Process Extraction from Development Artifacts

Abram Hindle, Michael W. Godfrey, Richard C. Holt

Software Architecture Group
David R. Cheriton School of Computer Science
University of Waterloo
Canada
http://swag.uwaterloo.ca/

{ahindle,migod,holt}@cs.uwaterloo.ca

Process Extraction

- Extract information about:
  - behavior
  - topics / focus
  - requirements
  - Software Development Life Cycle
  - repeating behaviors
- from...

Development Artifacts

- Wand of dispel demo demons
- Blessed wand of developer motivation (4:3)
- Rusty cursed plate mail named “contract deadline”
- Blessed dagger named “debug”
- Potion of Bad Smell Detection

Artifacts

- Mailing lists
- Bug tracker events
- Source Control Repositories
- Source Code
- Documentation
- Test
- Build system

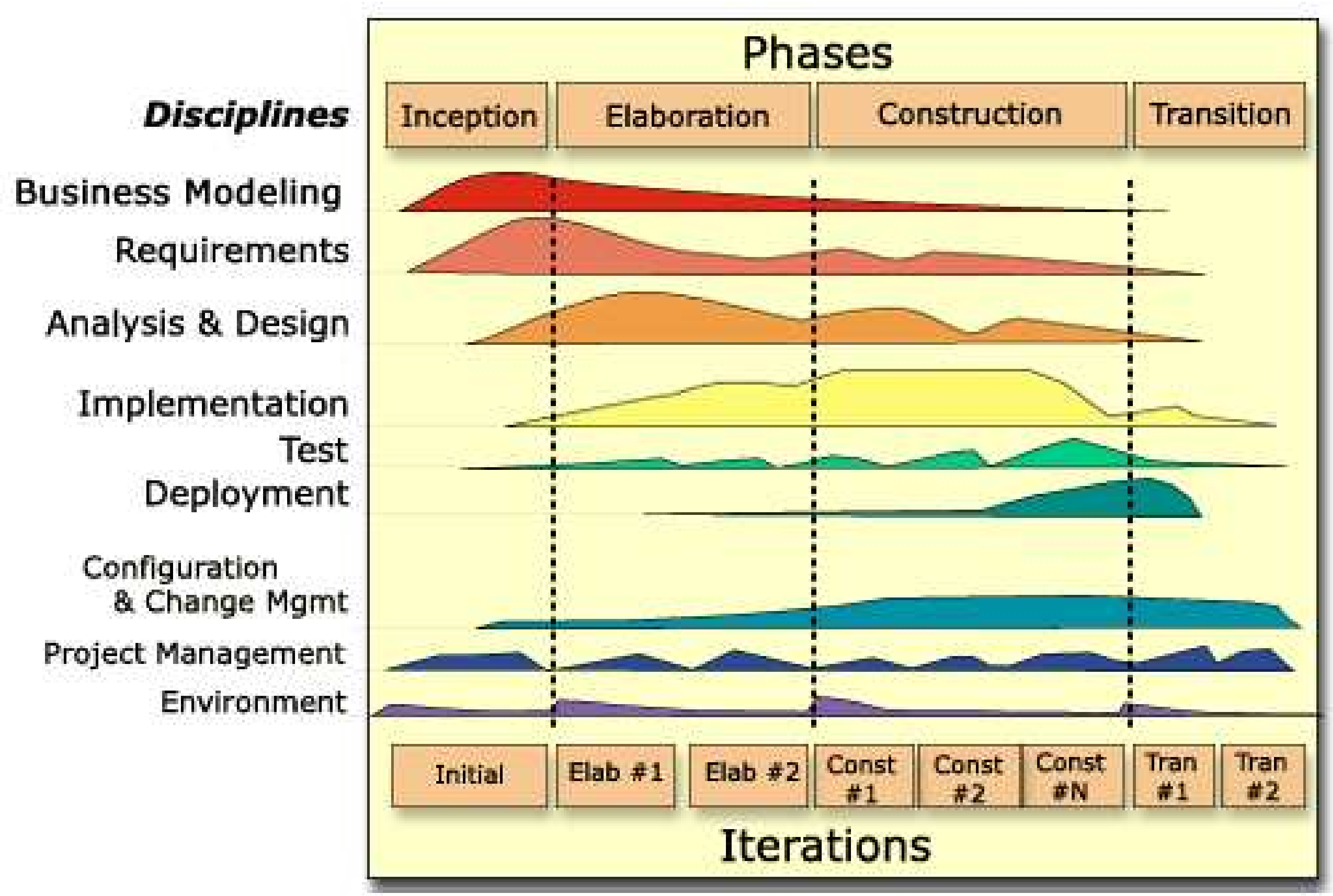


Figure 1: Remember the Rational Unified Process?

What is that diagram

- We have work-flows or areas of emphasis
- Plotted across time
  - across phases
- We could provide this to stake-holders

Self Reflection

- Not in OO terms, analysis of one’s self
- Analysis
- Can we reflect on a project’s processes, focuses, behaviors?
  - Can we do this automatically?

