

Prediction of Merge Conflicts

Olaf Leßenich

University of Passau

FOSD Meeting 2014

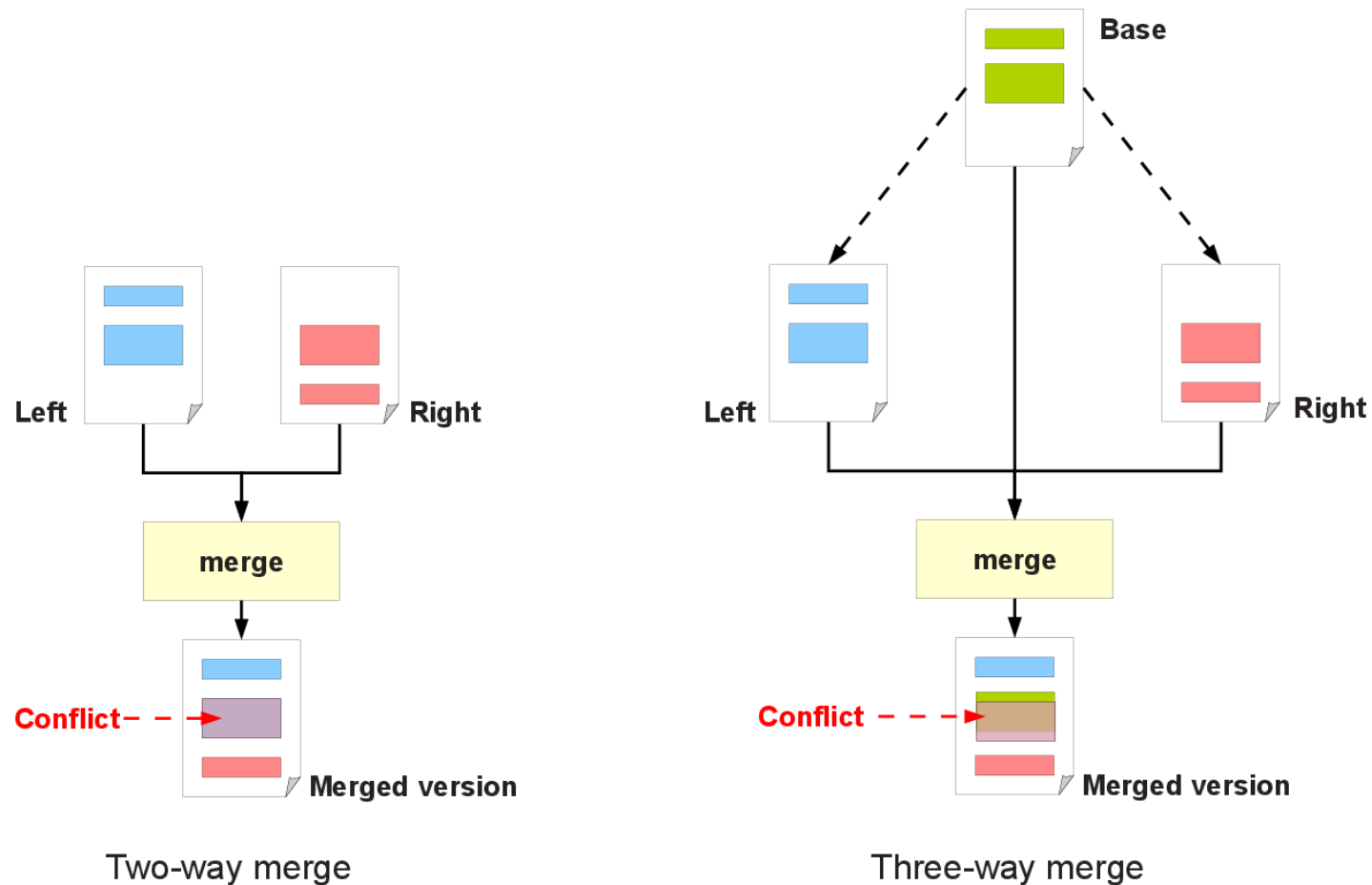
