Understanding Change

Where do we look at?

Filip Van Rysselberghe

Benevol 2005

Background

which change operations (and sequences) are applied?

➢ **Many changes**

 – Which ones are relevant?

➢ **Relevant changes**

 – Changes to remove design problems

 – Changes that influence the evolution
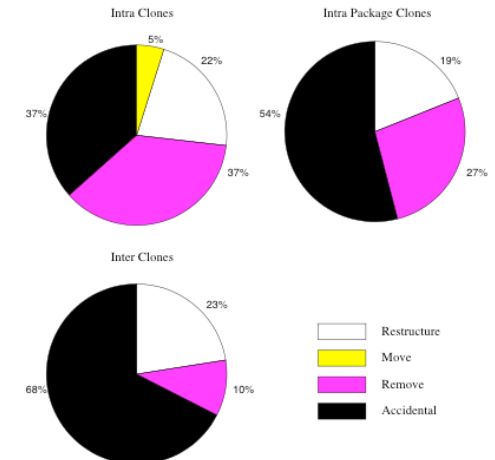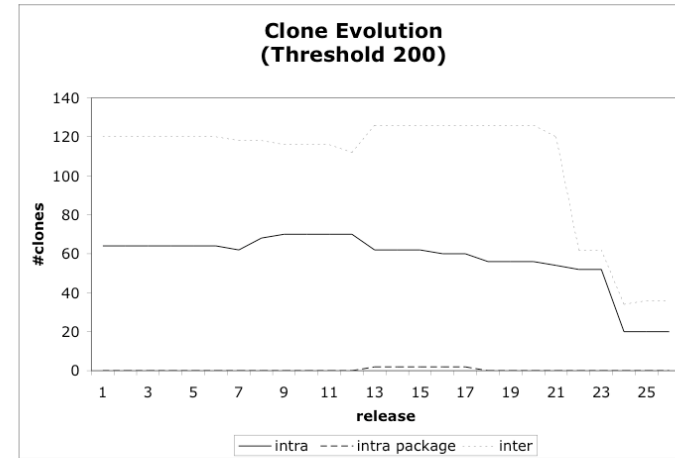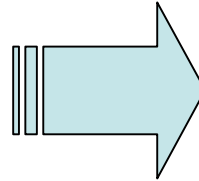
 – Refactorings