Dashboard versus Time-line

- Dashboard: An indicator of state
- Time-line: An indicator of past state correlated with time
- Speedometer versus Odometer

Who needs a Time-line

- Stake-holders not intimately familiar with development
  - Managers
  - New Developers
  - Developers after 2 weeks vacation
  - Acquisitions

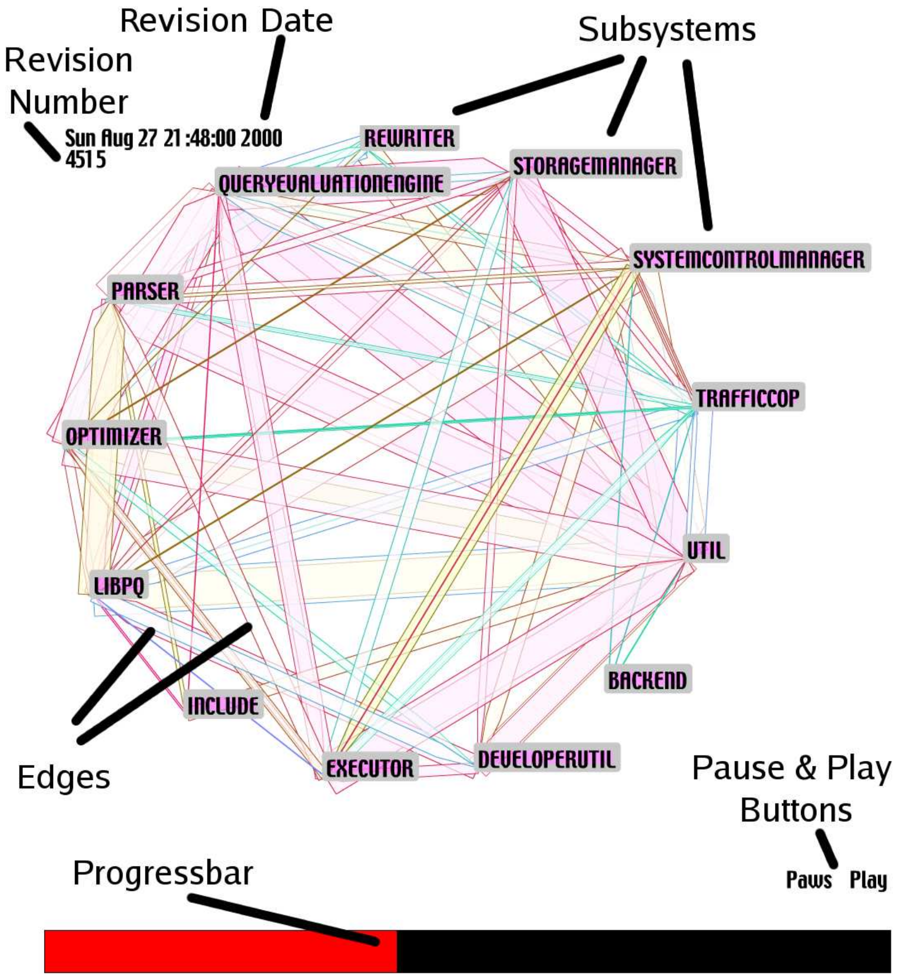
Proposal

- Automatically generate something like the RUP lumpy diagram
- Provide overviews in a time-line form but allow investigation
  - Allow zooming
  - Allow slicing

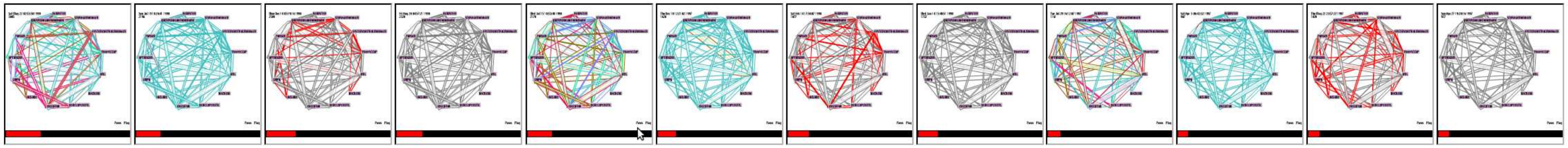
How does our previous work relate?

