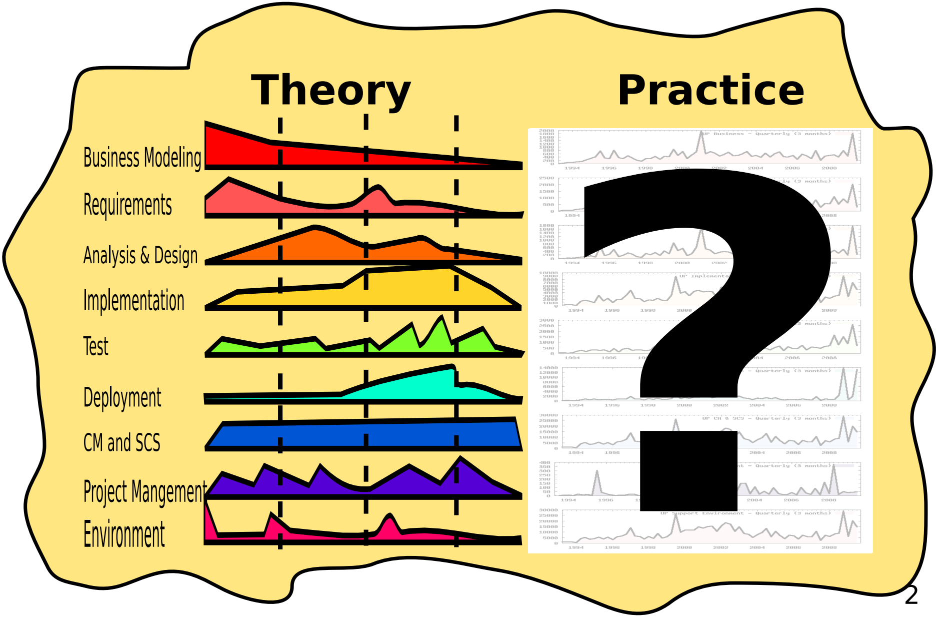
Evidence-based Software Process Recovery

Abram Hindle

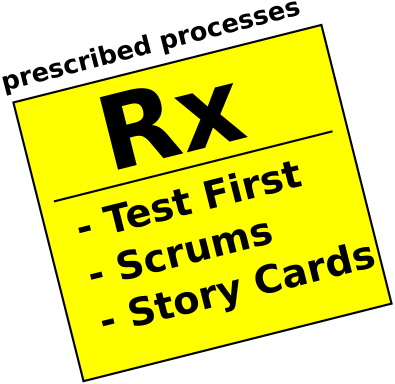
Software Architecture Group
David R. Cheriton School of Computer Science
University of Waterloo
Canada
http://swag.uwaterloo.ca/˜ahindle/

ahindle@cs.uwaterloo.ca

What are we going to do?

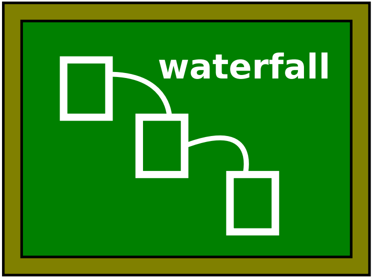


Process

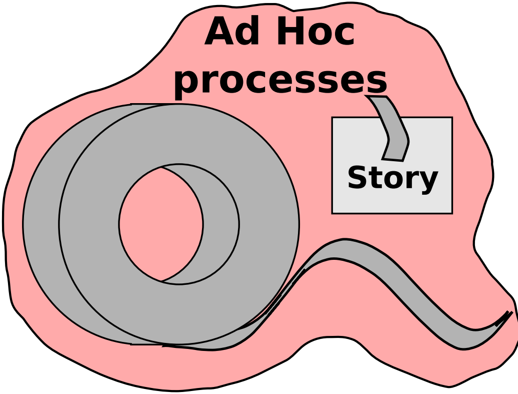

prescribed processes

Rx

- Test First
- Scrums
- Story Cards

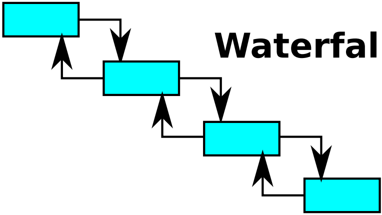


Formal Processes


Ad Hoc processes

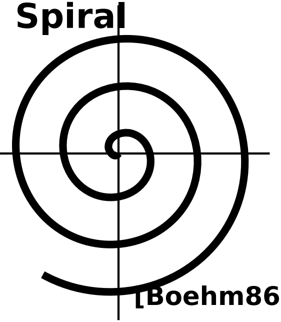
Software Development Processes

Waterfall



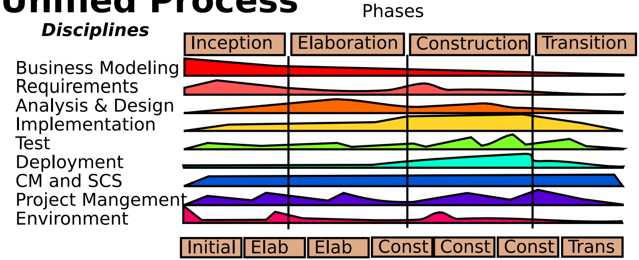
[Royce87]

Spiral



[Boehm86]

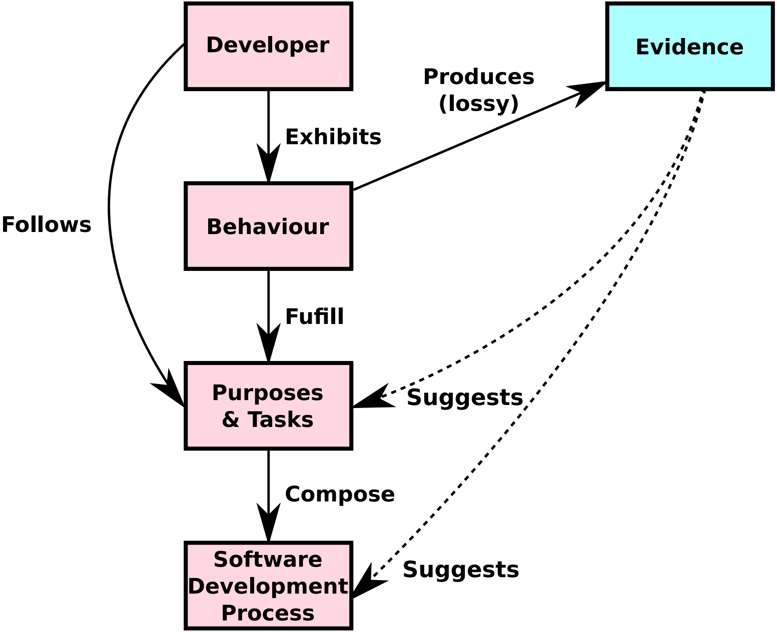
Unified Process



[Jacobson99]

*** CMM**

*** SDLC**



Developer

Exhibits

Behaviour

Produces (lossy)

Evidence

Follows

Fufill

Purposes & Tasks

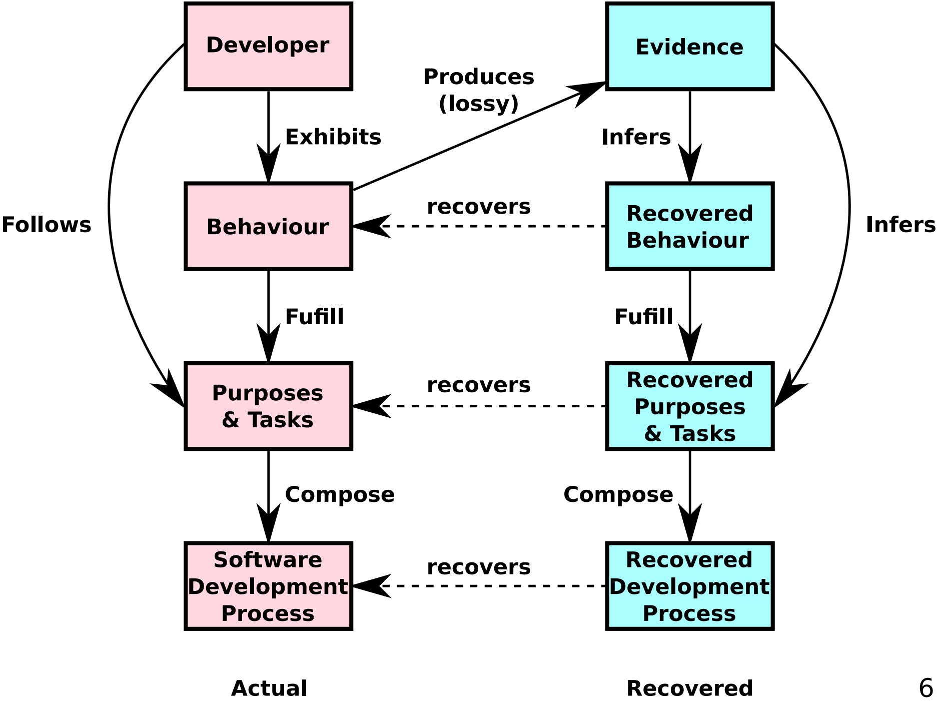
Suggests

Compose

Software Development Process

Suggests

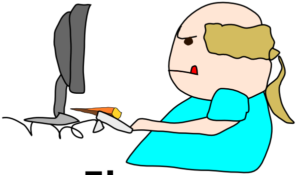
Actual


Developer
Evidence
Produces
(lossy)
Exhibits
Infers
Follows
Behaviour
recovers
Recovered
Behaviour
Infers
Fufill
Fufill
Purposes
& Tasks
recovers
Recovered
Purposes
& Tasks
Compose
Compose
Software
Development
Process
recovers
Recovered
Development
Process
Actual
Recovered

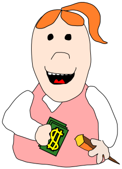
Research Relationships

	Behaviour	Intent, Purpose and Tasks	Software Development Process
Release Patterns: Source/Test/ Build/Docs	●		●
Large Changes		●	
Topic Analysis	●	●	
Recovered Unified Process Views			●

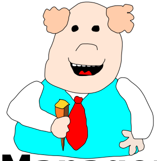
Motivation: Stakeholders



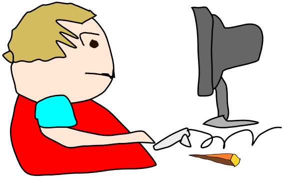
Fixers or Star Programmers



Investors and Acquisitions

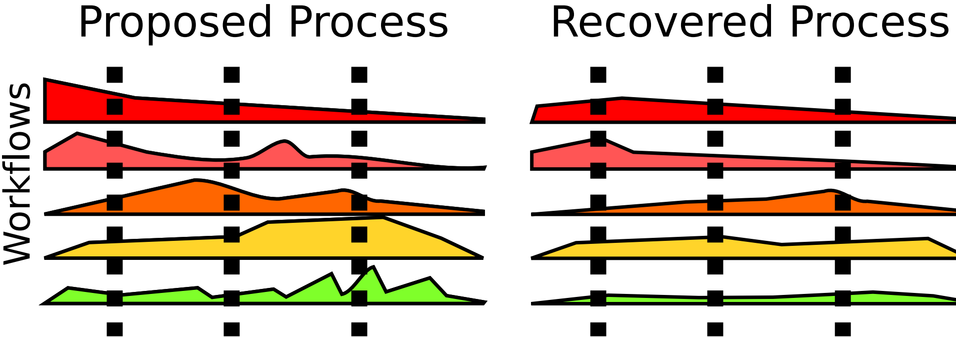


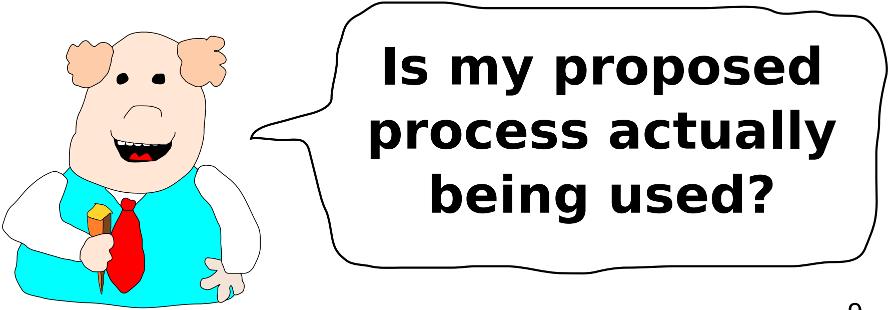
Managers



New Developers

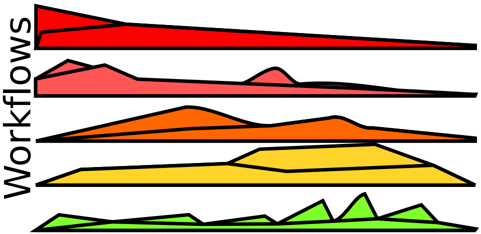
Employees assigned to a ISO9000 conformance project