Software
Product-Line
Group



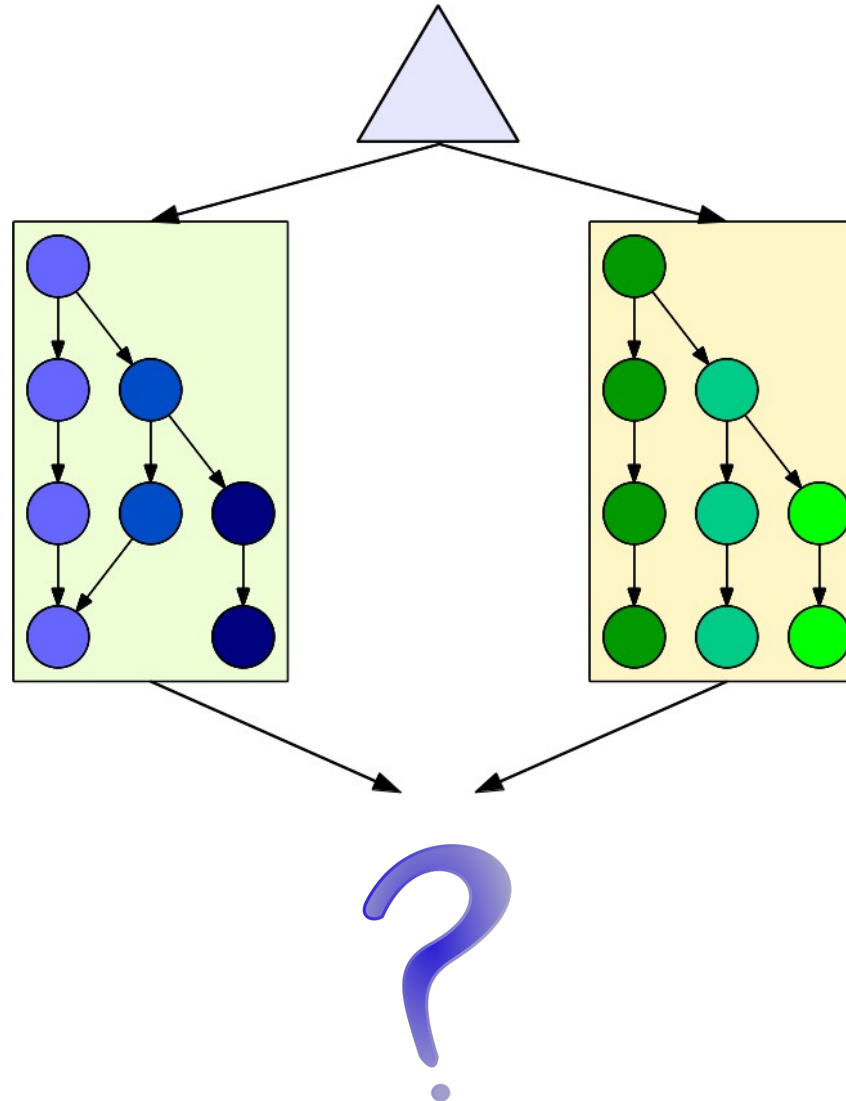
Context: Revision Control Systems



