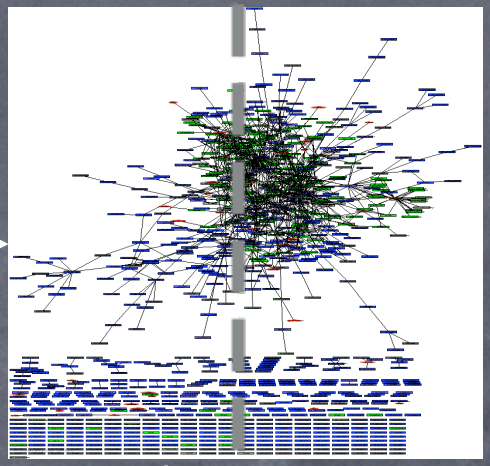
Seminar on open Source

Course Outline & Expectations.

Work Load

- Lots of reading---about 4-5 papers a week, read carefully, comment, think about questions, vigorous participation in Class discussions.
- Present one, perhaps 2 topics during the quarter
- Write a final report: survey, or original research on the topic you present in class.

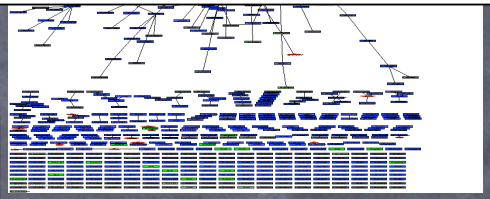
Large Software Projects:



Picture from Hendik Post, Uni-Tuebingen

Large Software Pr

Software is Large, Complex, under constant evolutionary pressure



Picture from Hendik Post, Uni-Tuebingen

Hard for both experienced programmers & new hands

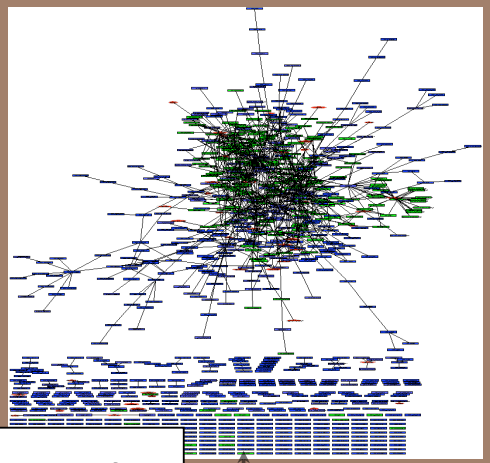
Therefore has lots of u-no-whats, that makes...

...users unhappy

Immigration:
Hard for new people to join & become productive.

ware

s:



Picture from Hendik Post, Uni-Tuebingen

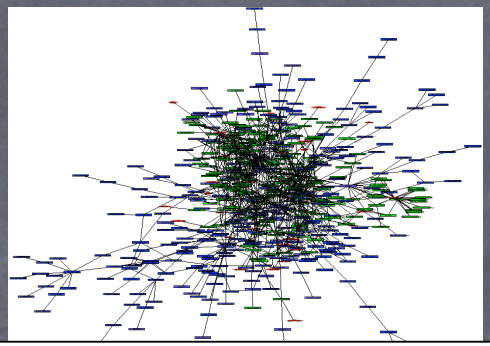
Tenure:
Experienced people become bored & territorial.

nication &
dination:
can help, or
nder!

Find
size, geography, culture....

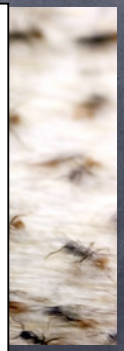
S comes in

Social: Large distributed teams, but Comprehensive interaction archives.



Technical: Big and complicated, but every version, and every change.

Bugs: Big and buggy, but every bug report and (sometimes) how



Research in OSS

Major Trends

- Goal (Science) : Understand surprisingly regular distributions in populations (power laws, etc).

Text

- Goal (Science): Understand socio-technical factors underlying open-source performance.
- Goal (Science): Explain emergent properties in software structures.