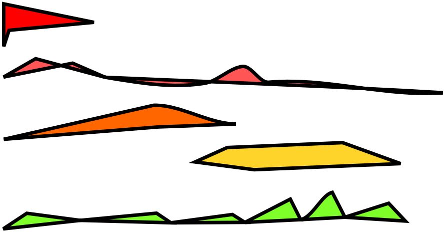


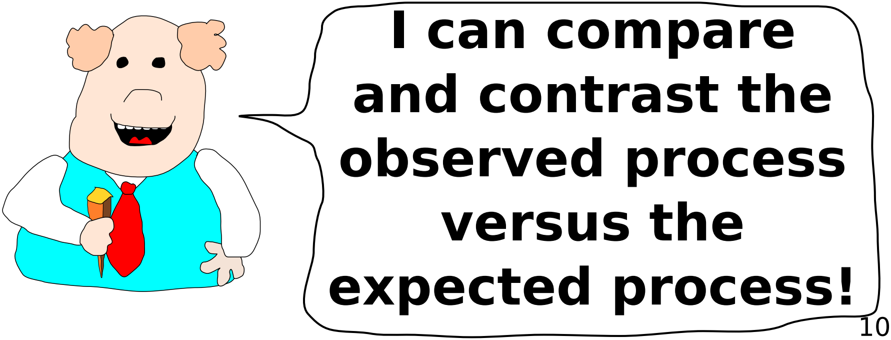
Is my proposed process actually being used?

Proposed and Recovered Process Overlayed

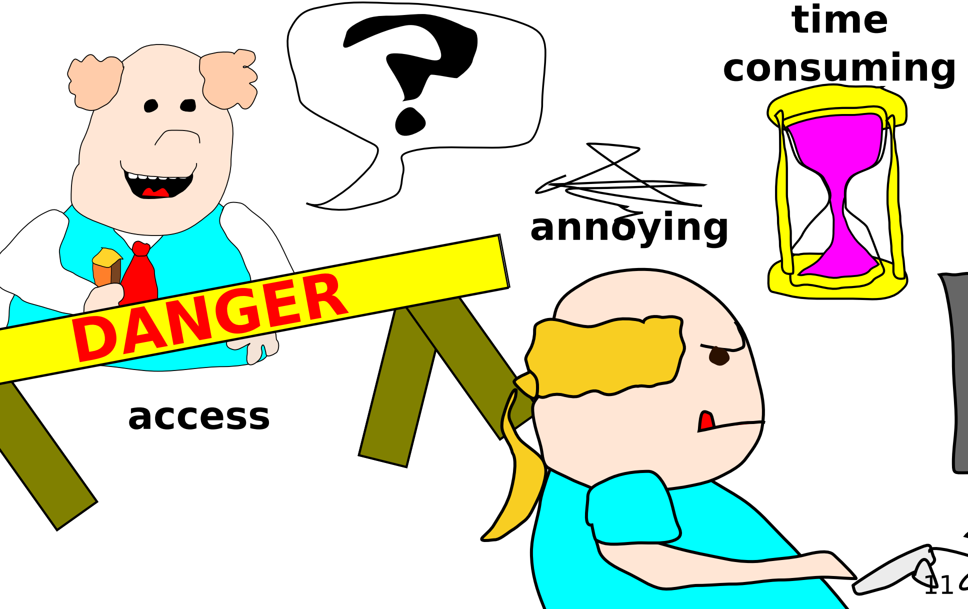


Differences between Proposed and Recovered




I can compare and contrast the observed process versus the expected process!

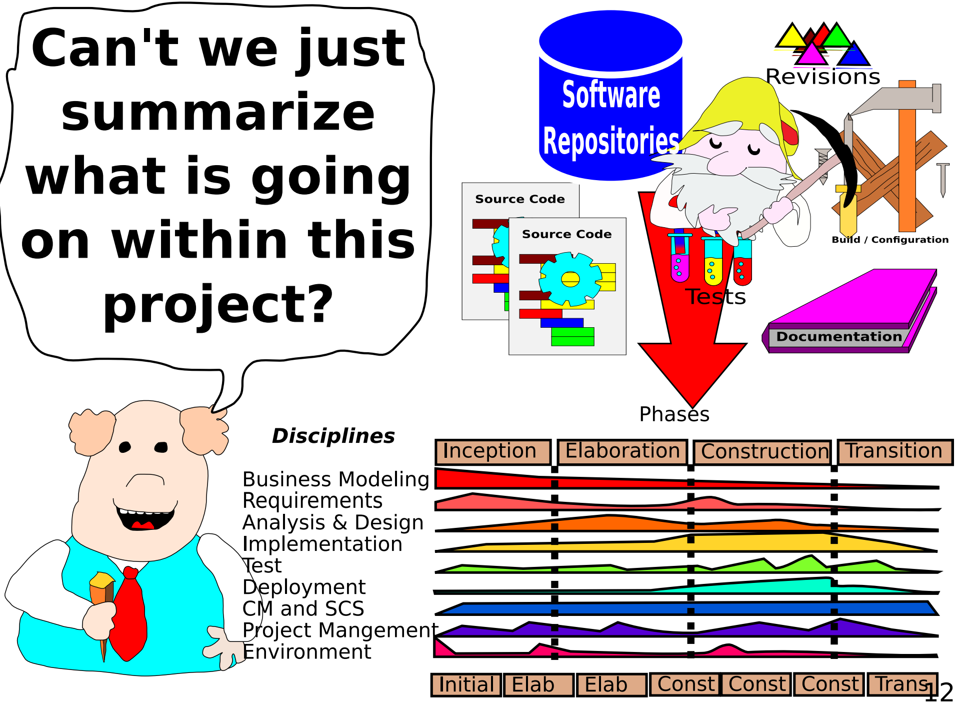
How to get an overview: Interviews



time consuming

annoying

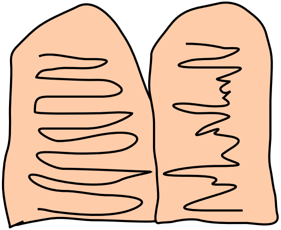
access


Can't we just summarize what is going on within this project?
Software Repositories
Revisions
Source Code
Source Code
Tests
Build / Configuration
Documentation
Phases
Disciplines
Inception
Elaboration
Construction
Transition
Business Modeling
Requirements
Analysis & Design
Implementation
Test
Deployment
CM and SCS
Project Mangement
Environment
Initial
Elab
Elab
Const
Const
Const
Trans

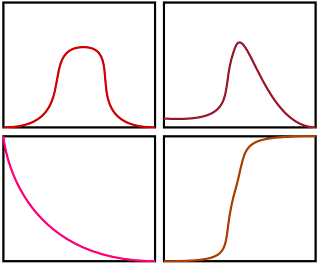
PREVIOUS WORK

RELATED RESEARCH

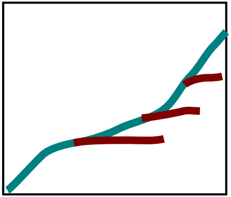
Stochastic Processes



[Lehman]



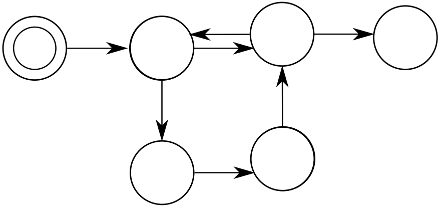
[Herraiz]



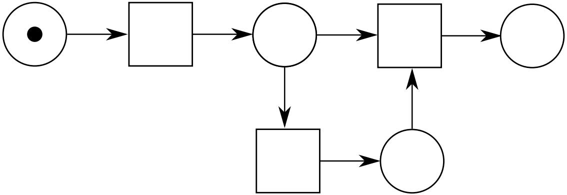
[Turksi]
[Tu]

Business Processes

Finite State Machines



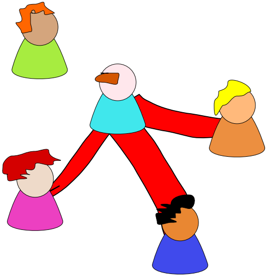
Petrinets



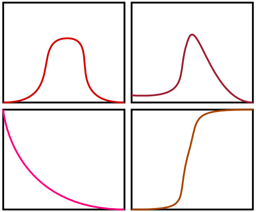
Execution of business goals

[Van der Aalst]

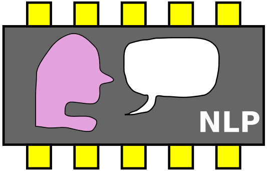
Analysis



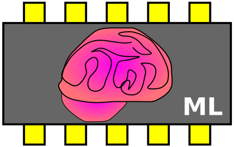
SNA



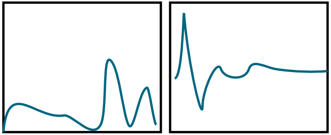
Statistics



NLP

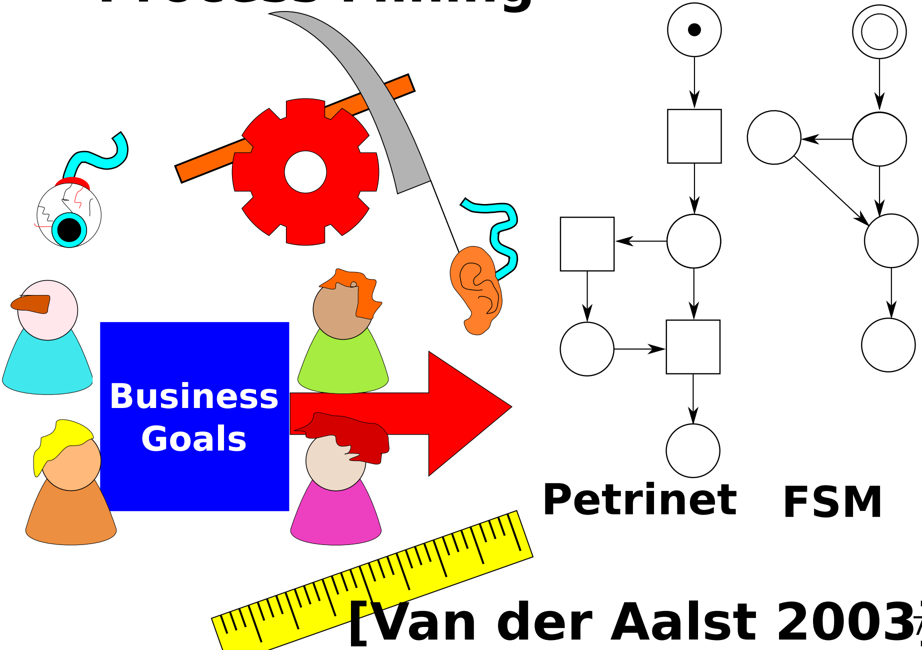


ML



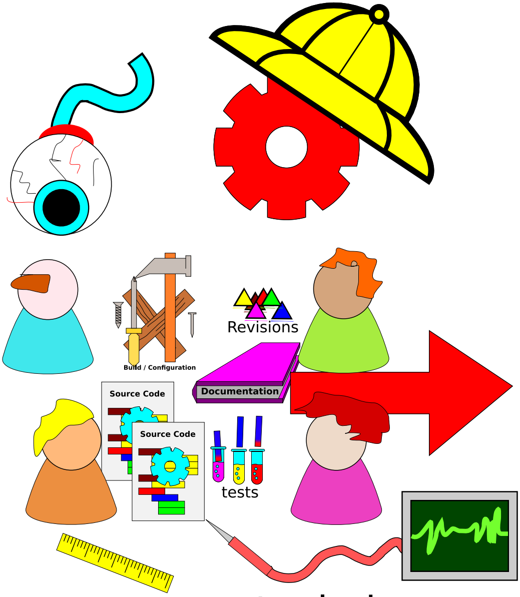
Timeseries

Process Mining

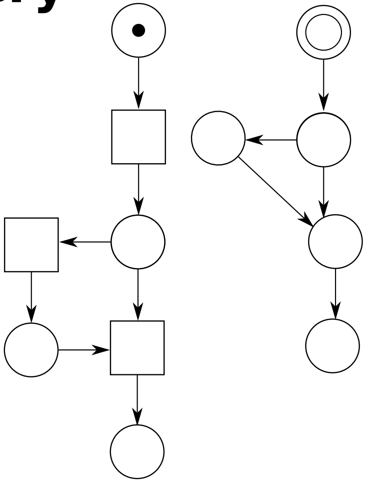


[Van der Aalst 2003]