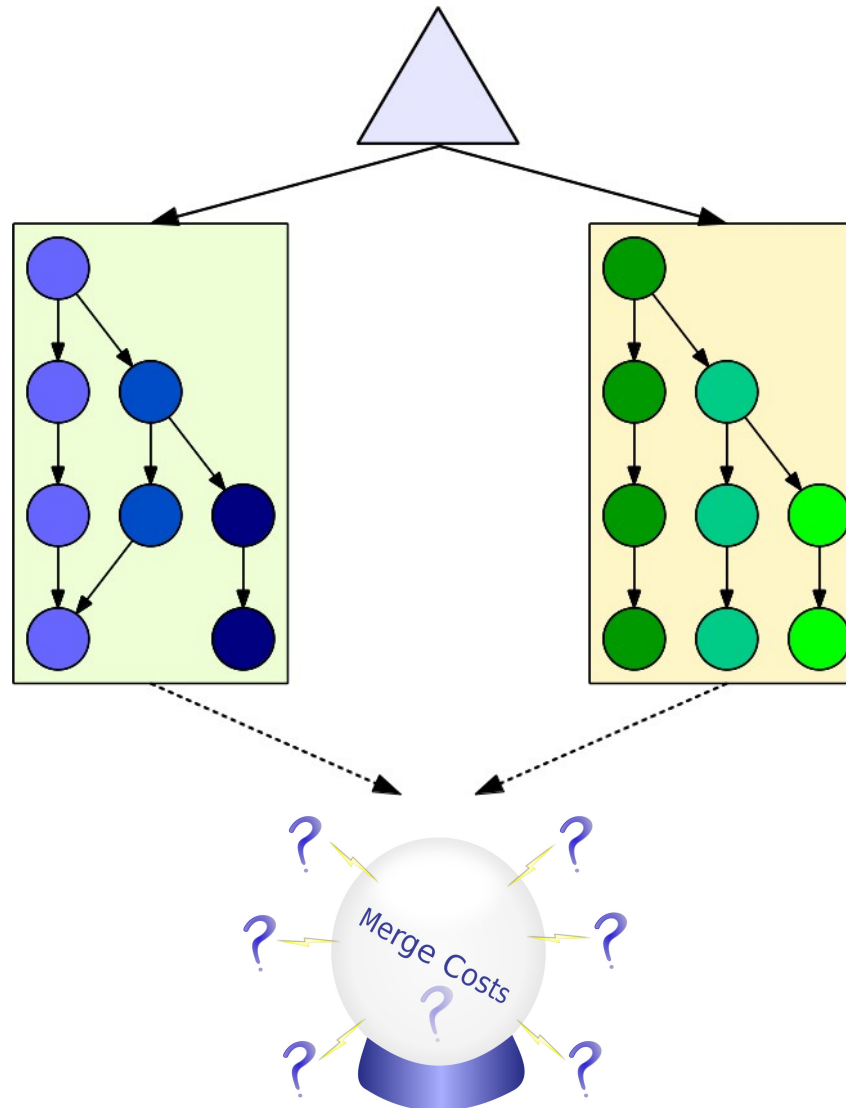
Merging in Revision Control



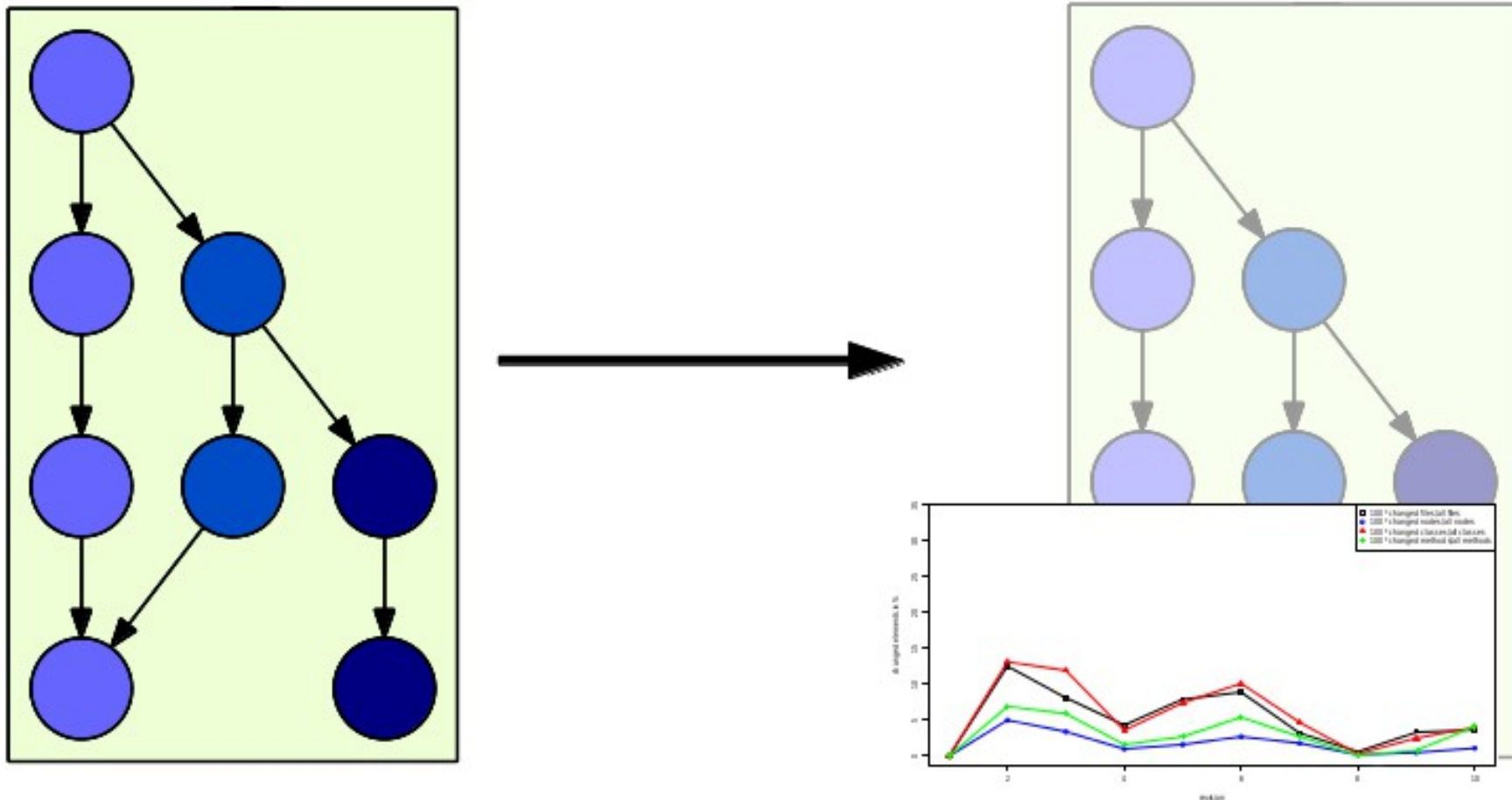
How Difficult is Merging?