- Time-lines
- Metrics
- Evolution

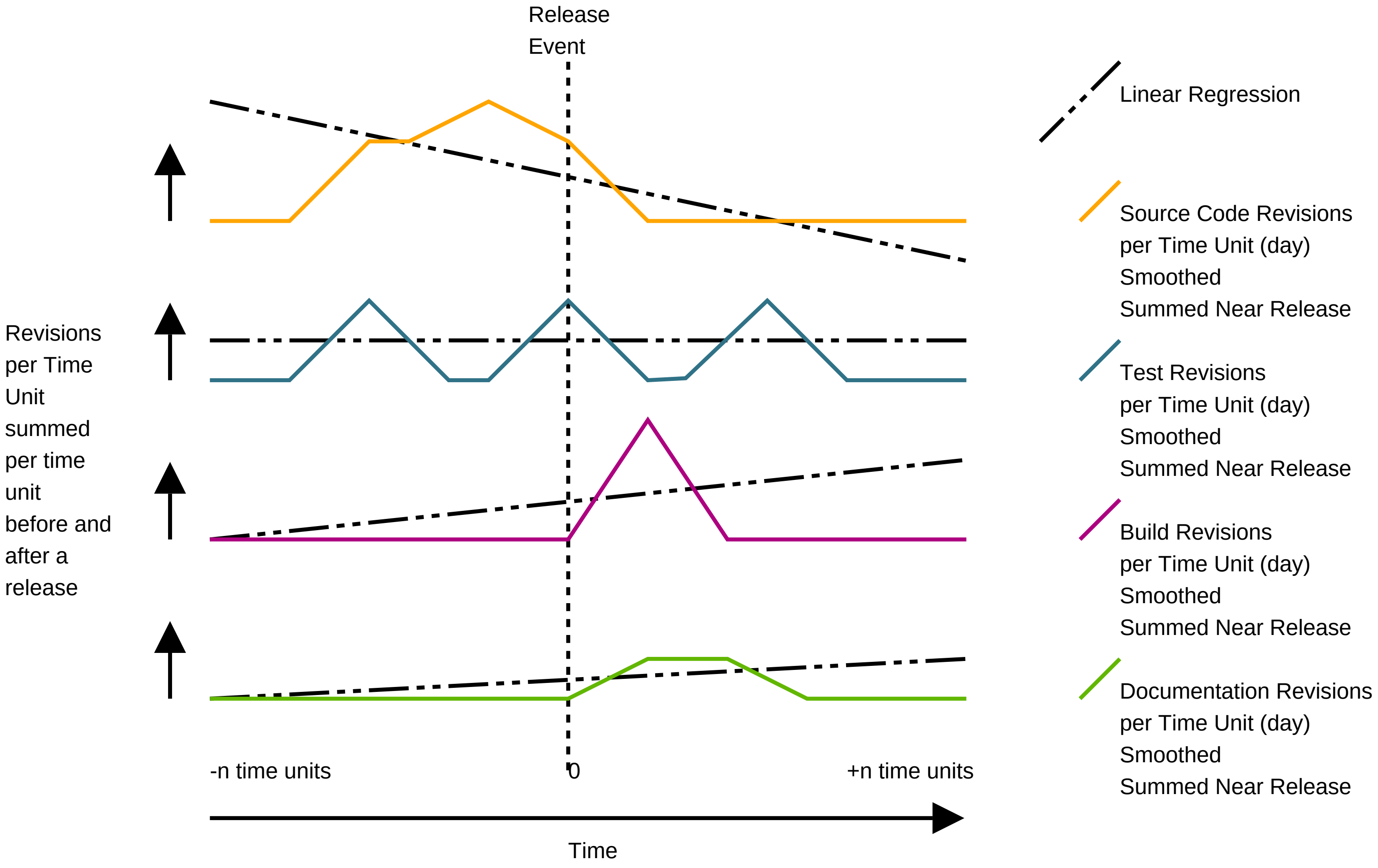
YARN


Revision Date
Revision Number
Subsystems
Sun Aug 27 21:48:00 2000
4515
REWRITER
STORAGEMANAGER
QUERYEVALUATIONENGINE
SYSTEMCONTROLMANAGER
PARSER
TRAFFICCOP
OPTIMIZER
UTIL
LIBPQ
BACKEND
INCLUDE
EXECUTOR
DEVELOPERUTIL
Edges
Pause & Play Buttons
Progressbar
Paws
Play

YARN on a Timeline



Release Patterns



Indentation Metrics

Get the Diff

```
> void square( int * arr, int n ) {
> □□□□int i = 0;
> □□□□for ( i = 0 ; i < n ; i++ ) {
> □□□□□□□□arr[ i ] *= arr[ i ];
> □□□□}
> }
```

