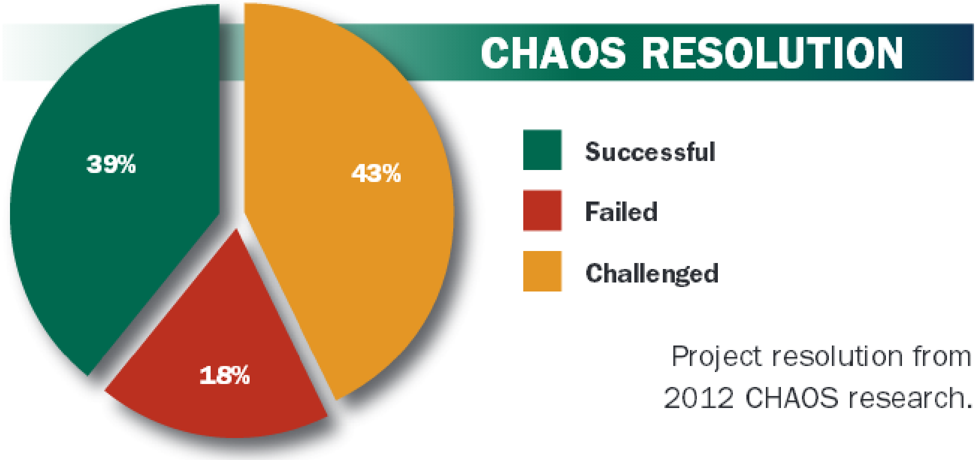
SOFTWARE REPOSITORIES AND THEIR USABILITY IN SOFTWARE PROCESS RECONSTRUCTION

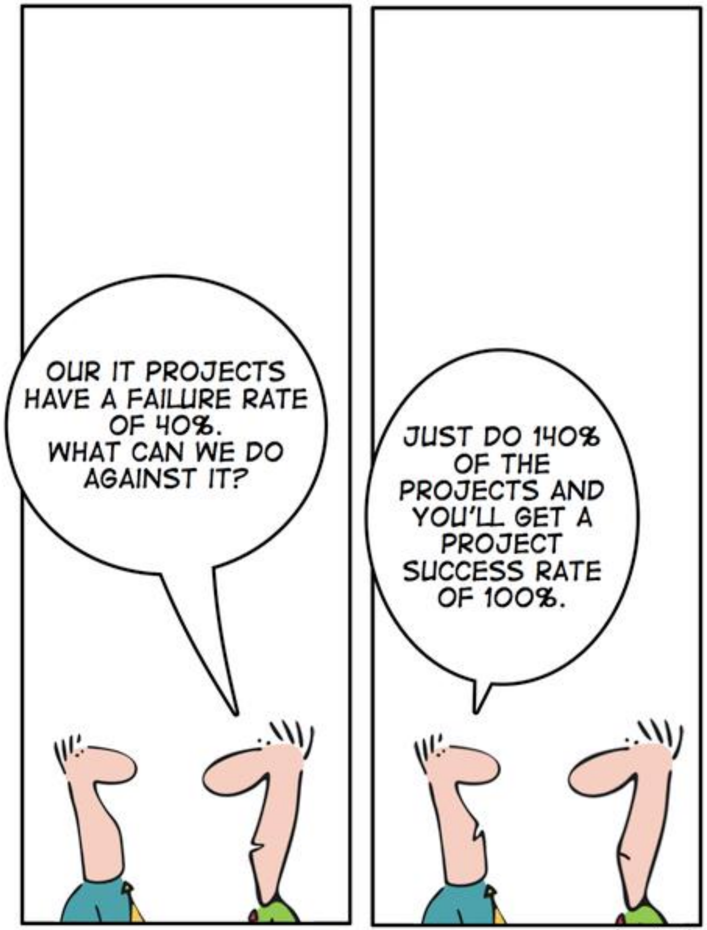
Marko Janković & Marko Bajec

University of Ljubljana

IT Project Performance



Project resolution from
2012 CHAOS research.

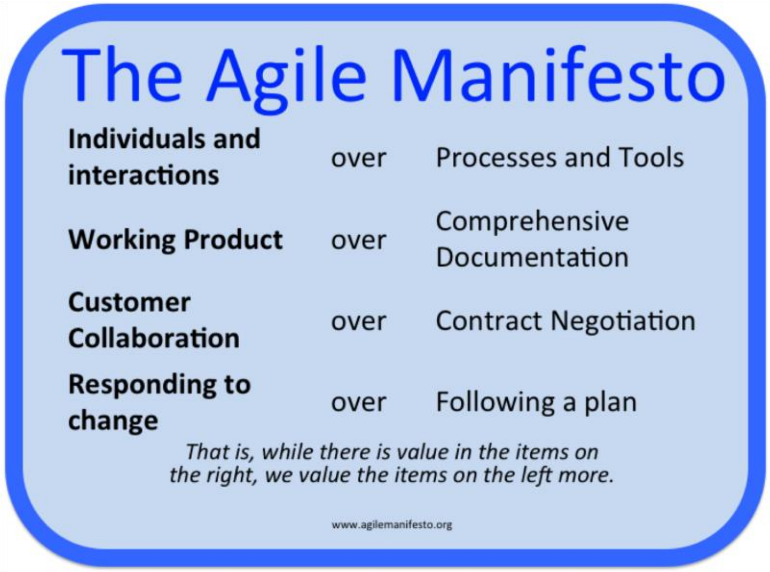

OUR IT PROJECTS HAVE A FAILURE RATE OF 40%. WHAT CAN WE DO AGAINST IT?
JUST DO 140% OF THE PROJECTS AND YOU'LL GET A PROJECT SUCCESS RATE OF 100%.
geek and poke

THE CONSULTANTS HANDBOOK PART 6:
ALWAYS DO THE MATH YOURSELF

Many reasons

- Social issues
- Technology challenges
- …
- The lack of discipline:
  - Many companies do not have any SDM in place
  - Prescribed SDMs not followed
  - Lack of motivation

ISD is about implementing IT into a human enterprise!



The Agile Manifesto

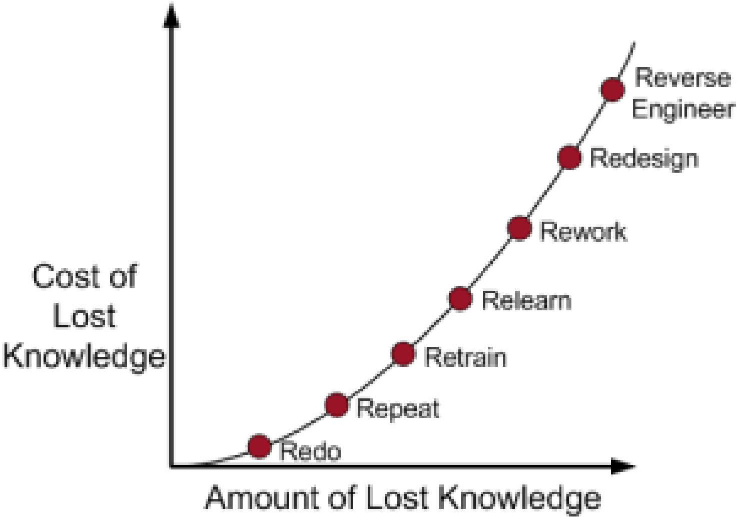
Individuals and interactions	over	Processes and Tools
Working Product	over	Comprehensive Documentation
Customer Collaboration	over	Contract Negotiation
Responding to change	over	Following a plan

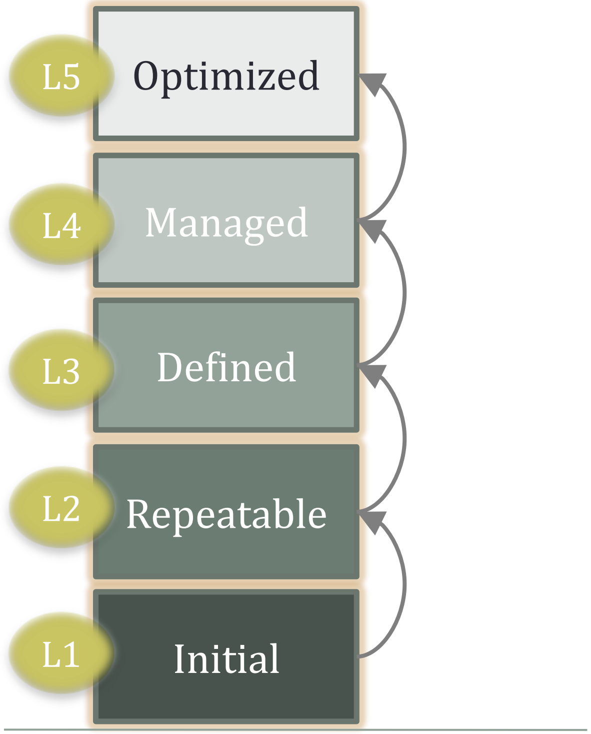
That is, while there is value in the items on the right, we value the items on the left more.

www.agilemanifesto.org

Problems and Limitations

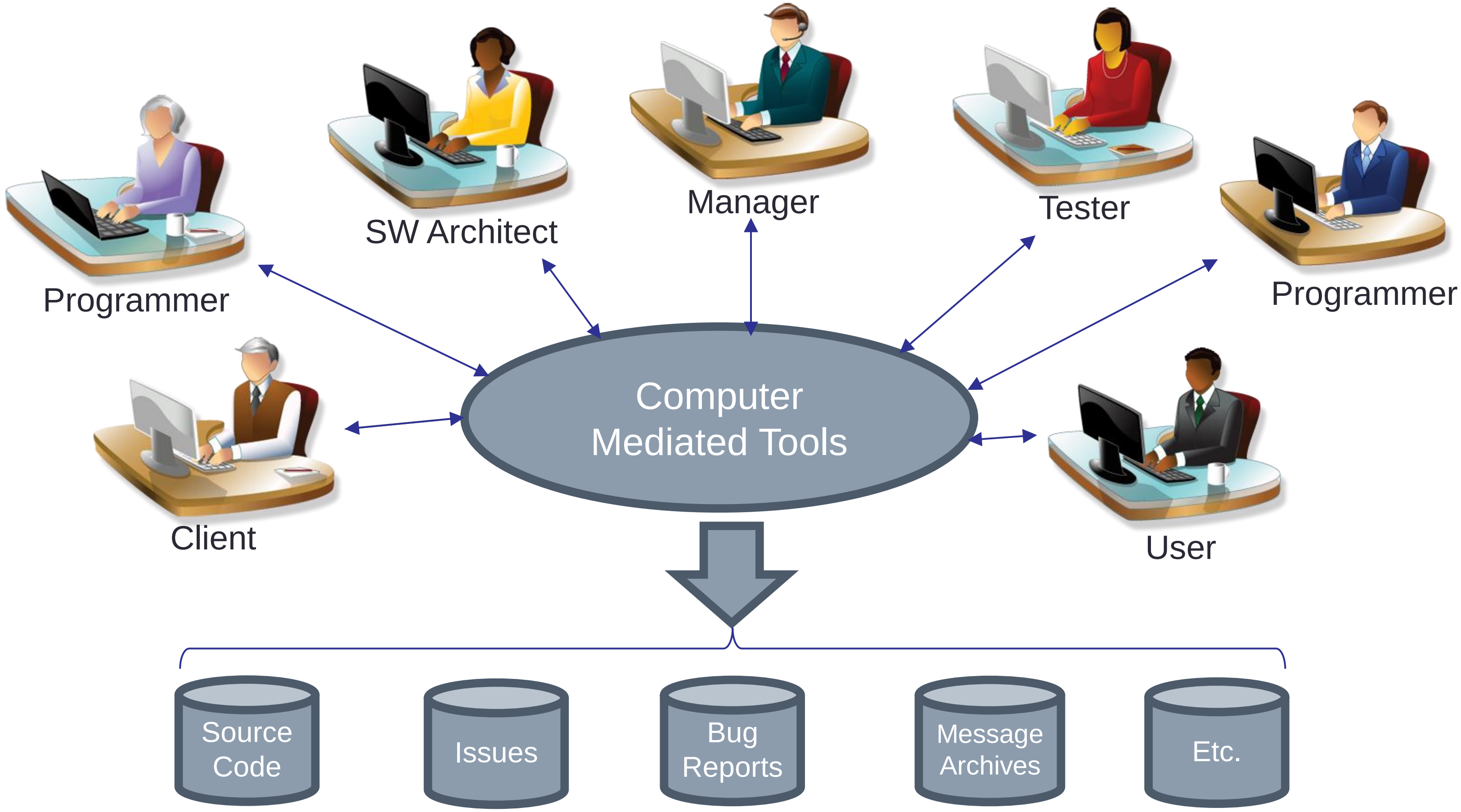
- Risk for knowledge loss…
- Repeating mistakes…
- Reinventing the wheel…