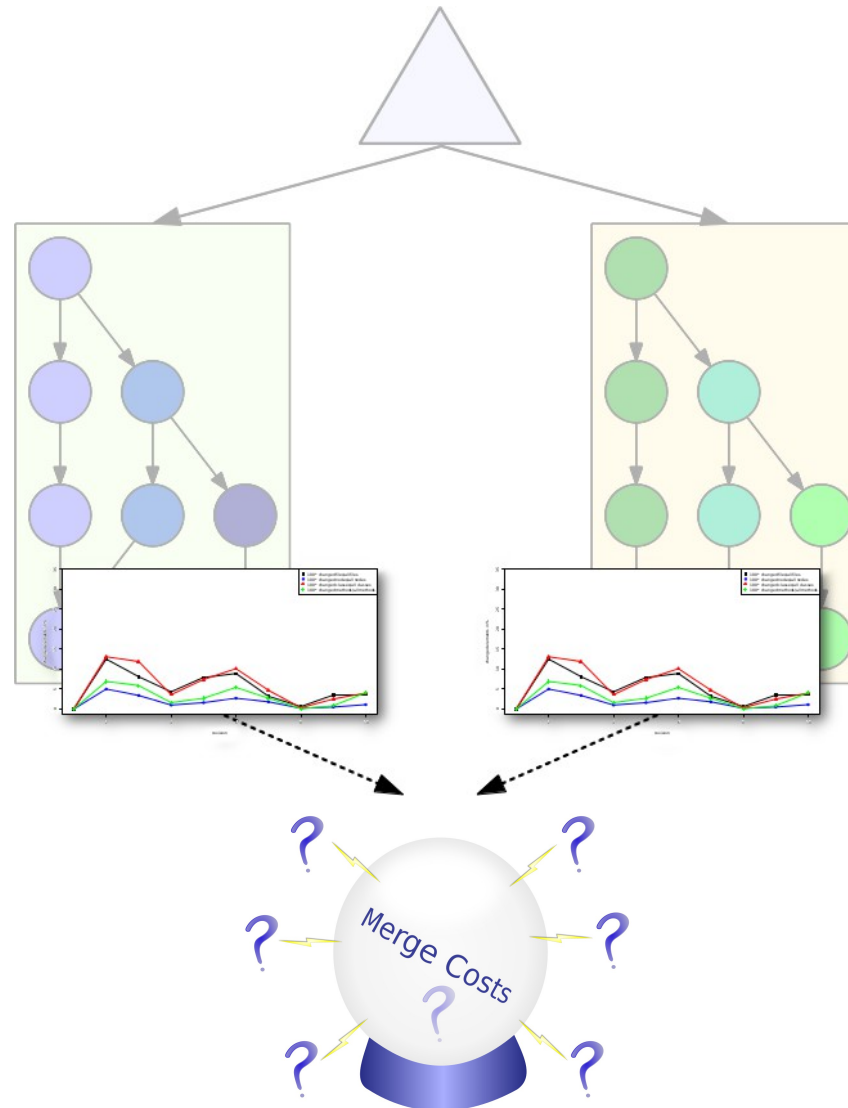
Goal: Forecast Merge Costs at any Time



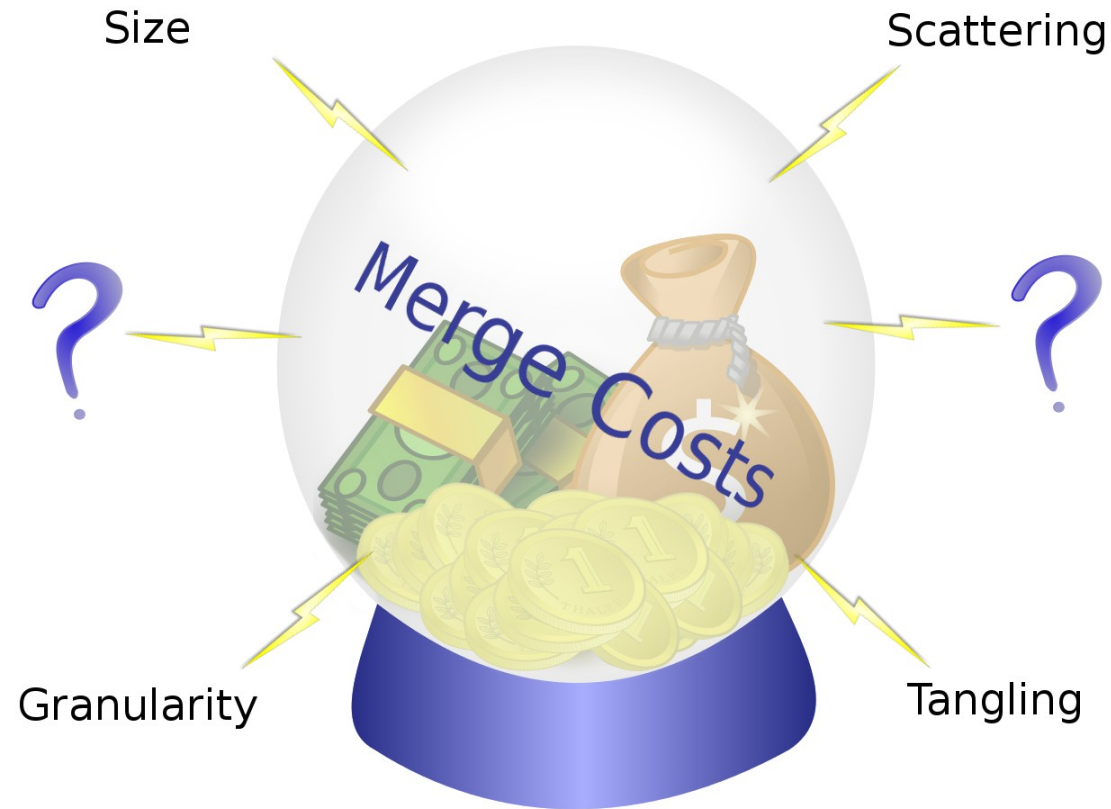