Process Discovery



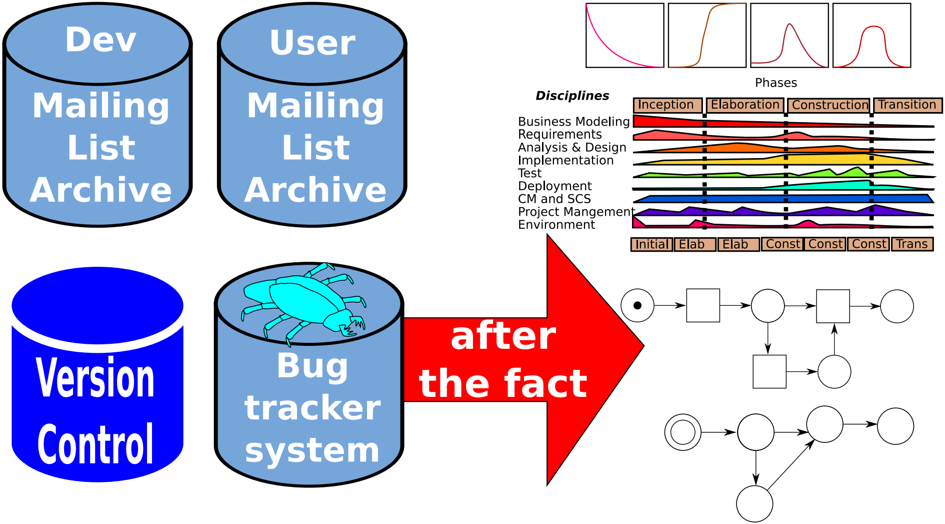
tooled process



Petrinet **FSM**

[Cook96]