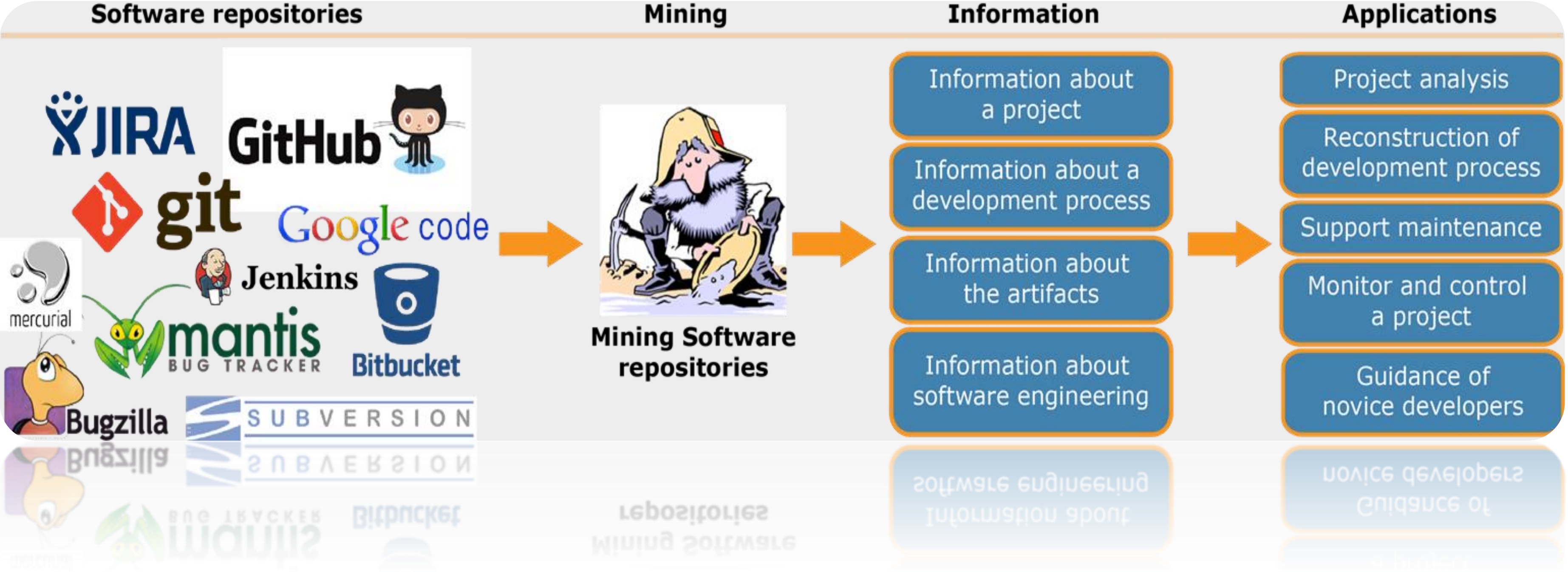
Maturity levels of the CMM

Software Repositories



Based on Marco Aurélio Gerosa, Mining Sociotechnical Information From Software Repositories, University of São Paulo, Brazil

Possible Applications



Software repositories → Mining → Information → Applications

Software repositories

Mining

Mining Software repositories

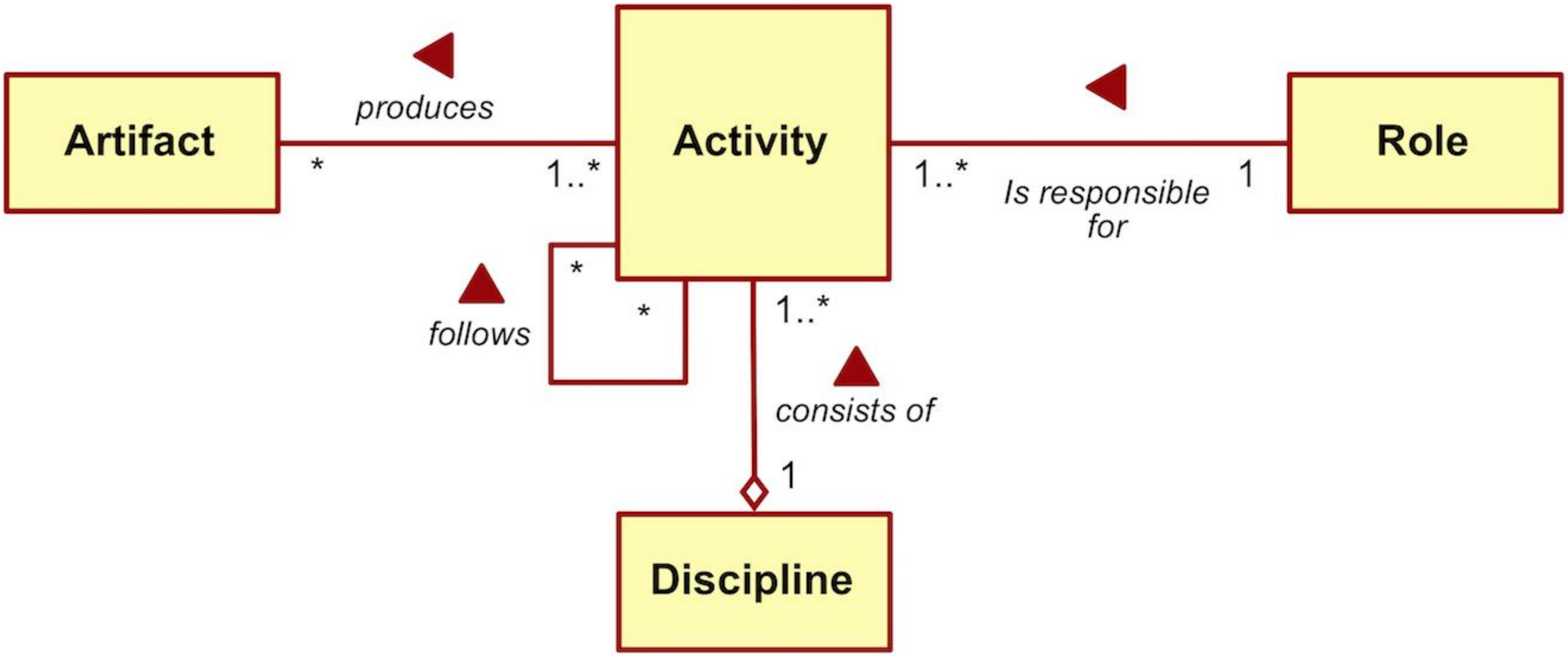
Information

- Information about a project
- Information about a development process
- Information about the artifacts
- Information about software engineering

Applications

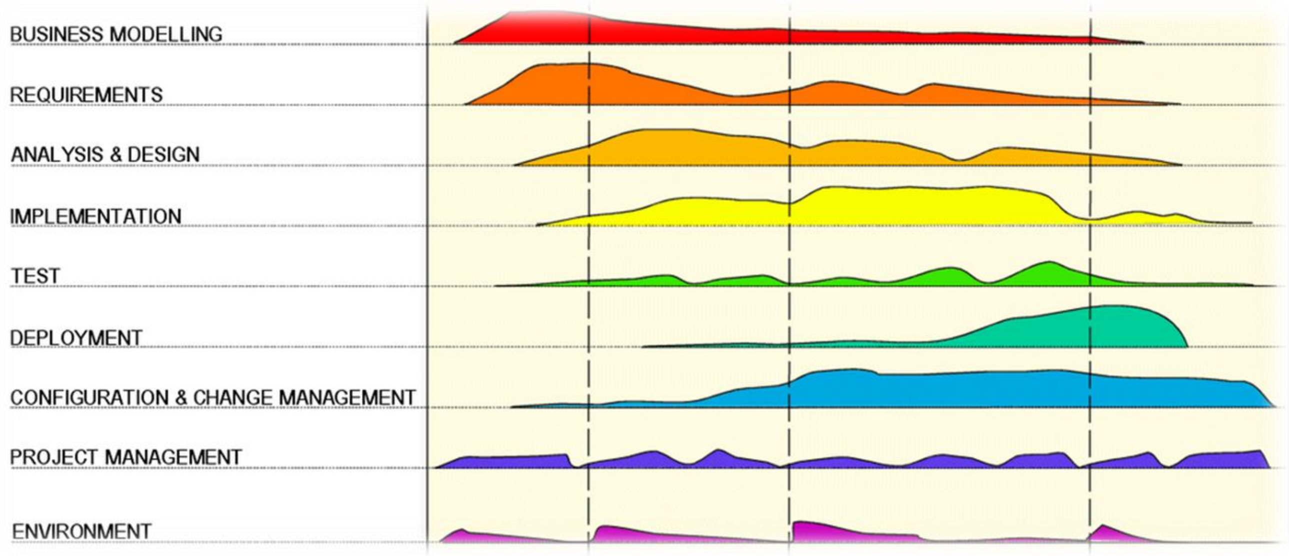
- Project analysis
- Reconstruction of development process
- Support maintenance
- Monitor and control a project
- Guidance of novice developers

Elements for Reconstruction



Software process recovery

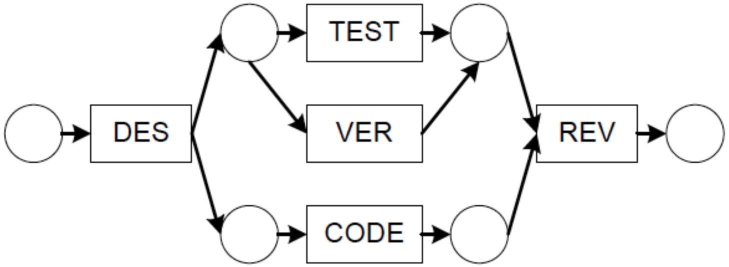
- Employs different semi-supervised techniques to recover UP diagram.
- Illustrates how the relative emphasis of different disciplines changes over the course of the project.