Software Evolution: Measuring Changes



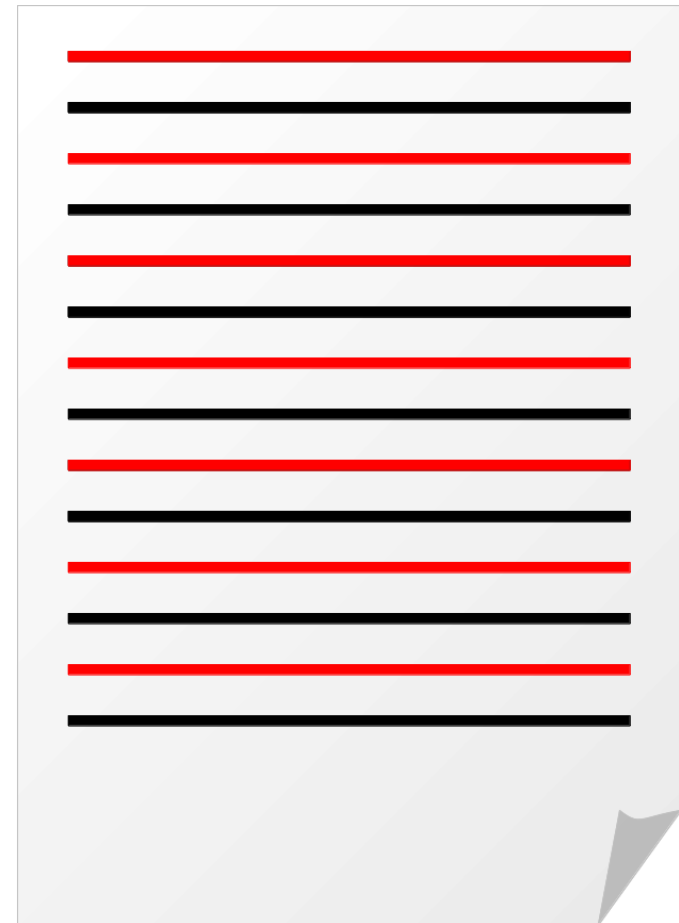
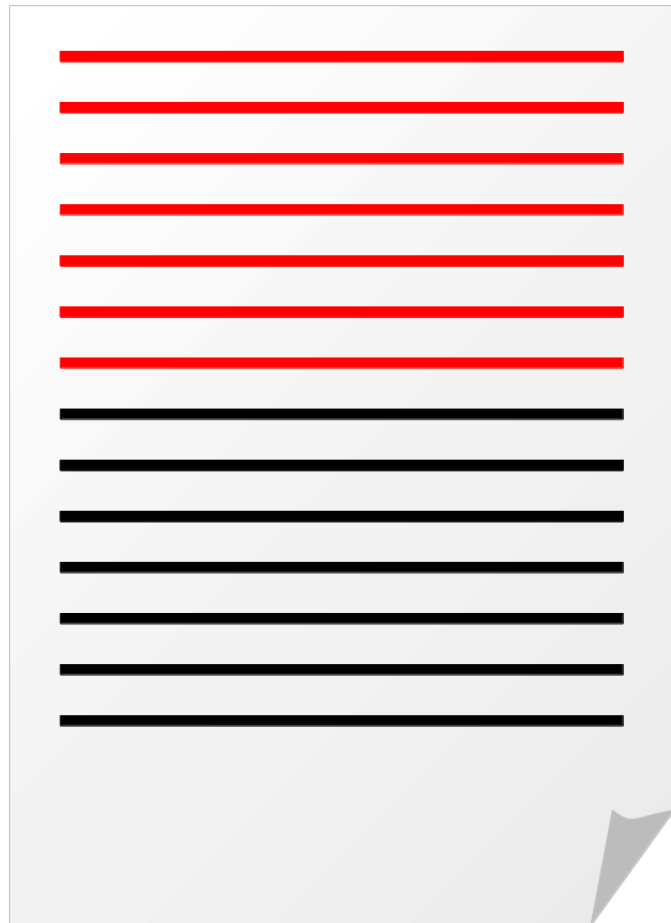
Goal: Forecast Merge Costs at any Time