Approaches

➢ **Problem-up**
  – Focus a change when it removes a problem

➢ **Evolution-up**
  – Focus a change when its evolution changes

➢ **Rationale-up**
  – Focus a change when the developer tags it

Problem-up

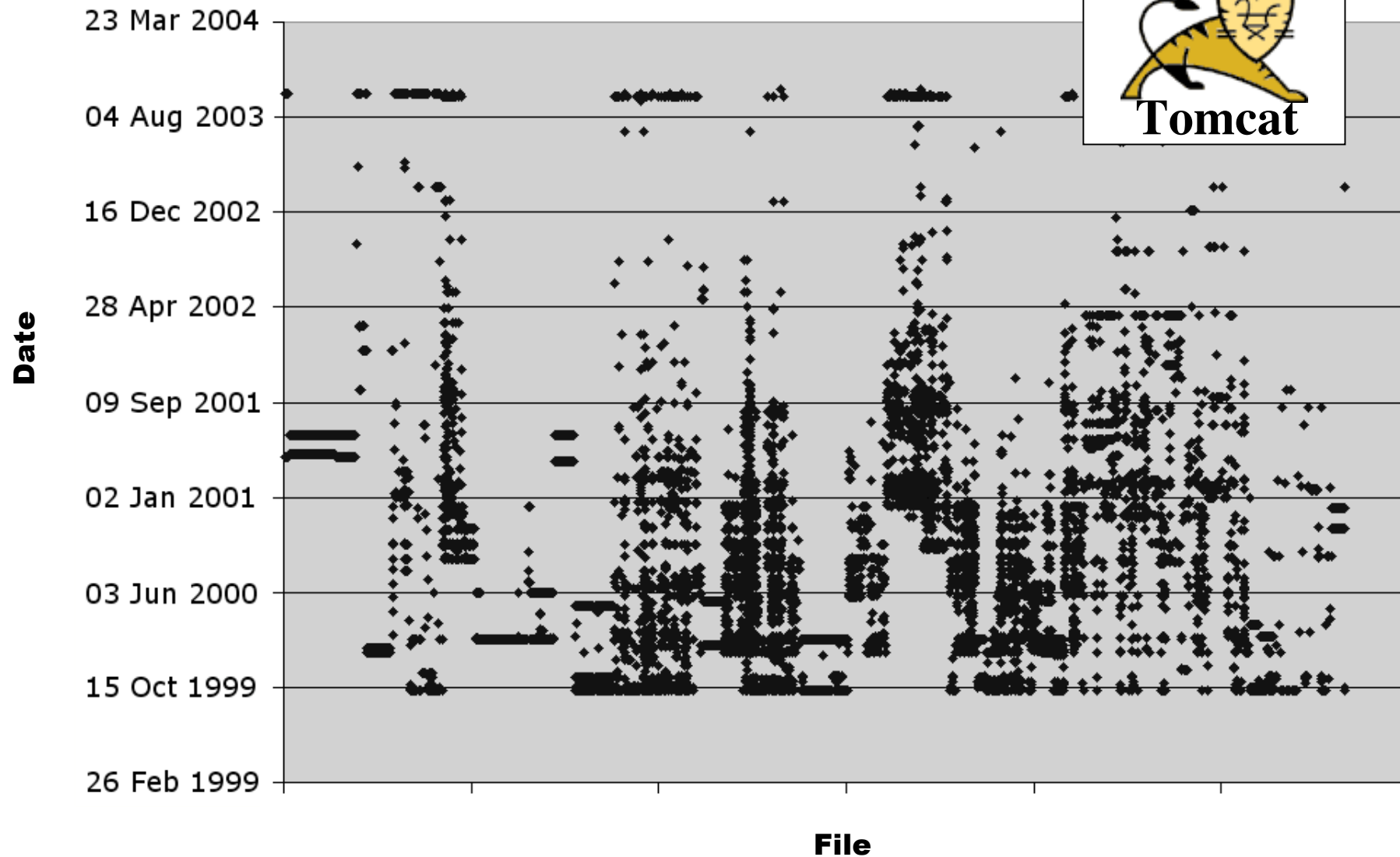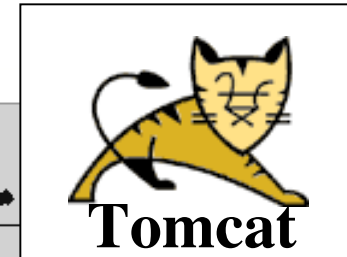|	R1			R2			R3			R4			R5			R6		
CookieTools.java	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Ajp12ConnectionHandler.java	2	0	0	2	0	0	2	0	0	2	0	0	2	0	0	2	0	0
IncludeGenerator.java	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0
EmbededServletOptions.java	2	0	0	2	0	0	2	0	0	2	0	0	2	0	0	2	0	0
FileRealmTool.java	4	0	0	4	0	0	4	0	0	0	0	0	0	0	0	0	0	0
SimpleTcpEndpoint.java	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
DefaultServlet.java	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
WarFileServlet.java	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
ForwardGenerator.java	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0
Parser.java	2	0	0	2	0	0	2	0	0	2	0	0	2	0	0	2	0	0
PoolTcpEndpoint.java	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
CookieUtils.java	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
TagLibraryInfoImpl.java	4	0	0	4	0	0	4	0	0	4	0	0	4	0	0	4	0	0
WebDescriptorFactoryImpl.java	2	0	0	2	0	0	2	0	0	2	0	0	2	0	0	2	0	0
LocaleToCharsetMap.java	2	0	0	2	0	0	2	0	0	2	45	0	2	45	0	2	45	0
ApacheConfig.java	0	0	0	0	0	0	0	0	0	2	0	0	2	0	0	2	0	0
MimeMap.java	0	0	0	0	0	0	0	0	0	2	45	0	2	45	0	2	45	0
ContextManager.java	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
DefaultCMSetter.java	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
LocaleToCharsetMap.java	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MimeMap.java	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ApacheConfig.java	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
JservConfig.java	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ErrorHandler.java	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL	18	8	0	18	8	0	18	8	0	18	90	0	18	90	0	18	92	2

Experimented with other code smells as well.

