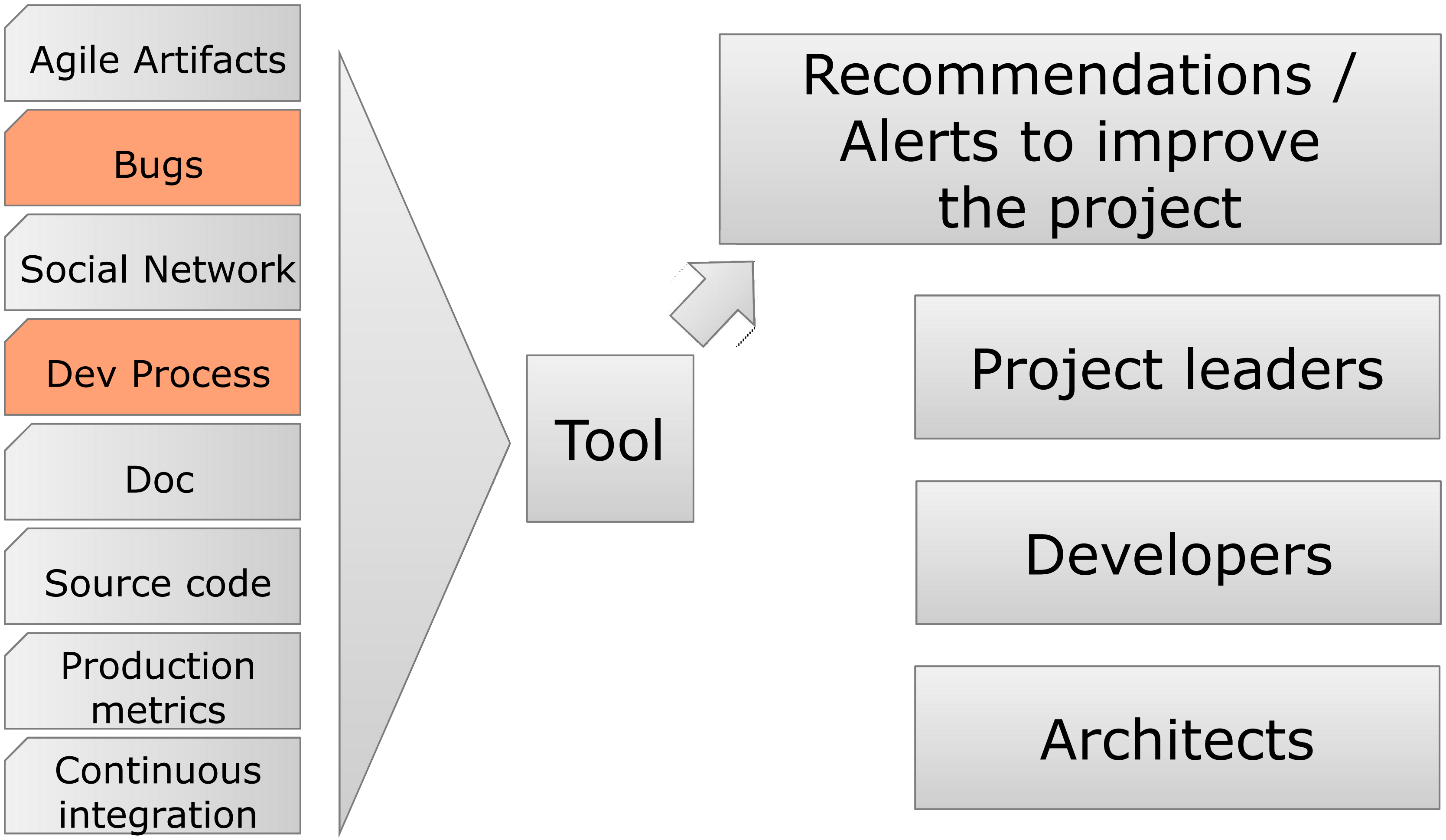
Context


Agile Artifacts
Bugs
Social Network
Dev Process
Doc
Source code
Production metrics
Continuous integration
Tool
Recommendations / Alerts to improve the project
Project leaders
Developers
Architects