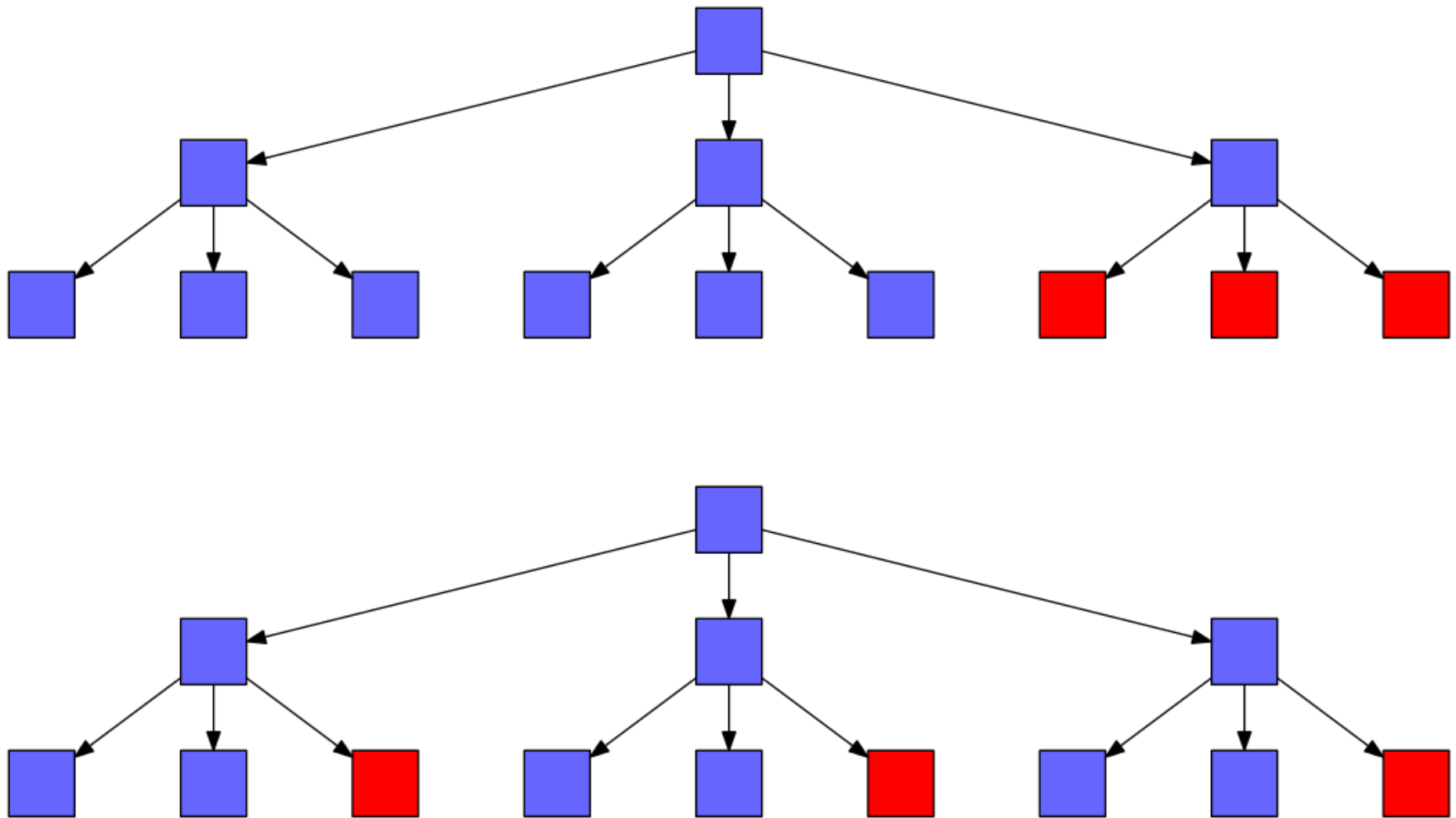
What are Indicators for Conflicts?