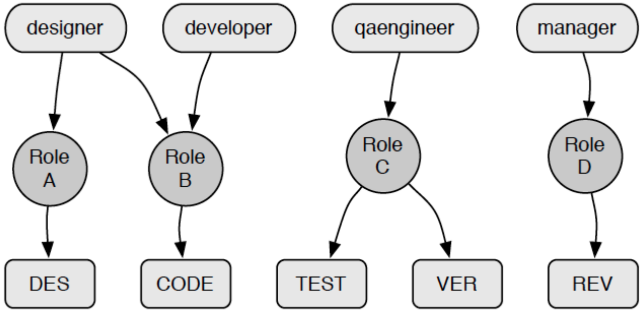


A. Hindle, Software process recovery, PhD thesis

Software process mining

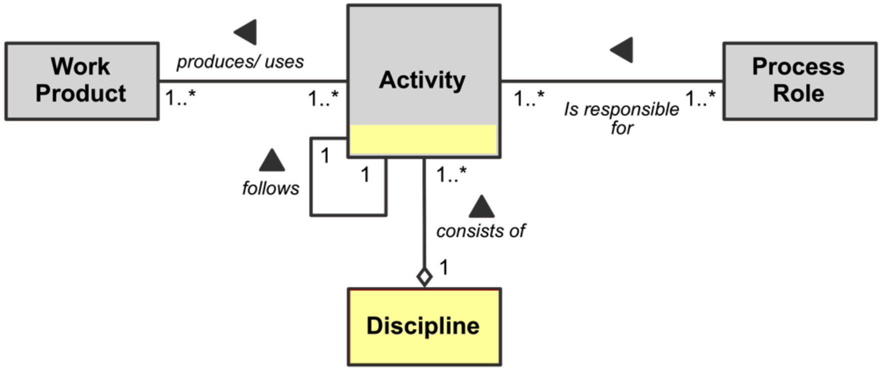
- Mainly apply techniques from process mining on the event log generated from software repositories.
  - document names mapped into abstract names…
  - e.g.: docs with "/src/" in the filepath and with an extension ".java" map to the activity "code"
- Focused on reconstruction of high-level elements (e.g. main activities/disciplines) and workflow mining…
- Data typically used from one repository only.



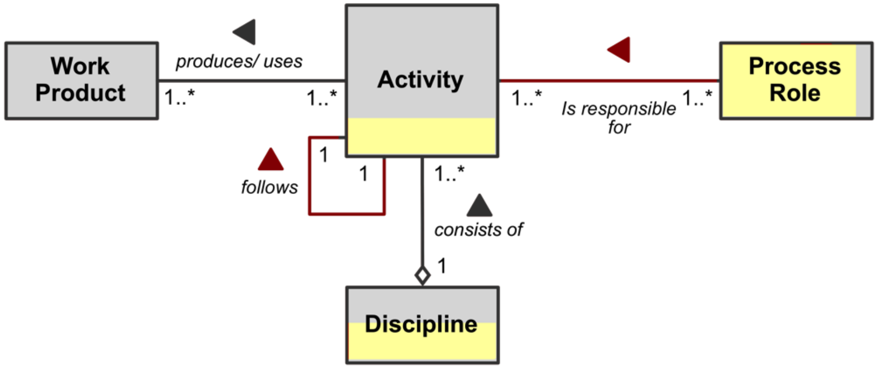


Limitations

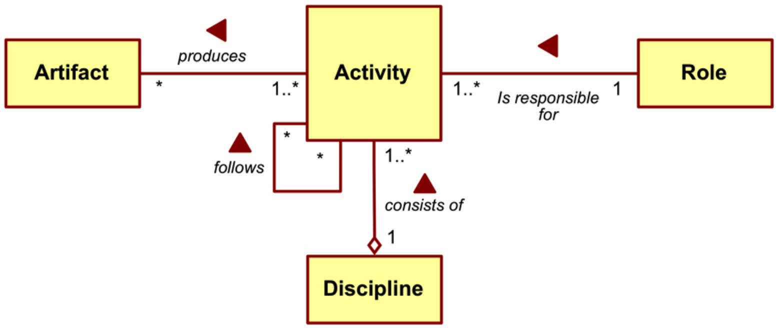
- Mining Software Repositories



- Software Process Mining



Approach



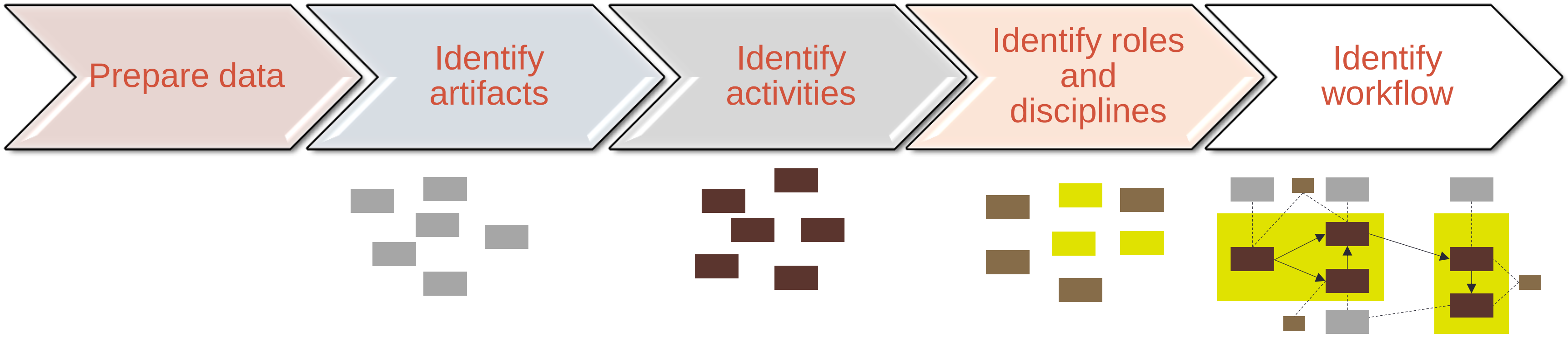
JIRA

github SOCIAL CODING

Google code

...

SUBVERSION


Prepare data

Identify artifacts

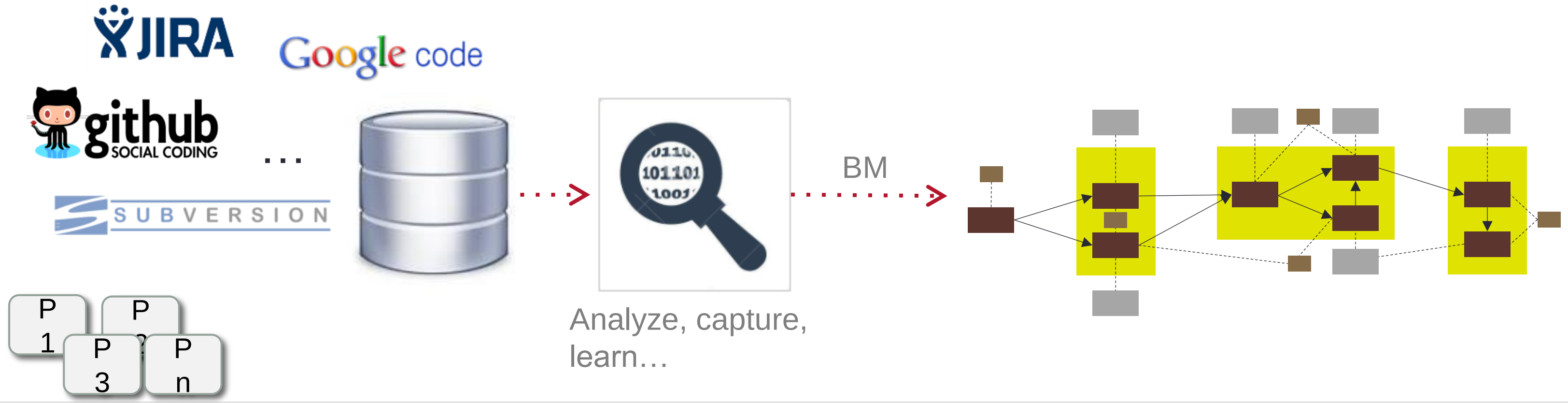
Identify activities

Identify roles and disciplines

Identify workflow

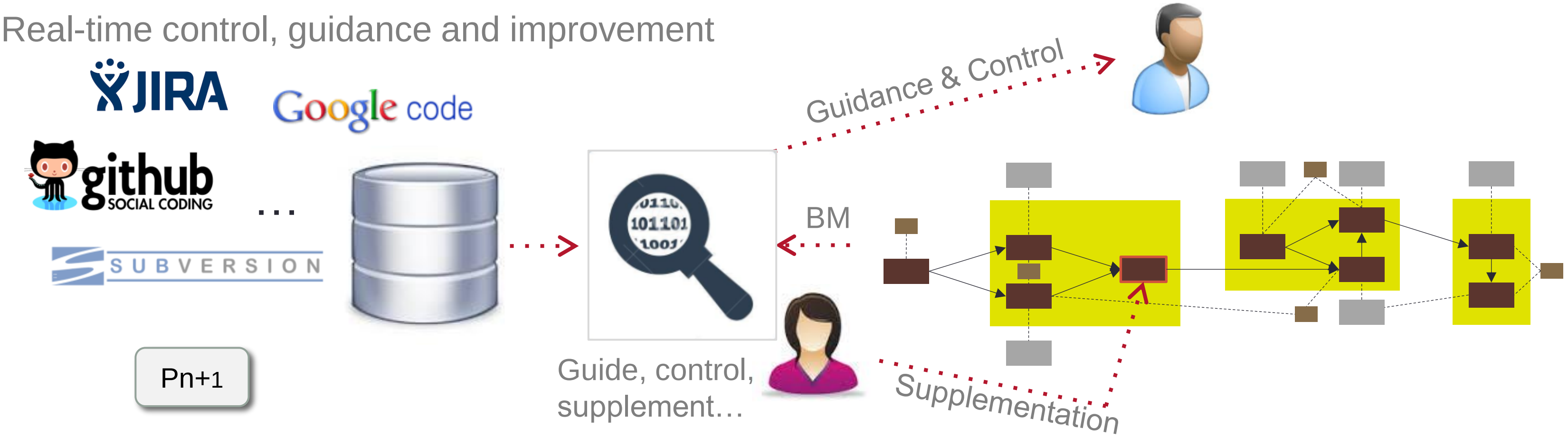
How it Works

Preparation: analysis of logs of past projects. Result: workflow of the base method



Analyze, capture, learn…

Real-time control, guidance and improvement



Guide, control, supplement…

Data Preparation

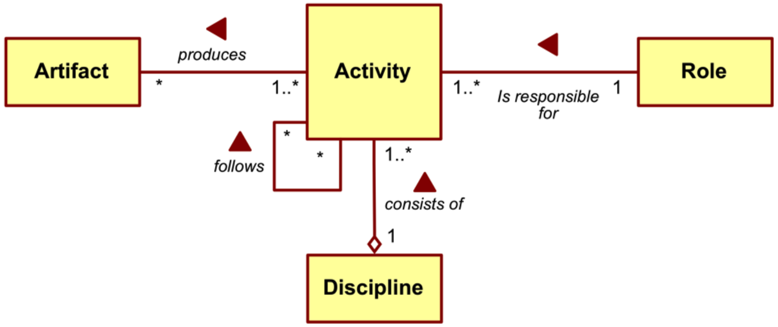
Prepare data

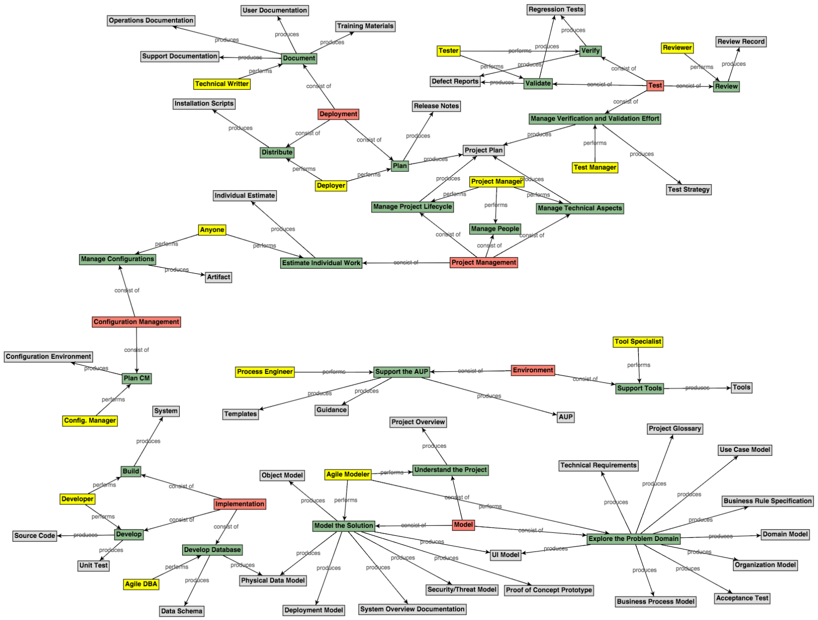
- Gather data from repositories:
  - Revision control systems
  - Document system
  - Issue/Bug tracking system
  - Code review systems…
- Link users of different repositories → entity resolution
- Link tasks/issues with commits (e.g. based on commit messages…)

Identification of artifacts

Identify artifacts

- Identification based on predefined ontology
  - Defines key elements (for each meta element of our interest)
  - Can be altered before or within the reconstruction process.