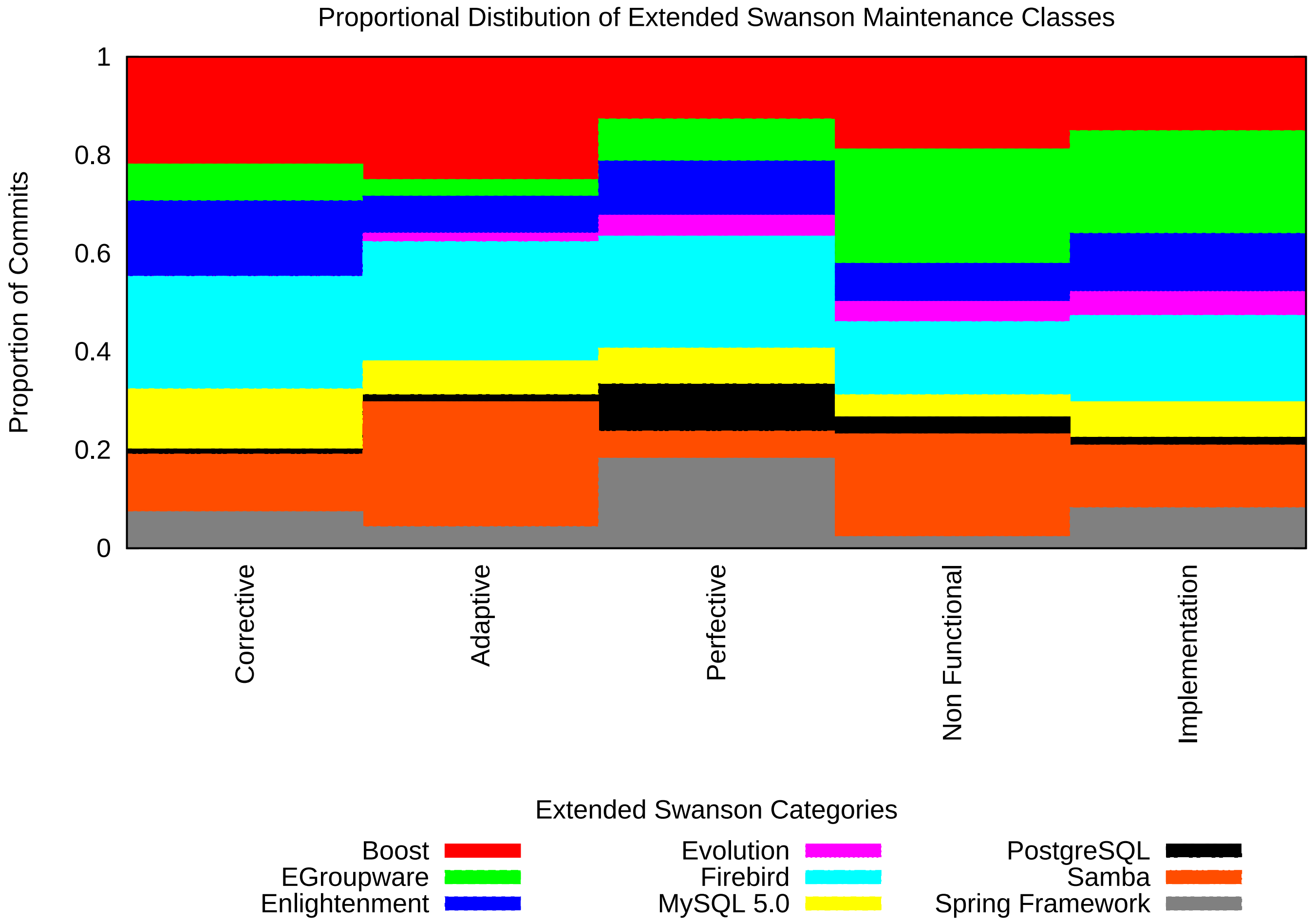
Measure the Indentation

Raw Indentation	0	4	4	8	4	0
Logical Indentation	0	1	1	2	1	0

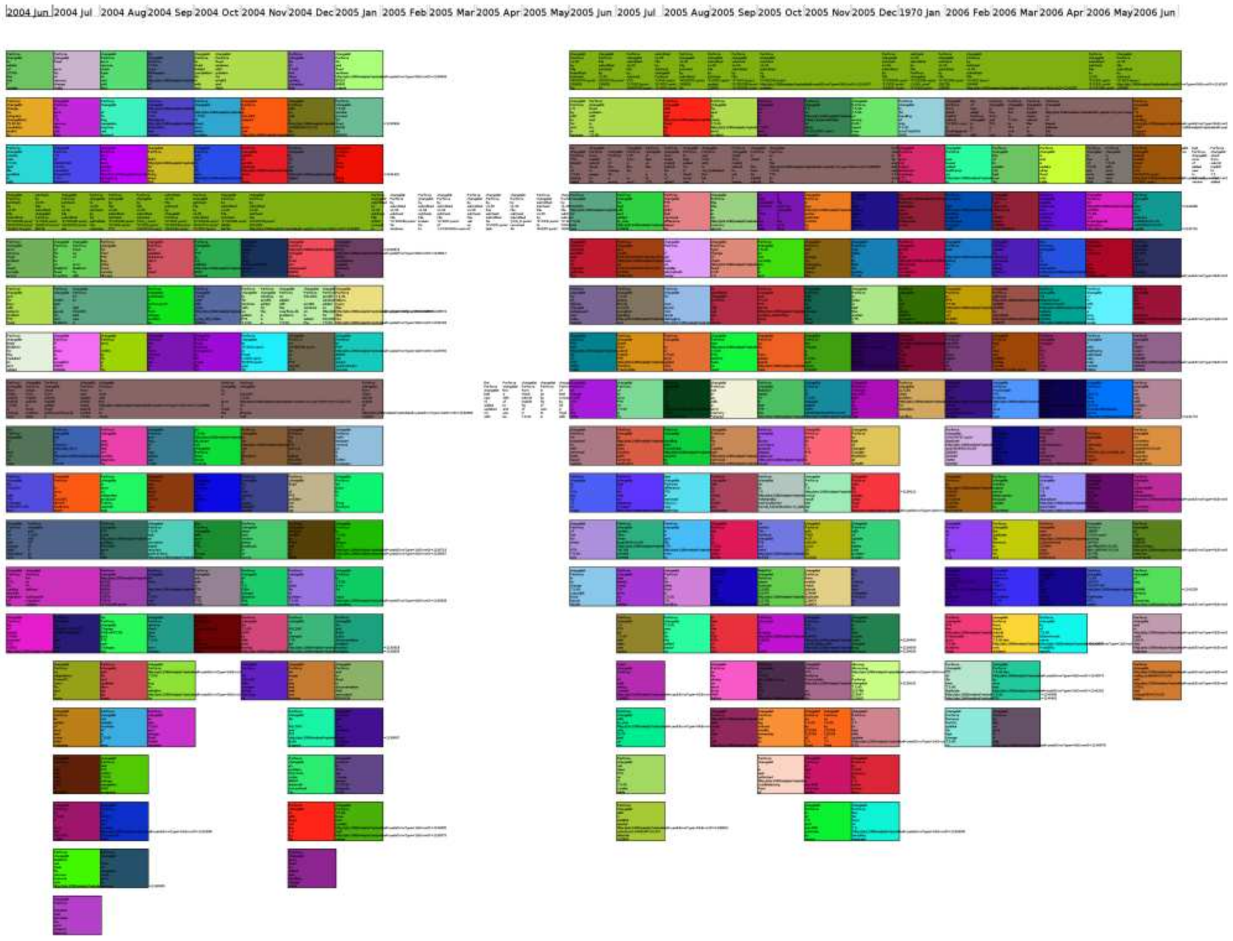
Produce Summary Statistics

Metric	Raw	Logical
LOC	6.000	6.000
AVG	3.330	0.833
MED	4.000	1.000
STD	2.750	0.687
VAR	9.070	0.567
SUM	20.000	5.000
MCC	2.000	2.000
HVOL	152.000	152.000
HDIFF	15.000	15.000
HEFFORT	2127.000	2127.000