Process Recovery



Dev Mailing List Archive

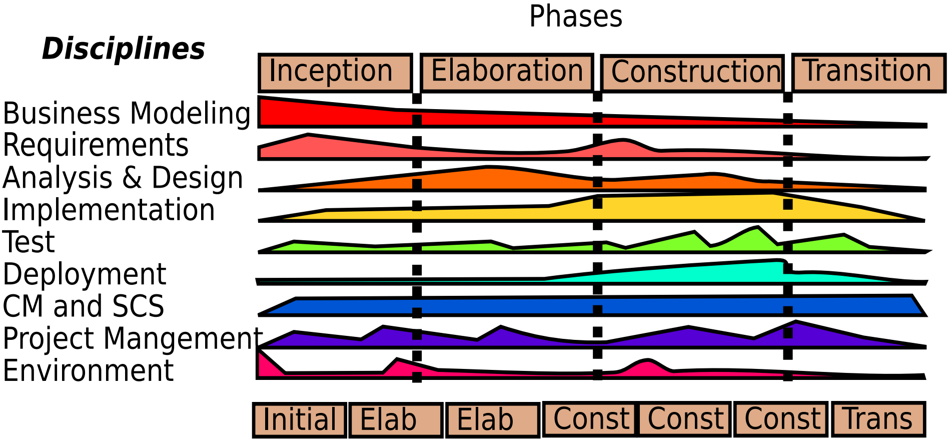
User Mailing List Archive

Version Control

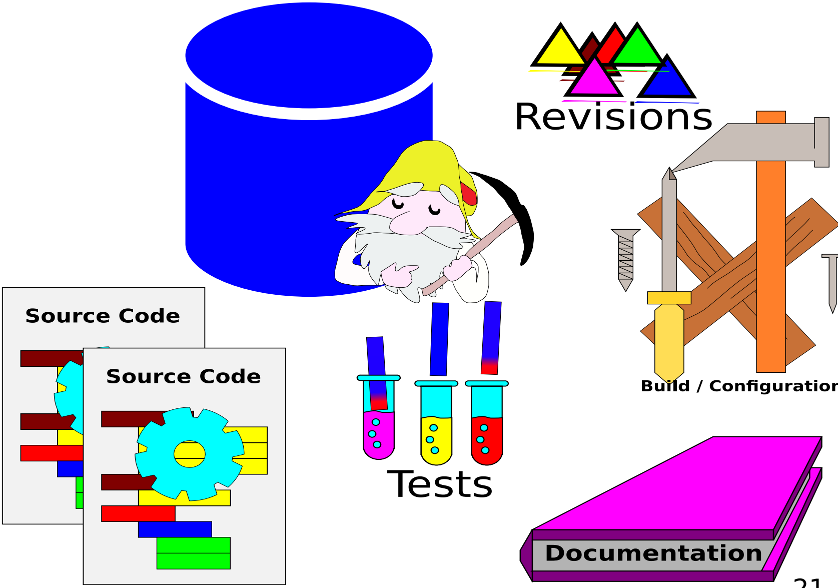
Bug tracker system

after the fact

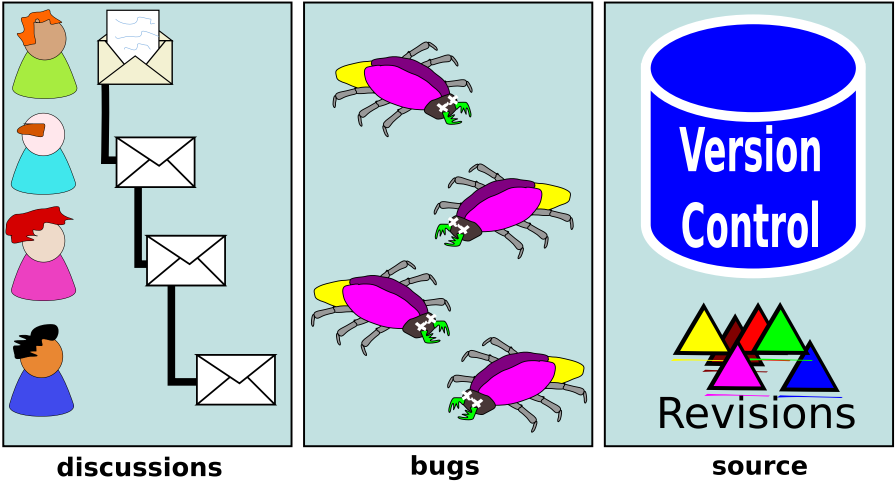
Unified Process Diagram



Mining Software Repositories

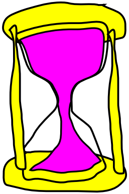


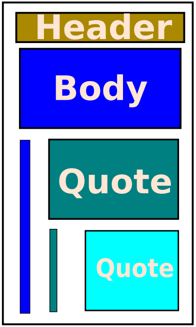
Source Acquisition



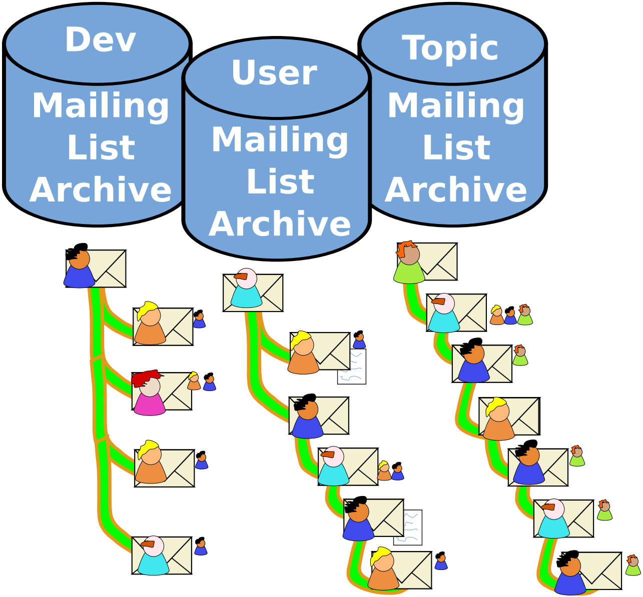
Initial Repositories and artifacts

Extraction: Mailing list archives

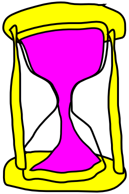


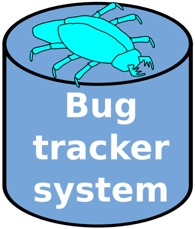


Email



Extraction: Bug trackers

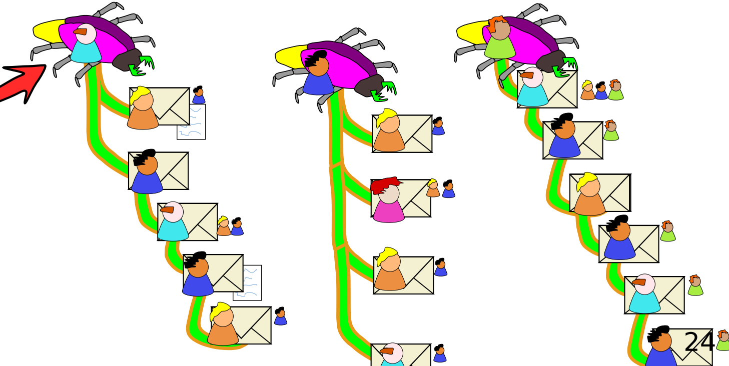




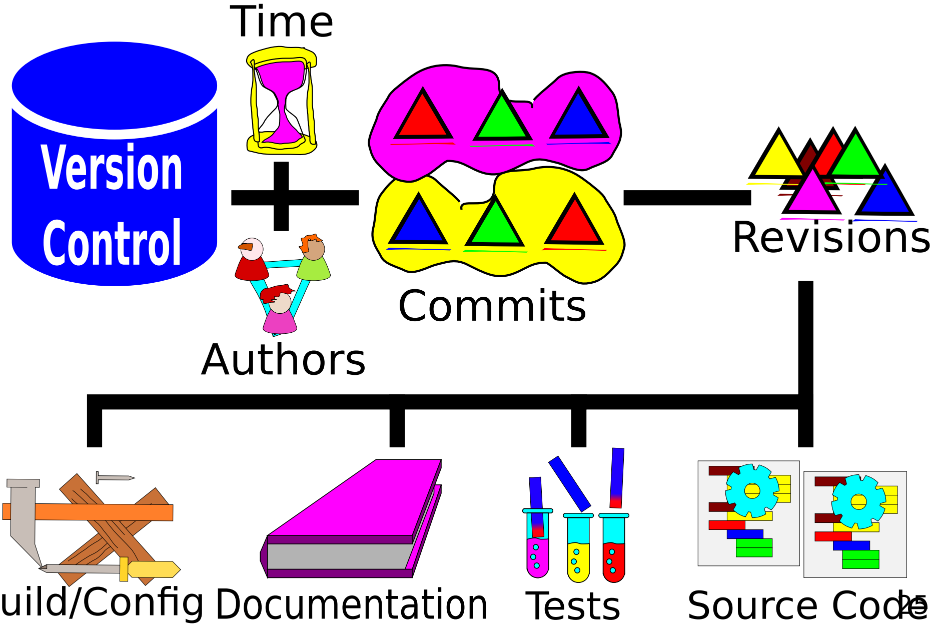
bug id

**Cricket,
Jim
#57981**

**Summary
Severity
Description
...**



Extraction: Version Control



Version Control

Time

Authors

Commits

Revisions

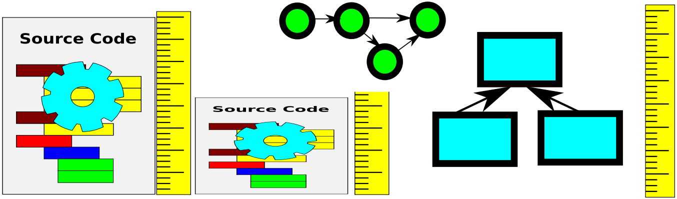
Build/Config

Documentation

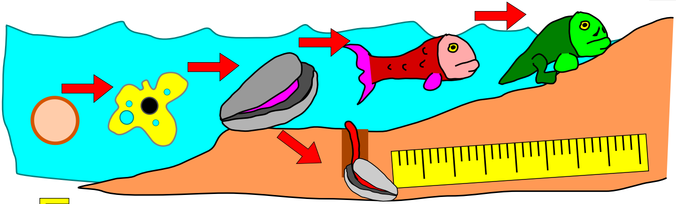
Tests

Source Code

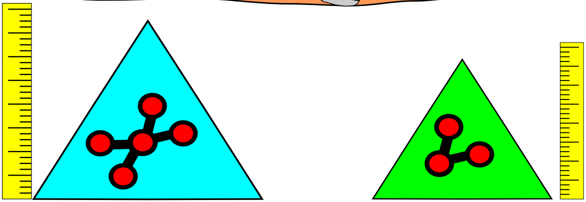
Metrics



Source code Metrics



Software Evolution Metrics

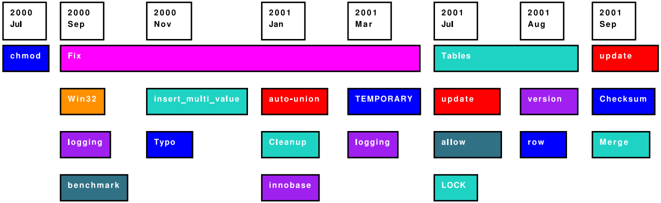


Coupling Metrics

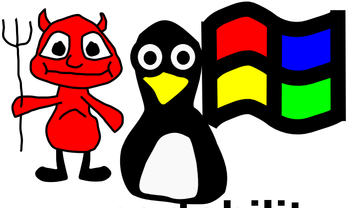
Topic/Concept Analysis

	Entity1	Entity2	Entity3	Entity4	Entity5	Entity6	Entity7	Entity8
Concept 1	●							●
Concept 2		●		●		●	●	●
Concept 3	●				●			
Concept 4			●		●		●	
Concept 5			●				●	

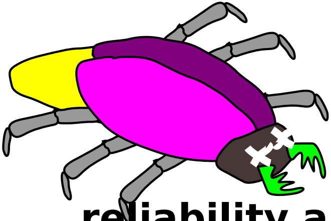
[Lukins]
[Linstead]
[Maletic]
[Poshyvanik]
[Marcus]
[Hindle]



Quality Related
Non functional requirements

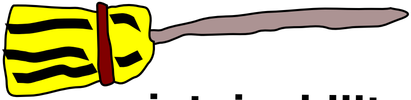


portability



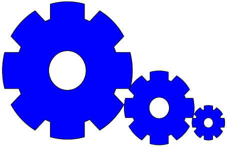
reliability and functionality (includes correctness)

usability



maintainability

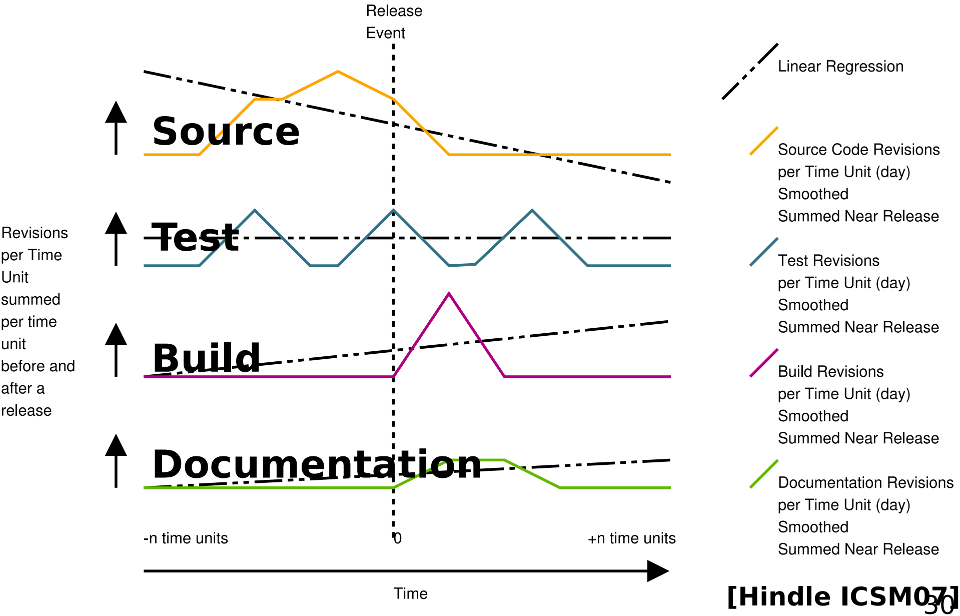
efficiency



[cleland-huang03]
[ernst10]

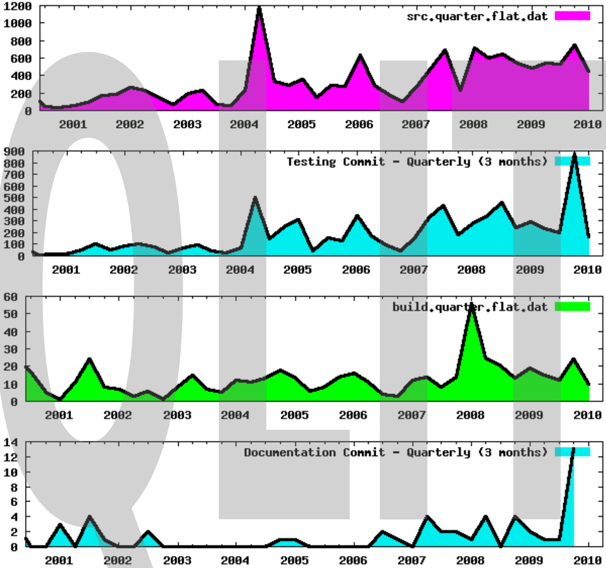
SOFTWARE PROCESS RECOVERY

Release Patterns: STBD

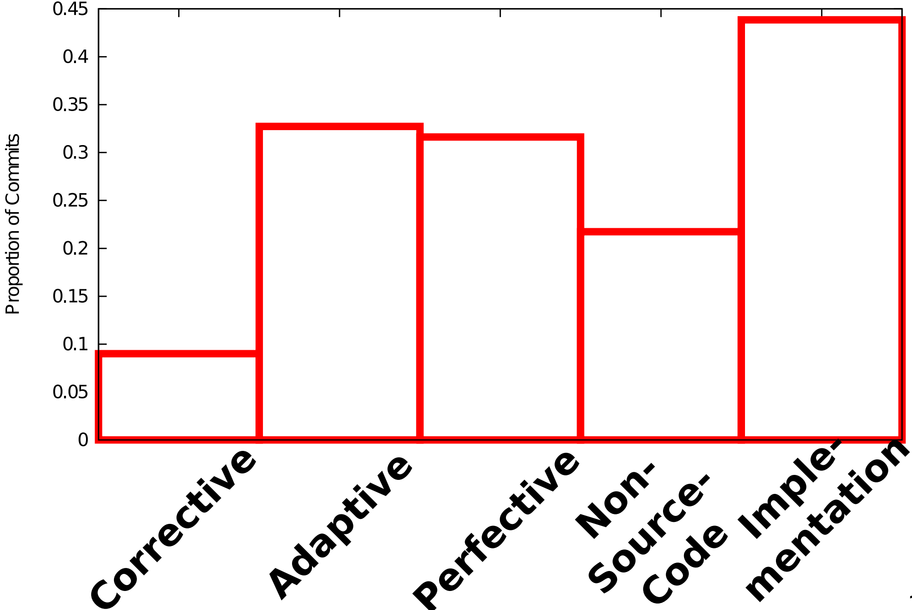


[Hindle ICSM07]

STBD applied to SQLite

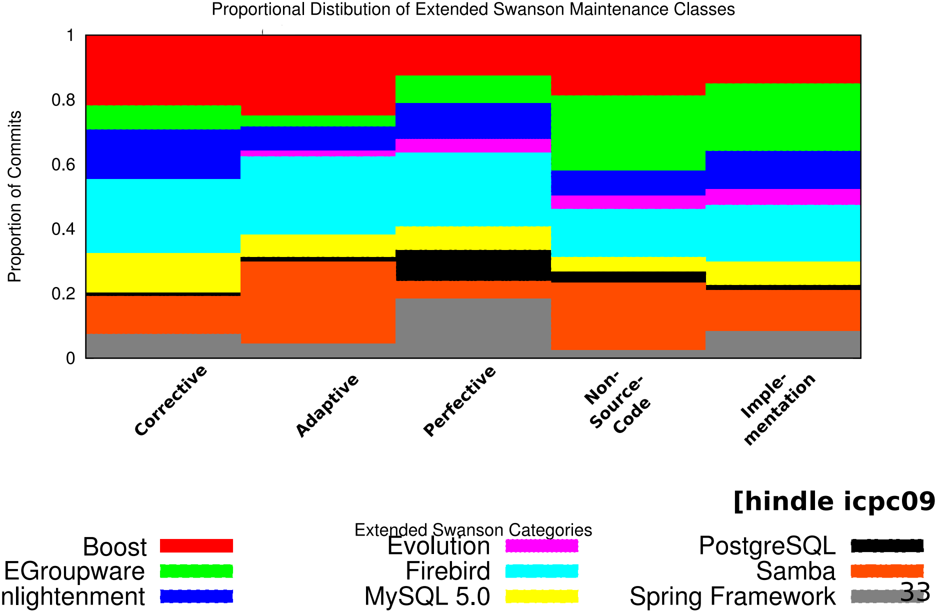


Maintenance Classes of Large Changes



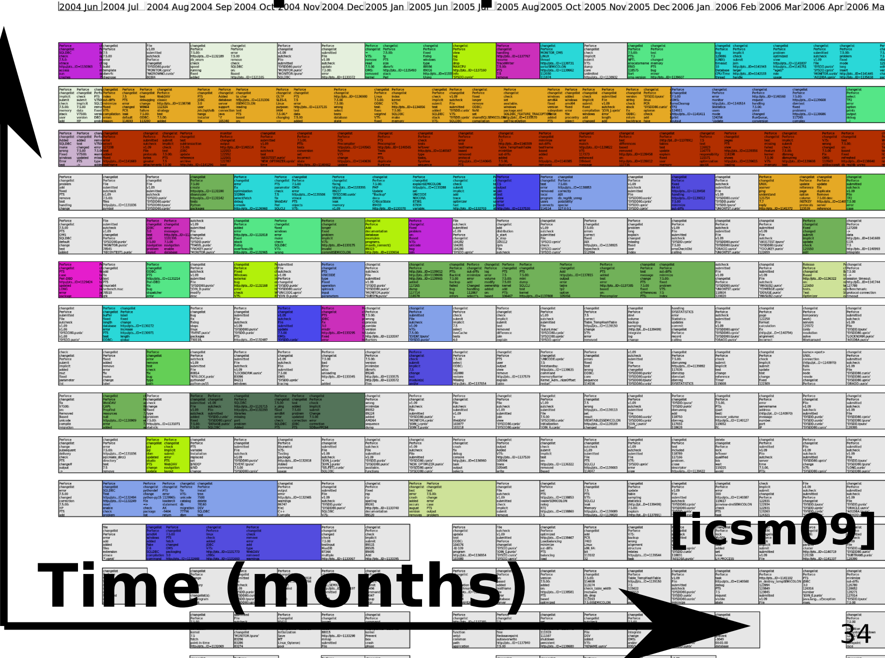
[Hindle ICPC09]

