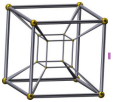




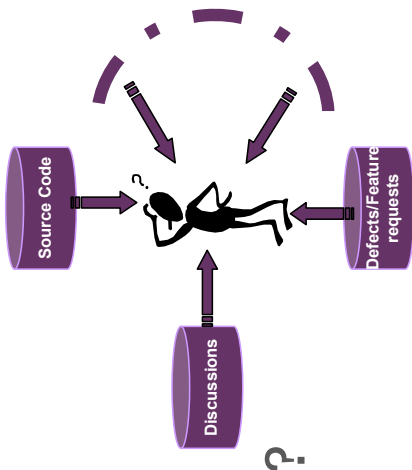
Tesseract: Interactive Visual Exploration of Socio-Technical Relationships in Software Development

Anita Sarma, Larry Maccherone, Patrick Wagstrom, and James Herbsleb
Institute for Software Research, School of Computer Science
Carnegie Mellon University

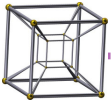


Some Questions in Software Development

- Who do I go to for help?
- Which other artifacts are affected by my changes?
- Which developer is affected by my changes?
- Whose changes are affecting my changes?
- Who should be assigned to this task?
- Which tasks need to be completed before the other?
- Which artifacts are brittle or buggy?
- ...

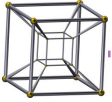


Answering these questions is non-trivial



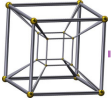
Problem

- Need for coordination arises because of complex relationships among project elements
- Data is siloed
- These relationships change over time
- Social and technical relationships are tightly coupled
- Lack of interactive exploratory environment for software projects



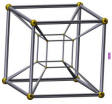
Objective

- Provide an interactive project exploration environment
 - aggregate data across data different sources
 - cross-link and visualize relationships
 - present changes in relationships over time
- treat social and technical relationships as first order objects
- Allow the 'lay user' to explore relationships and project dynamics



Research Challenges

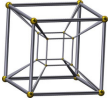
- Where to get the information?
- How to deal with scale?
- How to meaningfully extract and relate linkages?
- Allow investigating a particular problem
- Which information should be displayed?
- Help find interesting patterns



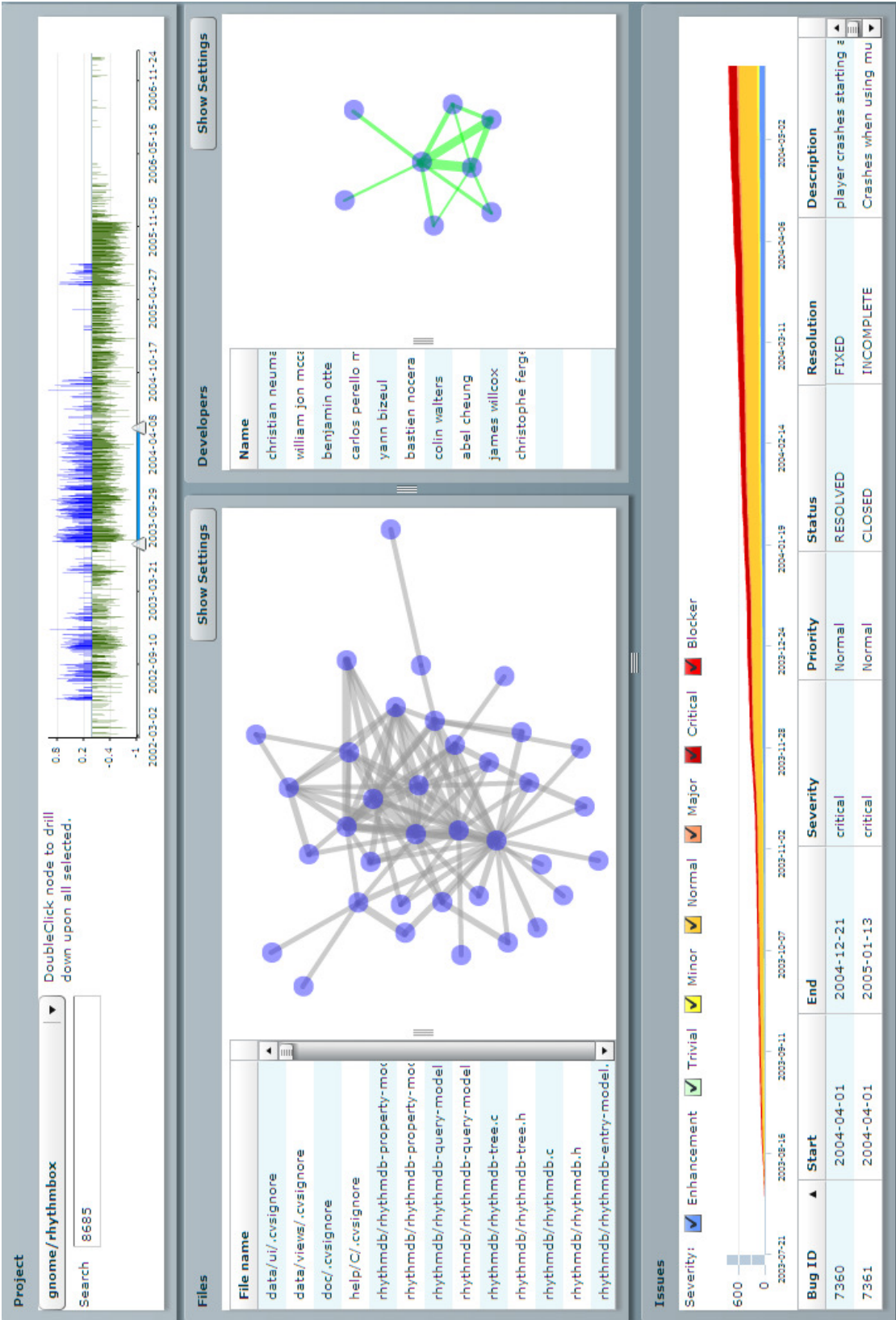
Tesseract

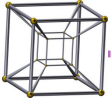
- Interactive and explorative environment to correlate and understand the complex relationships among:
 - code
 - developers
 - communication records
 - tasks (issues/ features)
 - time



