

Code Churn

YEGOR BUGAYENKO

Lecture #13 out of 24

80 minutes

All visual and text materials presented in this slidedeck are either originally made by the author or taken from public Internet sources, such as website. Copyright belongs to their respected authors.

+2pt



“We can measure the increase or decrease in system complexity as measured by a selected metric, code delta, or we can measure the total amount of change the system has undergone between builds, code churn.”

— *Code Churn: A Measure for Estimating the Impact of Code Change*, Sebastian G. Elbaum and John C. Munson, Proceedings of the International Conference on Software Maintenance (ICSM), 1998

Motivating Example

Commit #1:

0

```
1 class Book
2     private final int id;
3     public Book(int it)
4         this.id = i;
```

Commit #2:

+5

```
1 class Book
2     private int id;
3     public Book(int it)
4         this.id = i;
5     public int getId()
6         return this.id;
7     private int setId(int i)
8         this.id = i;
```

Commit #3:

+2 / -2

```
1 final class Book
2     private final int id;
3     public Book(int it)
4         this.id = i;
5     public int getId()
6         return this.id;
```

Delta: +3 / 0, Churn: +7 / -2

Code Delta vs. Code Churn

| BUILD | RCM | DELTA | CHURN |
|-------|-----------|---------|---------|
| 1 | 184643.38 | 5.72 | 7.12 |
| 2 | 184648.81 | 5.43 | 16.78 |
| 3 | 185702.10 | 1053.28 | 1547.93 |
| 4 | 185857.61 | 155.50 | 187.11 |
| 5 | 186236.58 | 378.97 | 429.30 |
| 6 | 186261.81 | 25.23 | 294.18 |
| 7 | 186083.21 | -178.60 | 791.87 |
| 8 | 186726.70 | 643.50 | 836.71 |
| 9 | 187072.37 | 345.65 | 426.24 |
| 10 | 187157.08 | 84.71 | 86.74 |
| 11 | 189945.91 | 2788.85 | 2804.02 |
| 12 | 190000.81 | 54.89 | 59.28 |
| 13 | 190007.10 | 6.29 | 6.38 |
| 14 | 190012.49 | 5.39 | 5.48 |
| 15 | 190069.36 | 56.87 | 75.08 |
| 16 | 190066.48 | -2.88 | 3.86 |
| 17 | 190067.65 | 1.17 | 1.17 |
| 18 | 190067.81 | 0.16 | 0.21 |

Table 2: Evolution Data for QTB

“A limitation of measuring code deltas is that it doesn’t give an indicator as to how much change the system has undergone. If several software modules are removed and are replaced by modules of roughly equivalent complexity, the code delta for the system will be close to zero.”

Source: *Code Churn: A Measure for Estimating the Impact of Code Change*, Sebastian G. Elbaum et al., ICSM, 1998

+2pt



“Our case study provides strong support for the following conclusion: increase in relative code churn measures is accompanied by an increase in system defect density.”

— *Use of Relative Code Churn Measures to Predict System Defect Density*, Nachiappan Nagappan and Thomas Ball, Proceedings of the International Conference on Software Engineering (ICSE), 2005

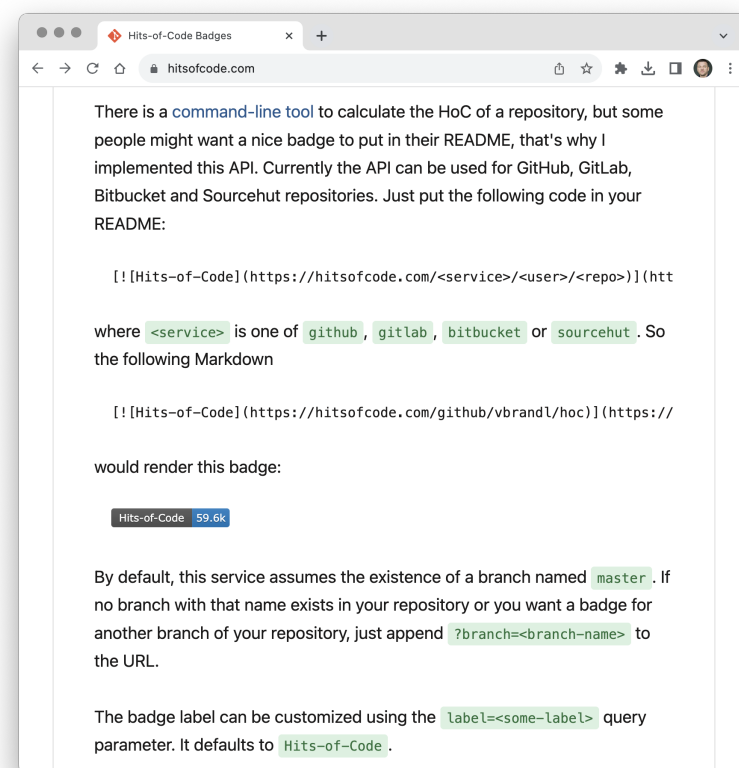
+2pt



“We can conclude that committing files with higher cumulative code churn values (i.e. those of longer change history) is more likely to result in negative maintainability change.”

— *Cumulative Code Churn: Impact on Maintainability*, Csaba Faragó, Péter Hegedűs, Rudolf Ferenc, Proceedings of the International Working Conference on Source Code Analysis and Manipulation (SCAM), 2015

HitsOfCode.com



Created by Valentin Brandl
(@vbrandl) and inspired by the
blog post of mine: Hits-of-Code
Instead of SLoC (2014)

Read this:

,

Hits-of-Code Instead of SLoC (2014)