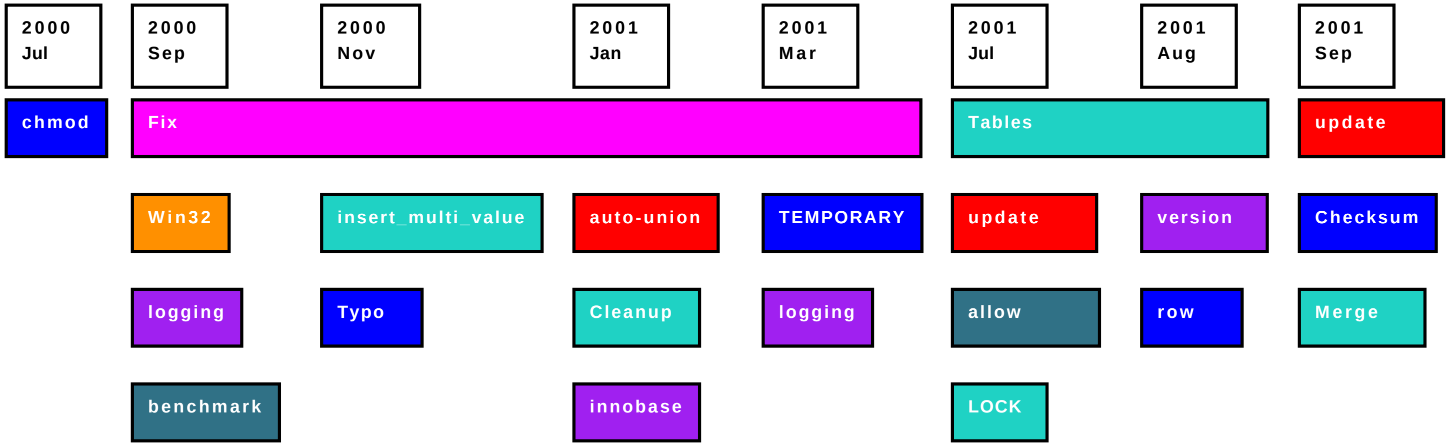
Large Changes



Topic Trends



Topic Trends Proposed



2000 Jul
chmod

2000 Sep
Fix
Win32
logging
benchmark

2000 Nov
insert_multi_value
Typo

2001 Jan
auto-union
Cleanup
innobase

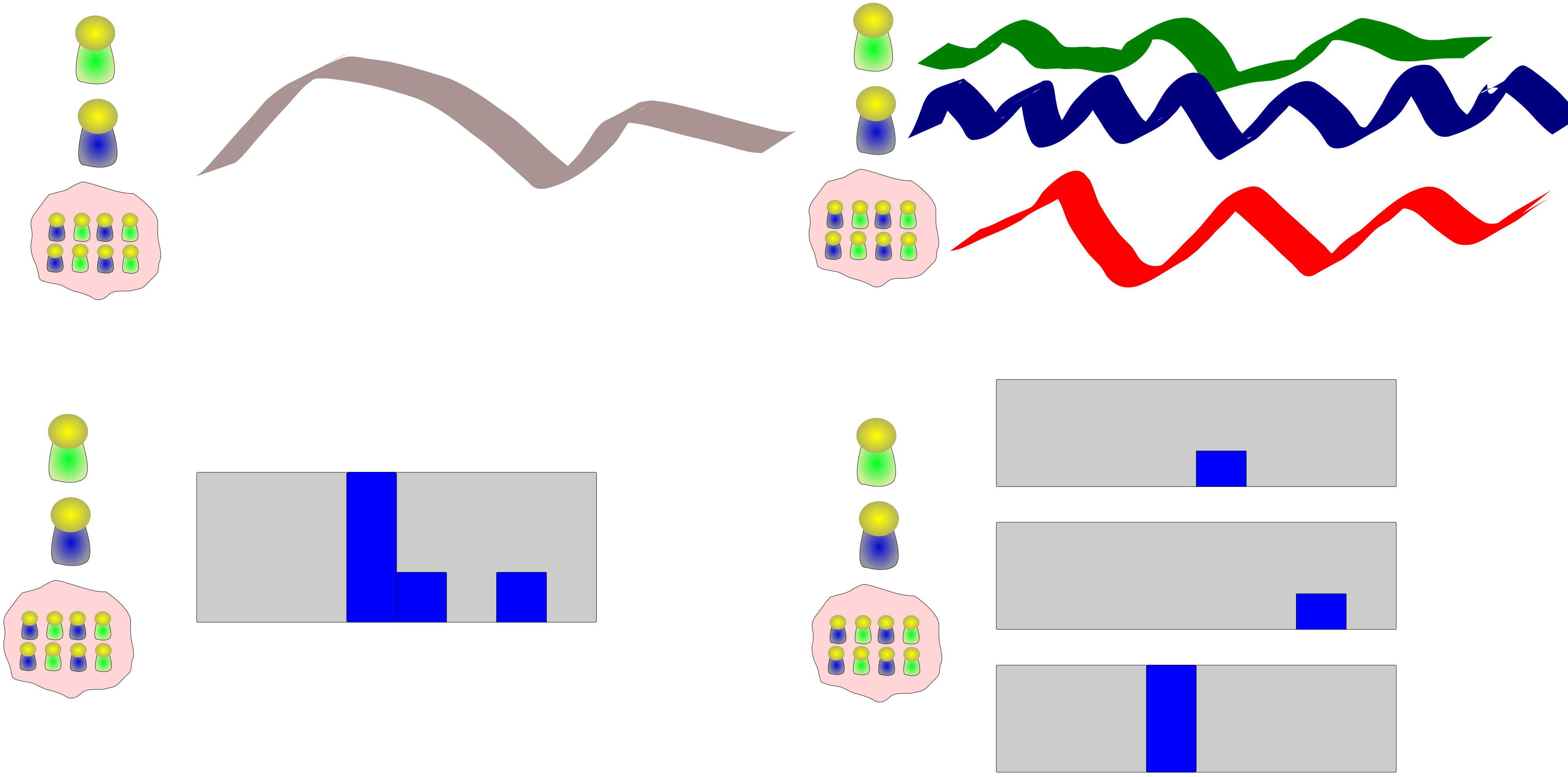
2001 Mar
TEMPORARY
logging

