Can we visualize how developers collaborated by applying process mining to VCS logs?

Leen Jooken  (leen.jooken@student.uhasselt.be)  Handelsingenieur in de Beleidsinformatica  Promotor: Prof. Dr. Jans  Supervisor: Dhr. Creemers

Research Problem
Many software projects = A lot of code
⚠ Lose overview of who knows which code
⚠ Code at risk of becoming unknown to any programmer

Proposed Solution
Tool that generates social network graph of how software developers collaborated
• Size shows importance
• Colour shows collaboration type
Data source = version control system logs

Business Relevance
• Better overview of the core teams
• Discover valuable & indispensable resources
• Take precautions against ‘brain drain’

Inspiration
Idea of extracting social network graph from VCS log
Rationale of applying process discovery to process event logs

Adaptation
Combination of metrics from fuzzy mining & graph theory
Calculation of weight that represents the importance of
Programmer:
• Unary frequency significance
• Betweenness centrality
• Eigenvector centrality
• Degree centrality

Collaboration relationship:
• Binary frequency significance
• Proximity correlation

Graph simplification approach:
1. Filter out weak relationships
2. Cluster less important but strongly connected programmers
3. Abstract insignificant programmers that are weakly connected to the graph

Results
This strong disjunct programming relation
The only ones working on a specific aspect of the code
Mitigate risk by:
• Pair programming
• Entrusting another programmer with the same task

Isolated groups
Risky if:
• Few members
• Members have a large importance
Why?
Valuable sources of knowledge & not easily replaced

Most important programmers
Important contribution but weak collaboration
Risk of ‘brain drain’ if:
• Illness
• Leaving the company

Regular node
Cluster node
Programming:
• Pair
• Disjunct
• Pair & Disjunct