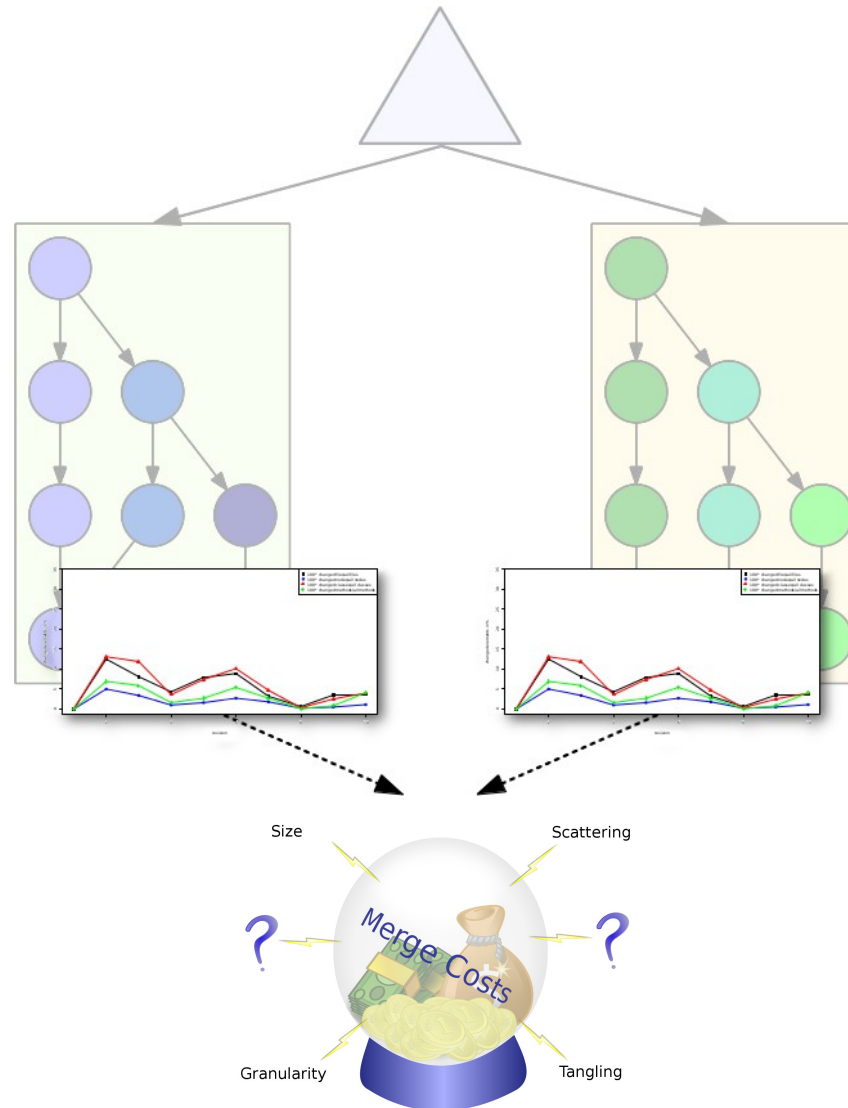
What is More Likely to Introduce Conflicts?



