# An Empirical Study on Evolution of Open Source JavaScript Projects

Shahriar Rostami
Everton da S. Maldonado

Goal

- Study evolution of JavaScript projects

  - LOC, directory structure, functions, complexity, etc.

- Community Contribution

  - Number of Developers, number of reported issues and the time issues take to fix.

- Study of code smells related to JavaScript

  - With different groups of severity (Critical, Major, Minor, etc).

Project selection → Define set of metrics → Analysis → Discuss results
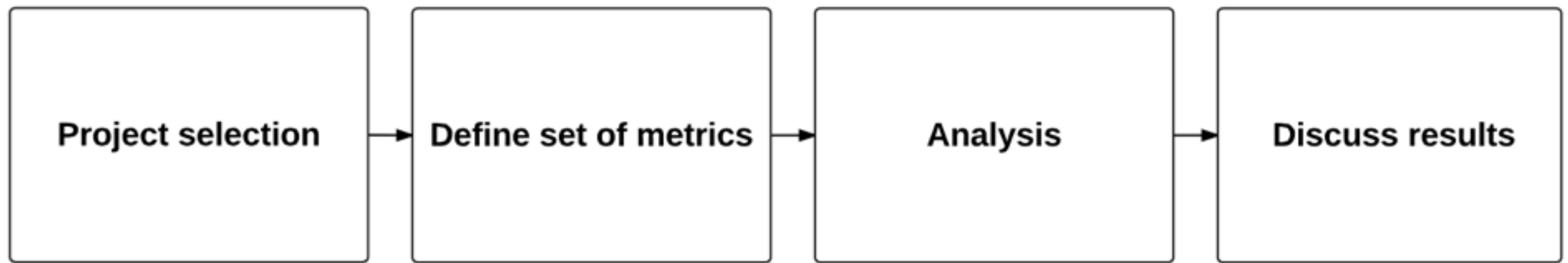
- 15 open source JavaScript projects

- Total 57 years of evolution

- 1065 releases

- 63473 commits

- 67787 submitted issues on Github issue tracker

Projects

| Project Name	First Release	Last Release	#Releases	#Commits
Coffee Script	2010	2015	31	3967
Less.js	2010	2015	38	2428
NPM	2010	2015	**298**	5509
Mongoose	2010	2015	197	4789
Underscore	2010	2015	30	1985
Node-mysql	2010	2015	38	914
Q	2010	2014	48	3359
Request	2011	2015	32	1644
Ember.js	2011	2015	87	9150
Source-map	2011	2015	29	**396**
Bootstrap	2011	2015	27	11195
Mocha	2011	2015	80	1702
Brackets	2011	2015	60	**15871**
Bower	2012	2015	65	1813
Grunt	2013	2014	**5**	1309

Metrics that we gather to study evolution

- **Lines of code:** The number of uncommented lines of code

- **Complexity:** McCabe complexity

- **Comment Lines:** Number of lines containing either comment or commented-out code.

- **Duplicated Lines(%):** Density of duplication = Duplicated lines / Lines * 100

- **Number of Directories**

- **Number of Files**

- **Code Smells:** Number of code smells with severity of blocker, critical, major and minor

Differences in first and last release

- Most growth in terms of <u>LOC</u> belongs to Brackets with total 260,520 lines added from first release in compare to last release.

- The general trend for <u>comment lines</u> is increasing, however CoffeeScript shows decreasing from first release with 836 lines to 124 lines in last release.

- Brackets starts with 14 <u>directories</u> and in the last release it contains 179 directories.

  - We observe that source organization grows as the size of project grows. It means the more code added to the project, the more directories and sub directories needed to organize them.