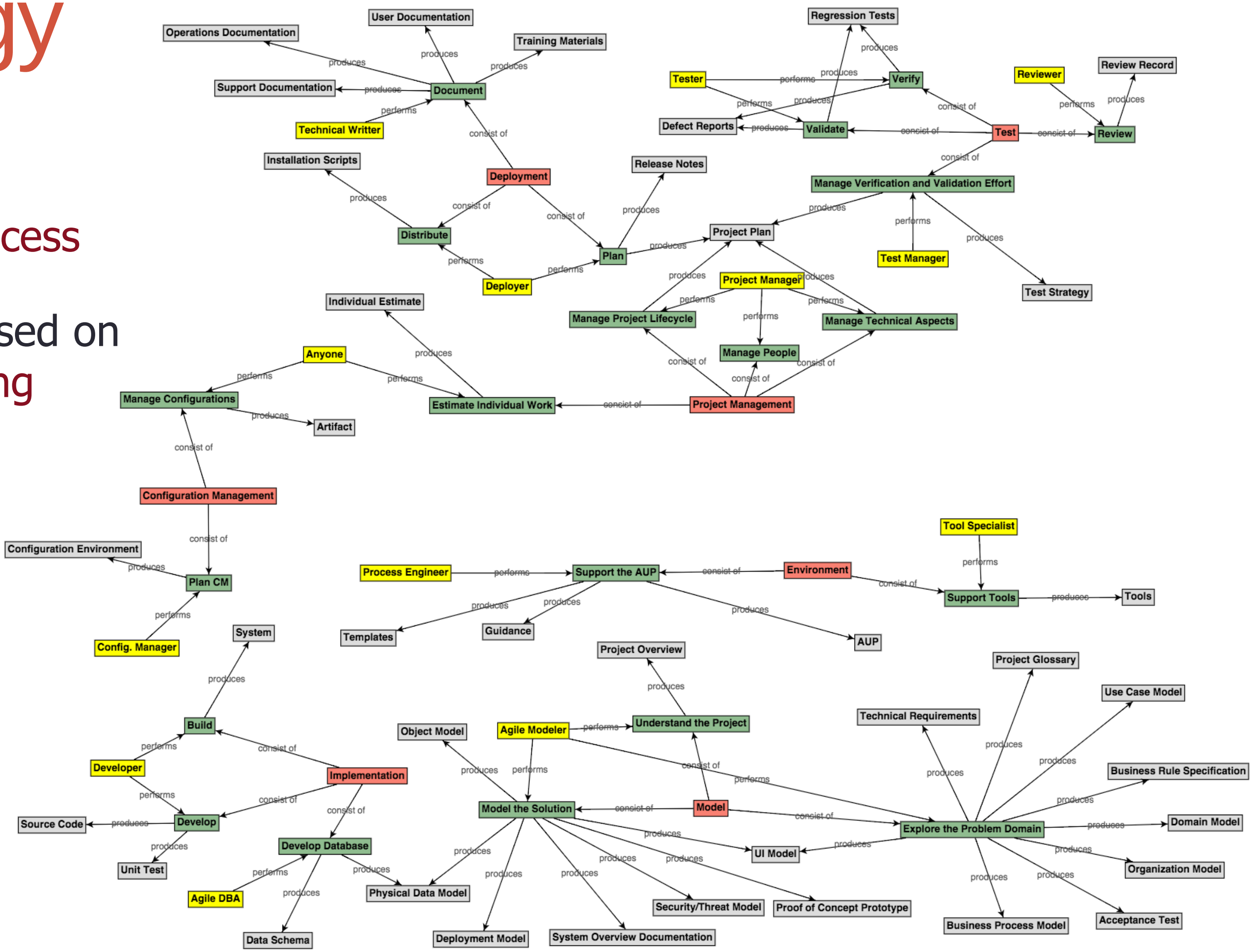




Ontology

Based on
Agile Unified Process

Identification based on
keyword matching



🟨 Process role

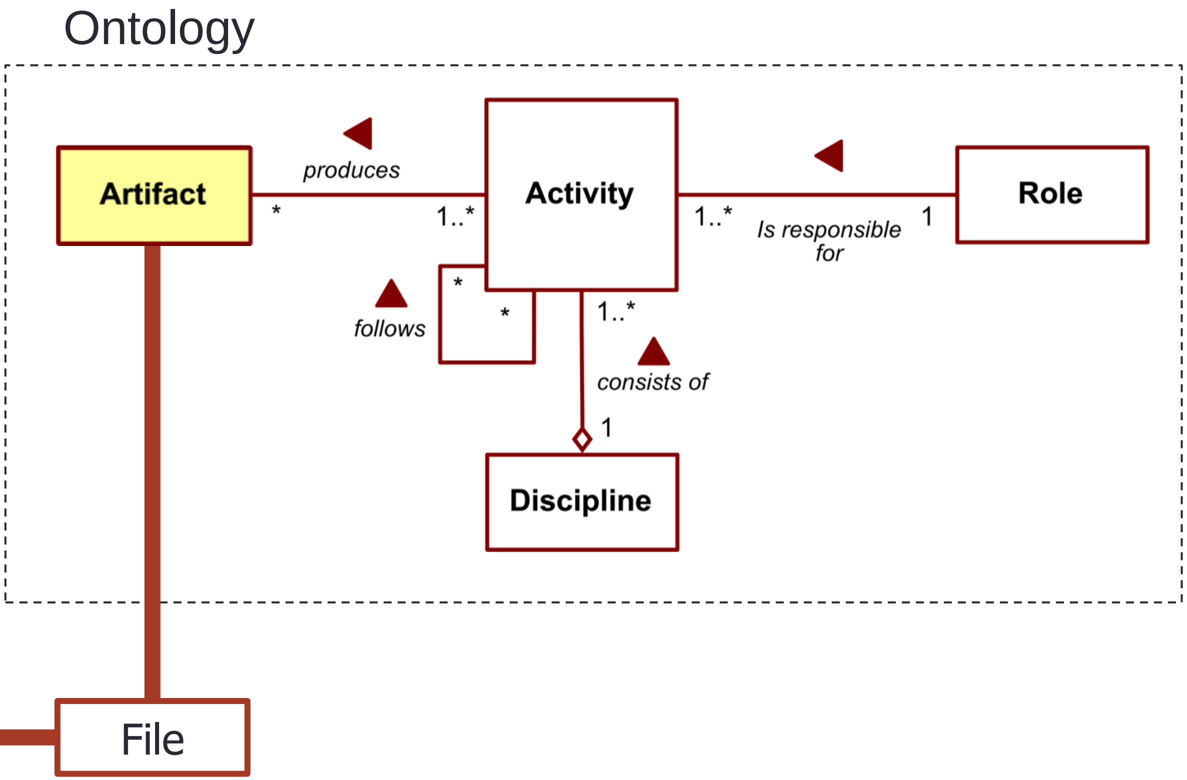
🟩 Activity

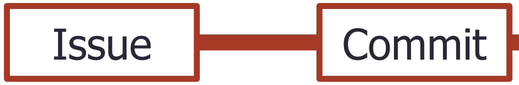
▨ Work product

🟥 Discipline

Connecting files with artifacts

If low classification confidence then ask user





JIRA

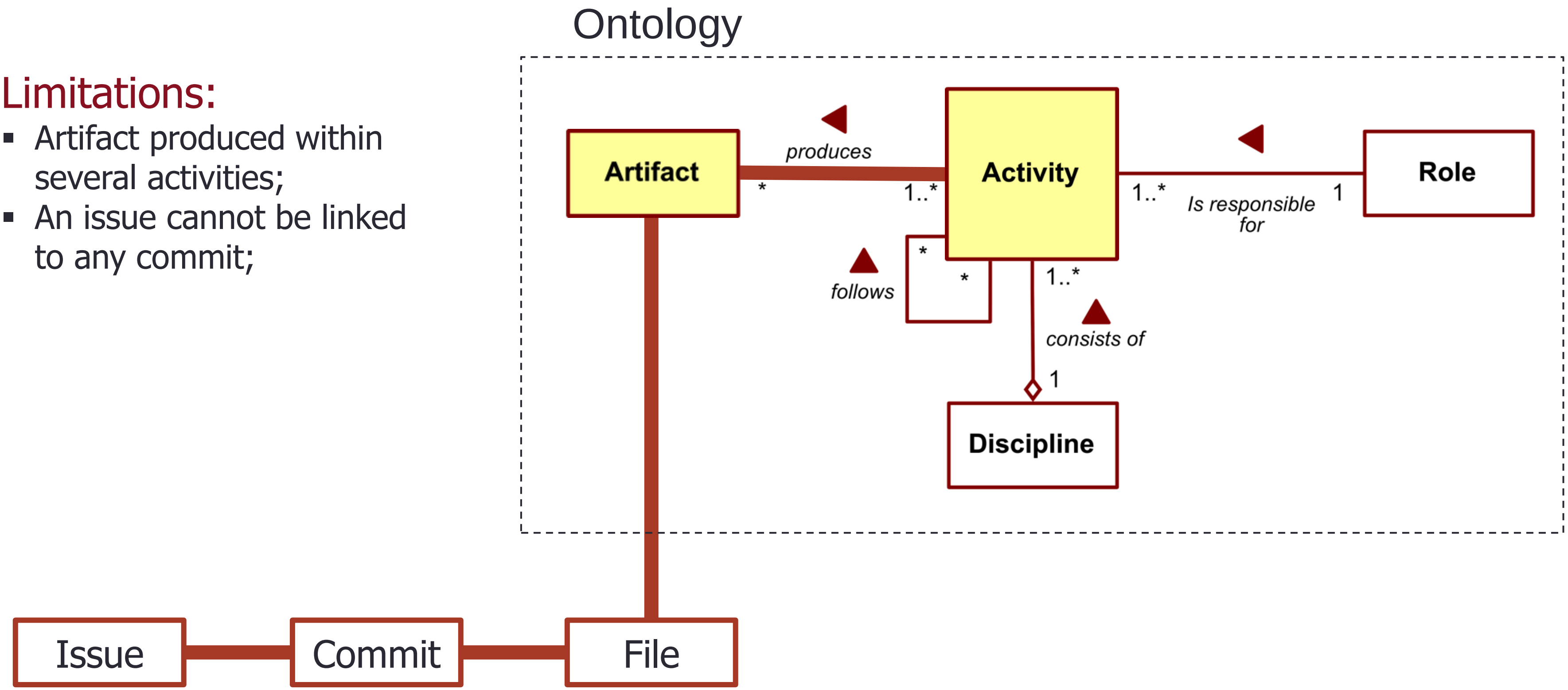
github SOCIAL CODING

Identifying activities

Identify activities

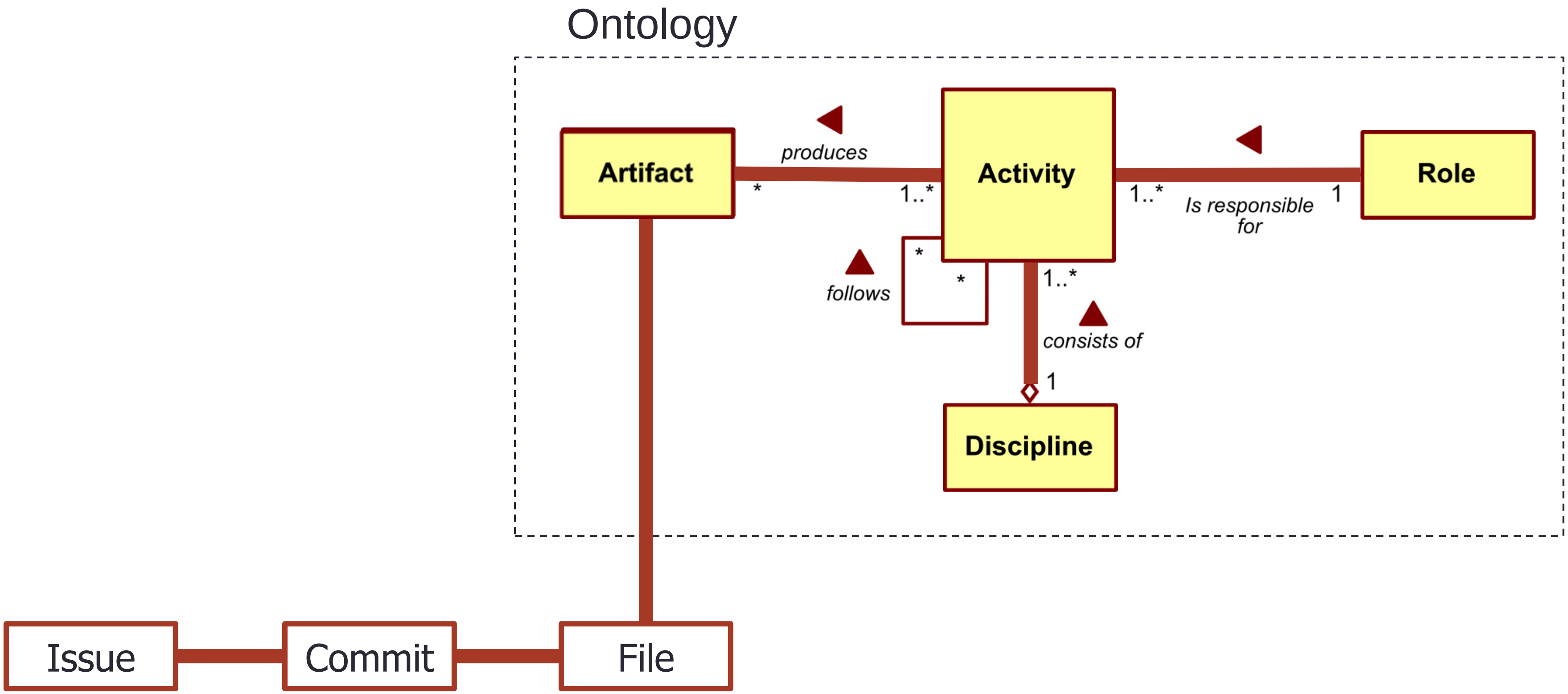
Limitations:

- Artifact produced within several activities;
- An issue cannot be linked to any commit;



Identifying roles and disciplines

Identify roles and disciplines

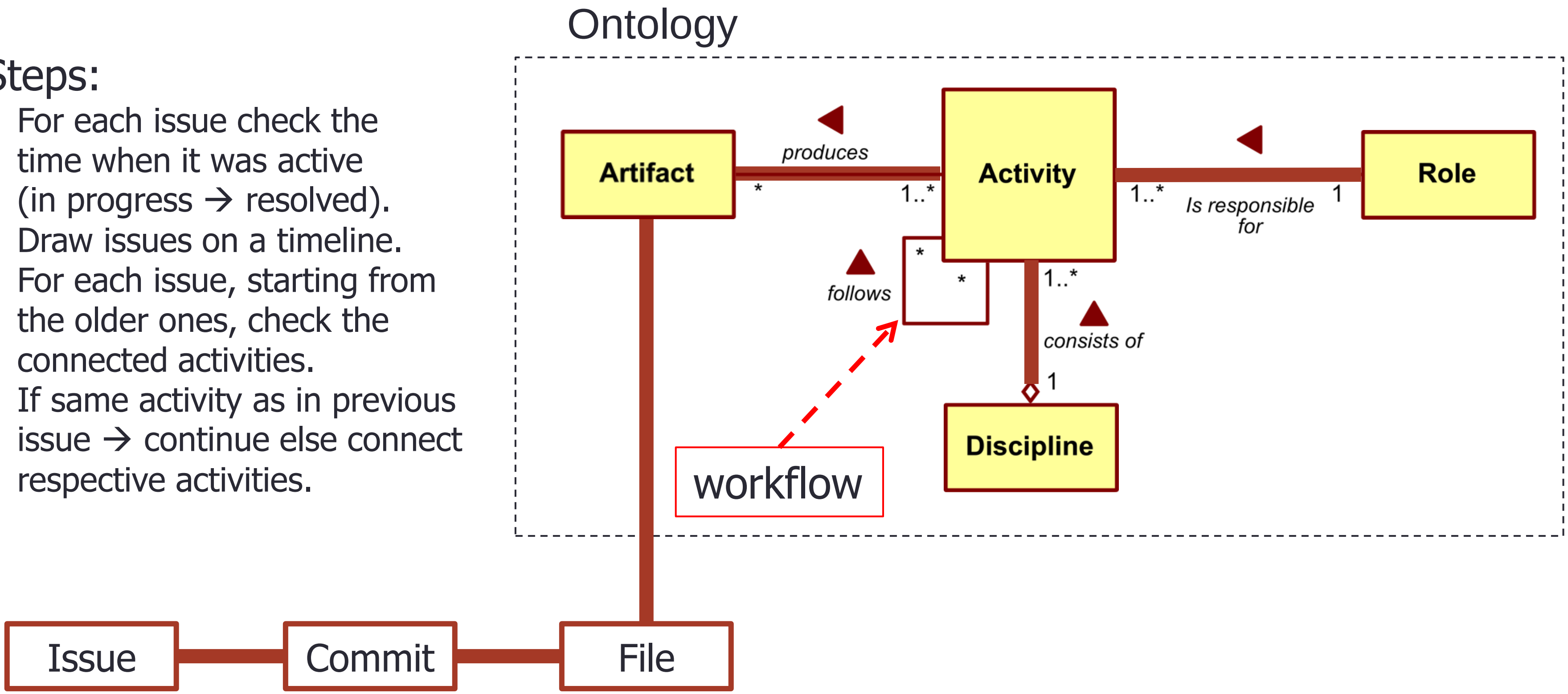


Identifying flow of activities

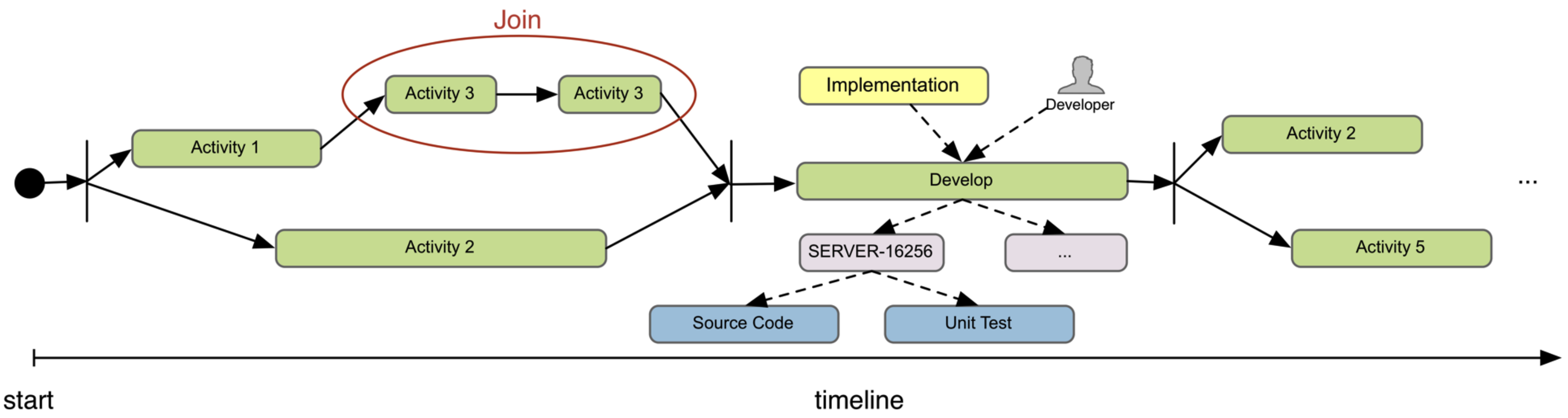
Identify workflow

