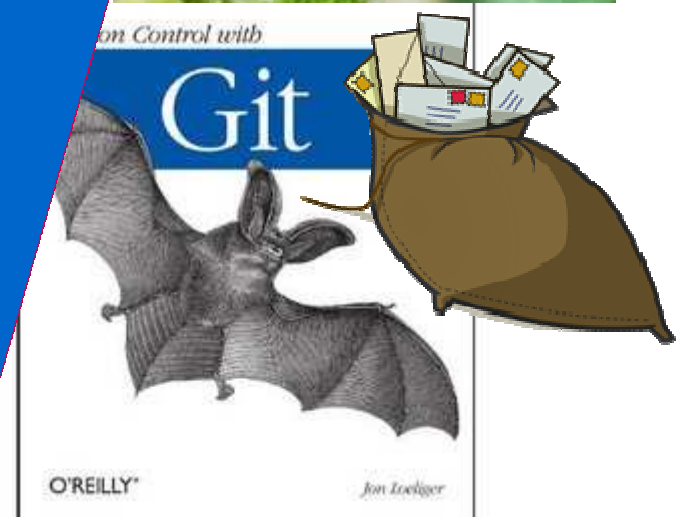


(Re-) discovering the history of your embedded software

Alexander Serebrenik
Mark van den Brand
Serguei Roubtsov
Tom Verhoeff



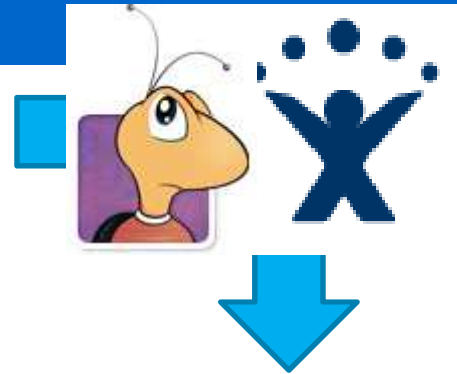
TU / **e** Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

It is all about communication...



Test #14352
fails sometimes



The error should be
somewhere here...
What does this code do?



I know how to fix it!

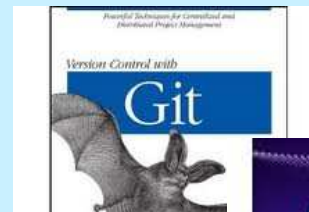


Tools record information

How can we
serve you?



Software repositories



How can we serve you?

- Is the documentation up-to-date?
- How fast are the bugs resolved?
- Who is responsible for
 - Bugs
 - Overtly complex code
 - Code guidelines violations?
- What parts are covered by tests?

**and many
more...**

Our studies so far

- **Open-Source software:**
 - developer roles
 - use of Bugzilla (intended vs. actual)
- **Student capstone projects**
 - adherence to guidelines
 - quality of the development process
 - developer roles
- **We are eager to cooperate with you and apply our techniques to your data!**

How does it work?

Sw Eng
Quest.