Overview

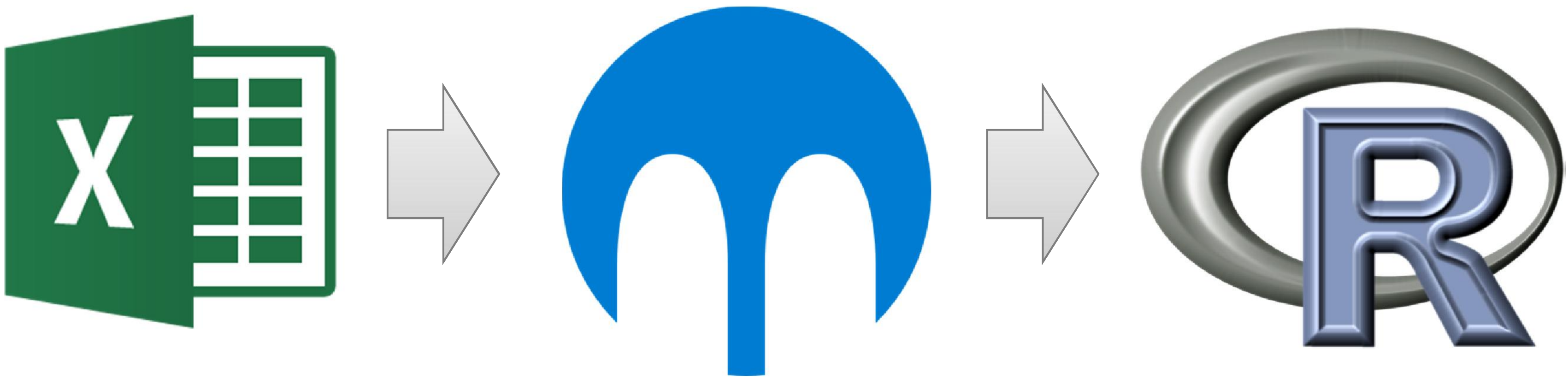
- **Data mining**
- Literature survey
- Meeting with team managers

Data mining


Agile Artifacts
Bugs
Social Network
Dev Process
Doc
Source code
Production metrics
Continuous integration
Tool
Recommendations / Alerts to improve the project
Project leaders
Developers
Architects

Project data mining

- Extracted from Excel files
  - Bugs: qualification / acceptance / prod
  - Budgets: projects and intermediate releases



Exploitable data

- 20 projects (out of 43)
  - 300 bugs / project on average
  - 1400 Men*Days / project on average

- 60 intermediate releases (out of 725)
  - 600 Men*Days / release on average
  - 92 bugs / release on average