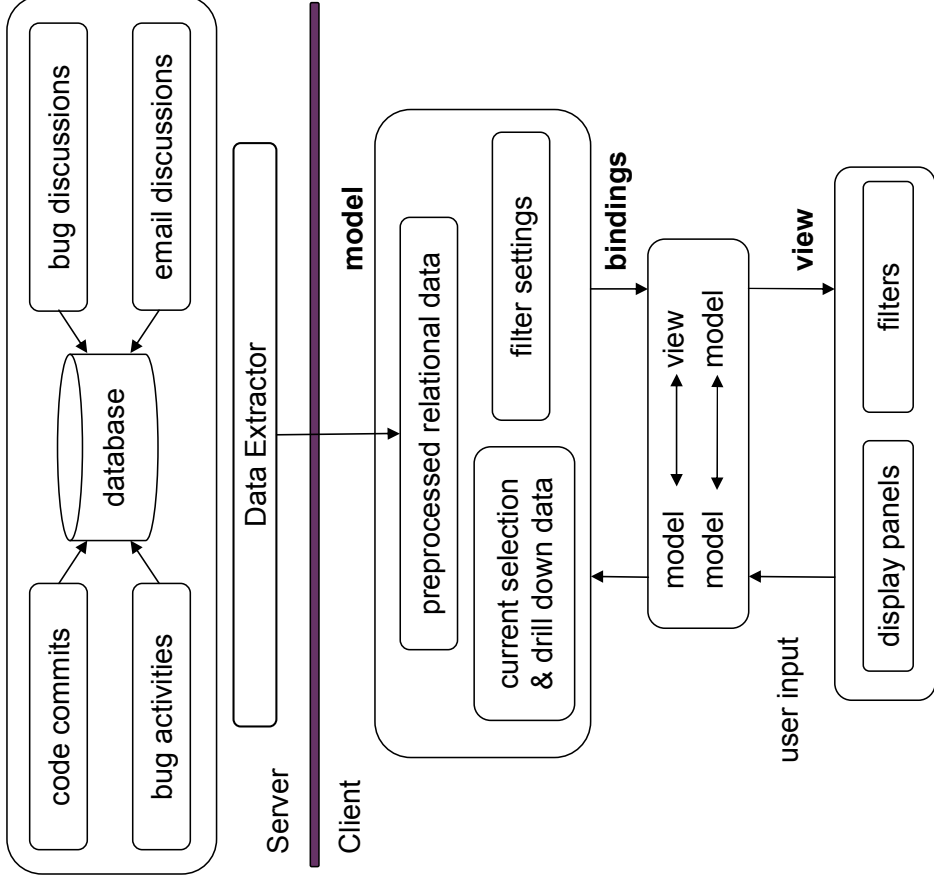


Tesseract

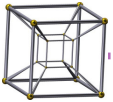




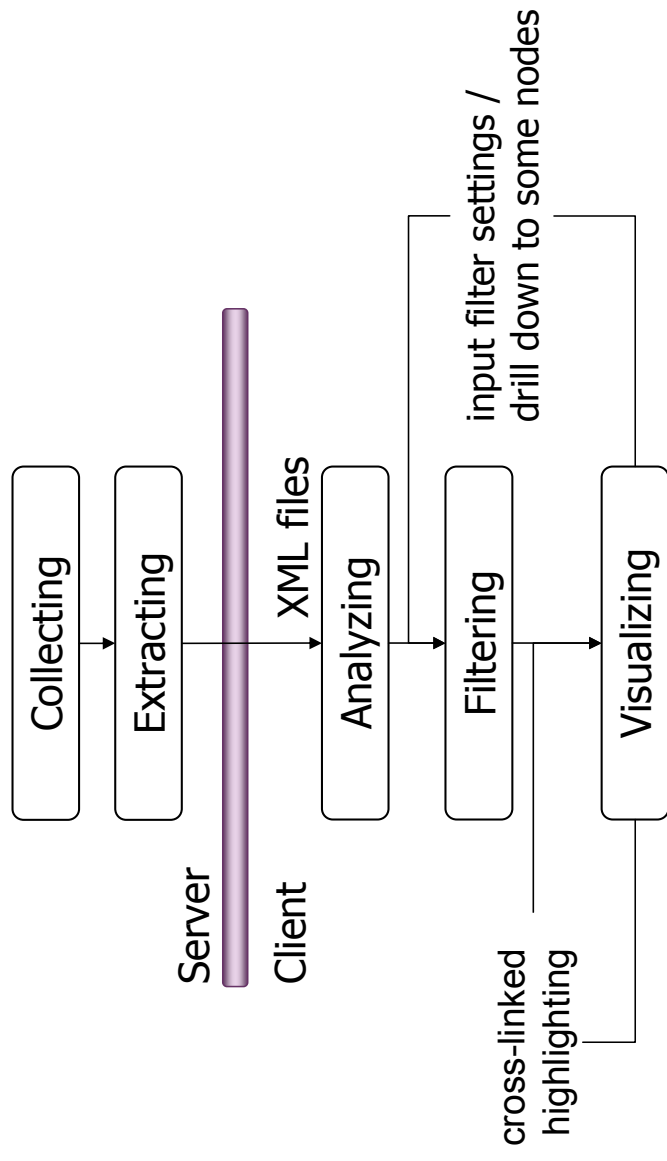
Design Decisions

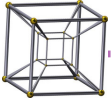


- Decoupling data collection from data consumption
- Easy substitution of linkage heuristics
- Easy substitution of visualization components