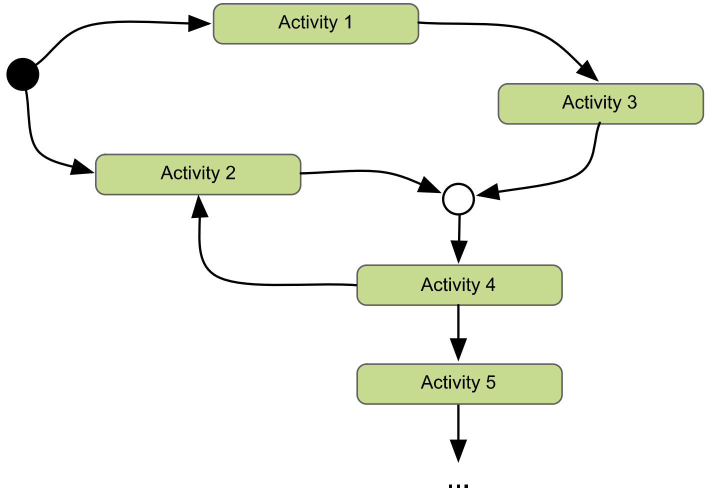
Steps:

- For each issue check the time when it was active (in progress → resolved).
- Draw issues on a timeline.
- For each issue, starting from the older ones, check the connected activities.
- If same activity as in previous issue → continue else connect respective activities.



Workflow visualization





Prerequisites

- For our approach to work the following is assumed:
  - Commits are a consequence of creating or changing artifacts through tasks defined as issues.
  - The majority of commits and associated artifacts can be traced back to an exact issue that triggered the creation/change of those artifacts.
  - An issue is a small piece of work usually assigned to one developer only.
  - Issue statuses (opened, in progress, …, closed) and links among issues are strictly logged by developers.