- Based our result, we found that the number of <u>statements, functions and cyclomatic complexity</u> increase as the lines of code increase.
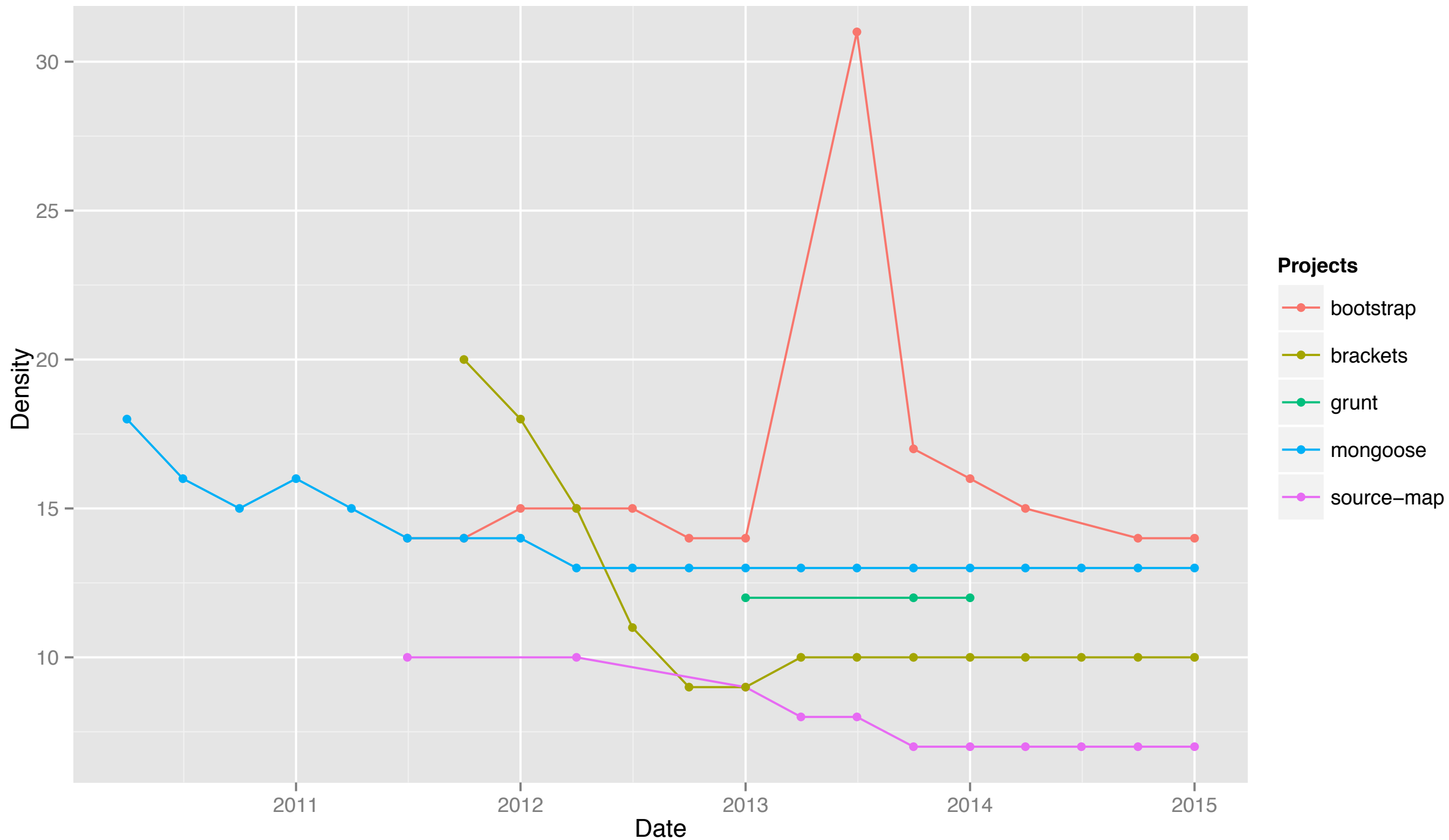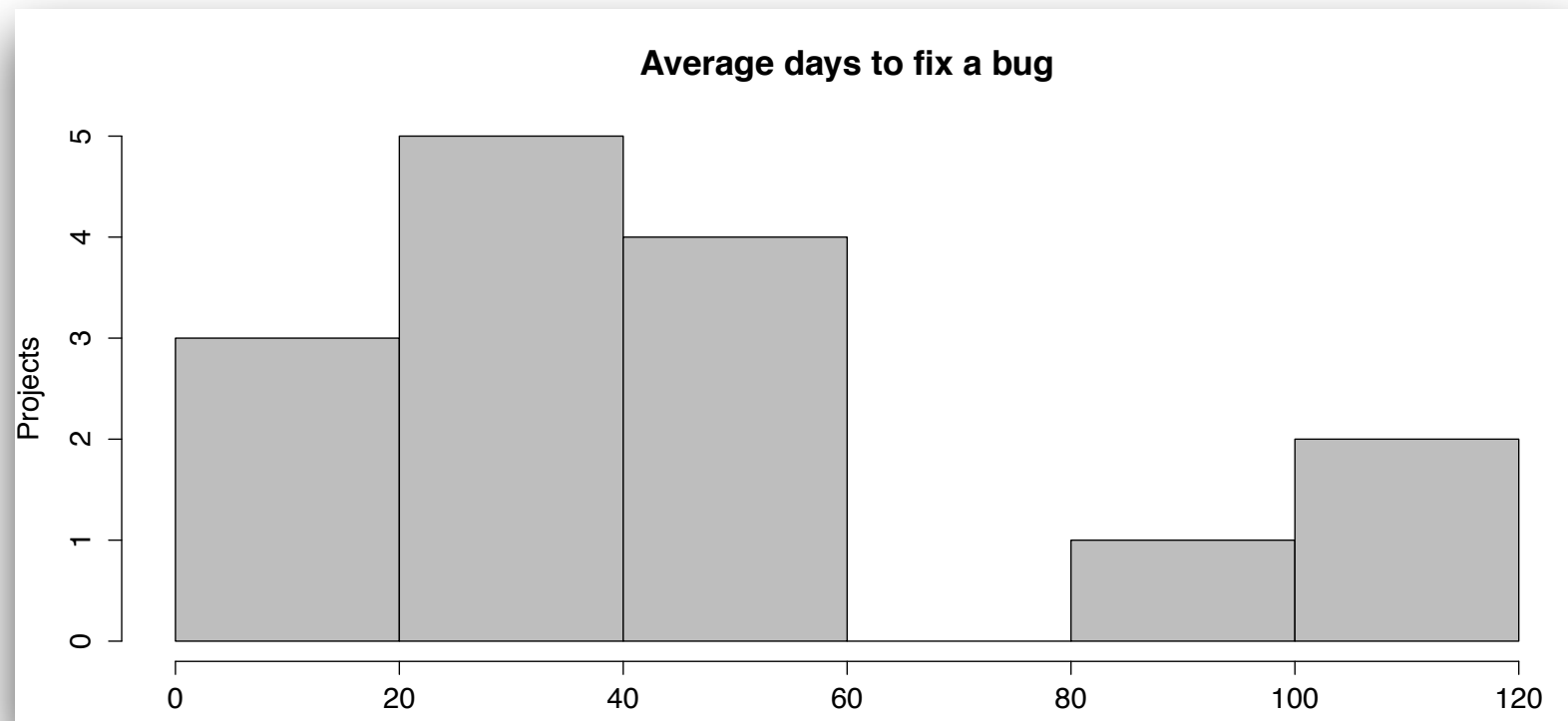
of Developers vs # of Issues