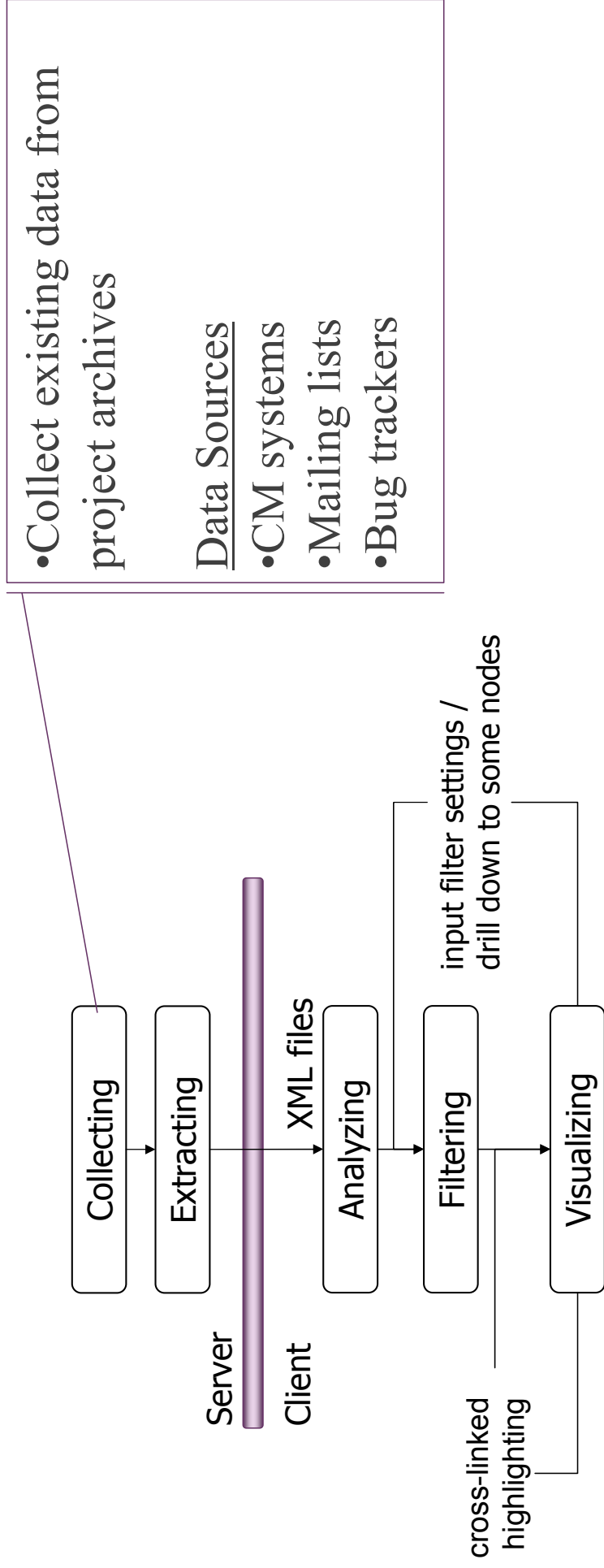


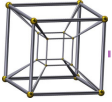
Information Flow



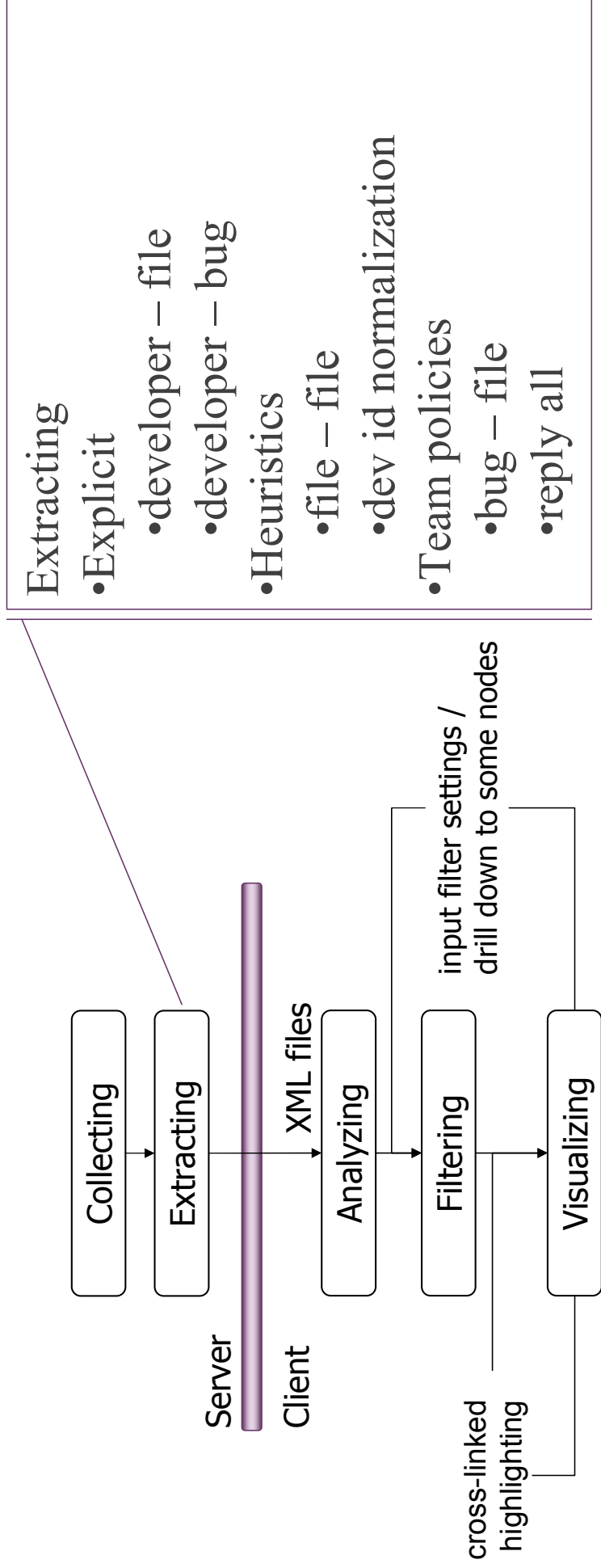


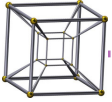
Information Flow



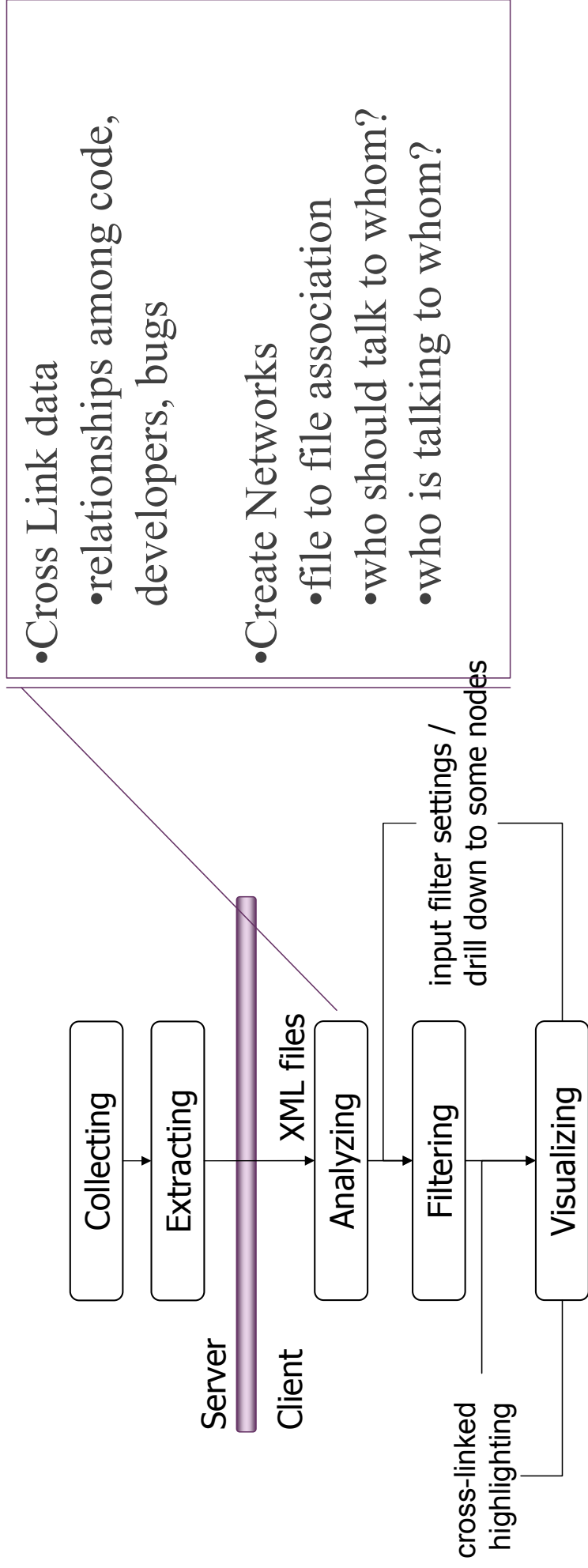


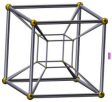
Information Flow





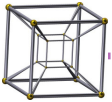
Information Flow





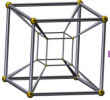
File Association

- Logical Coupling
 - Frequently co-committed files are logically associated with each other (Gall, Hajek, Jazayerri 1988)
- Especially useful in situations where
 - code base contains different programming language files
 - call site separated from target (e.g., network connection or even transmitted by event bus)
- Better measure of dependency for our purposes (Cataldo et al. 2006, 2008, 2009)