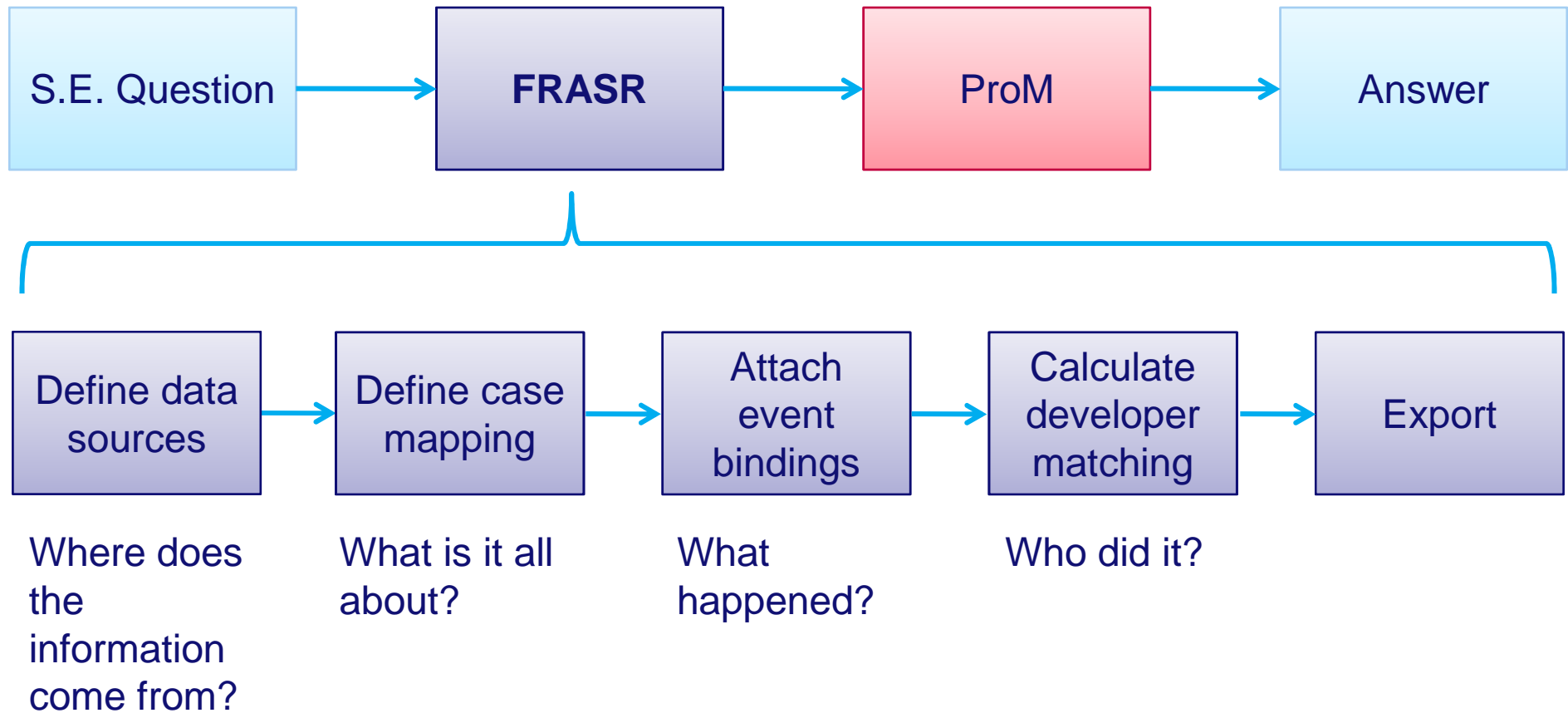
Sw Eng
Answer

The screenshot displays the FRASR (Framework for Analyzing Software Repositories) application. The interface includes a menu bar (Application, Project, Help) and a toolbar. The main window is divided into several sections:

- Data Sources:** A tree view on the left showing various sources like SVN, Bugtracker, SF bug repository, and Mailing list.
- Manage data sources:** A central panel with 'Add data source' and 'Delete data source' sections. The 'Add data source' section shows fields for Name, Comment, and Type (set to SVN).
- Modify data source:** A panel on the right for editing a selected source, showing 'SF bug repository - PhpMyAdmin (bugs)' with fields for Name and Comment.
- Analysis Results:** A bottom window titled 'Unstructured_example_mxml.gz' showing a 'Key data' sidebar, a 'Events per case' graph, 'Log info', and a 'Fuzzy Model Toolkit' with a network diagram.

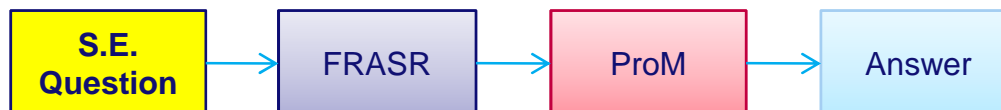
Combi-
ned log

How do we apply FRASR?



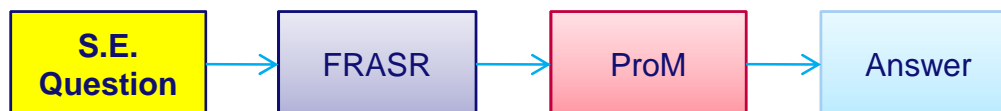
Case study 1: Developer roles

- Multiple sources + process mining (analysis)
- S.E. question
 - Classify developers according to their roles
- Classification of Nakakoji et al. IWPSE 2002: 8 roles
 - **Core member** involved for a relatively long period and made significant contributions to the development and evolution of the system
 - ≥ 3 years (project run: ≥ 8 years)
 - Version control: file added, file modified
 - More version control events than average



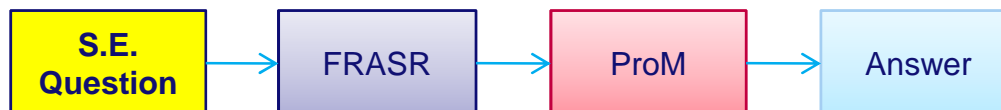
Case study 1: System under investigation

- **aMSN: instant messaging application**
 - **38 million downloads, 20th most popular at SourceForge**
 - **February 26, 2002 – July 9, 2010**
 - **7 bug repositories: 3137 bug reports**
 - **3 mail archives: 34947 messages**
 - **Subversion: 12062 commits**

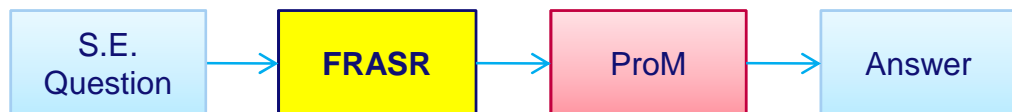


Case study 1: FRASR configuration

- We are interested in developers \Rightarrow case = developer
- Each data source type requires specific extraction technique \Rightarrow event-binding = type-specific
- 1725 developers \Rightarrow matching = heuristic