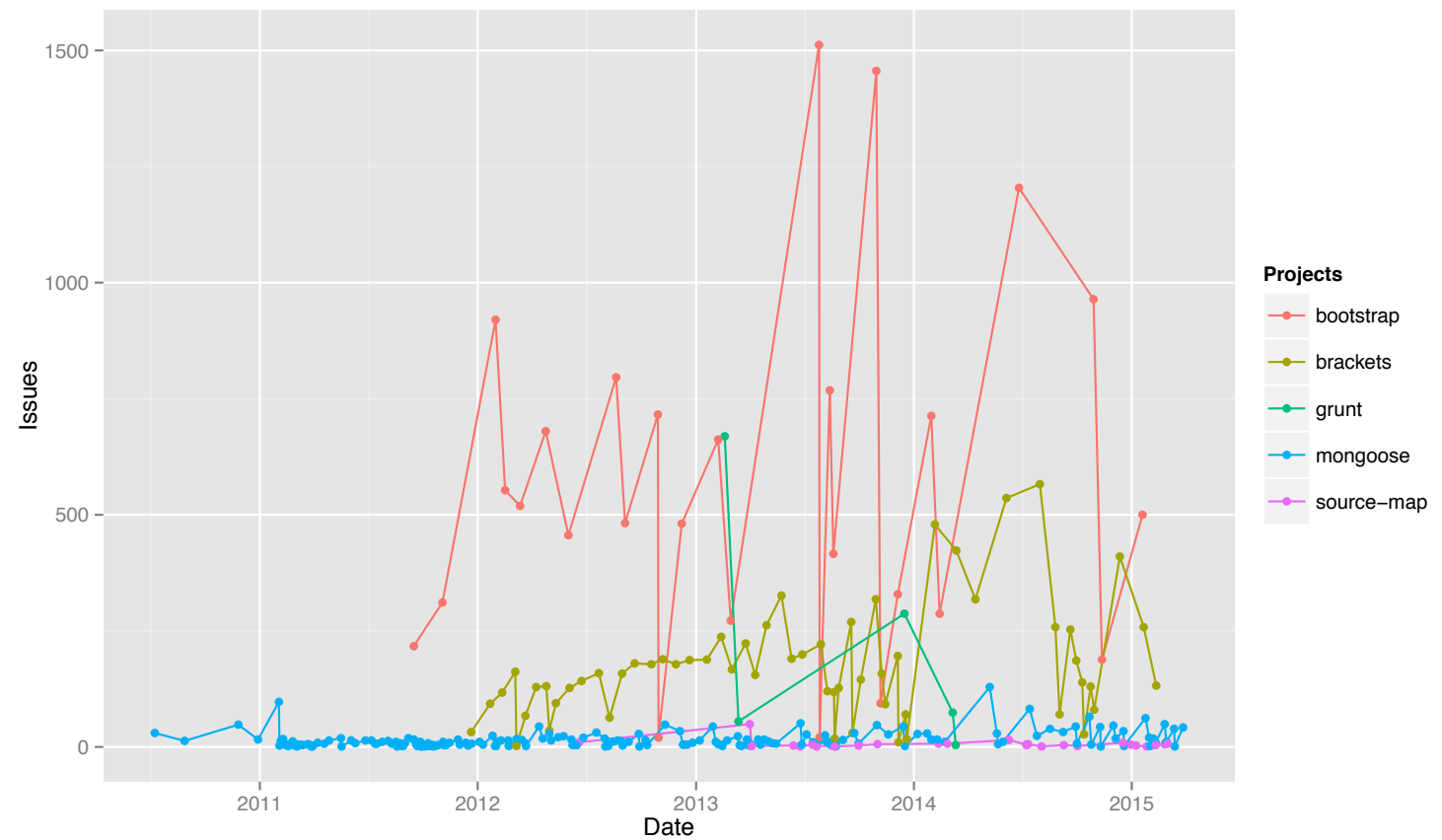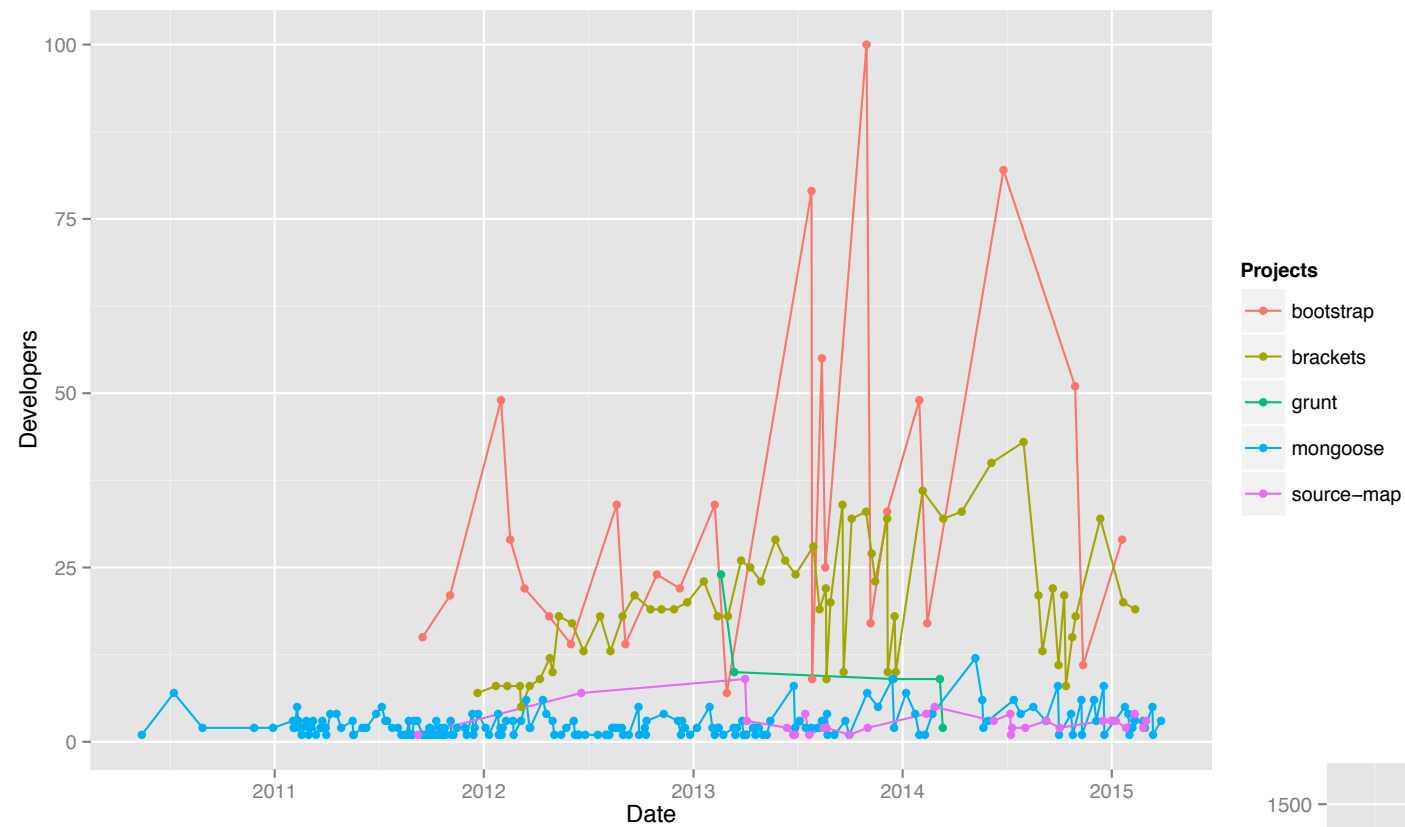
Average days to fix a bug