Project data mining

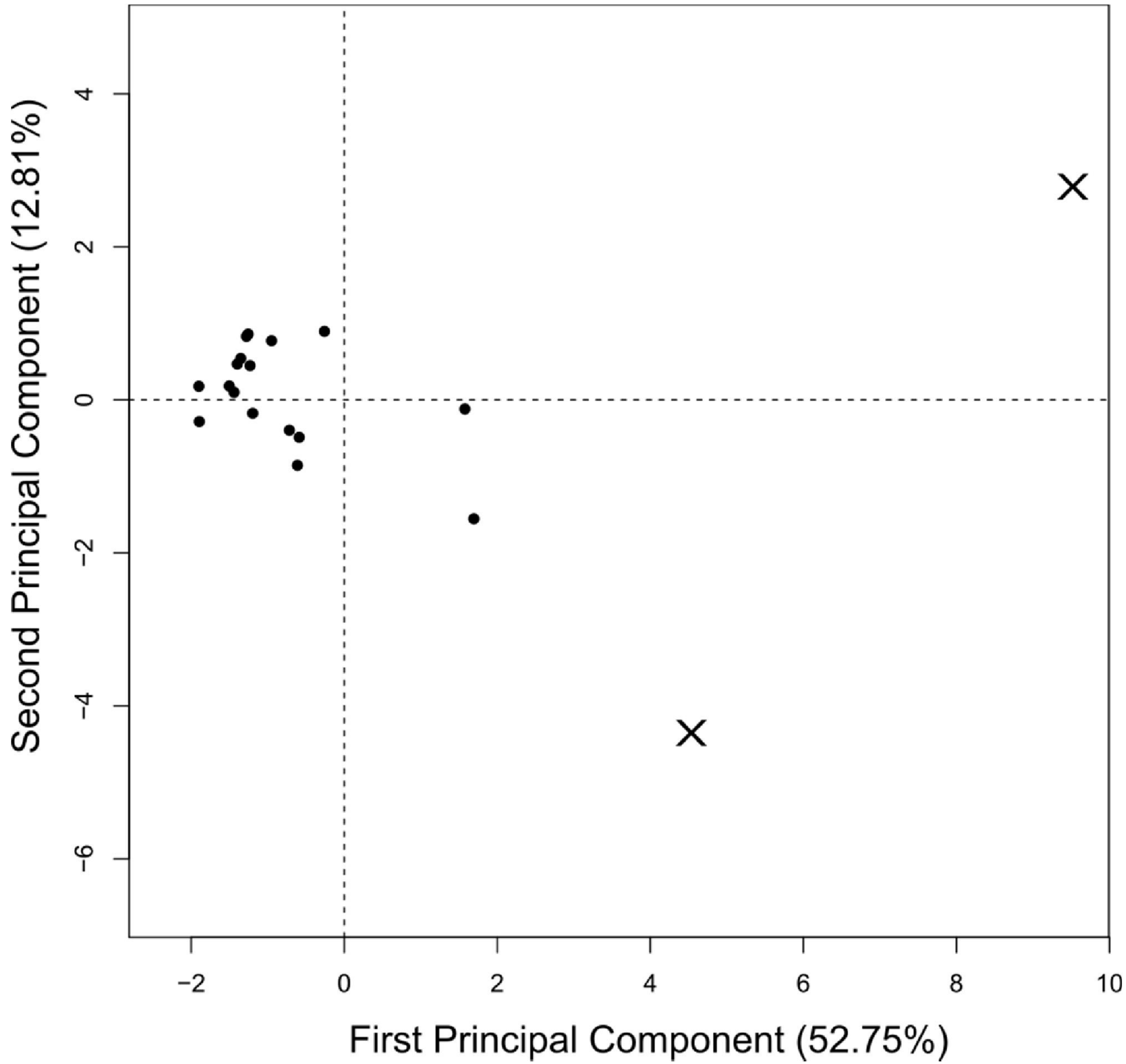
- Bugs
  - Critical, major, minor,
  - Qualification, acceptance, production
- Budget
  - Predicted, Realized
  - Delta Predicted / Realized
- Slippage
  - Yes / No
  - Number of months

Project data mining

- Bugs
  - Critical, major, minor,
  - Qualification, acceptance, production
- Budget
  - Predicted, Realized
  - Delta Predicted / Realized
- Slippage
  - Yes / No
  - Number of months