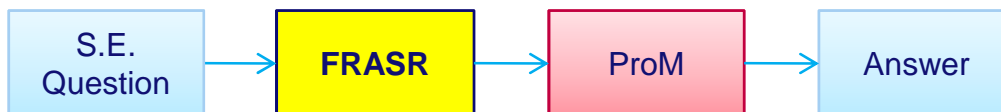


Case study 1: Results



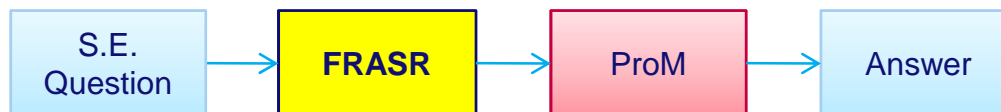
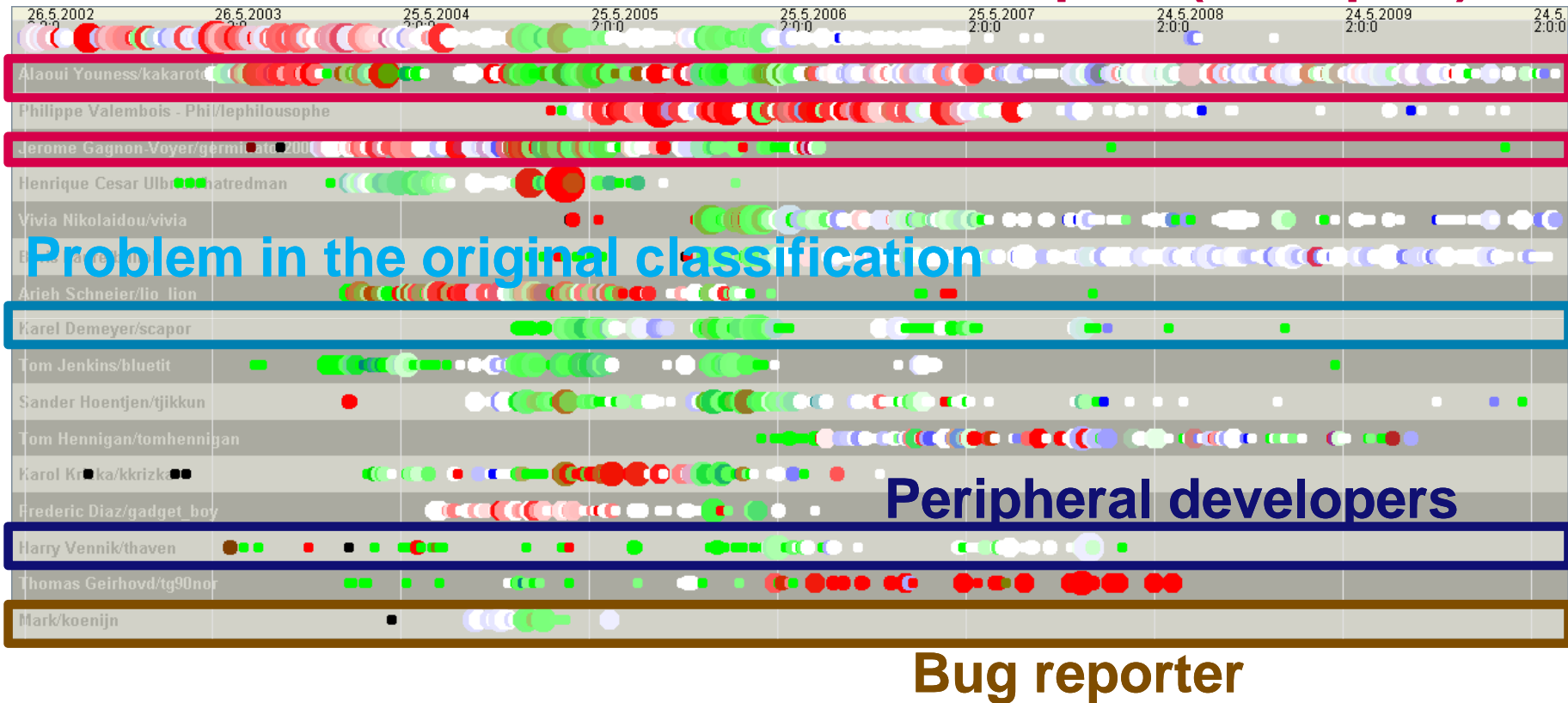
Case study 1: Results

ProM Dotted Chart visualization



Case study 1: Results

Core developers (examples)

