SQUAD: Software Quality Understanding through the Analysis of Design

Foutse Khomh and Yann-Gaël Guéhéneuc

Motivation

Understanding the impact of the design on the quality of systems

Problem

Building quality models that takes into account explicitly the design of systems

Solution

Study the relations between design patterns, design defects, code smells and changes, issues and Bugs

Antipatterns	1.0	2.0	2.1.1	2.1.2	2.1.3	3.0	3.0.1	3.0.2	3.2	3.2.1	3.2.2	3.3	3.3.1
AntiSingleton	0.03	0.30	< 0.01	0.03	< 0.01	0.17	0.96	0.34	< 0.01	< 0.01	0.23	< 0.01	0.08
Blob	< 0.01	< 0.01	0.16	< 0.01	0.71	0.05	< 0.01	< 0.01	0.23	0.01	0.16	< 0.01	< 0.01
ClassDataShouldBePrivate	0.09	< 0.01	0.23	< 0.01	< 0.01	< 0.01	0.79	0.89	0.49	< 0.01	0.05	0.03	< 0.01
ComplexClass	0.01	< 0.01	< 0.01	< 0.01	0.69	< 0.01	0.12	0.34	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
LargeClass	0.06	0.56	0.06	0.04	0.93	0.76	0.97	0.87	0.14	0.28	0.91	0.72	0.71
LazyClass	0.09	0.12	< 0.01	< 0.01	< 0.01	< 0.01	0.02	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
LongMethod	0.02	0.82	< 0.01	< 0.01	< 0.01	0.10	< 0.01	0.37	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
LongParameterList	< 0.01	< 0.01	0.12	0.24	< 0.01	< 0.01	0.09	0.34	0.06	< 0.01	0.87	< 0.01	< 0.01
SpaghettiCode	0.95	0.56	0.36	0.96	0.93	0.93	0.92	0.92				0.92	0.92
SpeculativeGenerality	0.08	0.48	0.08	0.14	0.63	0.31	0.16	0.11	0.16	0.05	0.19	0.27	0.03
SwitchArayKnife	0.08	0.79	0.01	0.01	0.89	< 0.01	0.01	0.01	0.01	0.15	0.68	0.38	< 0.01

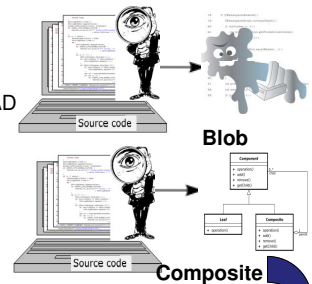
Logistic regression results for the correlations between change-proneness and kinds of antipatterns

Antipatterns	1.0	2.0	2.1.1	2.1.2	2.1.3	3.0	3.0.1	3.0.2	3.2	3.2.1	3.2.2	3.3	3.3.1
AntiSingleton	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
Blob	0.59	0.35	0.24	0.02	0.20	0.77	< 0.01	< 0.01	0.85	0.27	0.06	0.08	0.20
ClassDataShouldBePrivate	< 0.01	< 0.01	0.96	< 0.01	0.26	0.16	< 0.01	< 0.01	0.26	0.48	0.96	0.29	0.36
ComplexClass	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
LargeClass	0.35	0.27	0.37	0.34	0.27	0.37	0.46	0.39	0.26	0.71	0.34	0.37	0.27
LazyClass	0.35	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
LongMethod	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
LongParameterList	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	0.27	0.25	0.10	0.56	0.92
SpaghettiCode	0.97	0.97	0.28	0.98	0.94	0.98	0.96	0.93				0.98	0.96
SpeculativeGenerality	0.79	0.35	0.14	0.94	0.91	0.10	0.02	0.18	< 0.01	< 0.01	0.38	0.16	0.36
SwitchArayKnife	0.49	0.78	0.39	0.30	0.39	0.64	0.42	0.02	0.27	0.05	0.28	0.81	< 0.01

Logistic regression results for the correlations between issue-proneness and kinds of antipatterns

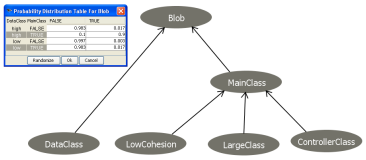
Data extraction with the frameworks SAD and PTIDEJ

- Computing design defects and code smells on systems with SAD
- Computing design patterns on Systems with PTIDEJ

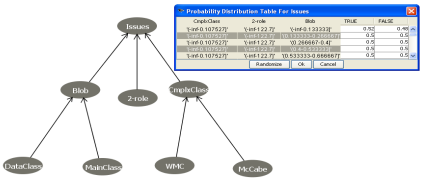


Building models for the prediction of quality, change, issues, and bug proneness

- Building models to predict design defects on the basis of metrics

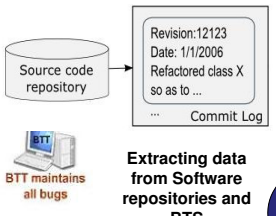


- Building models to predict changes, issues and bug-proneness using Information on metrics, design patterns, design defects and code smells