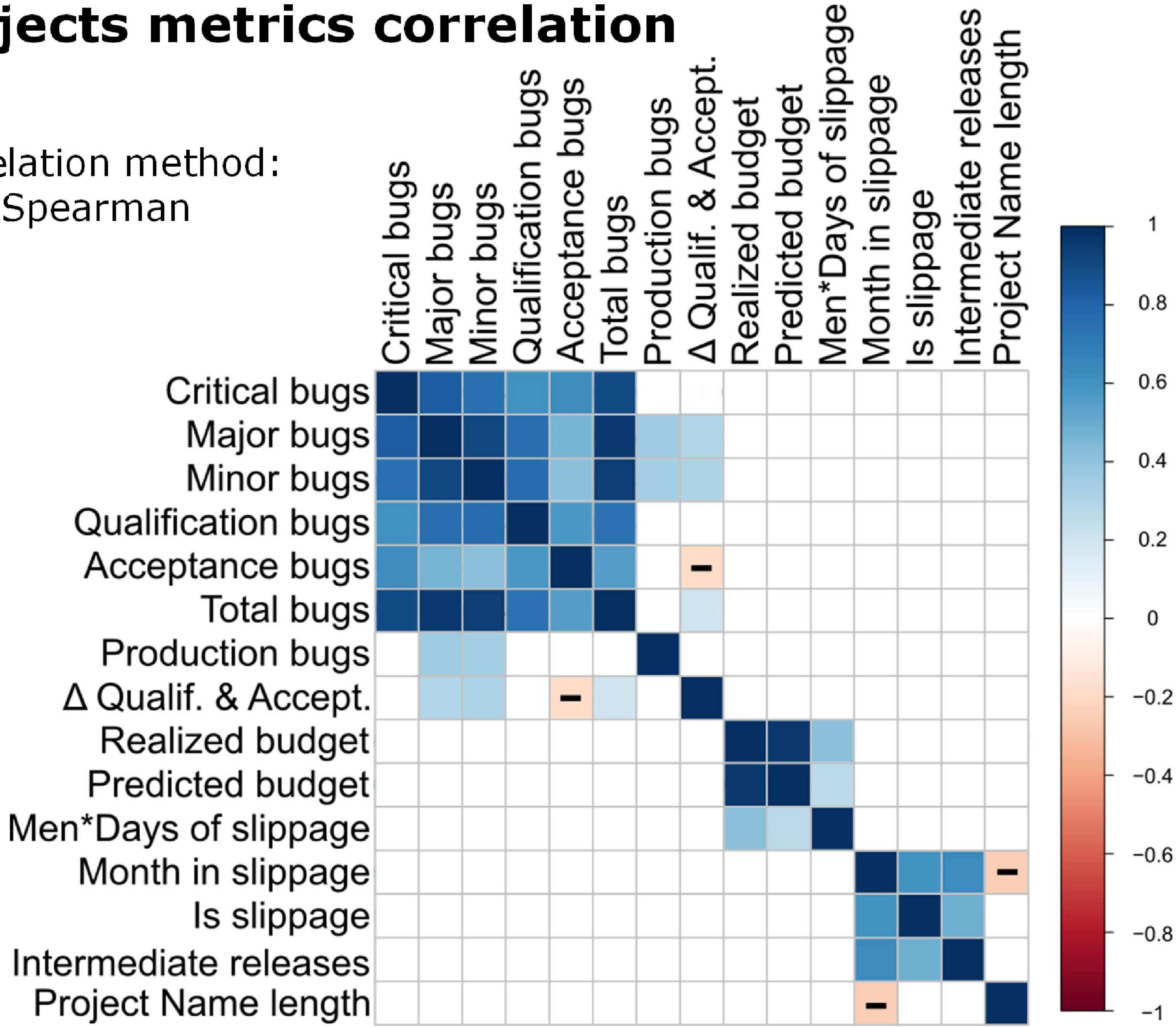
Project Name Length

Projects metrics correlation



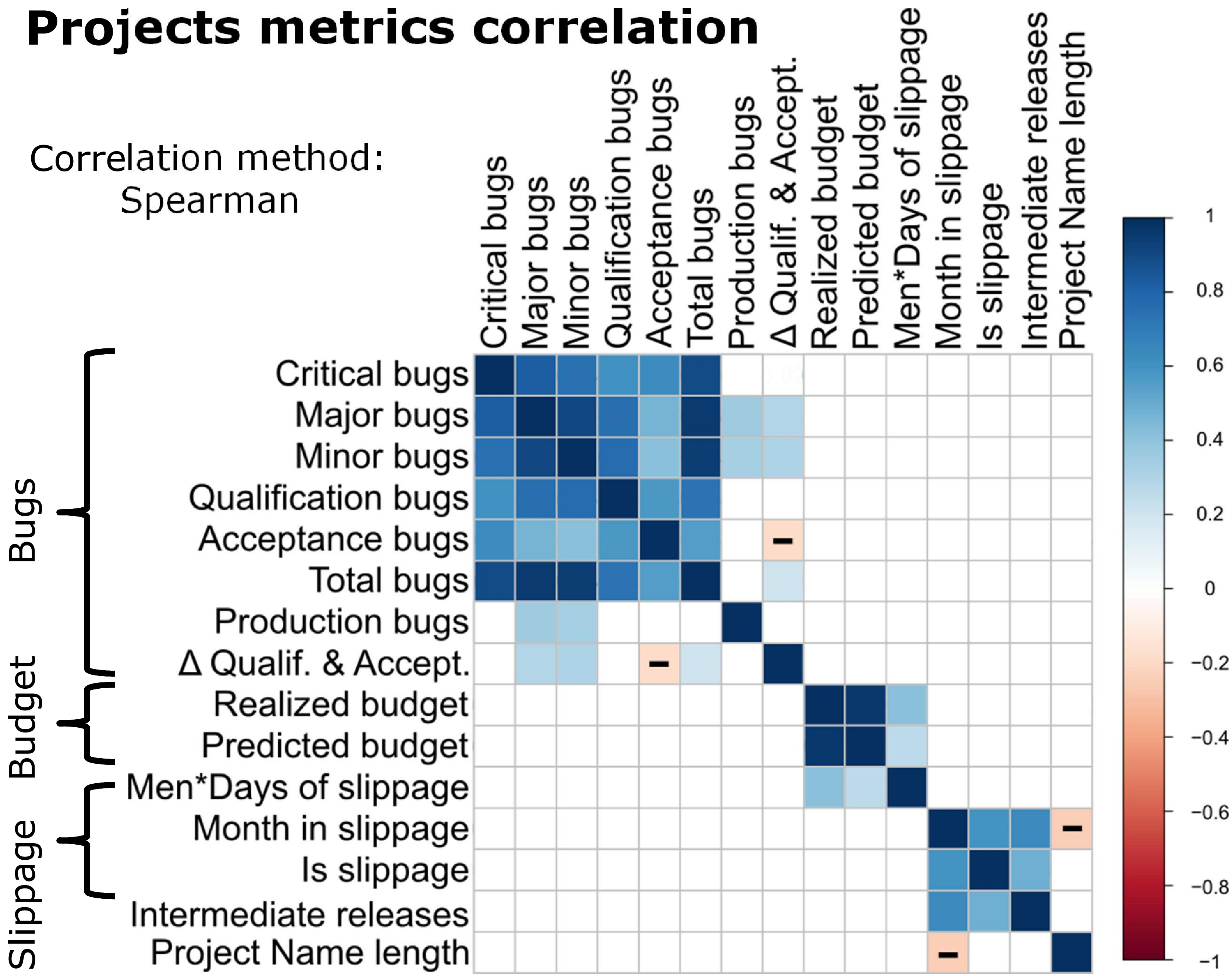
Projects metrics correlation

Correlation method:
Spearman

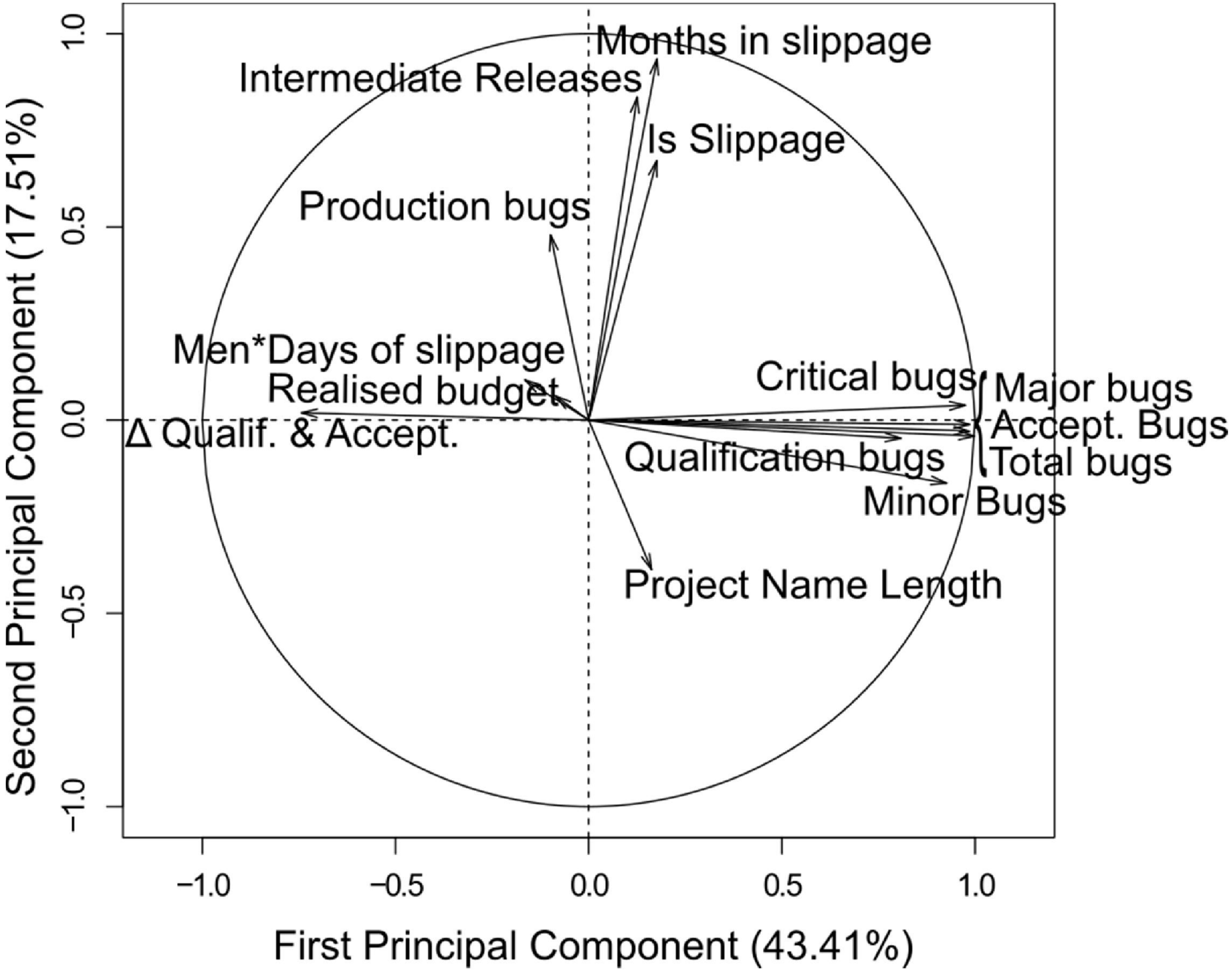


Projects metrics correlation

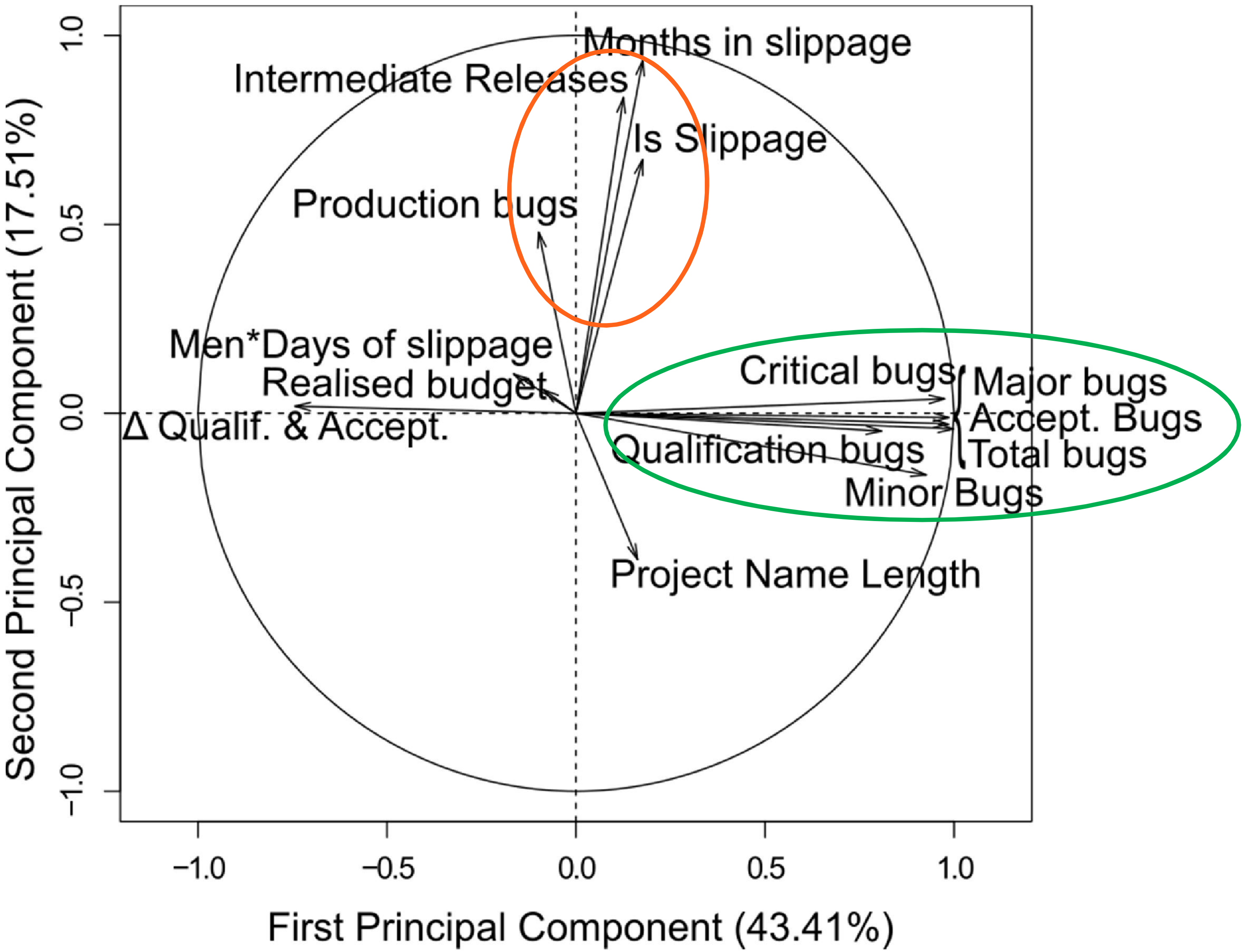
Correlation method:
Spearman



Projects metrics correlation



Projects metrics correlation



Data mining results

- Bugs ⇒ Bugs

- Slippage ⇏ Bugs
- Bugs ⇏ Slippage

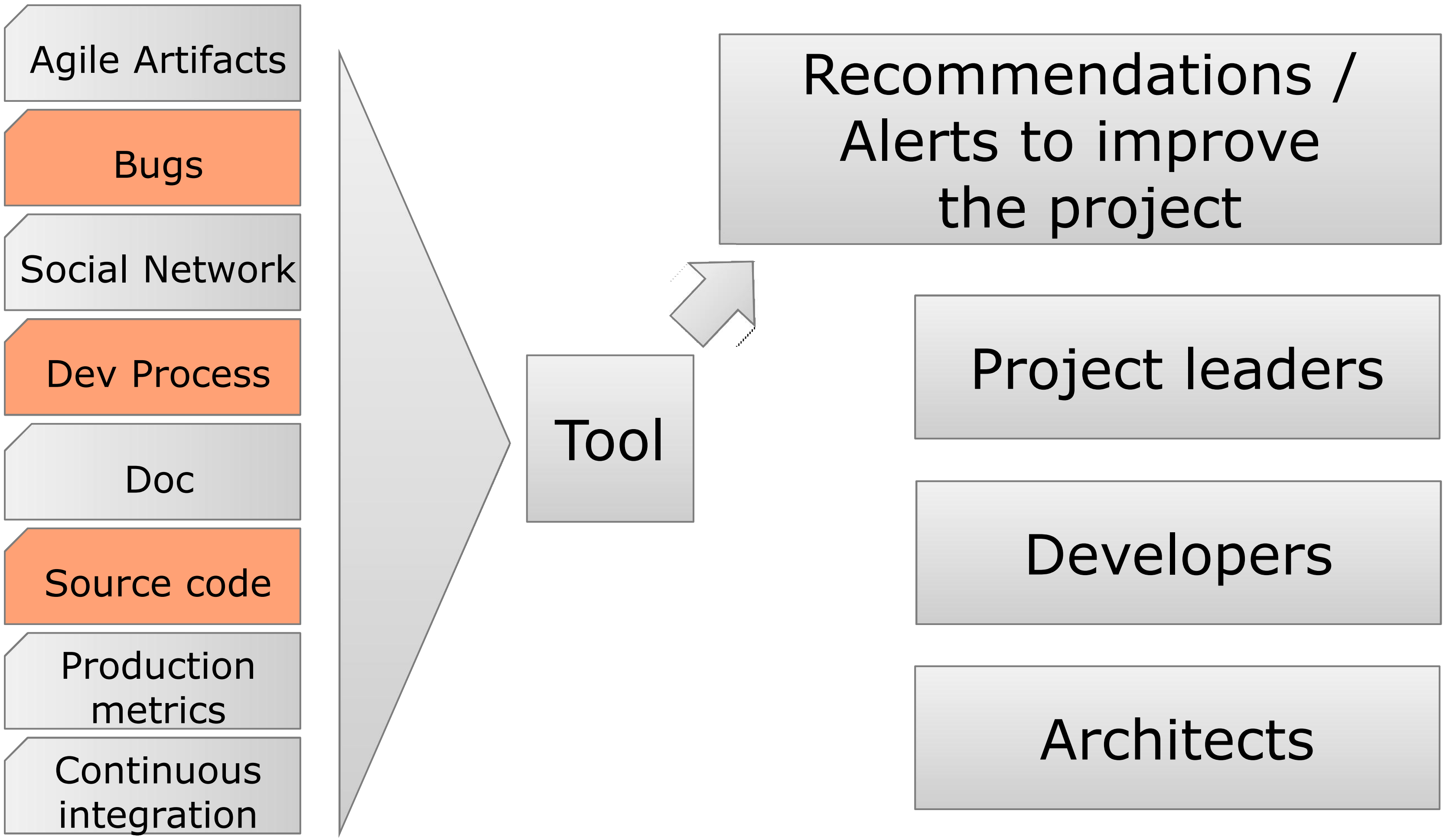
- Production Bugs ⇒ Slippage

- Name length ⇒ Less bugs

Overview

- Data mining
- **Literature survey**
- Meeting with team managers

Literature survey