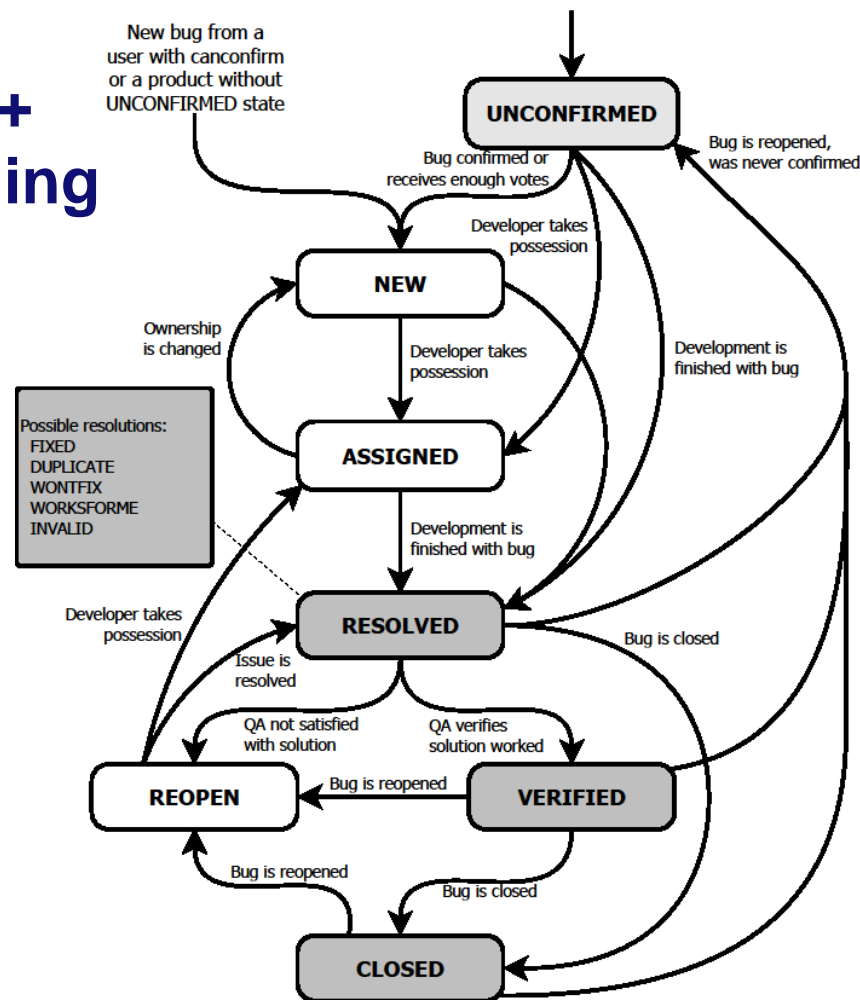


Case study 1: Classification

<i>Role</i>	<i>#developers</i>	
Bug reporter	1443	
Bug fixer	3	Bugs are usually fixed by peripheral developers
Peripheral developer	29	
Active developer	6	
Core member	7	
Project leader	3	Only ticket-commented or mail-reply
Other	234	
Total	1725	

Case study 2: Bug life cycle in Bugzilla Theory according to the Bugzilla Guide

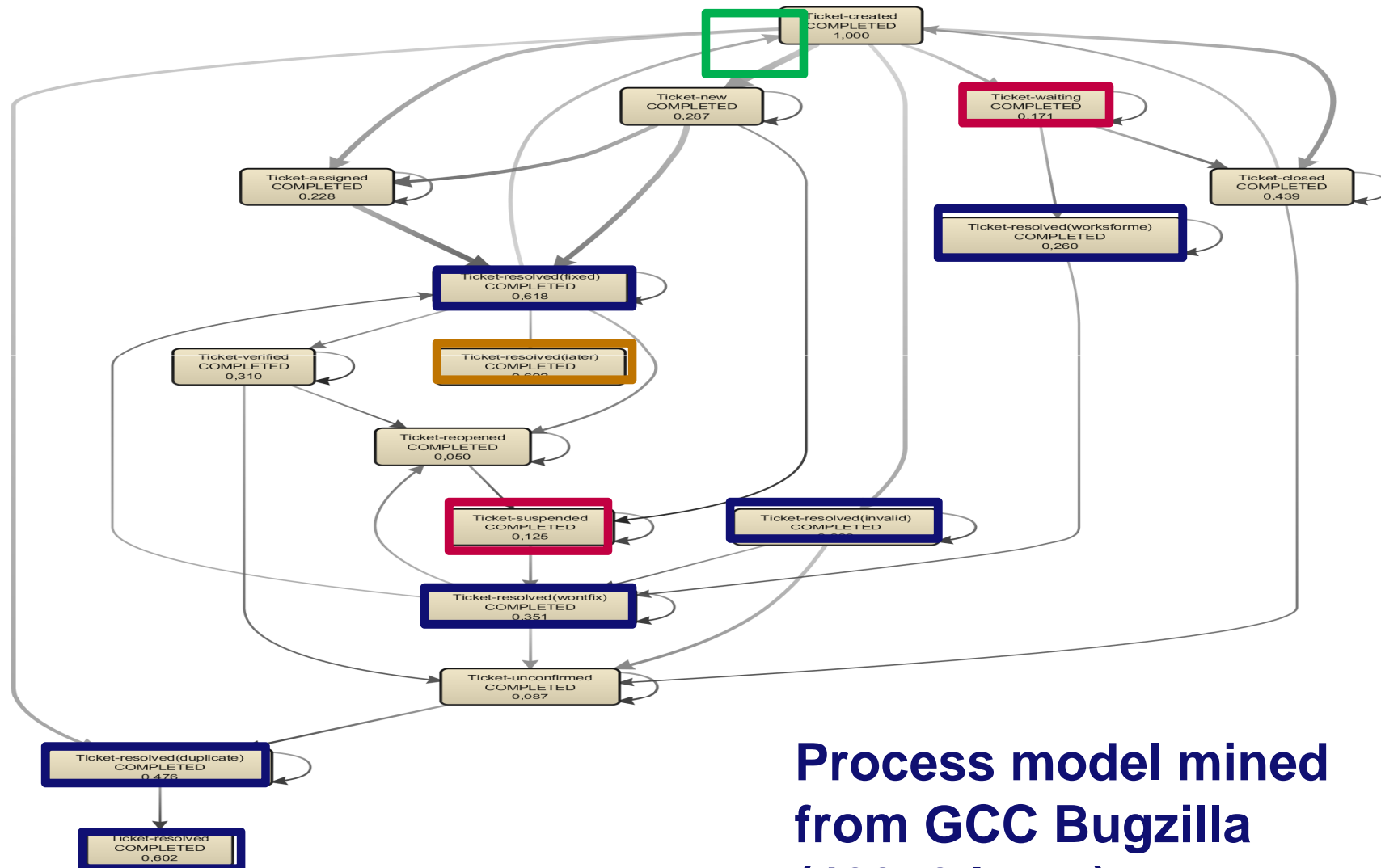
One source +
process mining
(mining)