Developer Testimony

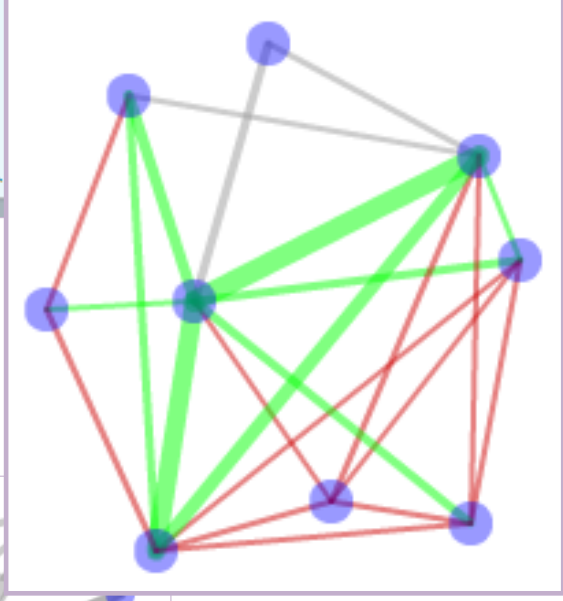
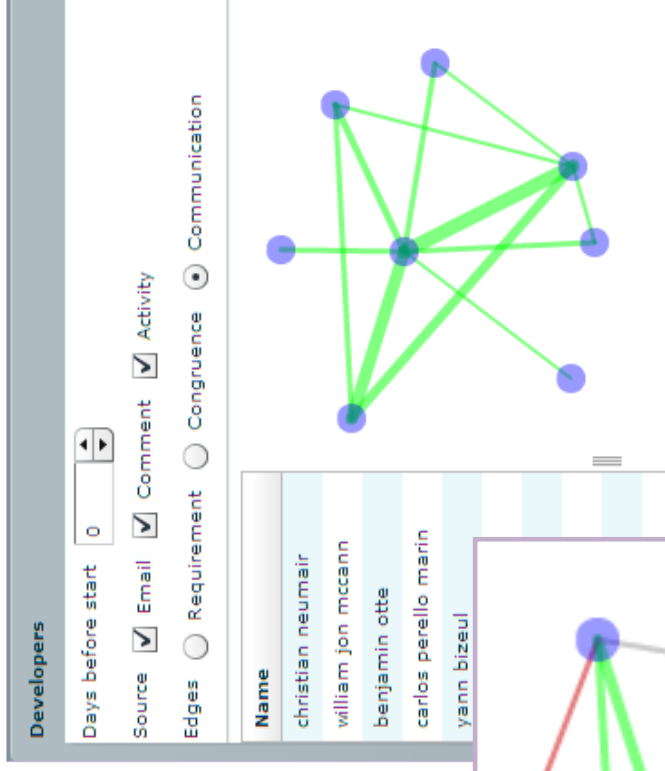
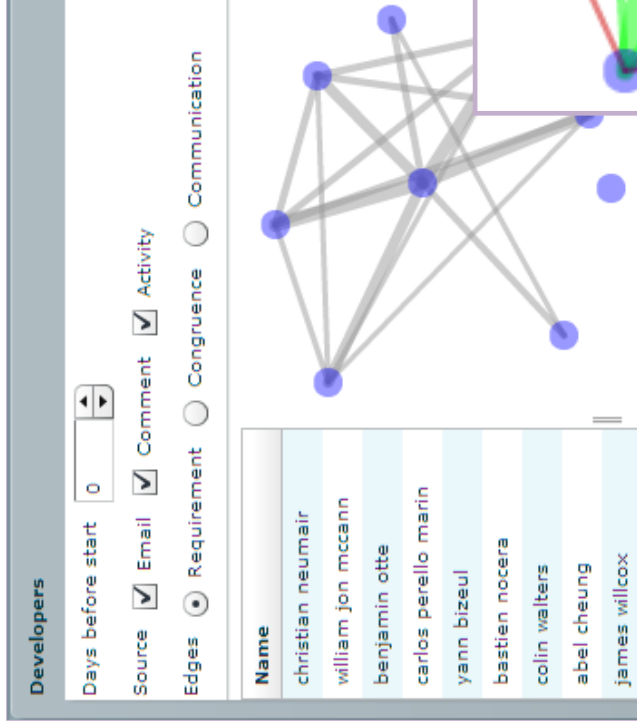
“The implicit dependency stuff, that, I think could be really useful in and of itself. So things that which end up being changed together but don't necessarily have an inheritance relationship, or compositional -- knowing that, I've changed this thing it looks like something in isolation, but in reality whenever someone changes something here, these thirty other things change because of **some ripple effect, that would be useful...”**