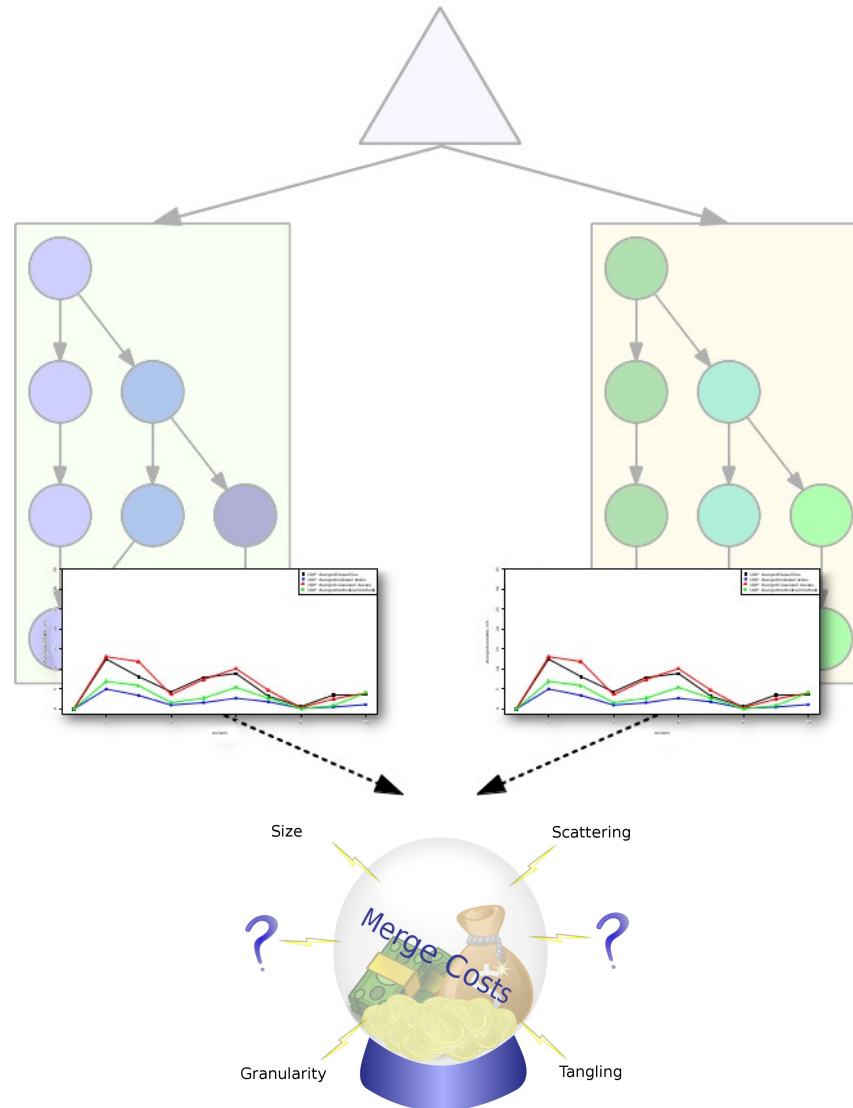
What is More Likely to Introduce Conflicts?



Goal: Forecast Merge Costs at any Time



Goal: Forecast Merge Costs at any Time



Tools

- Ötzi * FOSD 2013 Tool Name Award *
Repository analyzer
- JDime
Structured diff (and merge)
tool for Java programs

Hypotheses

- Large changes are more likely to introduce conflicts
- Scenarios with a small common ancestor lead more often to conflicts (more 2-way merges)
- Tangled changes lead more often to conflicts
- Scattered changes lead more often to conflicts
- Most conflicts occur within methods when using structured merge

Evaluation

Retrieve 15 popular repositories from Github



Extract 92 merge scenarios from history



Run merges with structured merge tool & collect data

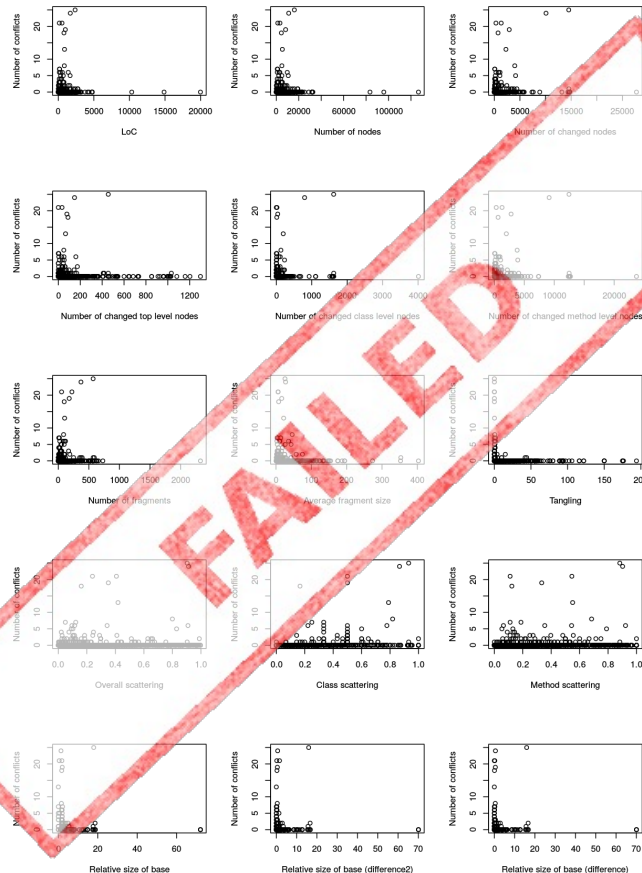


Wait some time



Check hypotheses

Preliminary Results



- Size metrics:
no correlation
- Weak common ancestor:
no correlation
- Tangling:
no correlation
- Scattering:
no correlation
- Most conflicts within methods:
YES !!!

What to Measure Instead



Summary