Supervised: Maintenance Classes



[hindle icpc09]

Developer Topics



Unique Topics

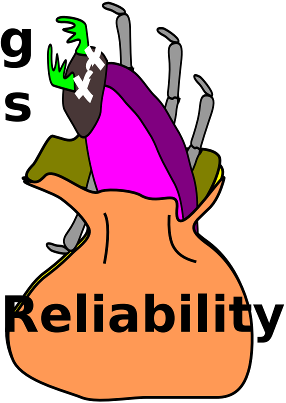
Time (months)

[icsm09]

Word Bag Examples

Portability

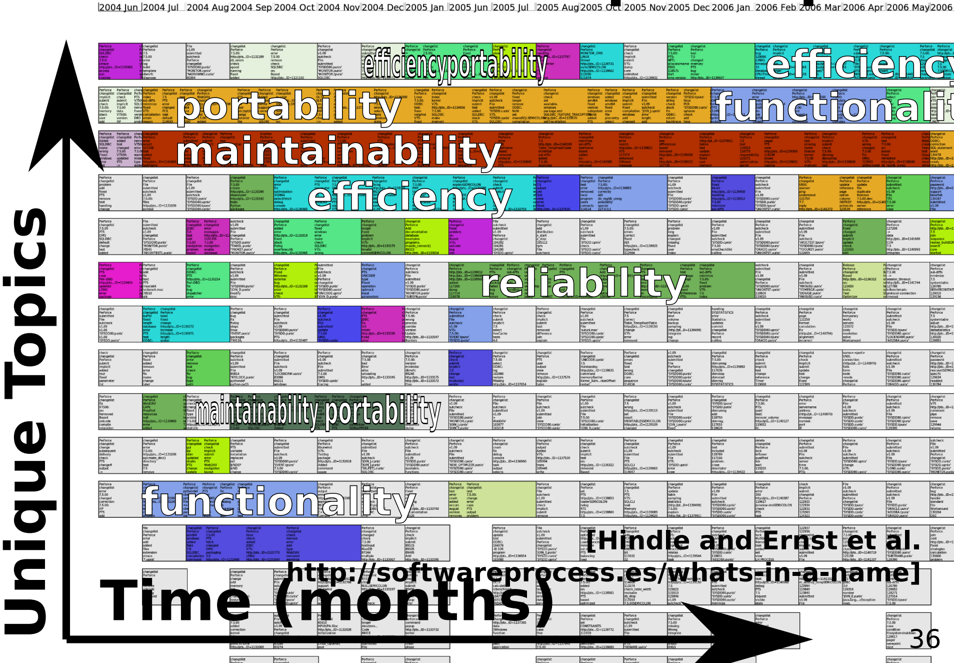
portability
transferability
interoperability
documentation
internationalization
i18n
...



Reliability

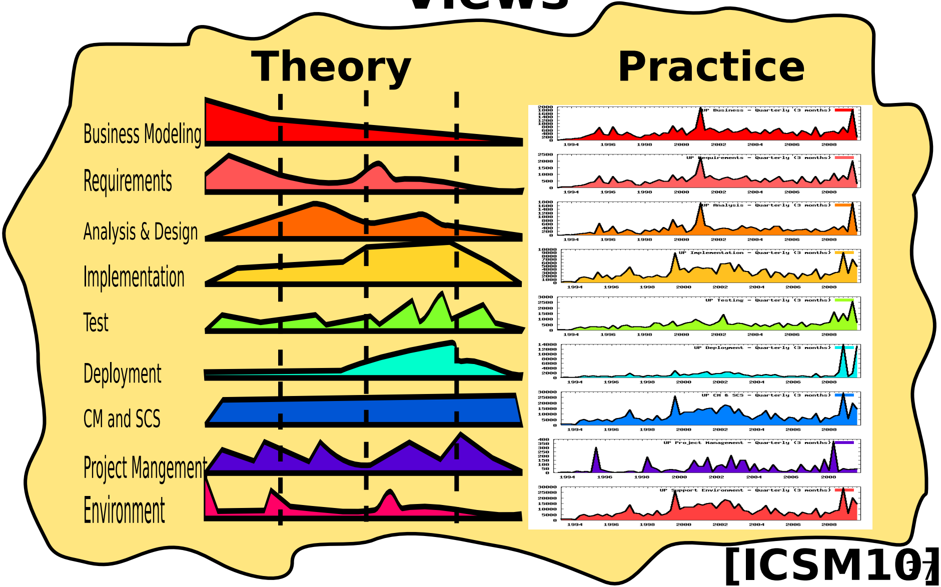
reliability
failure
error
redundancy
fails
bug
...

Labelled Developer Topics



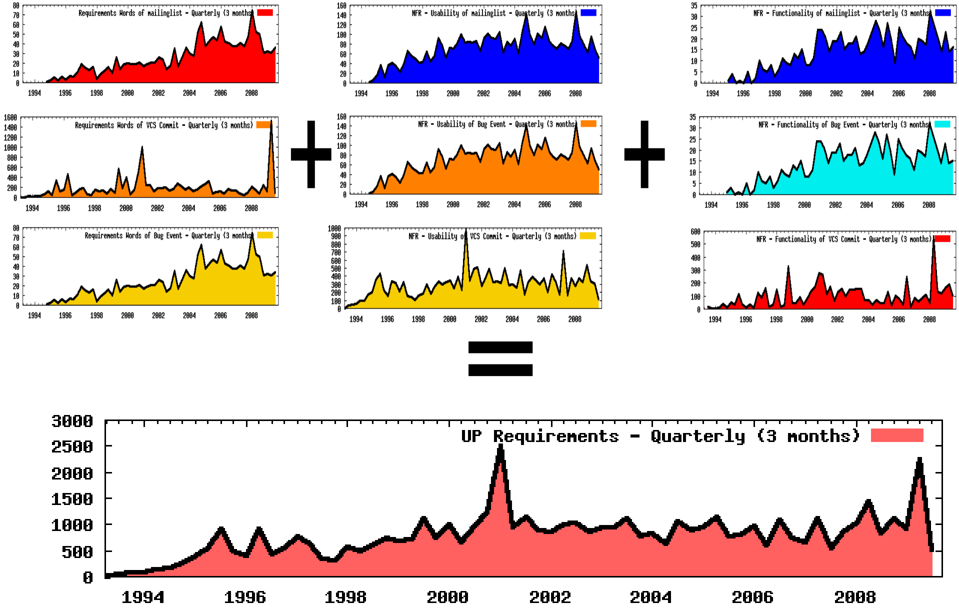
[Hindle and Ernst et al.
http://softwareprocess.es/whats-in-a-name]

Recovered Unified Process Views

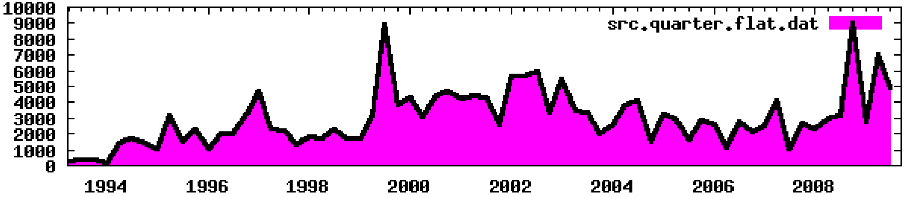


[ICSM10]