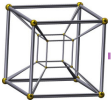


Developer Network

Coordination Requirements

Communication Pattern

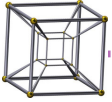




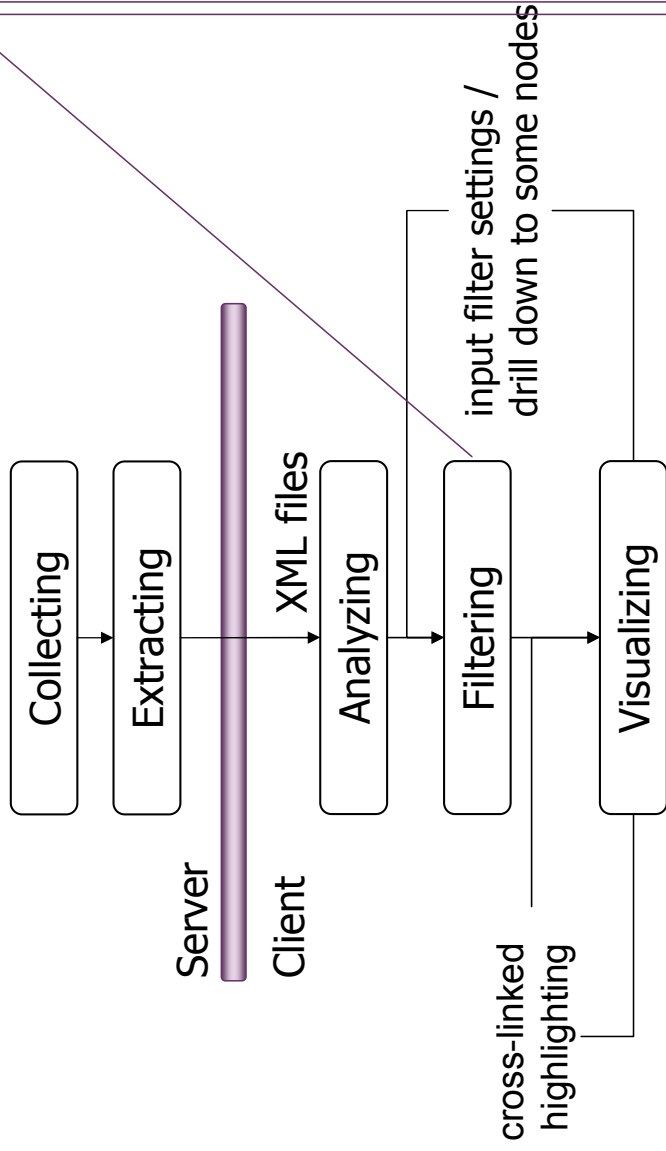
Developer Testimony



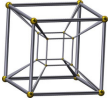
“this [developer pane] is a project manager view.
What I know is, I am this person, three people have
red flag and one person has green flag. **My
dashboard says you need to talk to
[developer] because he made these
changes...**”



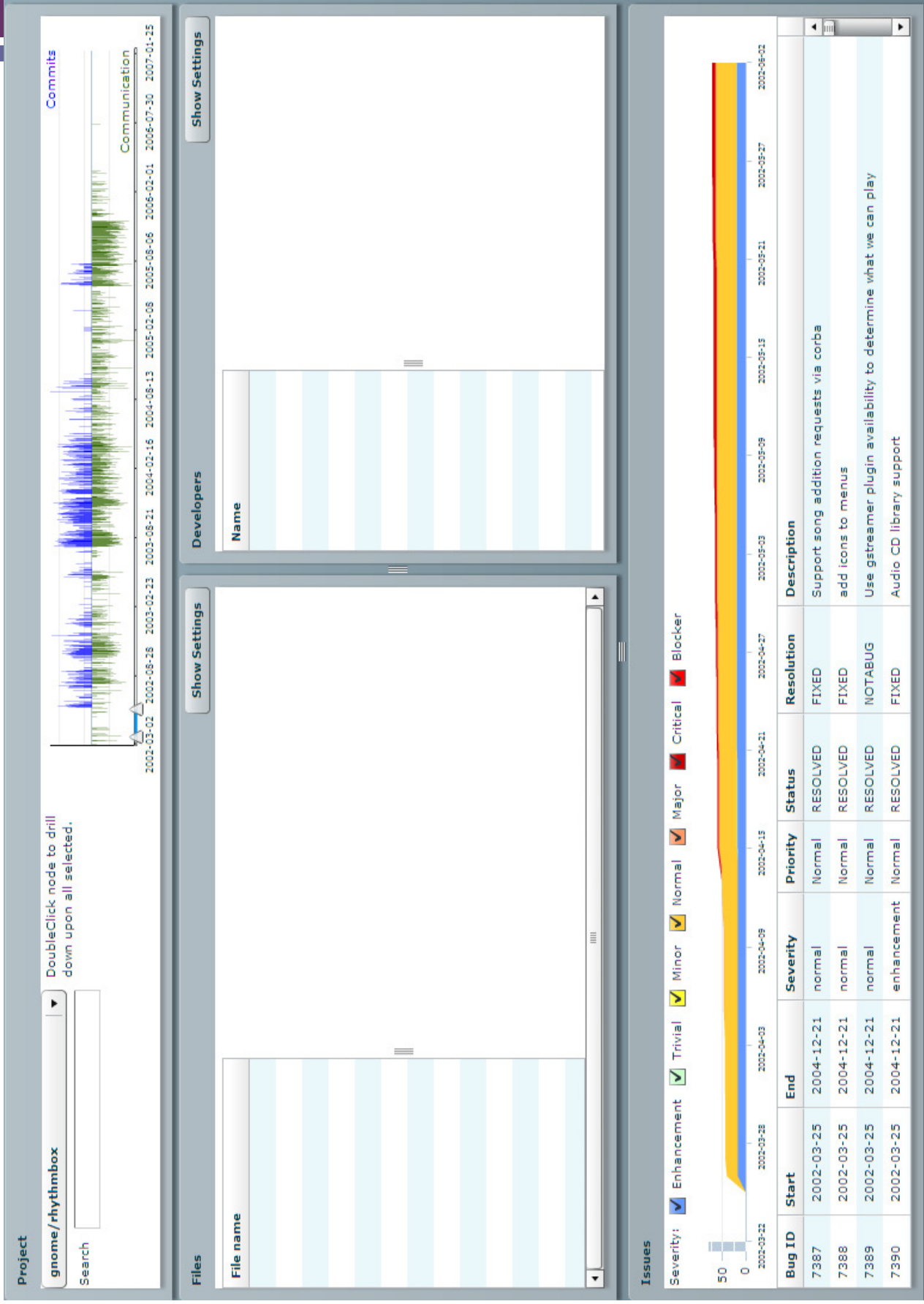
Information Flow

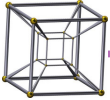


- Project activity view
- Drill down
- Thresholding
 - density of file, developer association
 - total commits in a set
 - file types
 - communication type
- Text search