How limiting are the prerequisites…

- Five projects analyzed, three open source and two commercial.

Open source project

Mongo DB

Started in Oct 2007
15.292 issues in Jira
28.374 commits in GitHub
Code Review in Rietveld

Open source project

Spring Framework

Started in 2003
12.467 issues in Jira
9.696 commits in GitHub

Open source project

Hibernate ORM

Started in 2003
9.419 issues in Jira
5.673 commits in GitHub

Commercial project

IS for insurance industry

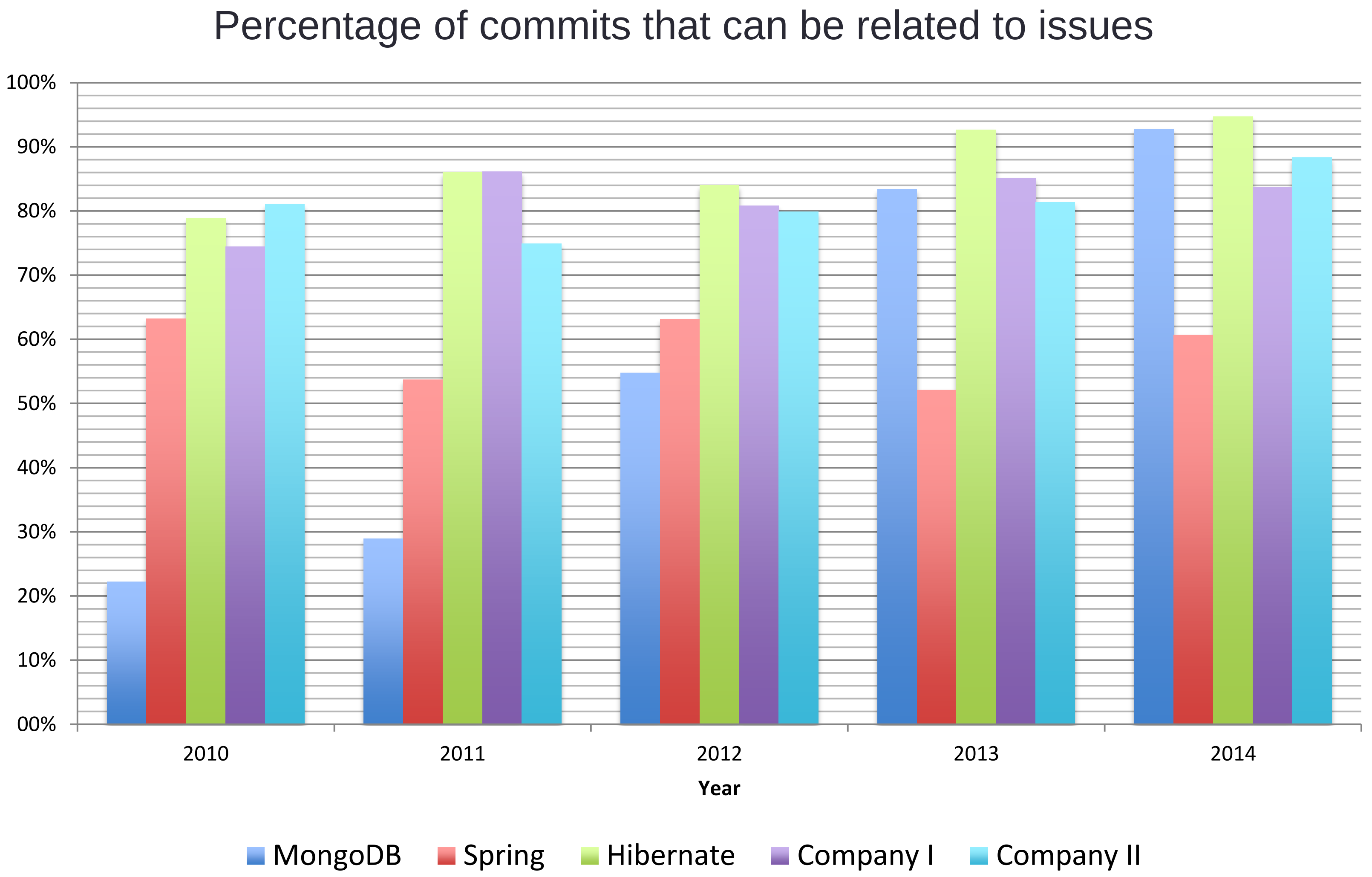
Company with 250 emp.
Project started in 2007
Deployed to 15+ organiz.
13.389 issues in Jira
18.571 commits in SVN
Project mngm: SCRUM

Commercial project

Billing for Utilities

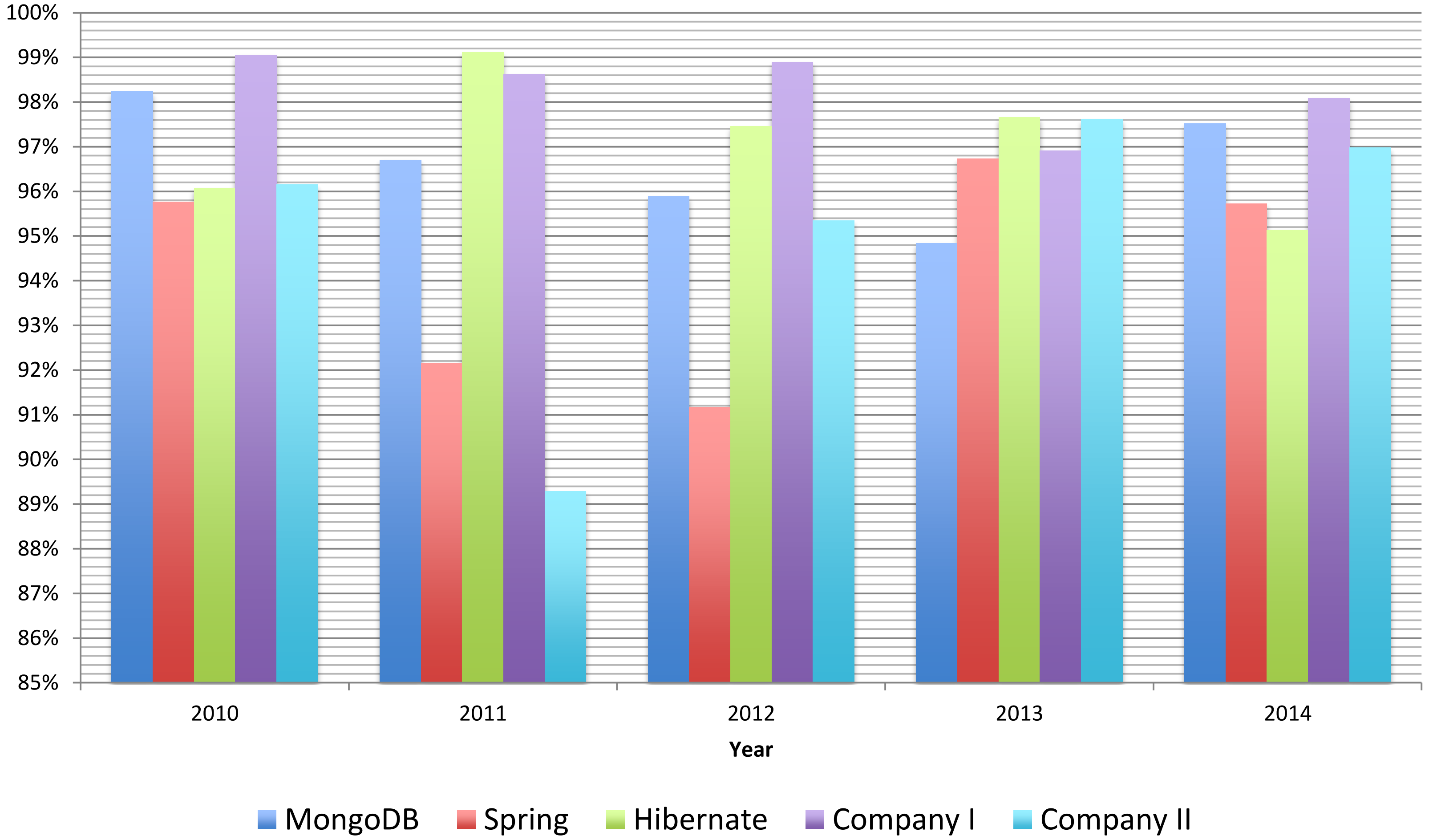
Company with 30 emp.
Project started in 2008
5.148 issues in Jira
13.735 commits in SVN
Project mngm: SCRUM

Results

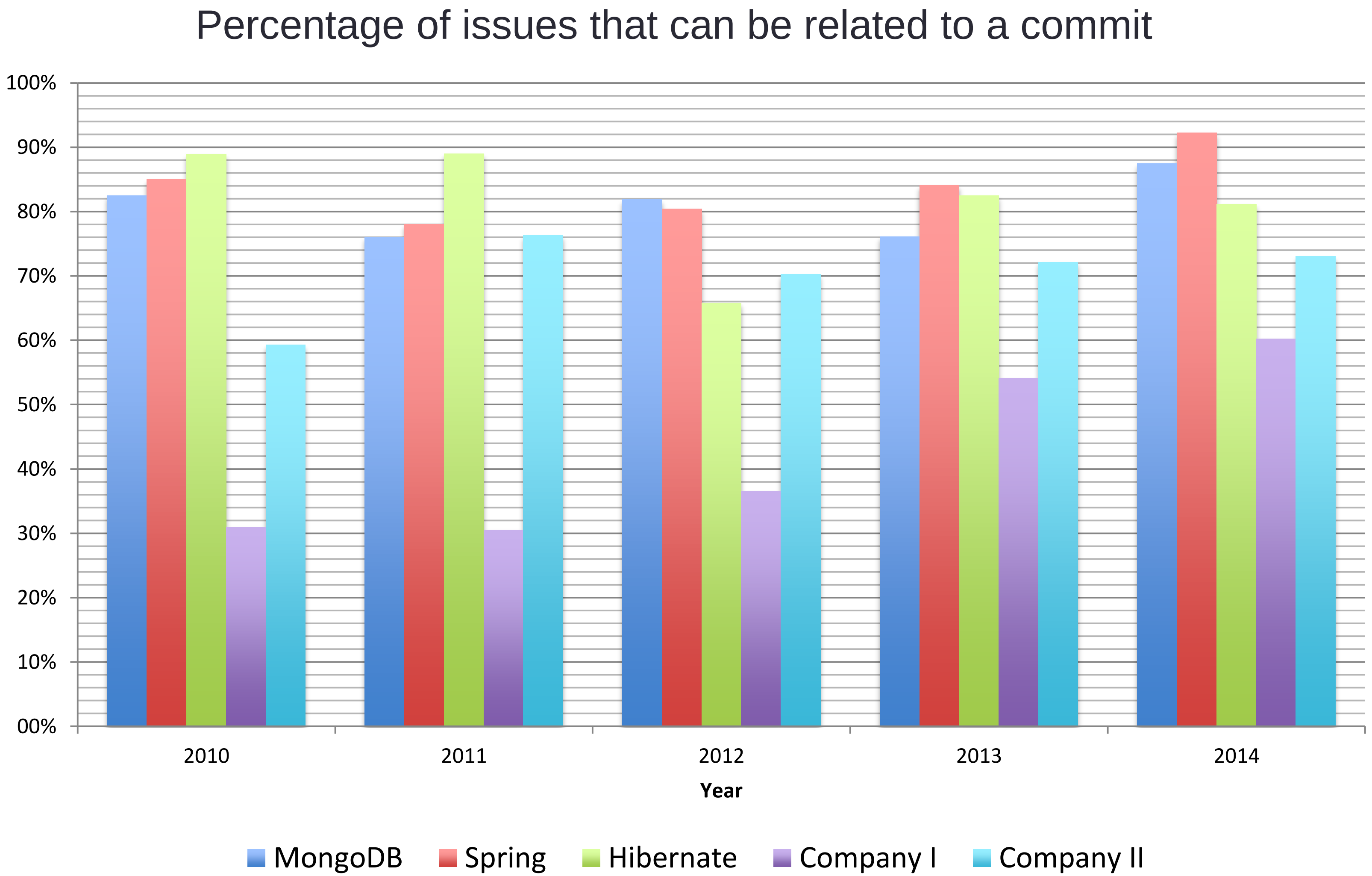

Percentage of commits that can be related to issues
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
00%
2010
2011
2012
2013
2014
Year
MongoDB
Spring
Hibernate
Company I
Company II