Data extraction with the framework IBDOOS

- Extracting change data of systems from CVS and SVN.
- Extracting issues and bugs data of systems from Bugzilla and others



Extracting data from Software repositories and BTS

Dates	Releases	Number of				
		LOC	Classes	Changes	Issues	UBE
2001-11-07	1.0	781,480	4,647	21,553	2,208	166
2002-06-27	2.0	1,249,840	6,742	26,378	3,307	204
2003-06-27	2.1.1	1,797,917	8,730	10,397	2,211	165
2003-11-03	2.1.2	1,799,037	8,732	11,534	1,923	133
2004-03-10	2.1.3	1,799,702	8,736	15,560	3,137	309
2004-06-28	3.0	2,260,105	11,166	17,582	798	113
2004-09-10	3.0.1	2,268,058	11,192	24,150	4,225	248
2005-03-11	3.0.2	2,272,852	11,252	49,758	10,000	1,061
2005-06-20	3.2	3,271,516	15,153	2,745	550	111
2006-09-21	3.2.1	3,284,732	15,176	11,854	4,078	229
2007-03-12	3.2.2	3,286,300	15,184	10,682	2,137	153
2007-06-25	3.3	3,752,212	17,162	7,386	1,822	244
2007-09-21	3.3.1	3,756,164	17,167	40,314	14,915	792
Total	13	31,379,975	151,939	243,903	51,371	4,018

OCEAN: sOftware Change and Evolution ANalysis

Salima Hassaine, Yann-Gaël Guéhéneuc and Sylvie Hamel

Motivation

Software repositories such as versioning systems are used to help manage the progress of software projects. Analyzing these repositories can help to support predictions about software development, and to plan various evolutionary aspects of software projects.

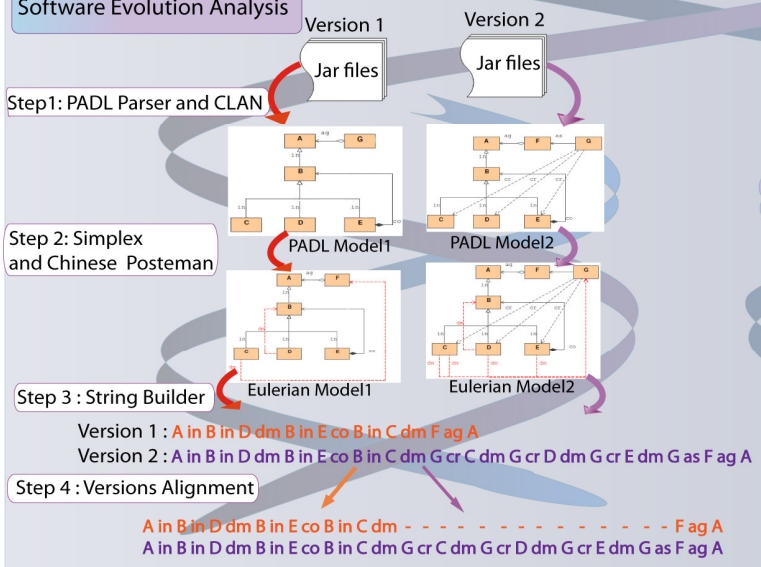
Problem

The complexity and the amount of data requires the use of efficient techniques for version comparisons.

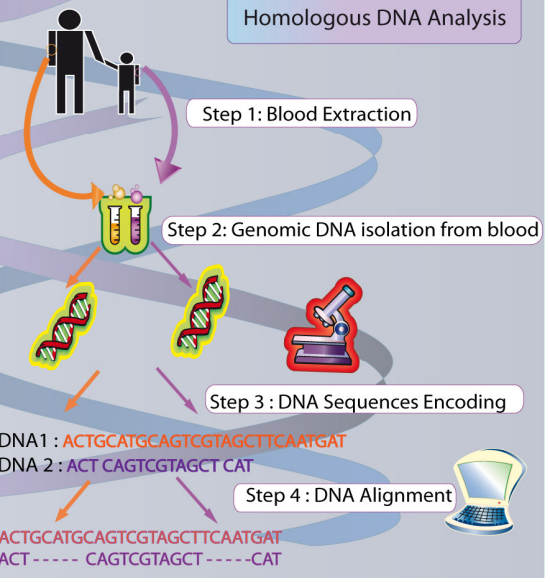
Solution

We propose a metaphor of homology to study the evolution of a program.

Software Evolution Analysis



Homologous DNA Analysis



Université de Montréal

ÉCOLE POLYTECHNIQUE MONTRÉAL

Ptidej — Pattern Trace Identification, Detection, and Enhancement in Java

Pattern Trace Identification, Detection, and Enhancement in Java