Mining Metrics to Predict Component Failures
Nachiappan Nagappan, Thomas Ball, Andreas Zeller
2006, ICSE

- Goal: Predict after release bugs
- 5 C++ Microsoft projects
- 18 source code metrics
- Correlations, PCA, regression models

  - ∃ some metrics correlated to bugs
  - ∄ metrics for all the projects
  - The prediction seems accurate on the same kind of project

A model to predict anti-regressive effort in Open Source Software

Andrea Capiluppi, Juan Fernández-Ramil

2007, ICSM

- Goal: Find metrics to identify regressions
- 8 C/C++ Open Source Systems (OSS)
- 4 source code metrics
  - $\nexists$ factor which alone makes a best predictor
  - Each system needs to determine individually which measurement is best

Exploring the relationship between cumulative change and complexity in an Open Source system
Andrea Capiluppi, Alvaro E. Faria, Juan F. Ramil - 2005, CSMR

- Goal: Find classes to refactor
- 62 releases of ARLA (AFS file system)
- 4 code source metrics

  - 50% of classes with frequent changes are the more complex and have the higher number of methods

Cross-project defect prediction

A Large Scale Experiment on Data vs. Domain vs. Process

Thomas Zimmermann, Nachiappan Nagappan – 2009, ESEC/FSE

- Goal: predict defects
- 28 releases of open and closed source software
- 40 project and source code metrics