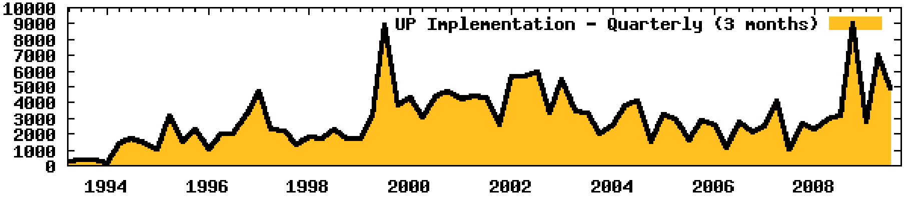
UP Requirements Signal



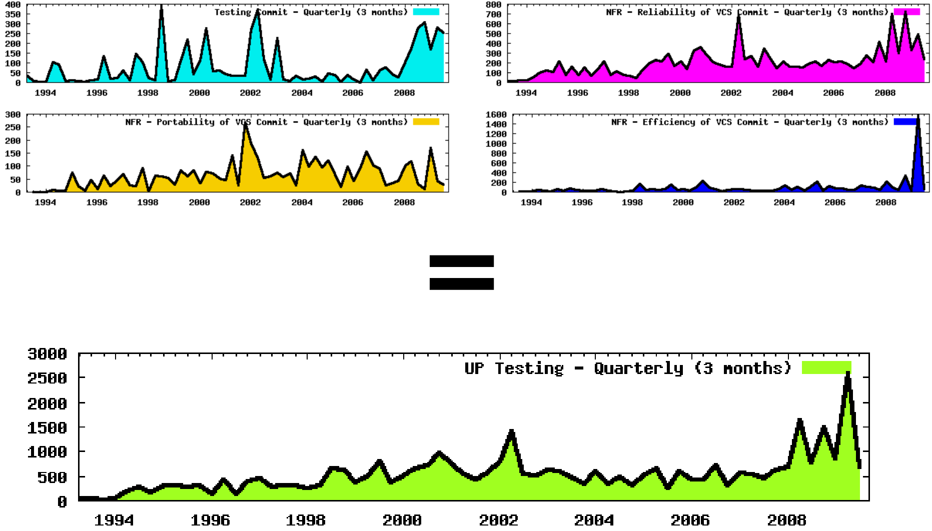
UP Implementation Signal



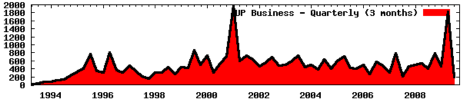
=

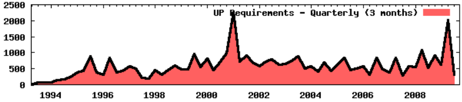


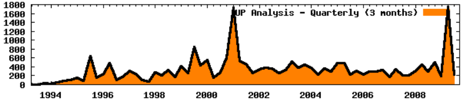
UP Testing Signal

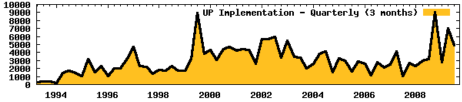


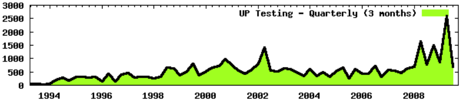
FreeBSD Case Study

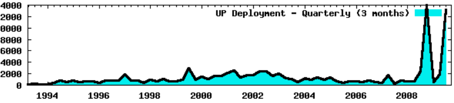


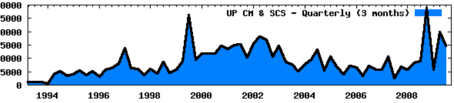


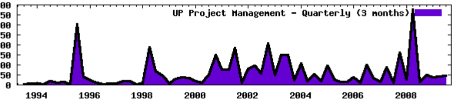


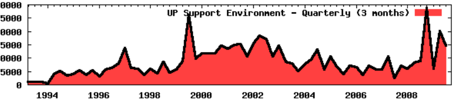




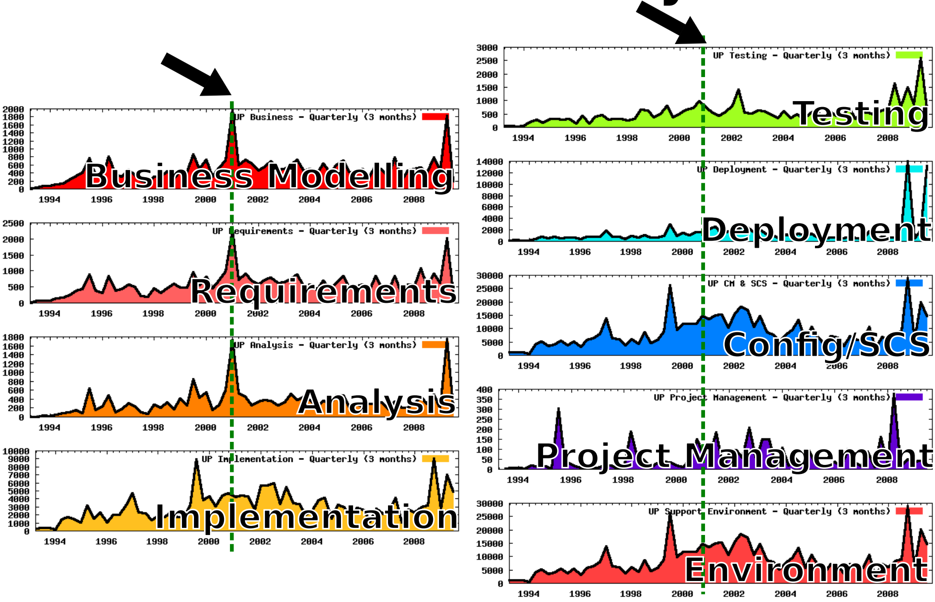




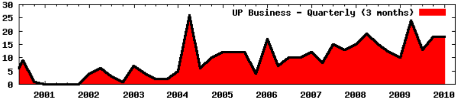


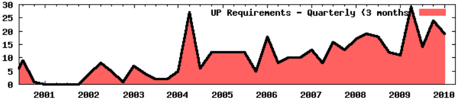


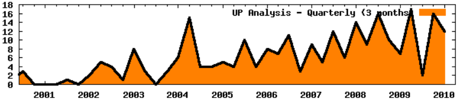
FreeBSD Case Study: 2001

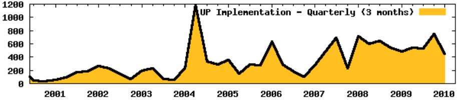


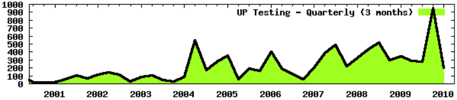
SQLite Case Study

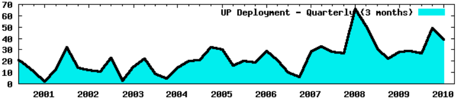


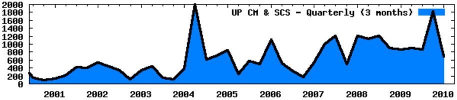


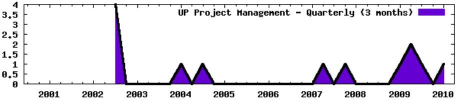


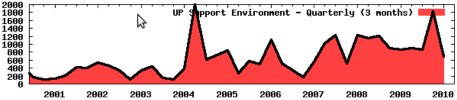




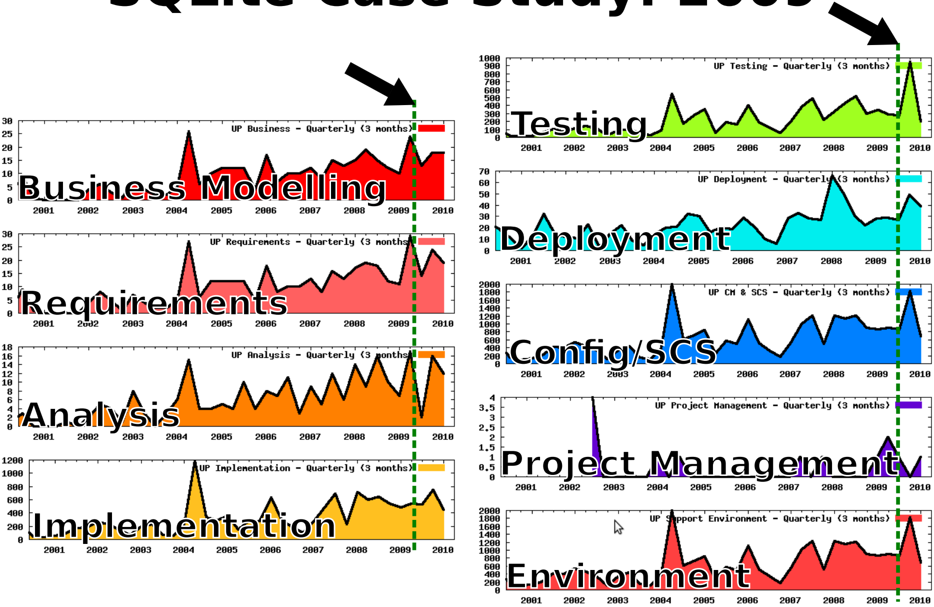




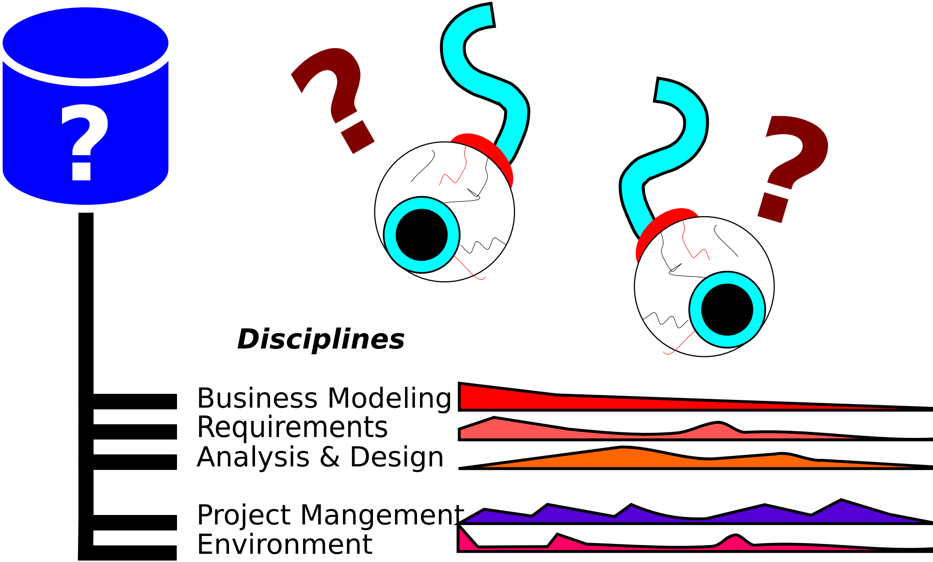




SQLite Case Study: 2009



UP Observability

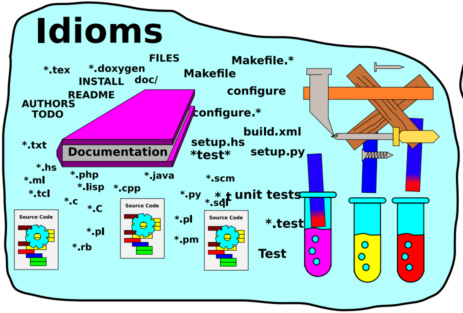


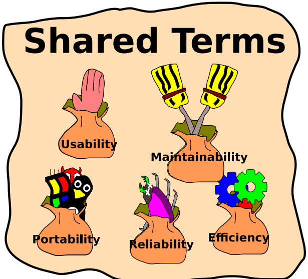
Disciplines

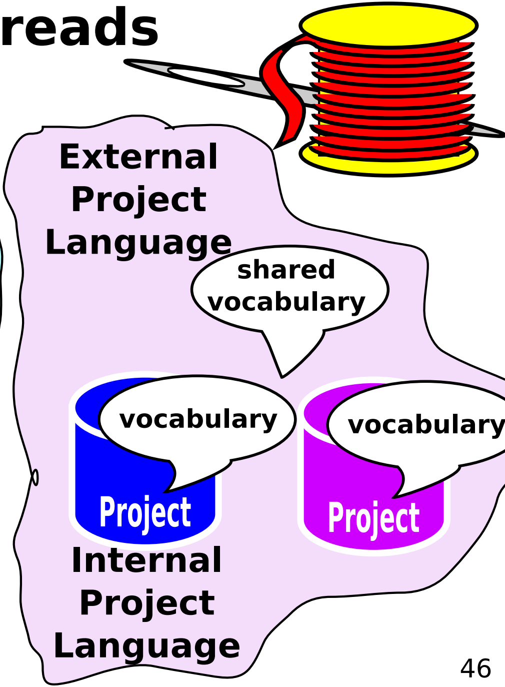
Business Modeling
Requirements
Analysis & Design

Project Mangement
Environment

Common Threads

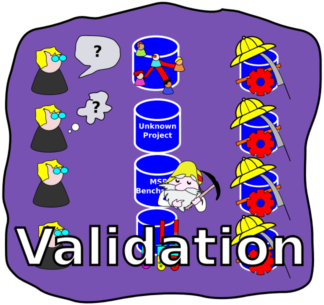






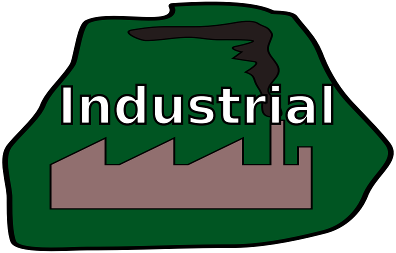
Future Work

People and teams



Validation

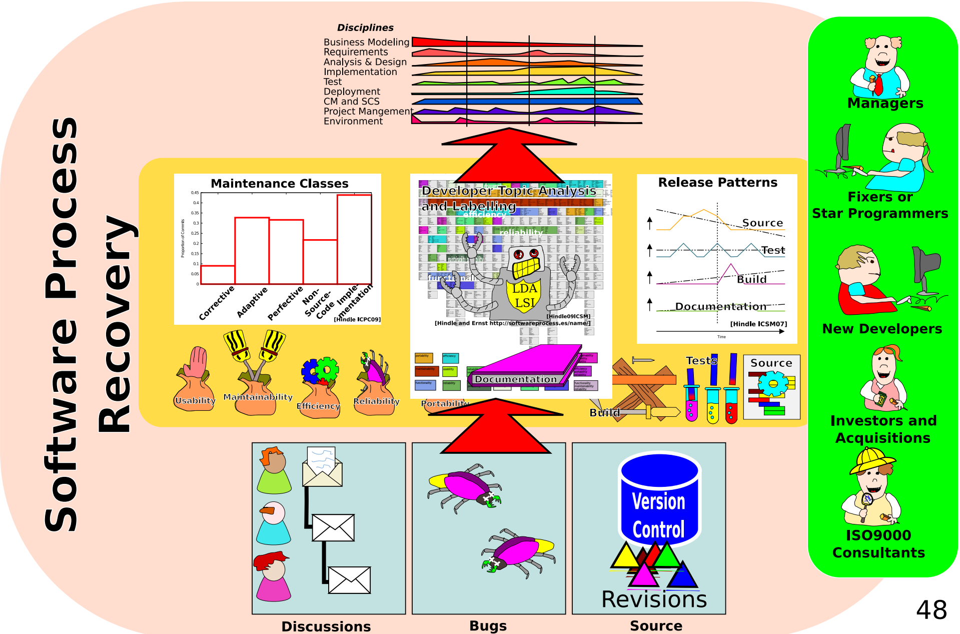
Accuracy



Industrial

Iteration Identification

Conclusions



Software Process Recovery

[Hindle ICPC09]

[Hindle09ICSM]
[Hindle and Ernst http://softwareprocess.es/name/]

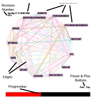
[Hindle ICSM07]

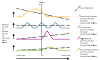
Research Timeline

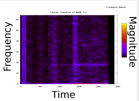
The Past

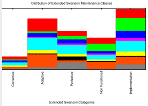
Published

- YARN - visualization [VISSOFT 2007]
- Release Discovery (MLM) [MSR & ICSM 2007]
- evolution metrics [ICPC/SCAM 2008, SSP]
- study of large changes [MSR 2008]
- change classification [ICPC 2009]
- recurrent behaviour [ICSE 2009]
- topic analysis [ICSM 2009]
- RUPV [ICSM 2010]
- Topic Naming [ICSE Submission]