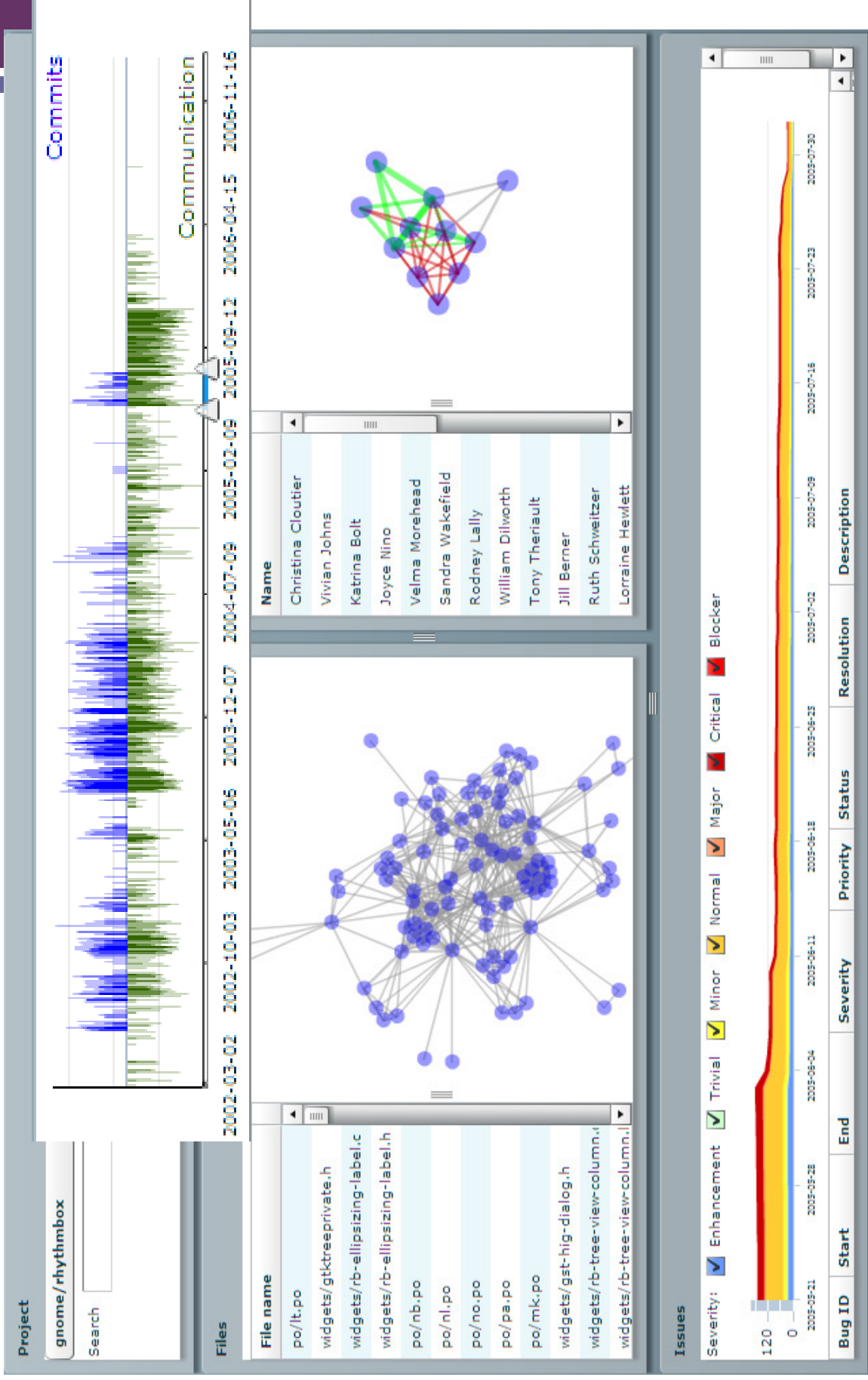


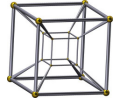
Tesseract Visualization



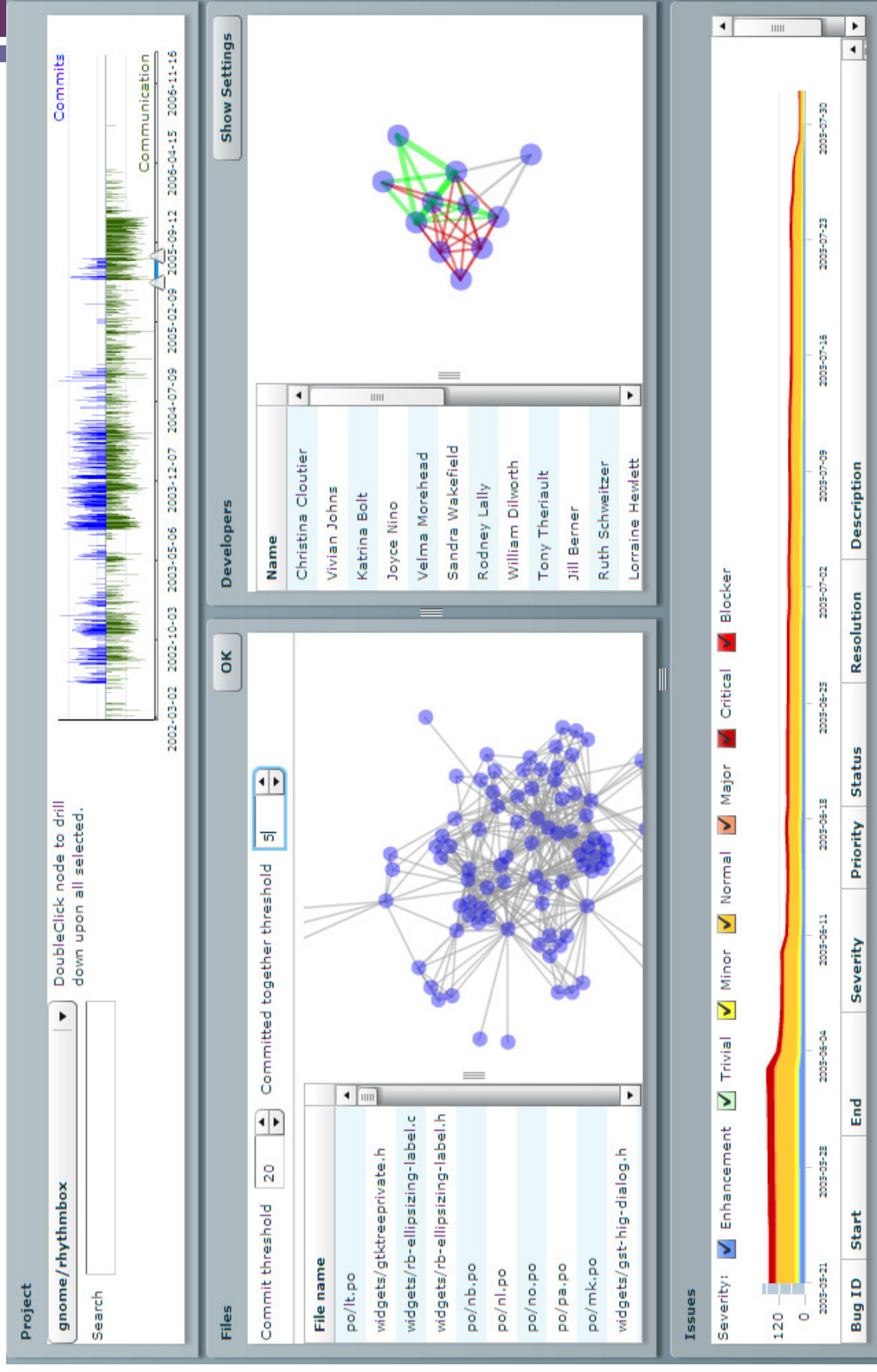


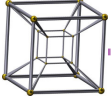
Tesseract Visualization



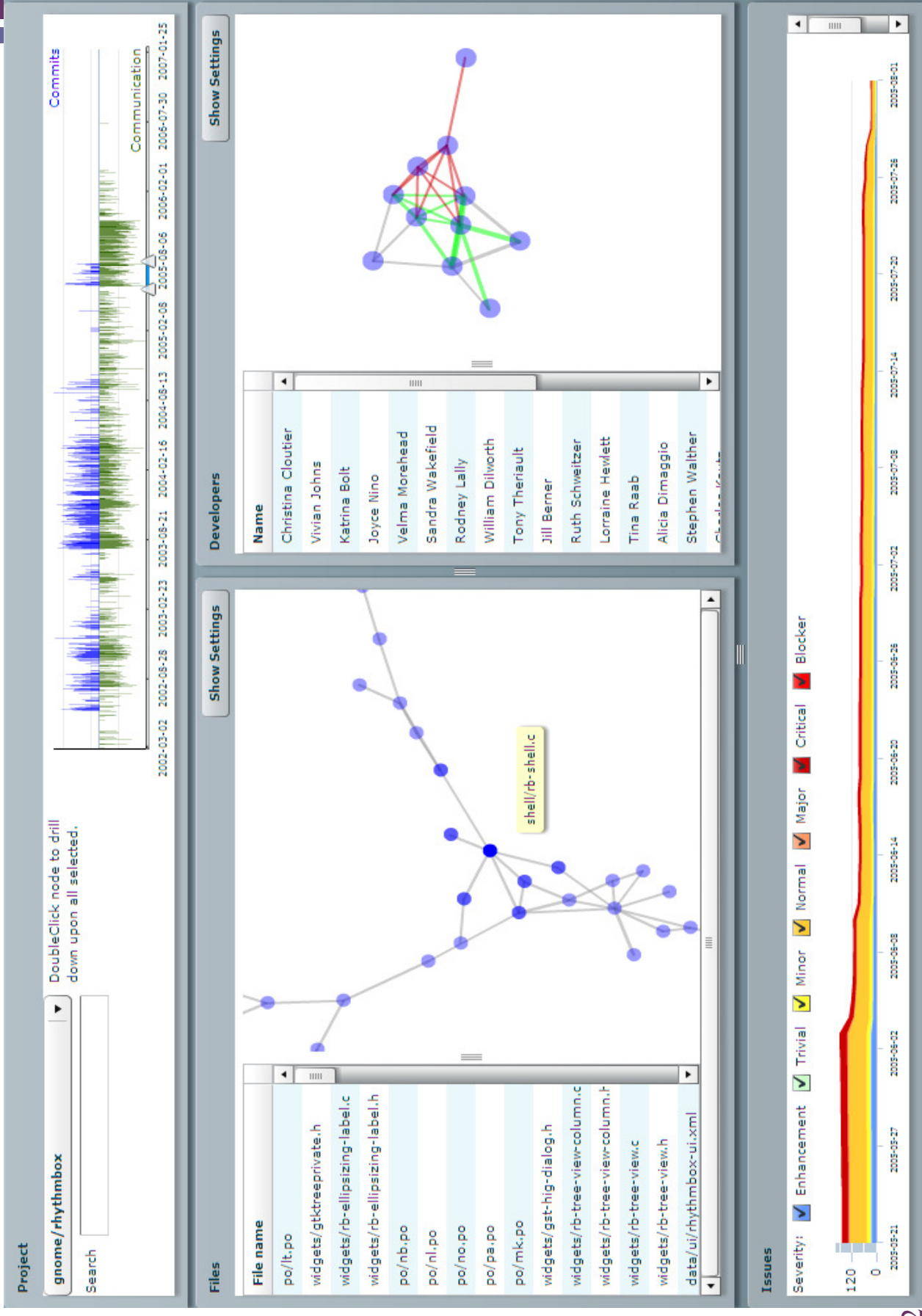


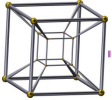
Tesseract Visualization



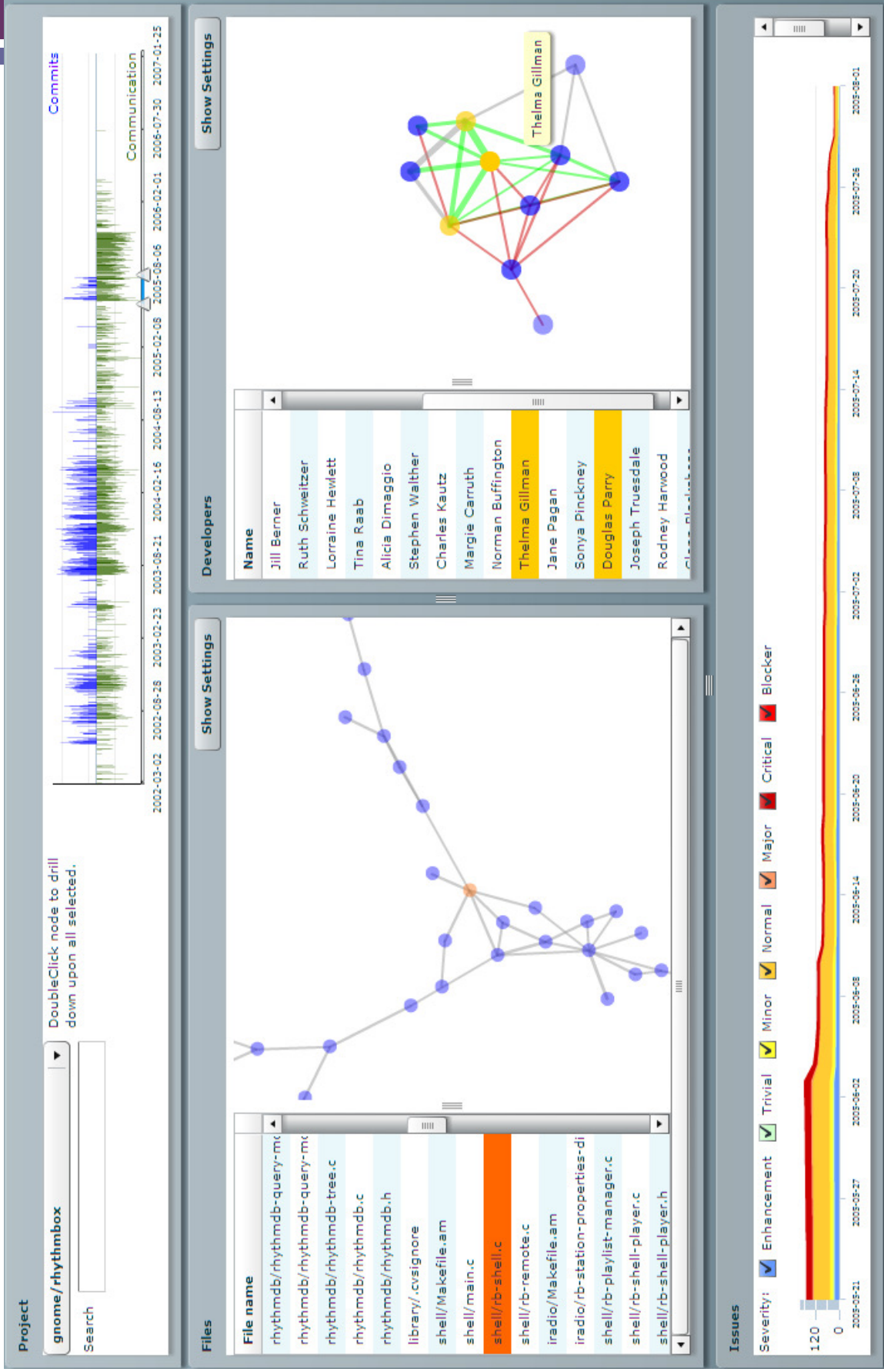


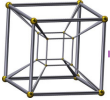
Tesseract Visualization



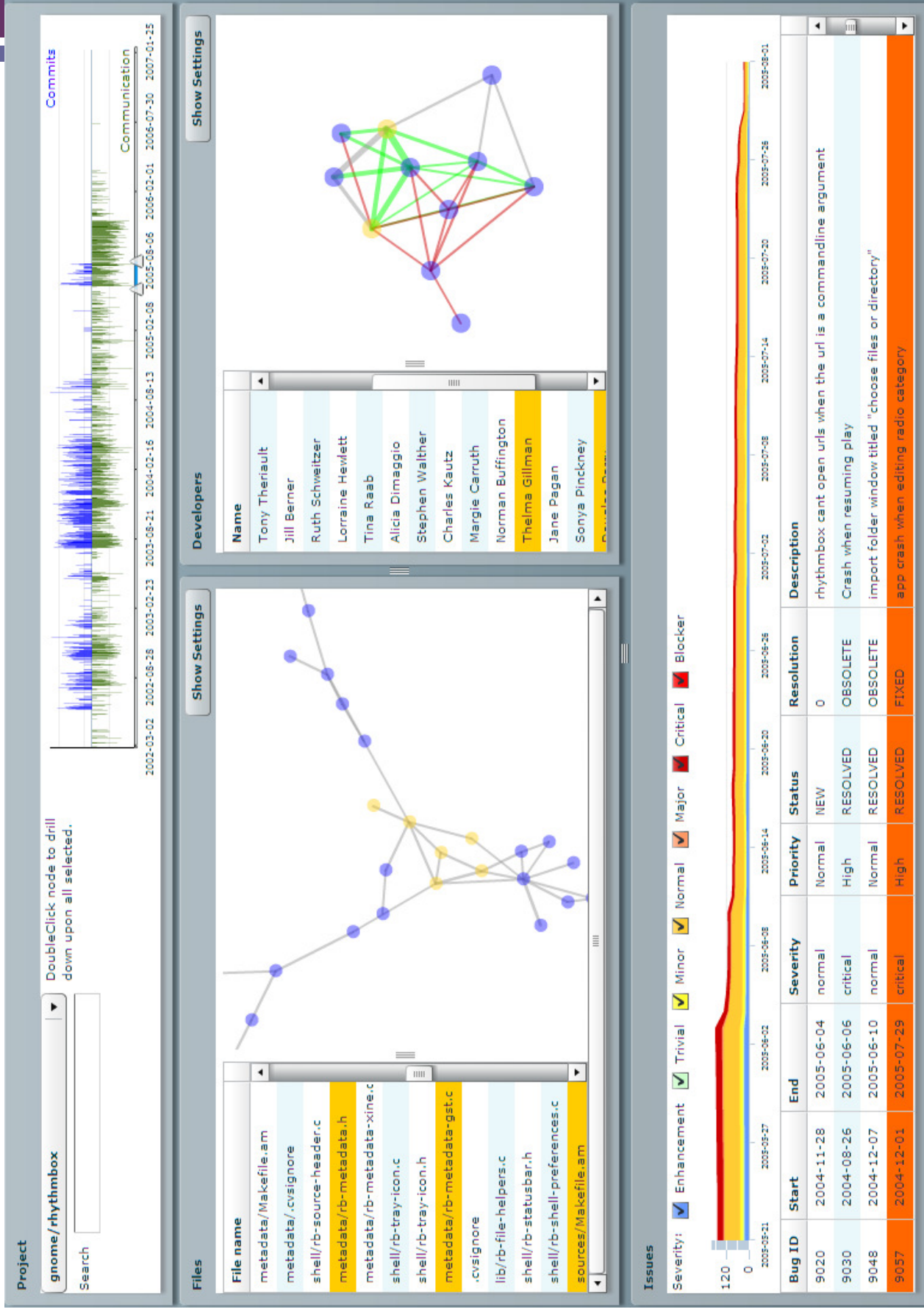


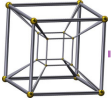
Tesseract Visualization





Tesseract Visualization





Developer Testimony



Without Tesseract

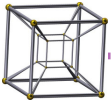
“It’s usually just talking to people about what happened, going back to the CVS and trying to see what happened with the file changes [is] **kinda fruitless.**”

With Tesseract