Project Name	Days
Coffee Script	46
Less.js	113
NPM	58
Mongoose	44
Underscore	14
Node-mysql	47
Q	30
Request	105
Ember.js	22
Source-map	17
Bootstrap	11
Mocha	83
Brackets	26
Bower	30
Grunt	28

Average days to fix a bug

Code Smells in JavaScript

Critical

continue should not be used:
continue is an unstructured control flow statement. It makes code less testable, less readable and less maintainable. Structured control flow statements such as **if should be used instead**.

Major

switch statements should end with a **default** clause
The requirement for a final default clause is defensive programming. The clause should either take appropriate action or contain a suitable comment as to why no action is taken.

Minor

Statements should be on separate lines:
For better readability, do not put more than one statement on a single line.

Projects

Project Name	Major Bad Smells		Minor Bad Smells	
	First Release	Last Release	First Release	Last Release
Coffee Script	1692	3528	31	48
Less.js	336	4285	24	317
NPM	1300	12535	10	49
Mongoose	316	4773	4	161
Underscore	465	1000	10	19
Node-mysql	230	479	10	72
Q	15	319	1	41
Request	115	3798	2	20
Ember.js	8469	1489	586	343
Source-map	91	2686	5	7
Bootstrap	523	4777	7	49
Mocha	233	1675	2	76
Brackets	1167	40985	35	5248
Bower	169	226	1	44
Grunt	137	155	11	10

Questions?