S.E. question:
Is Bugzilla used
the way it is
supposed to be?

Case study 2: Bug life cycle in Bugzilla

Practice vs. Theory



Process model mined
from GCC Bugzilla
(42373 bugs) TU/e

TU/e
Eindhoven
University of Technology

Case study 3: Capstone projects at TU/e

- **Customer:**
 - **SME, multinationals, research institutions, non-profit org.**
- **Task:**
 - **Middle-sized SW development**
 - **ESA standard**

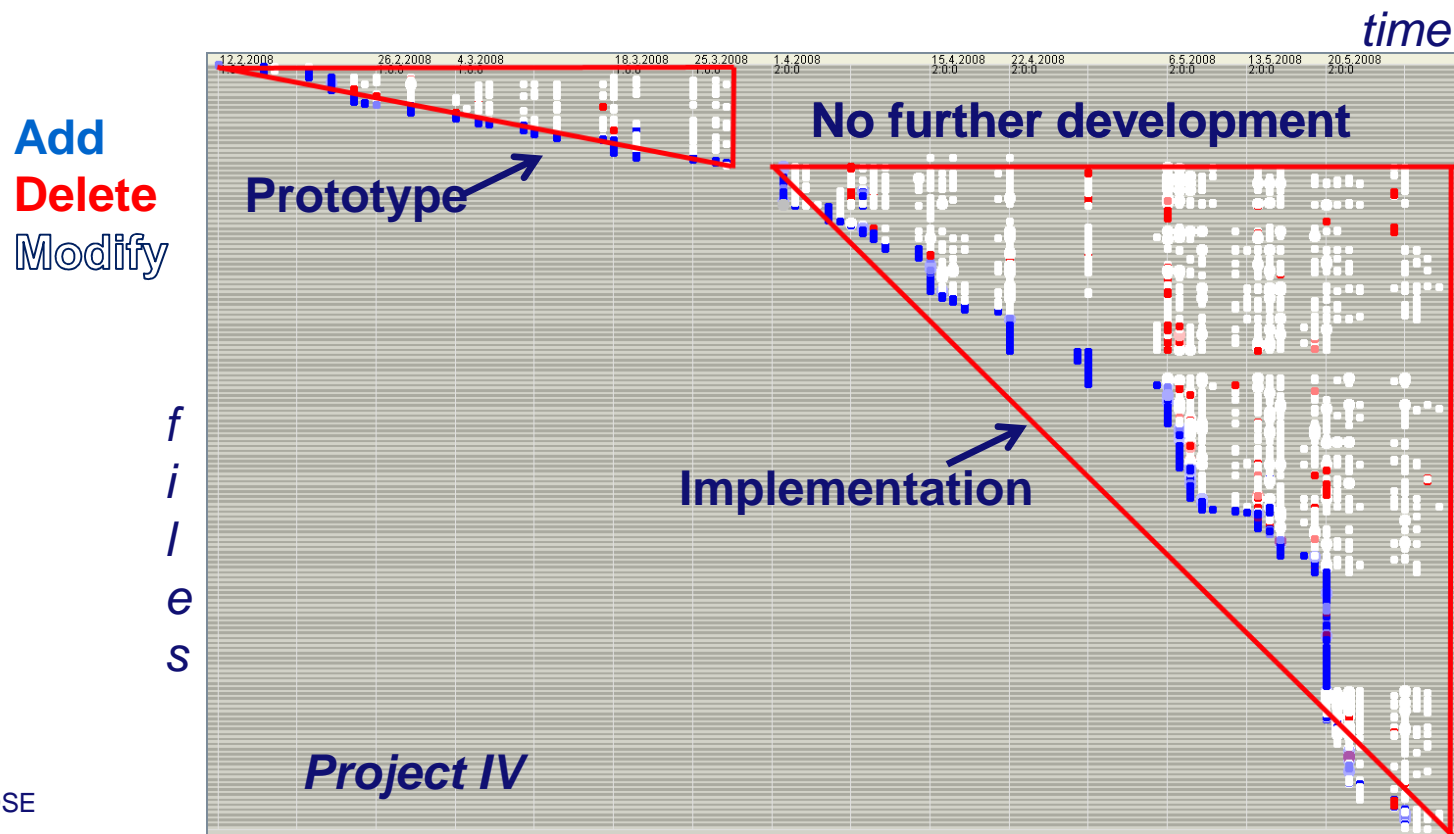
- **7-10 3rd year bachelor students**
- **PM: master student**
- **Technical advisor: staff member**
- **Senior management**