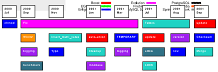




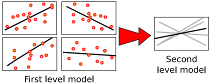


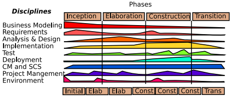


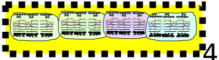
Present and ~~Future~~



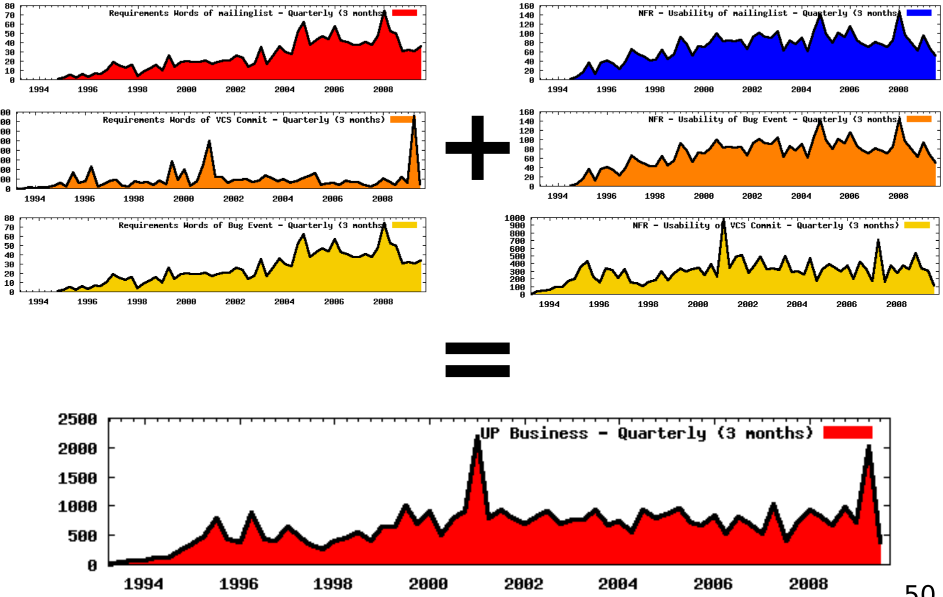
- finish Topic Naming paper [writeup]
- ~~finished MLM journal paper [casestudy]~~
- ~~slicing [writeup]~~
- multi-timeline correlation [some imp]
- Unified process diagram summary [imp]
- ~~phase & iteration identification [imp]~~
- Thesis