2001 Jul
Tables
update
allow
LOCK

2001 Aug
version
row

2001 Sep
update
Checksum
Merge

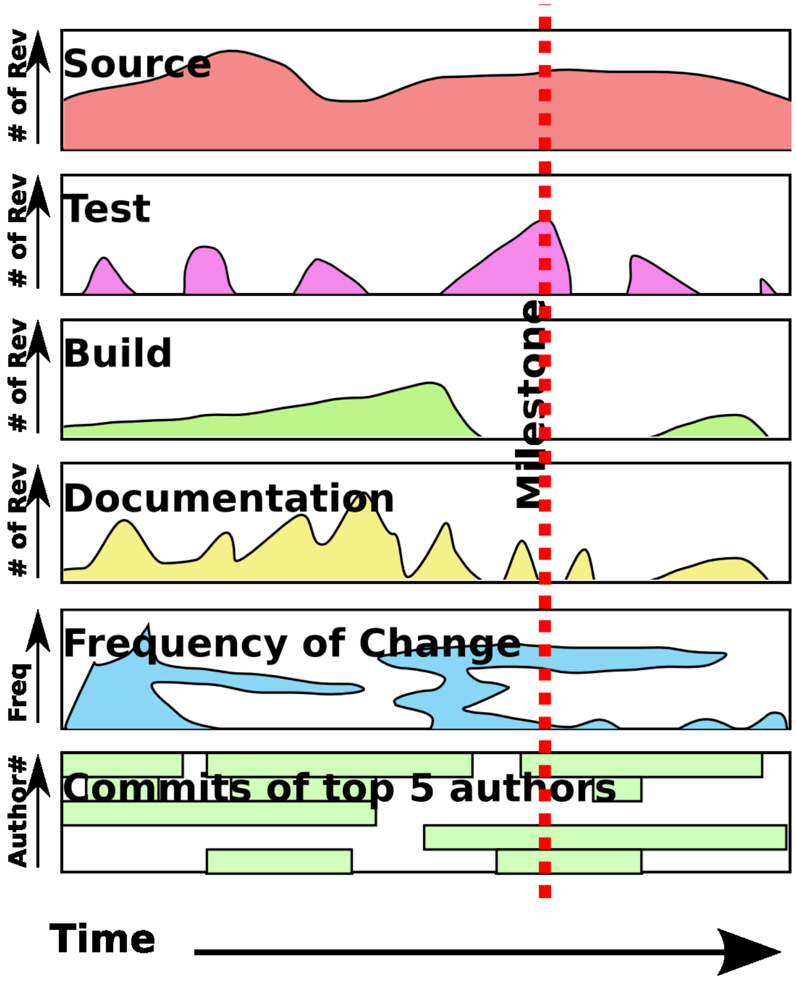
Frequency Analysis



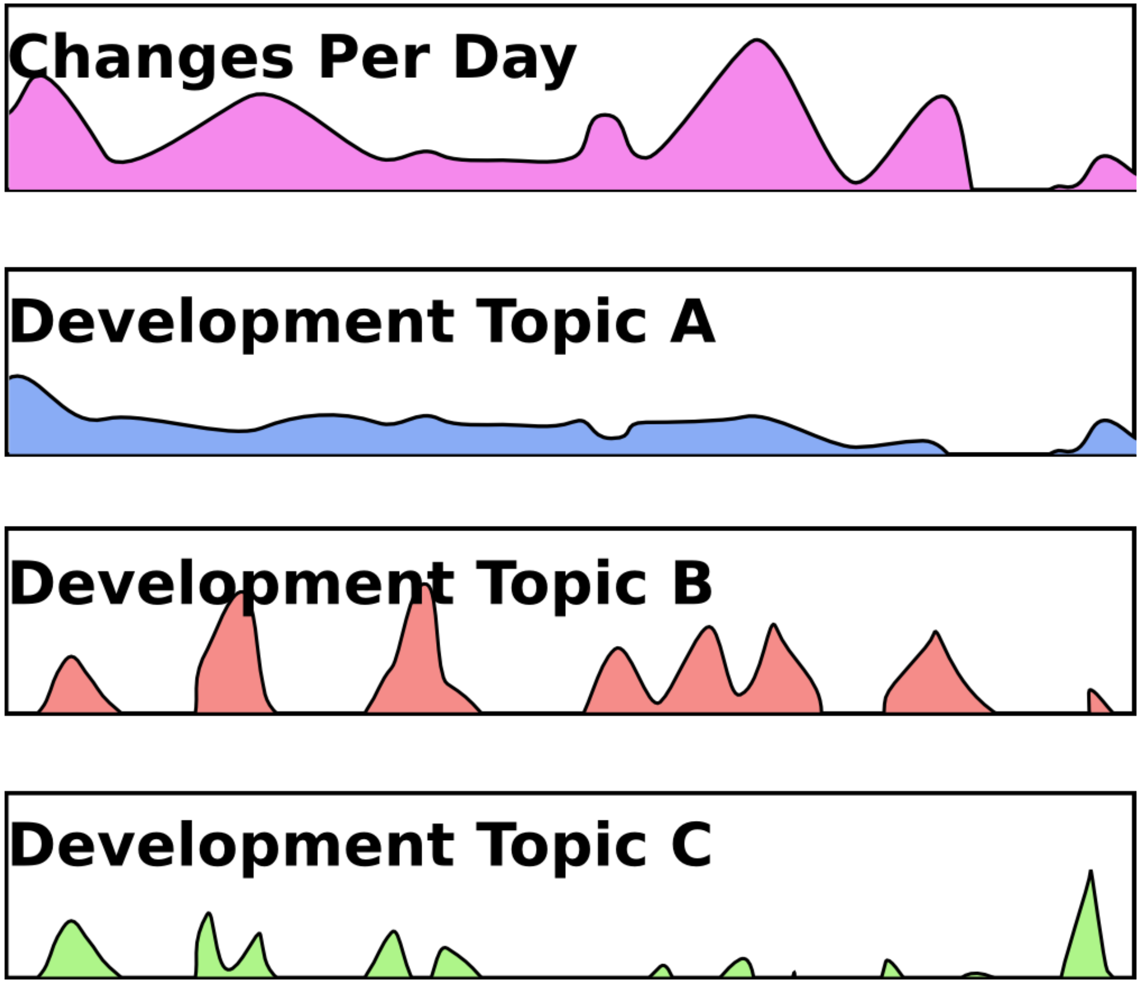
How do we combine these?

- Luckily we have local time correlations of events
- We can slice up data sources