- **6 projects**

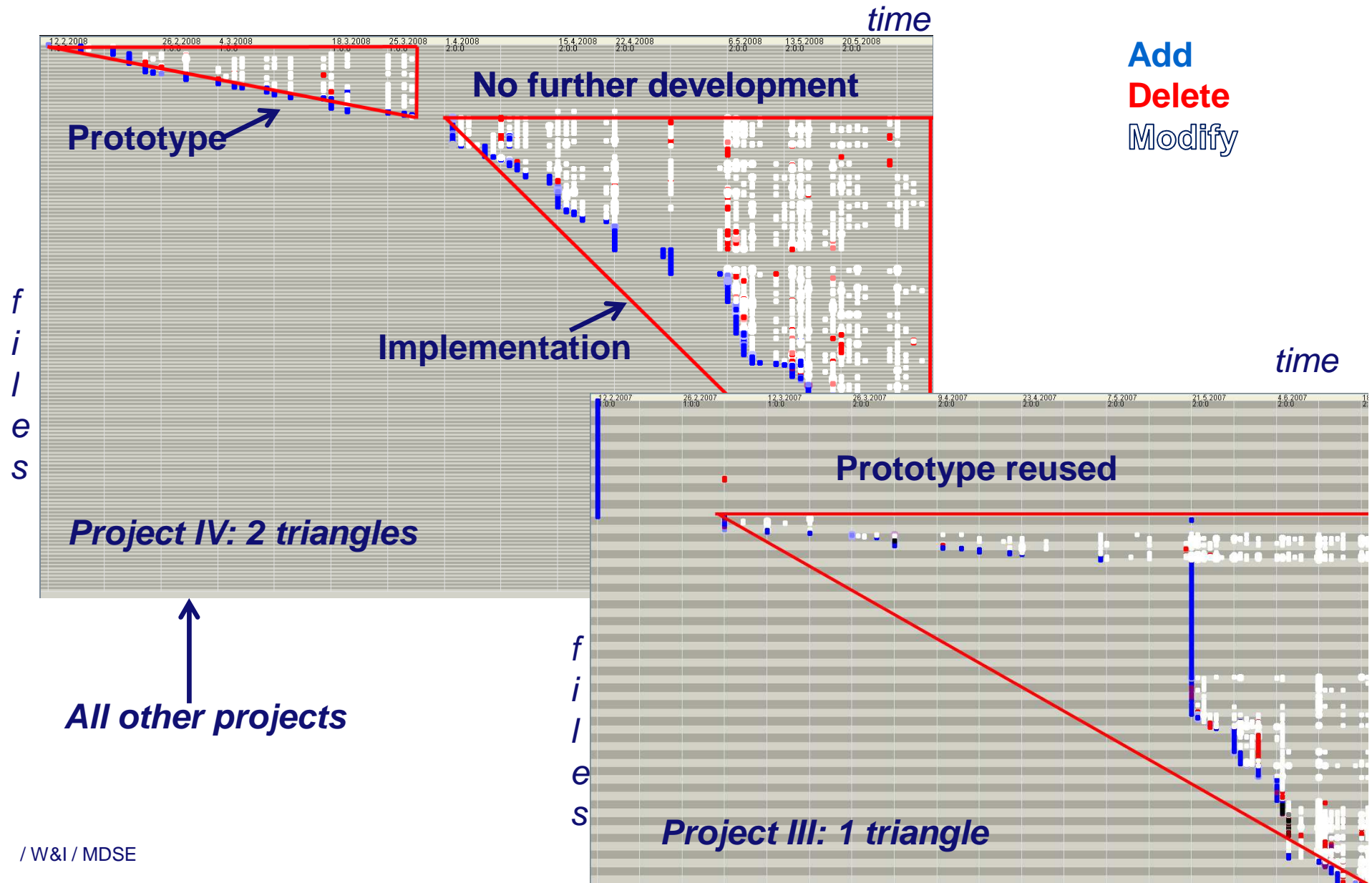


SE question 1: Did the students adhere to ESA guidelines in their development process?