  - OSS $\Rightarrow$ closed source (CS)
  - OSS, CS $\not\Rightarrow$ OSS
  - $CS_1 \Rightarrow CS_2$ or $CS_1 \not\Rightarrow CS_2$

21 out of 622 (3,4%) cross-project predictions worked

“There was no single factor that led to success”

Literature review results

- Individually, ∃ metrics to make predictions
- No unique metric for all the projects
- Predictions *at posteriori*

Overview

- Data mining
- Literature survey
- **Meeting with team managers**

Meeting with team managers

- 3 in Retail team
- 1 in Telecoms team

  - What are their problems?
  - How they detect them?
  - How they resolve them?

Roots Causes of bad health of a project

- **Delay** at the start of the project
- **Collaboration** between the team and the client
- Lack of team **cohesion**

- Bad understanding of the **specifications**
- Bad knowledge of the **functional concepts**

- Change of the **framework** during the development
- **Experience** with the used frameworks

- **Bypass** the qualification **tests**
- High number of **bugs** listed by the client

Conclusion

- Literature survey
  - No correlation
- Data mining
  - No correlation
- Wrong metrics studied at first

Conclusion

- Literature survey
  - No correlation
- Data mining
  - No correlation
- Wrong metrics studied at first

Next step:

Survey to validate these root causes

Help to test software