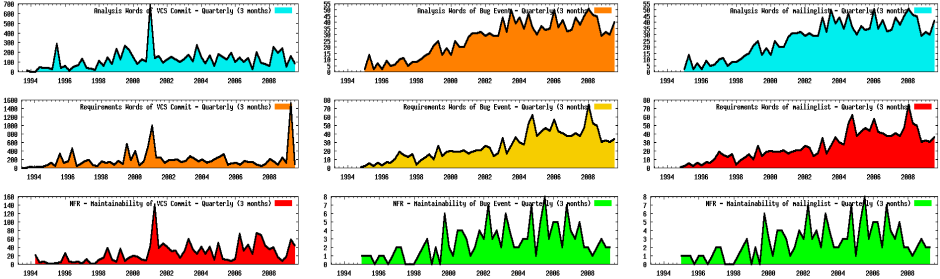




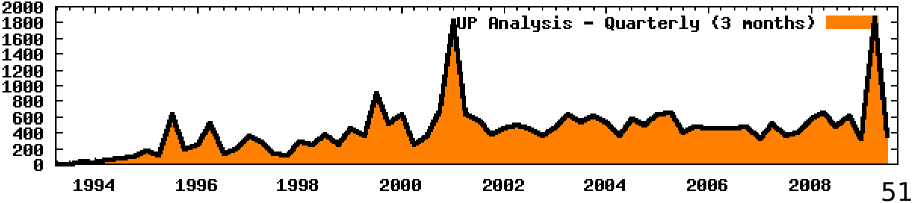
UP Business Modelling Signal



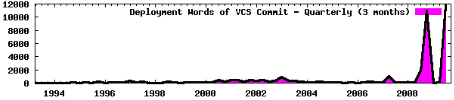
UP Analysis Signal

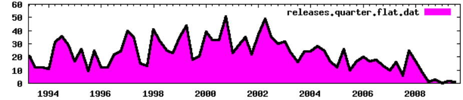


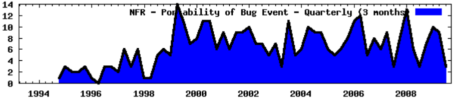
=

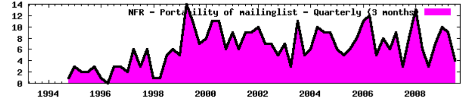


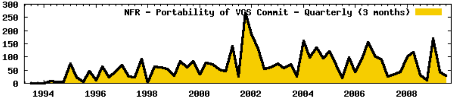
UP Deployment Signal

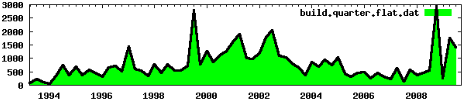




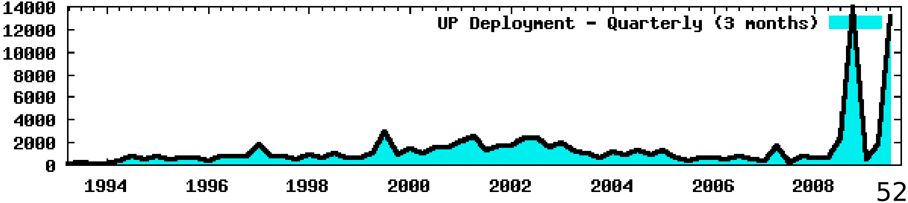




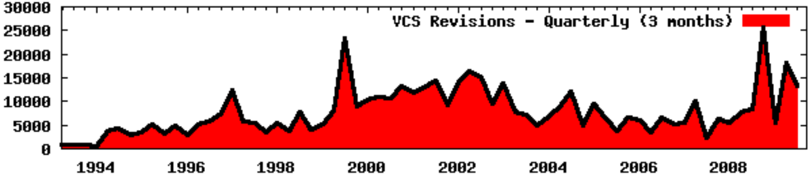


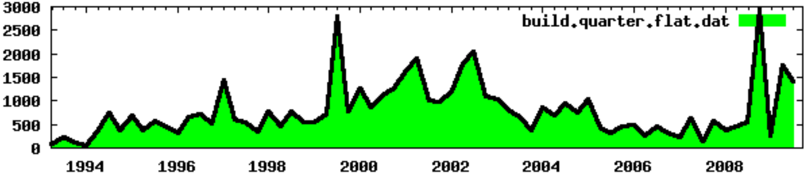


=

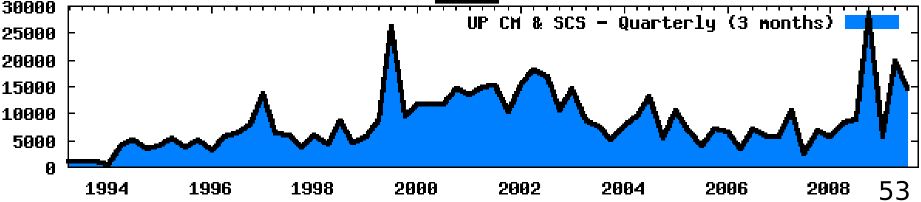


UP Configuration Managment and SCS

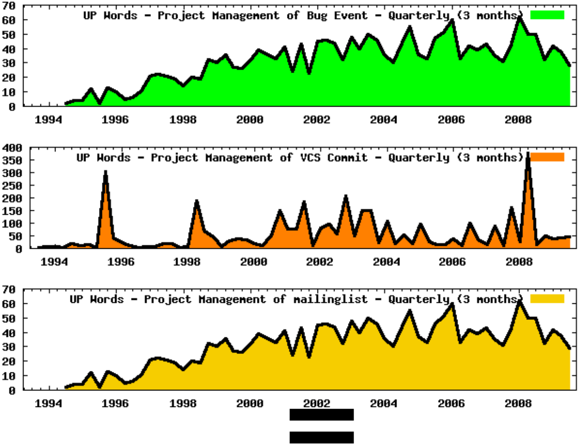


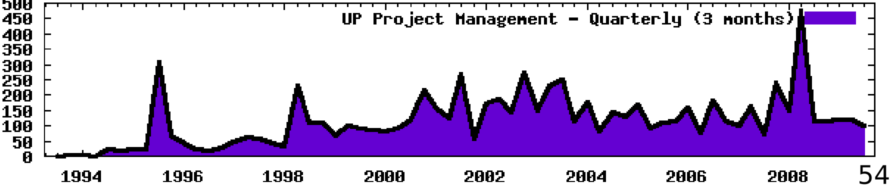


=

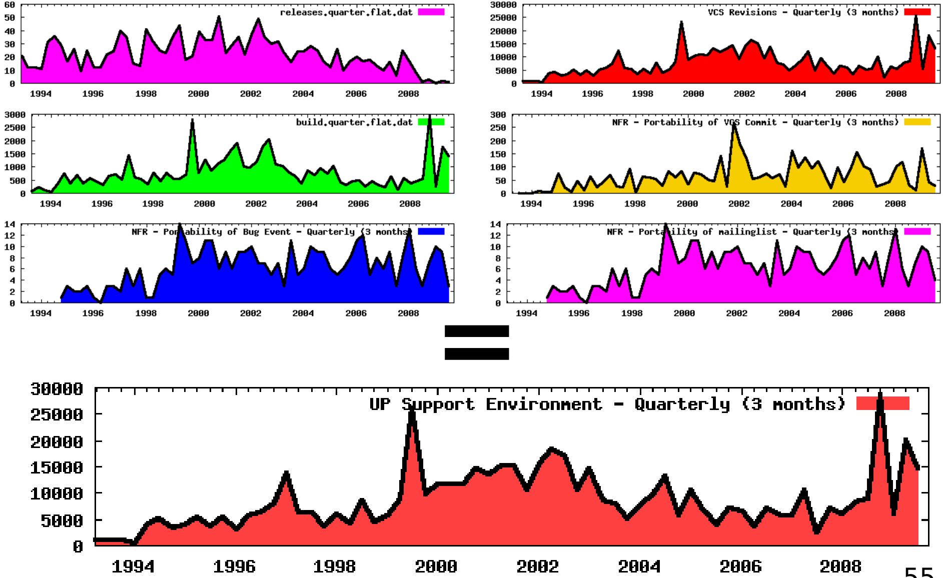


UP Project Management Signal

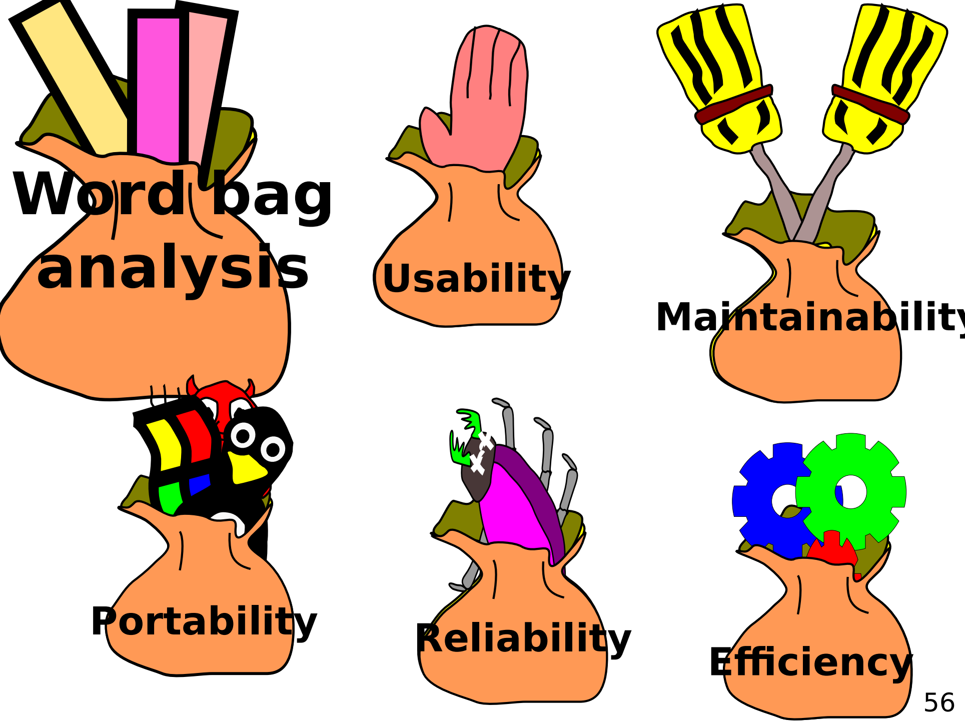




UP Environment Signal



Word bag analysis



Usability

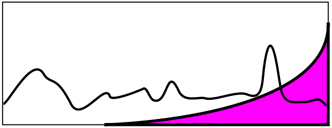
Maintainability

Portability

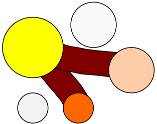
Reliability

Efficiency

Prediction



window decay

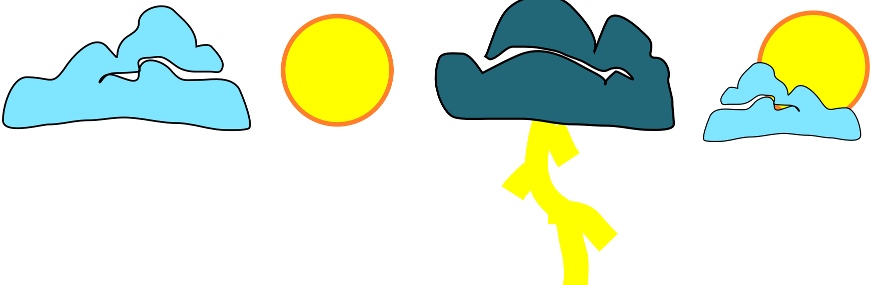


co-change

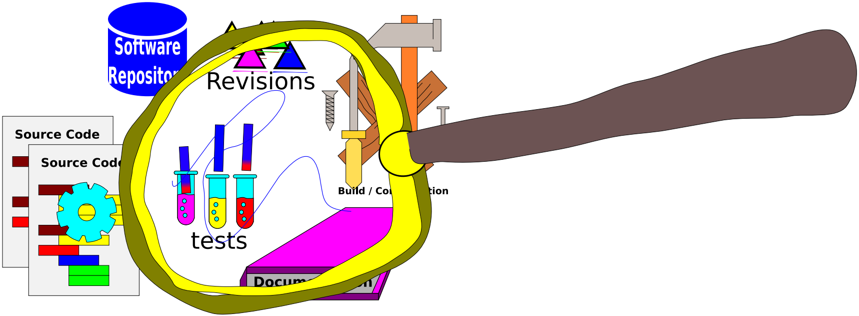
[Girba]

[Askari]

[Hassan]



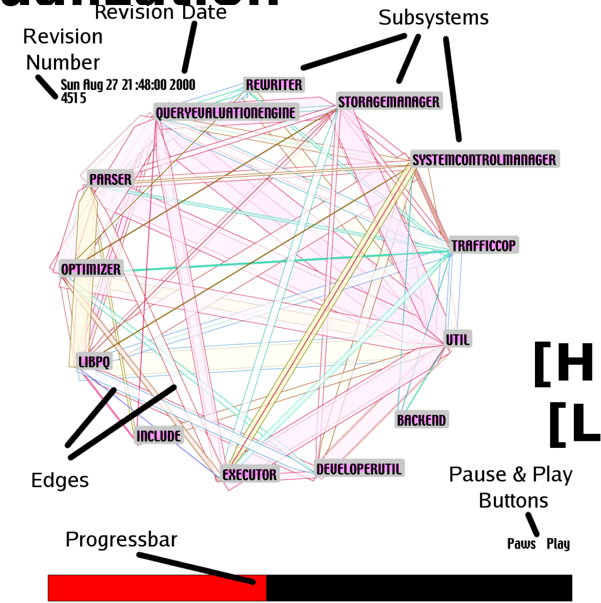
Querying



- * 1st Order Logic
- * Temporal Logic
- * Unification

[Cubranic]
[Kim]
[Hindle]

Visualization



Revision Date

Revision Number

Subsystems

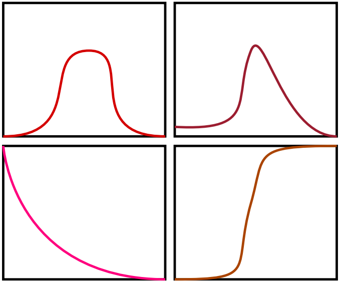
Edges

Pause & Play Buttons

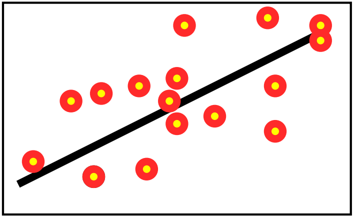
Progressbar

[Hindle]
[Lanza]

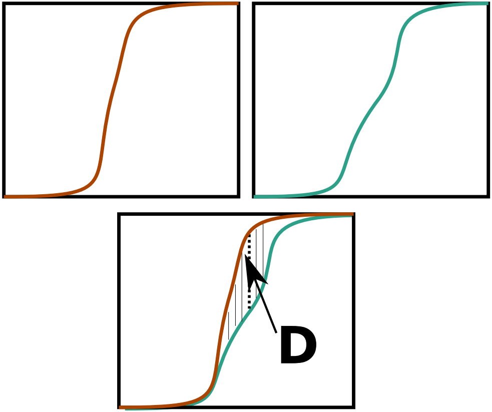
Statistics



Distributions

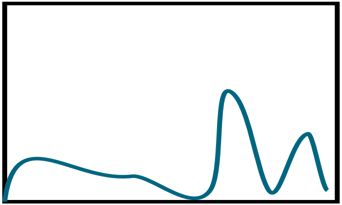


Linear Regression

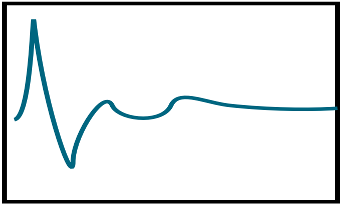


Compare distributions

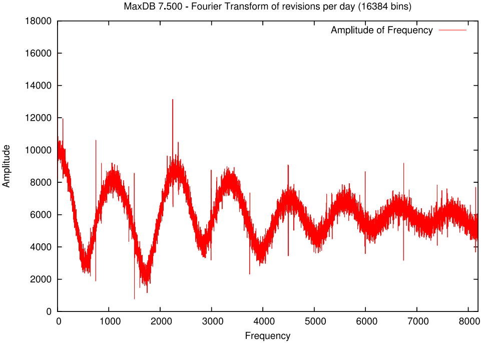
Timeseries



Timeseries

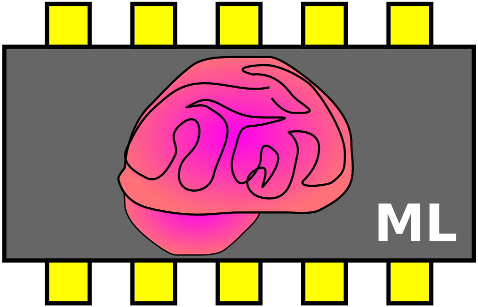


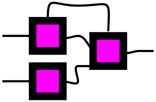
Autocorrelation

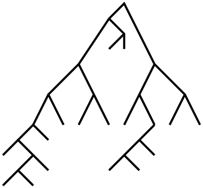


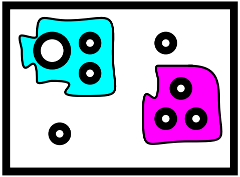
[Herraiz]
[Hindle]

Machine Learning

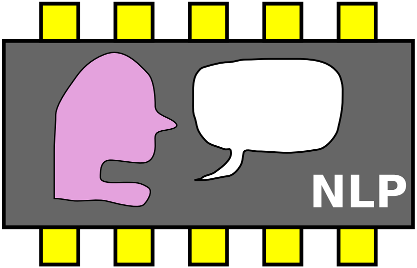


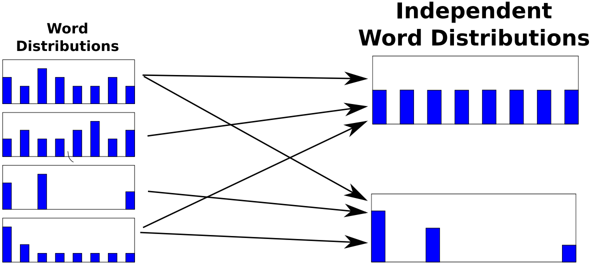




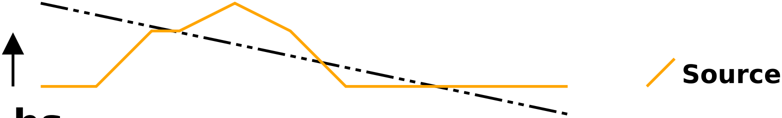


Natural Language Processing





Release Patterns: Source revisions



*.hs

*.ml

*.tcl

*.php

*.lisp

*.c

*.C

*.pl

*.rb

*.cpp

*.java

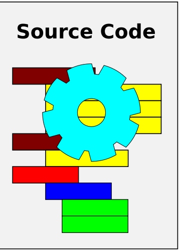
*.py

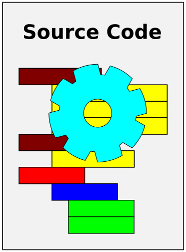
*.pl

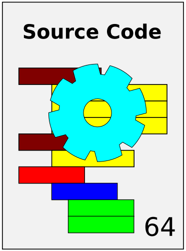
*.pm

*.scm

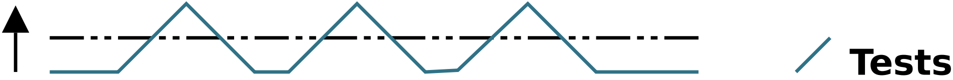
*.sql







Release Patterns: Test Revisions



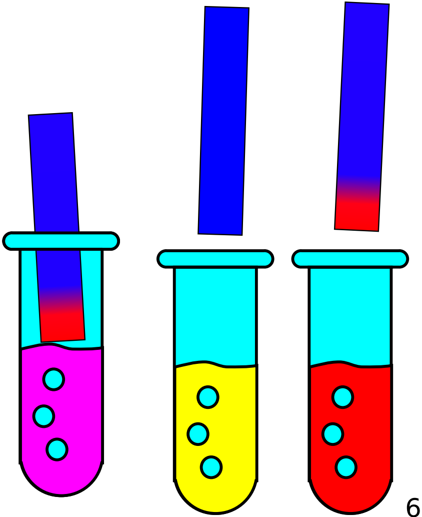
*.t

test

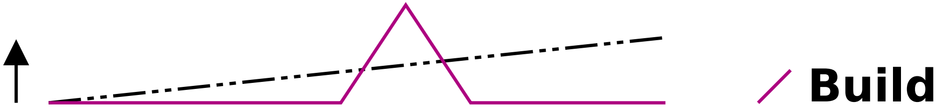
unit tests

*.test

Test



Release Patterns: Build file revisions



Makefile.*

Makefile

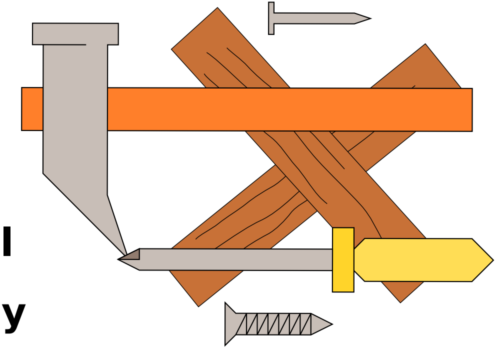
configure

configure.*

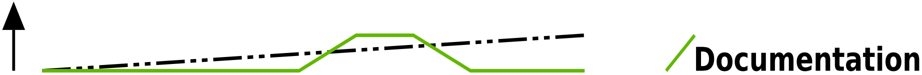
build.xml

setup.hs

setup.py



Release Patterns: Documentation Revisions



FILES

*.tex

*.doxygen

doc/

INSTALL

AUTHORS

README

*.txt

TODO

maybe: *.html
*.png
*.svg