- **Guideline:** “Do not reuse the prototype”!
- **Data source:** Version control system (Subversion)
- **Case:** Implementation files
- **Technique:** Dotted chart visualization

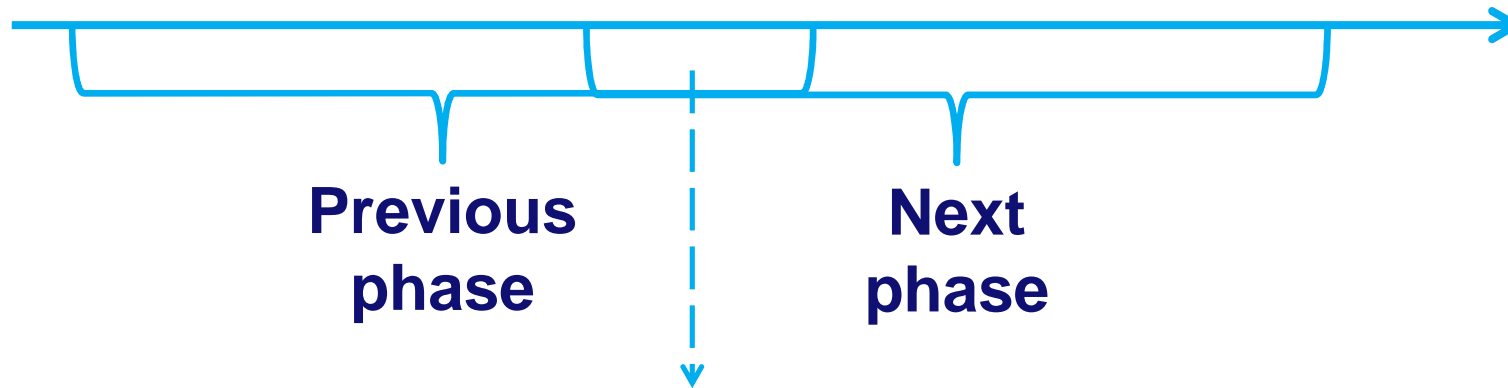


SE question 1: Did the students adhere to ESA guidelines in their development process?



SE question 2: Did the students adhere to the V-model?

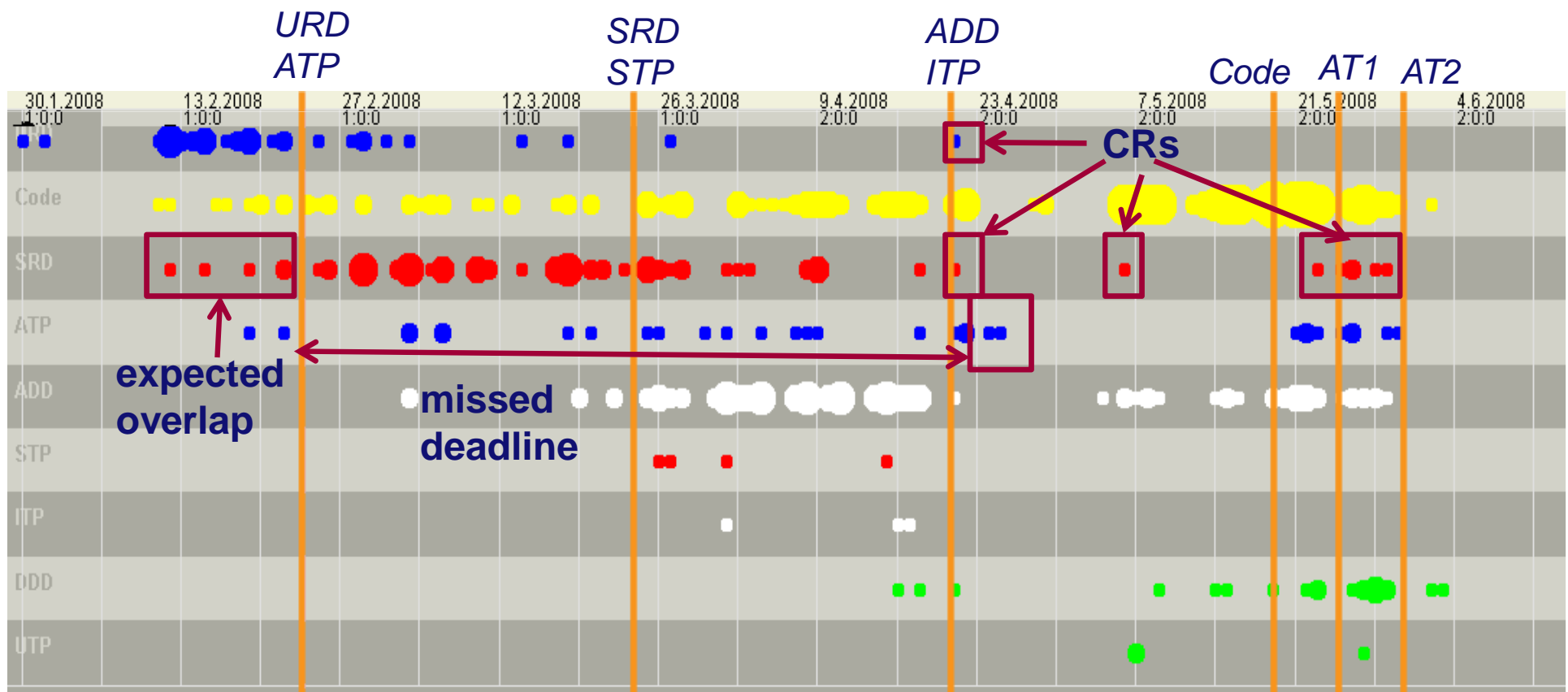
- Prescribed: V-model with limited overlap betw. phases