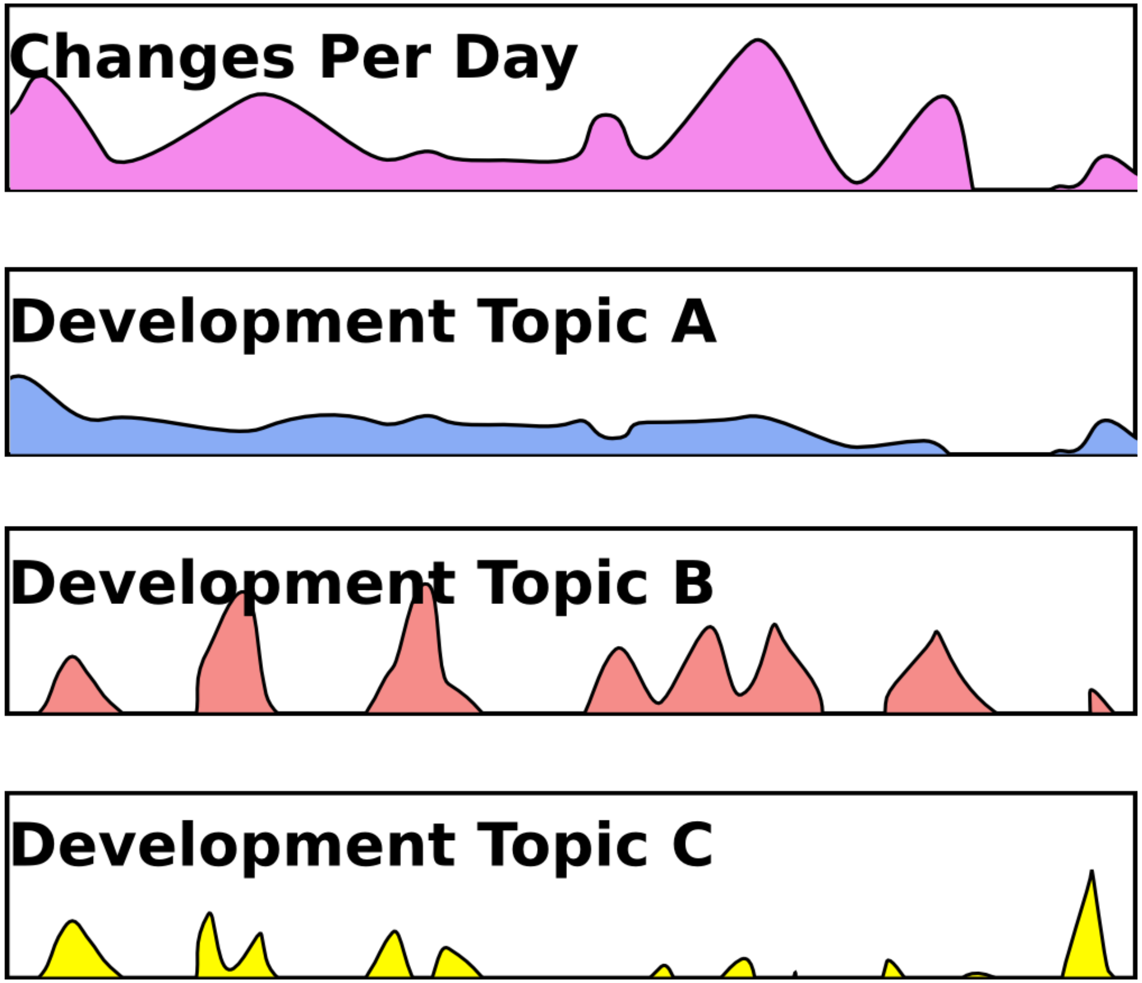
Time-line Example



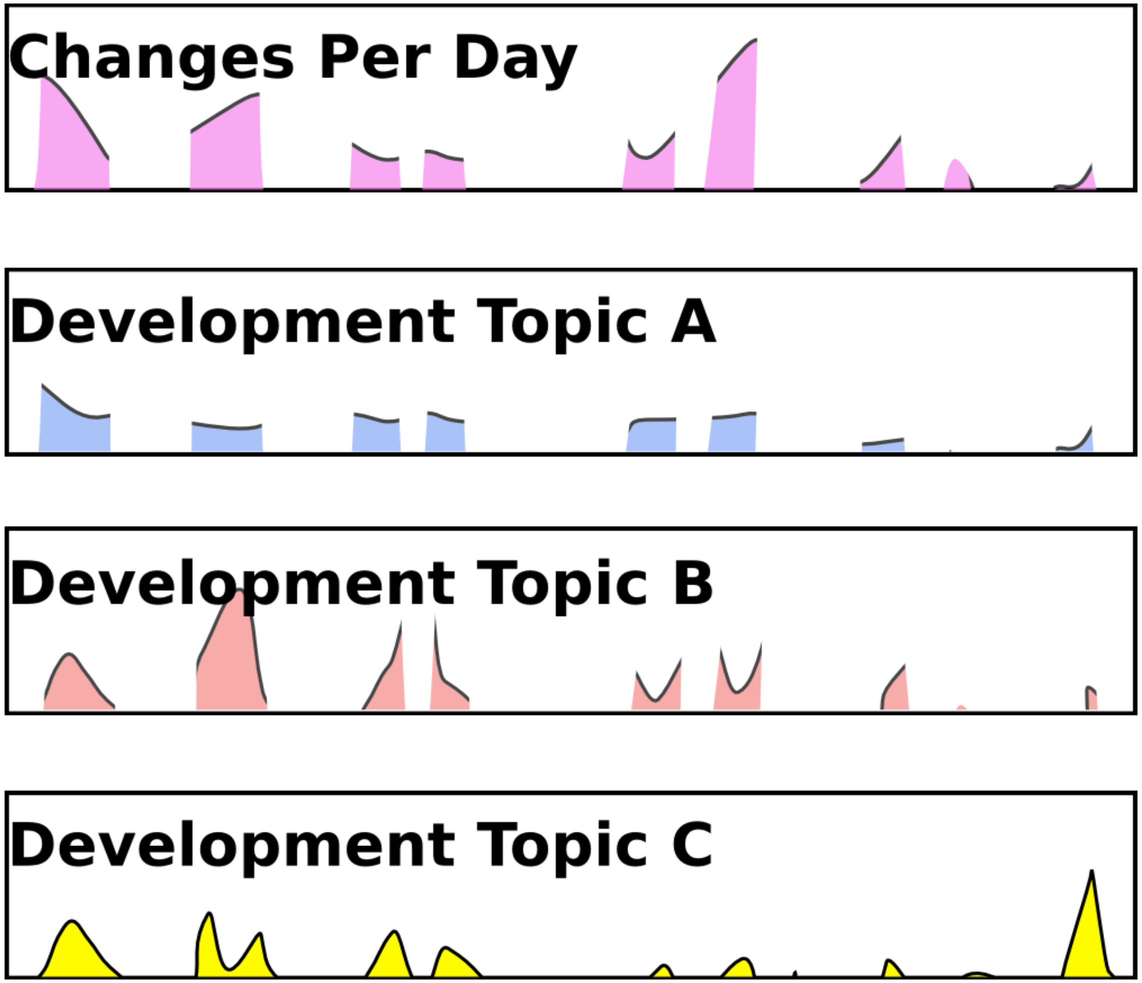
Time-line Topics



Time-line Topics Select



Time-line Topics Sliced



What are work-flows we can look for

- STBD
- Communication measurements
  - Design & Implementation
  - Bugs, fixing, debugging
  - “Nonfunctional”

What to do with the data

- Visualization
  - Dashboard / Time-line
- Analysis
  - Event analysis
  - Windowed Analysis
  - Correlation
  - Tagging

Further Analysis

- Can we spot patterns of phases
  - Identify phases
  - Find repeating sequences of phases?

Conclusions

- Aim to visualize and analyze data about a project with respect to entities and time
- Produce a tool to act as dashboard but also a time-line of the topics and behaviors of a project
- Produce a methodology and tool communicate project activities to stake-holders
- Tool should help automate parts of project retrospectives