Results

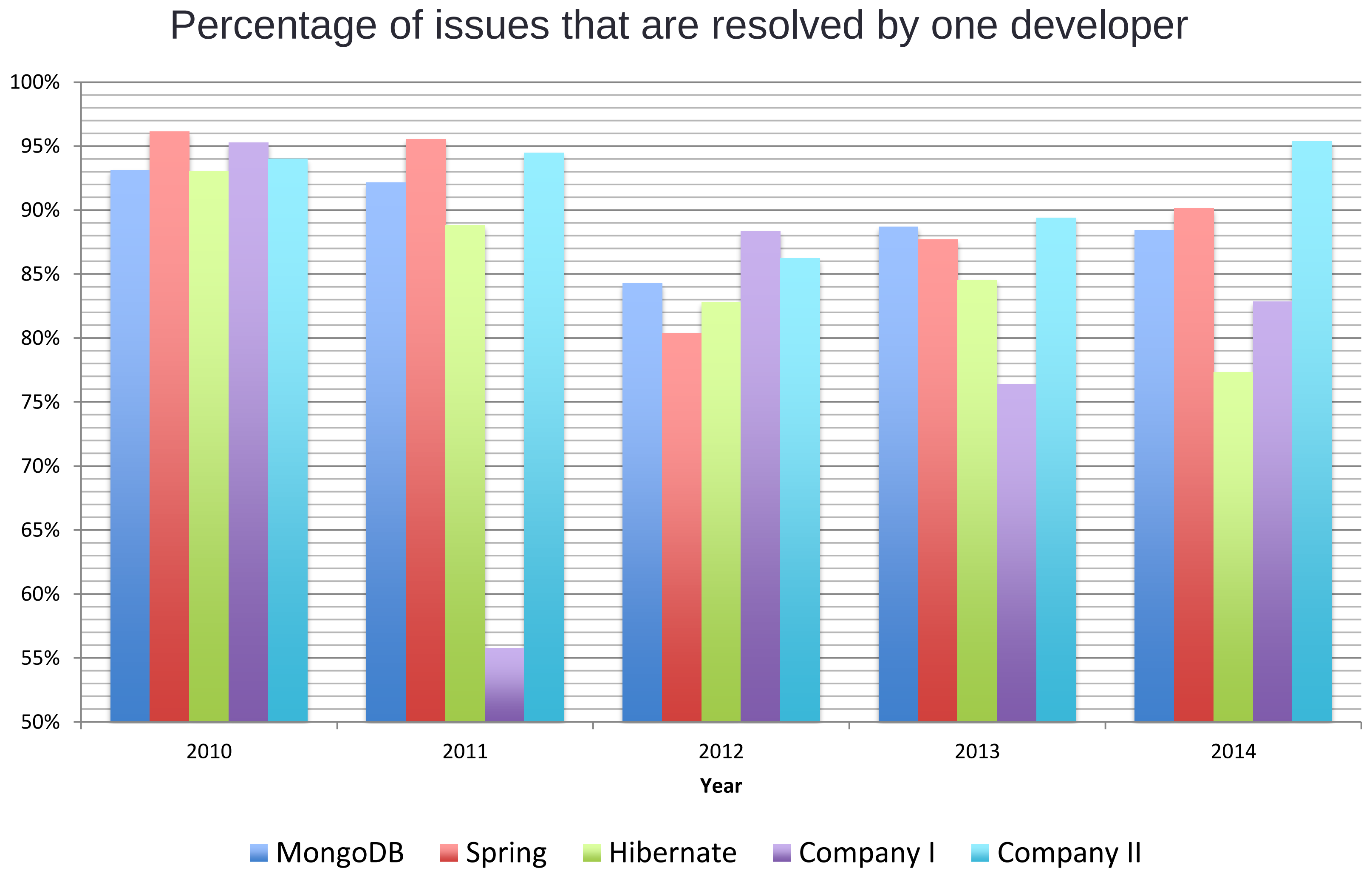
Percentage of commits that can be related to exactly one issue



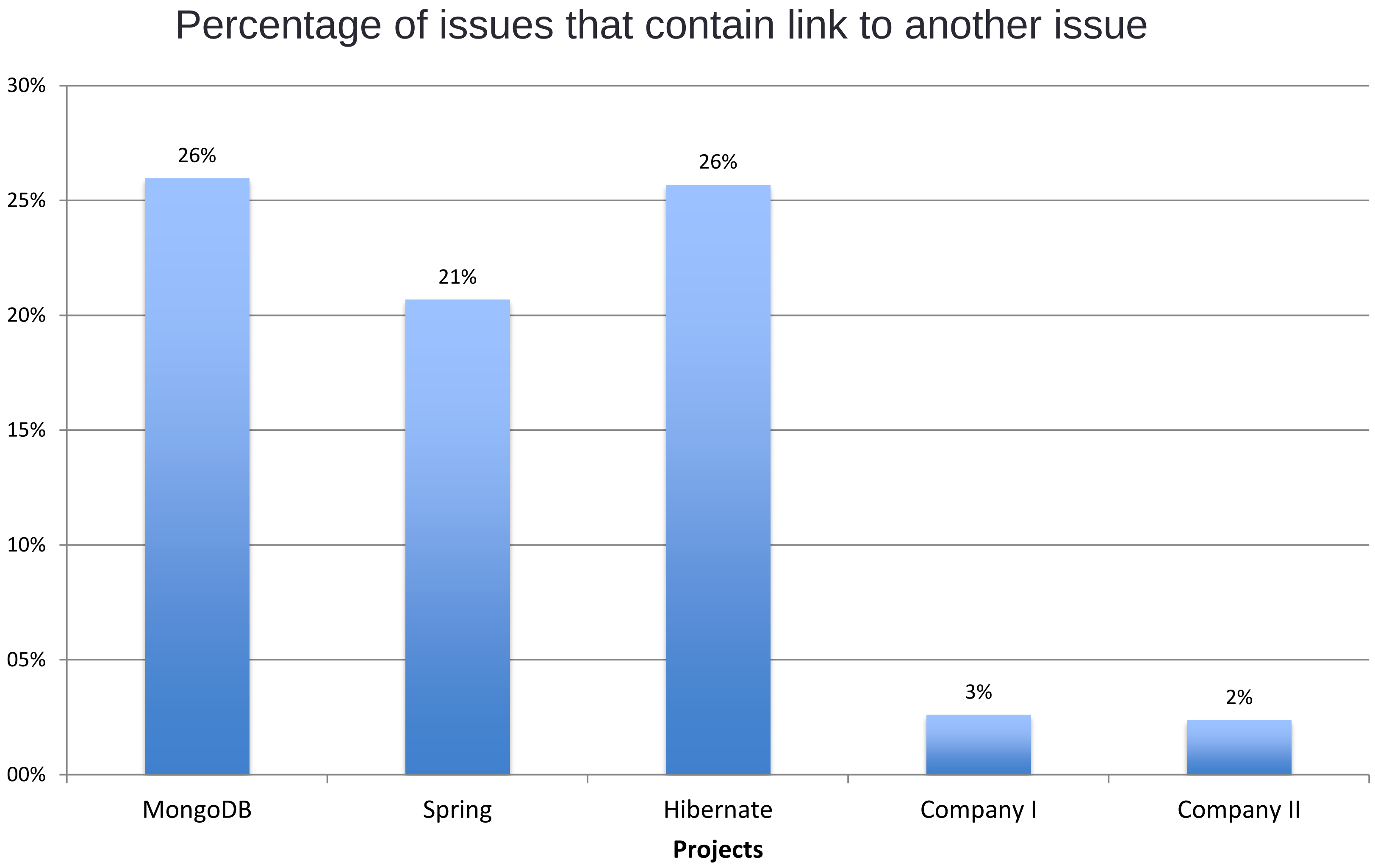
Results


Percentage of issues that can be related to a commit
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
00%
2010
2011
2012
2013
2014
Year
MongoDB
Spring
Hibernate
Company I
Company II

Results

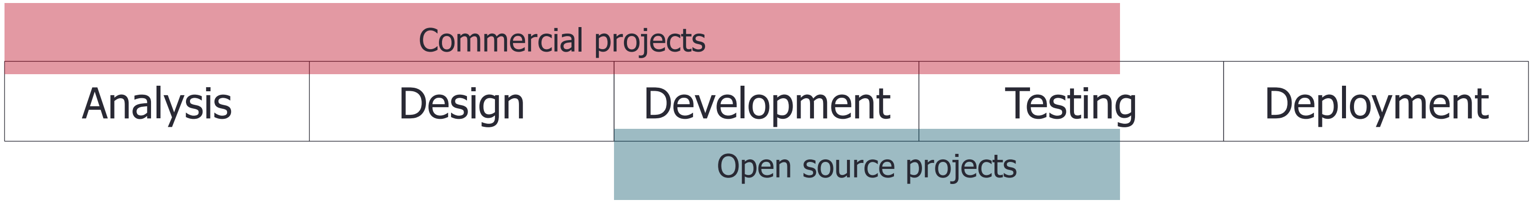

Percentage of issues that are resolved by one developer
100%
95%
90%
85%
80%
75%
70%
65%
60%
55%
50%
2010
2011
2012
2013
2014
Year
MongoDB
Spring
Hibernate
Company I
Company II

Results


Percentage of issues that contain link to another issue
30%
25%
20%
15%
10%
05%
00%
26%
21%
26%
3%
2%
MongoDB
Spring
Hibernate
Company I
Company II
Projects

Additional findings

- Commercial projects usually keep detailed worklogs (e.g. time spent for an issue → date, hours, user…).
- Commercial projects have wider coverage:

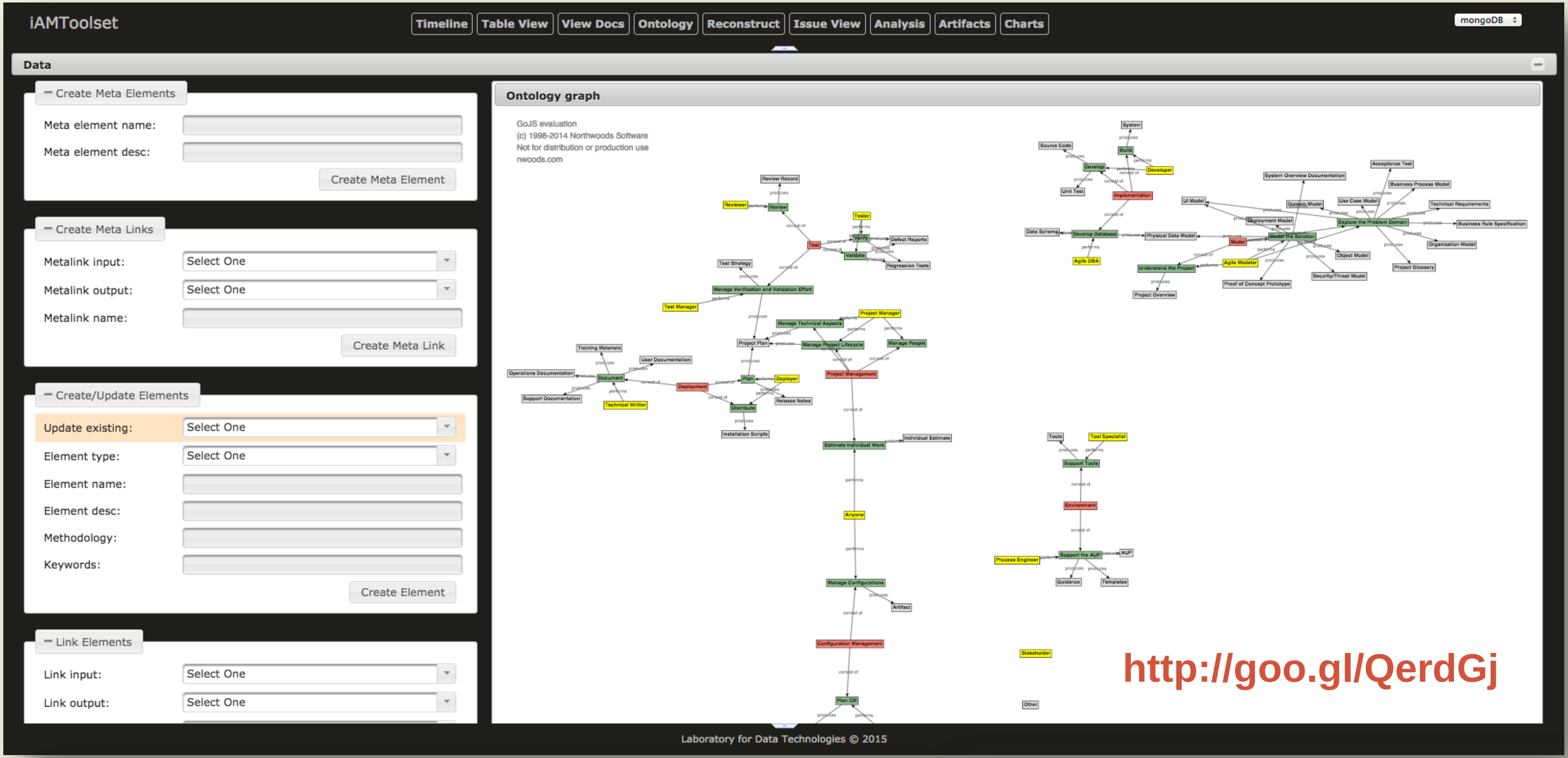


Commercial projects

Analysis Design Development Testing Deployment

Open source projects

- Users on open source projects are more disciplined in logging information to software repositories (e.g. issue status).
- Different tools of same software repositories store the all the data needed for reconstruction.



Data

Create Meta Elements

Meta element name:

Meta element desc:

Create Meta Element

Create Meta Links

Metalink input: Select One

Metalink output: Select One

Metalink name:

Create Meta Link

Create/Update Elements

Update existing: Select One

Element type: Select One

Element name:

Element desc:

Methodology:

Keywords:

Create Element

Link Elements

Link input: Select One

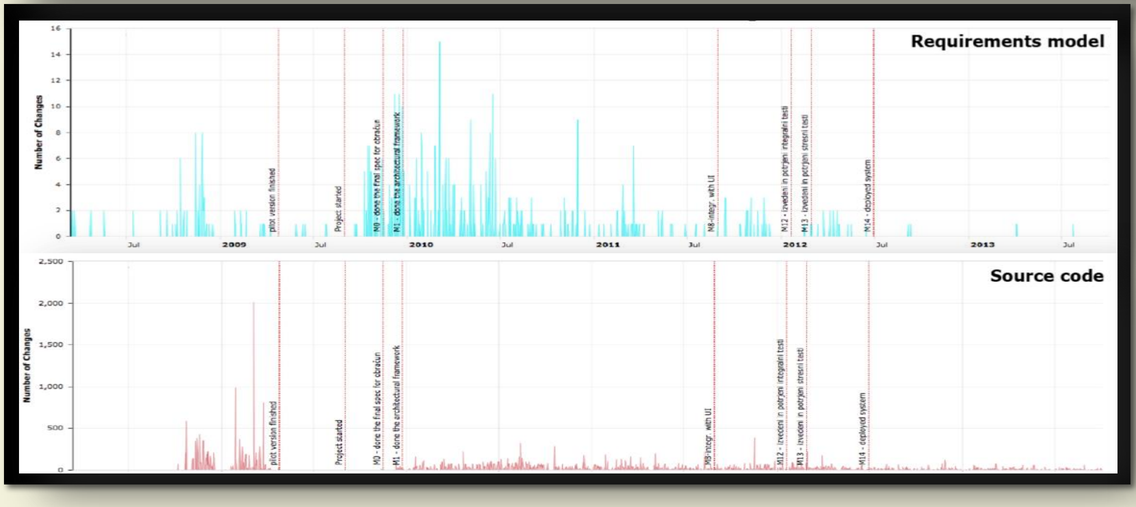
Link output: Select One

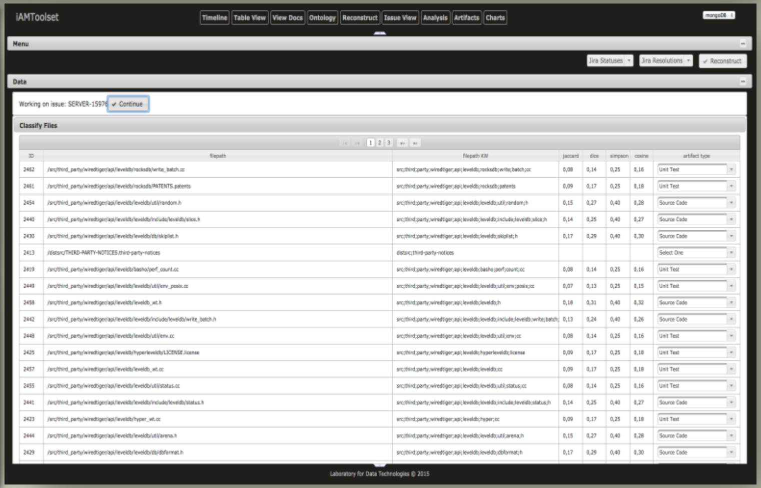
Ontology graph

GoJS evaluation
(c) 1998-2014 Northwoods Software
Not for distribution or production use
nwoods.com

http://goo.gl/QerdGj

Laboratory for Data Technologies © 2015





Working on issue: SERVER-15976

Classify Files

ID	filepath	filepath KW	jaccard	dice	simpson	cosine	artifact type
2462	/src/third_party/wiredtiger/api/leveldb/rocksdb/write_batch.cc	src;third;party;wiredtiger;api;leveldb;rocksdb;write;batch;cc	0,08	0,14	0,25	0,16	Unit Test
2461	/src/third_party/wiredtiger/api/leveldb/rocksdb/PATENTS.patents	src;third;party;wiredtiger;api;leveldb;rocksdb;patents	0,09	0,17	0,25	0,18	Unit Test
2454	/src/third_party/wiredtiger/api/leveldb/leveldb/util/random.h	src;third;party;wiredtiger;api;leveldb;leveldb;util;random;h	0,15	0,27	0,40	0,28	Source Code
2440	/src/third_party/wiredtiger/api/leveldb/leveldb/include/leveldb/slice.h	src;third;party;wiredtiger;api;leveldb;leveldb;include;leveldb;slice;h	0,14	0,25	0,40	0,27	Source Code
2430	/src/third_party/wiredtiger/api/leveldb/leveldb/db/skiplist.h	src;third;party;wiredtiger;api;leveldb;leveldb;skiplist;h	0,17	0,29	0,40	0,30	Source Code
2413	/distsrc/THIRD-PARTY-NOTICES.third-party-notices	distsrc;third-party-notices					Select One
2419	/src/third_party/wiredtiger/api/leveldb/basho/perf_count.cc	src;third;party;wiredtiger;api;leveldb;basho;perf;count;cc	0,08	0,14	0,25	0,16	Unit Test
2449	/src/third_party/wiredtiger/api/leveldb/leveldb/util/env_posix.cc	src;third;party;wiredtiger;api;leveldb;leveldb;util;env;posix;cc	0,07	0,13	0,25	0,15	Unit Test
2458	/src/third_party/wiredtiger/api/leveldb/leveldb_wt.h	src;third;party;wiredtiger;api;leveldb;leveldb;h	0,18	0,31	0,40	0,32	Source Code
2442	/src/third_party/wiredtiger/api/leveldb/leveldb/include/leveldb/write_batch.h	src;third;party;wiredtiger;api;leveldb;leveldb;include;leveldb;write;batch;	0,13	0,24	0,40	0,26	Source Code
2448	/src/third_party/wiredtiger/api/leveldb/leveldb/util/env.cc	src;third;party;wiredtiger;api;leveldb;leveldb;util;env;cc	0,08	0,14	0,25	0,16	Unit Test
2425	/src/third_party/wiredtiger/api/leveldb/hyperleveldb/LICENSE.license	src;third;party;wiredtiger;api;leveldb;hyperleveldb;license	0,09	0,17	0,25	0,18	Unit Test
2457	/src/third_party/wiredtiger/api/leveldb/leveldb_wt.cc	src;third;party;wiredtiger;api;leveldb;leveldb;cc	0,09	0,17	0,25	0,18	Unit Test
2455	/src/third_party/wiredtiger/api/leveldb/leveldb/util/status.cc	src;third;party;wiredtiger;api;leveldb;leveldb;util;status;cc	0,08	0,14	0,25	0,16	Unit Test
2441	/src/third_party/wiredtiger/api/leveldb/leveldb/include/leveldb/status.h	src;third;party;wiredtiger;api;leveldb;leveldb;include;leveldb;status;h	0,14	0,25	0,40	0,27	Source Code
2423	/src/third_party/wiredtiger/api/leveldb/hyper_wt.cc	src;third;party;wiredtiger;api;leveldb;hyper;cc	0,09	0,17	0,25	0,18	Unit Test
2444	/src/third_party/wiredtiger/api/leveldb/leveldb/util/arena.h	src;third;party;wiredtiger;api;leveldb;leveldb;util;arena;h	0,15	0,27	0,40	0,28	Source Code
2429	/src/third_party/wiredtiger/api/leveldb/leveldb/db/dbformat.h	src;third;party;wiredtiger;api;leveldb;leveldb;dbformat;h	0,17	0,29	0,40	0,30	Source Code

Laboratory for Data Technologies © 2015

Next steps

- POC – accuracy of the reconstructed workflows – qualitative analysis with IT/Project managers;
- POC – usability of the approach for:
  - Guidance & Control (interviews with developers),
  - Knowledge acquisition and continuous improvement of the SDM (interviews with IT/Project managers),
  - Project quality analysis
- Workflow analysis: comparison of successful and failed projects.

Questions

University of Ljubljana
Faculty of Computer & Information Science
Vecna pot 113,
1000 Ljubljana

Marko Janković

Laboratory for Data Technologies

http://lpt.fri.uni-lj.si/

Contact:
e-mail: marko.jankovic@fri.uni-lj.si