- Experts know constraints previous imposes on next
- Students have to learn these
 - If constraints are discovered before the completion of the preceding phase, the deliverables of the preceding phase can be easily adapted
 - Otherwise, ESA prescribes a CR procedure.

SE question 2: Did the students adhere to the V-model?

- Data sources: version control system (*Project IV*)
- Case: Files, grouped
- Technique: Dotted chart visualization
 - Deadlines added manually



SE question 2: Did the students adhere to the V-model?

- Other projects:

I	URD \cap SRD, ADD \cap DDD
II	No significant overlap
III	SRD \cap ADD more than 50% of the time! Corrective action could have been considered
V	SRD \cap ADD, ADD \cap DDD
VI	URD \cap SRD

SE question 3: Do the students experience all aspects of a software dev process?

- **Intention: students should play *all* roles**
 - Req engineer, architect, developer, tester, tech writer
- **Challenge: how to assess students individually?**
 - Well-known challenge in SE education
- **SE question: How were the tasks distributed?**
- **Data sources (Project IV):**
 - Subversion, Trac tickets and Wiki, mail archive
- **Case: Person**
- **Technique: “Originator-by-task” matrix**

SE question 3: Task distribution

Student	Management documents				Req.	Design	Code	Testing plans				Transfer	Process execution				Misc.			
	Project Management Plan (SPMP)	Configuration Management Plan (SCMP)	Quality Assurance Plan (SQAP)	Verification and Validation Plan (SVVP)	User Requirements Document (URD)	Software Requirements Document (SRD)	Architectural Design Document (ADD)	Detailed Design Document (DDD)	Code	Unit Testing Plan (UTP)	Integration Testing Plan (ITP)	System Testing Plan (STP)	Acceptance Testing Plan (ATP)	Software Transfer Document (STD)	Software User Manual (SUM)	Agenda	Minutes	Presentation	Process report	null
A	12	2	0	0	5	18	52	0	90	1	0	0	4	1	1	0	13	0	1	4
B	6	11	4	3	25	51	25	2	184	1	0	1	11	5	6	1	28	16	1	21
C	0	0	0	0	9	0	65	1	57	0	0	0	0	0	7	0	0	5	0	10
D	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	1	0	0	4
E	0	0	0	0	1	0	17	1	82	0	0	0	0	0	0	0	0	0	0	8
F	0	0	0	14	1	7	2	0	65	3	3	5	21	6	1	0	0	5	0	8
G	0	0	0	0	10	28	8	5	108	0	0	0	0	0	0	0	0	3	0	8
H	0	0	0	0	1	1	31	6	285	0	0	0	0	2	4	0	0	0	0	14
I	0	0	16	0	8	30	65	2	97	0	0	0	0	9	21	0	0	4	0	25
J	1	23	0	0	11	33	52	2	106	0	0	0	0	7	5	0	0	2	0	4
K	0	0	0	0	1	56	65	6	84	0	0	0	2	1	22	0	3	4	2	34

1. Calculate the matrix
2. Convert to shares per person
3. Calculate cosine similarity

Students prefer to specialize!

Our studies so far: Summarized

- **Open-Source software:**
 - developer roles
 - use of Bugzilla (intended vs. actual)
- **Student capstone projects**
 - adherence to guidelines
 - quality of the development process
 - developer roles
- **We are eager to cooperate with you and apply our techniques to your data!**



Mining...

- **Information is available in software repositories**
 - **Just waiting to be mined**
- **Numerous opportunities and chances**
- **Interested? Join us!**



LaQuSo
★ Laboratory for Quality Software
★