Problem–up experiences

- Manual validation
  - Time intensive
  - Need for a high level representation
  - ®Can graphs help us out?
- Change rationale
  - Why is not always as obvious
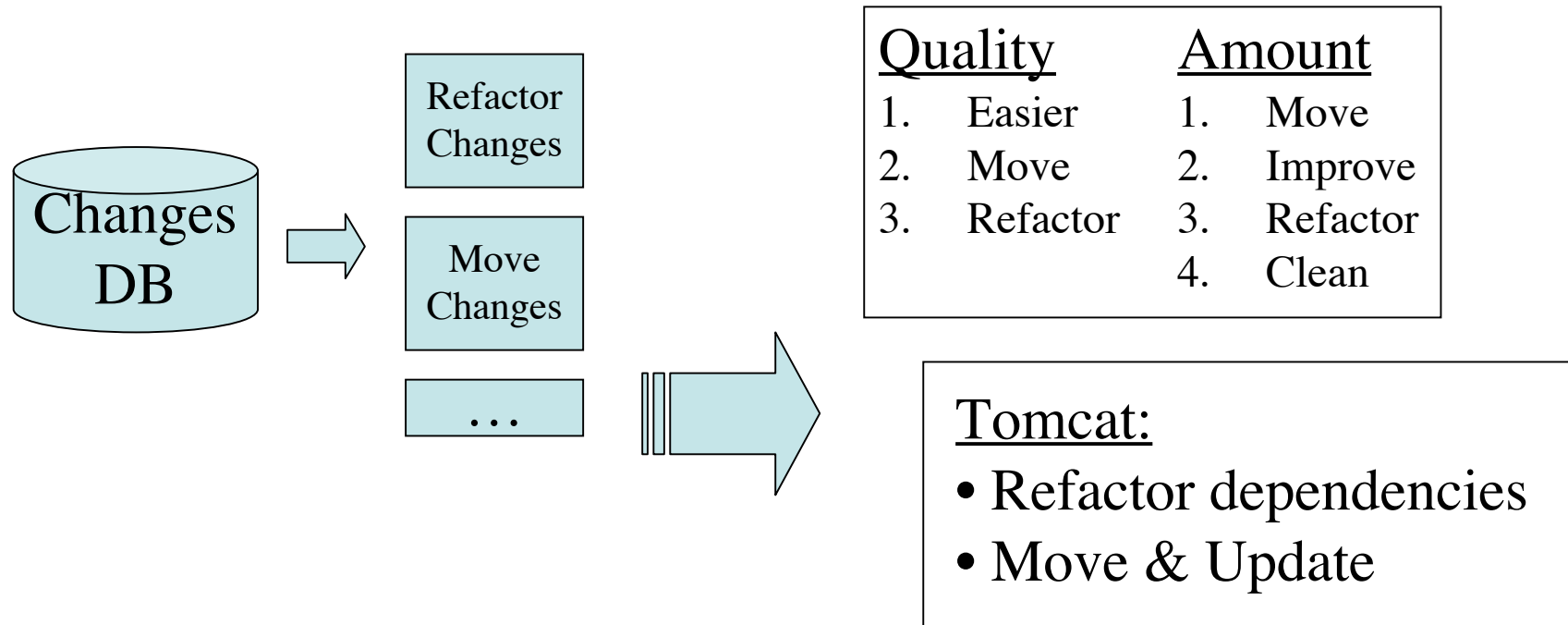 - Link with additional info! Developers!

Evolution-up

Evolution–up experiences

- ➢ Can we put it in a number?
  - – Change rate?
  - – Patterns?
- ➢ Suitable to study the effect of …
  - – A change?
  - – A code smell/problem?
- ➢ Links to the rationale
  - – Learns us more about the why!
- ➢ Still have to study actual changes
  - – High-level change representation!

Rationale-up

Changes DB → **Refactor Changes** / **Move Changes** / **…**

Quality		Amount	
1.	Easier	1.	Move
2.	Move	2.	Improve
3.	Refactor	3.	Refactor
		4.	Clean

Tomcat:
- Refactor dependencies
- Move & Update

project	MRs	refactorMRs	percentage
ArgoUML	8127	210	5%
Gaim	9906	4	0,04%
Jakarta-Tomcat	4098	60	1,40%
Jboss	7437	63	0,84%
PostgreSQL	20013	21	0,10%

Rationale–up Experiences

- ➢ Differences in CVSquality
  - – Quick assessment of messages
- ➢ Branching is a problem
- ➢ High level change view!
- ➢ Rationale!

Conclusion