“...from a grunt developer standpoint, the file listing and cross reference of who has worked before – **that would be very, very nice.**”

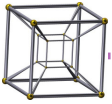
But probably not much use for experienced developers

“..This stuff most useful for the initial developers...they err... for getting to know the code base. **I have been for eight years...have this stuff in mind.**...If someone new, help in how to find stuff...”



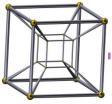
Formative Evaluation

- Instrumented with GNOME data
 - 10 years data
 - 1,000 developers, 48,000 commits
 - 200,000 bugs
- Usability studies
 - five tasks to evaluate the understanding of cross-panel referencing
 - five participants
- Open source developer feedback
 - interview to verify the need and usage scenario
 - five open source developers from different projects



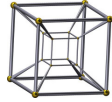
Conclusions

- Allow Interactive explorations of project relationships
 - cross-linked across different data sources
 - over time
- Treat both social and technical relationships as first order elements
- Use logical coupling for file associations
- Determine fit between communication needs and behavior
- Formative evaluations that demonstrate the need for such an approach



Future Work

- Summative user evaluation
- Visualization Enhancements
 - hierarchical grouping of nodes (e.g., packages, directories)
 - clustering algorithm (Newman grouping)
 - sticky layout of networks
 - integration with source repositories
- Analysis
 - other analyses (temporal consideration for congruence, SNA metrics)
 - difference in networks between two time slices



Questions!



Demo presentation on Friday 11am, Salon C

Authors gratefully acknowledge support from the following agencies:

- NSF IIS-0414698, IIS-0534656
- IBM Jazz Innovation and Faculty grants
- Alfred P. Sloan Foundation (2008-09)