Research in OSS

Major trends:

- Goal (Engineering) : Improve software quality
- Goal (Engineering): Improve software evolvability, understandability and maintainability.
- Goal (Engineering): Improve social/personnel performance (immigration, retention, communication & co-ordination)
- Goal (Engineering): Predict likelihood of project success/failure

Research in Data Extraction

- Understanding/analyzing software changes.
- Extracting information from Emails
- Extracting information from Bug reports
- From source code: reference data, clones, etc.

Research in Metrics

- A metric is a measure of some property.
- Uses: assessment & planning
- Variety of measures for quality, complexity, cost....

Software "Physics"

Principle: Regular, large-scale properties can be precisely predicted from random, but theory-governed small-scale events (statistical physics). Examples...

- Studying properties of software graphs.
- Properties of source code sizes & evolution.
- Studying design as an optimization problem.

Code Clones

Principle: People copy and paste code (reasons...)

- Effects: code bloat, maintenance nightmares, defect replications....
- Remedies:
  - Finding clones automatically
  - Defect finding
  - Refactoring code.

Mining Generalities from specifics

General Principle: When things happen together more than they should, this means something.

- When do function calls occur together (uses..)
- When are calls added together ? (uses...)
- In what order to function calls occur? (uses...)
- How do I get from object type A to ... B? (uses..)

Recommenders

General Principle: Give programmers useful suggestions to help navigate very large information spaces.

- What bug reports/documents are similar
- Who should fix this bug ?
- Who should join this discussion?
- What other pieces of code should I look at?

Software Evolution

Theory: Software Evolution is affected by i) Design and ii) Requirements Creep and iii) People

- Which parts of the system are stable/unstable? why?
- What kinds of evolutionary changes create problems?
- Are there different evolutionary "epochs"? How and when do transitions happen?
- Does modular design affect evolution (stability analysis)

R

Social Network theories of OSS

General Principle: Graph theory provides a useful abstraction to study.

- Who are the important people? how did they get that way?
- Who interacts with whom? Who helps whom?
- Are relationships reciprocal? specifically or generally?
- How do social structures and artifact structures relate? (socio-technical congruence)? Does this relationship evolve? does it matter?

Role Transition in Open Source

General Principle: Several roles exist within OSS, and healthy, timely transitions between roles in vital to both project and participant success.

- What are the roles? What are the stated policies of transitions?
- Do the policies and the observed transitions match?
  - When do the transitions happen?
  - What factors influence these transitions?

Data Visualization

General Principle: Visualization to provide insights not readily gained using traditional analyses.

- Visualizing the social "climbing" of individuals
- Visualizing the evolution of large software.
- Social & technical activity joint visualization.
- Visualizing evolution of discussions and the behaviour of individuals

Design patterns & Evolution

General Principle: Design patterns are supposed to provide certain features of evolvability, decoupling etc. Do they?.

- How to find pattern instances.
- Outcomes of interest.
- How to do the statistics.

Micro-designs

General Principle: The structure of large graphs/networks is influenced by the "preferences" of individual participants in these graphs (examples...). These individual preferences give rise to frequent occurrence of "graphlets".

- Do graphlet frequencies describe programmer social networks, collaborations, and design networks?
- Are specific graphlets associated with interesting outcomes: e.g., stability, defect occurrence, social boundaries, etc.
- Are software "genuses" associated with graphlet frequencies.

Innovation in OSS

General Principle: Evangelicals claim that OSS Is HIGHLY INNOVATIVE. Oh Really ?

- How does population matter?
- Does the "gift culture"? .
- Does the "ego boo?"
- Does the "freedom of expression/choice"? (within and between projects)

Cognitive Studies of Program understanding

General Principle: Developers spend between 50-80% of their time reading & understanding code. Why? How? Can we improve this??

- What questions do programmers ask during reading?
- What are the cognitive mechanisms of program understanding? .
- What makes some programs hard to understand?
- What are the effects of understanding levels?

Interaction of OSS & Business

General Principle: OSS is increasingly connect to Commercial interests. What are the causes/effects?

- What are the effects of commercial involvement?
- Which projects have commercial involvement?
- Are compensated programmers and pure volunteers different in effectiveness/productivity/social status?

Conclusion

Come talk to me now or tomorrow during office hours (11-12:30) about what you want to work on. Need to tie this up right away.