Standard Grimoire Report
AIMA-PYTHON Project
15-01-18-06

June 6, 2018

This report would not exist without the effort of the people involved in the development of the Grimoire toolset.

Executive Summary

This report provides a quantitative analysis of the current and past situation of the AIMA-PYTHON project. All the data presented in it is based on information retrieved from the software development repositories of the project. The analysis includes a summary of the general situation of the project, and specific analysis of some of its development processes and communication channels. Data from previous periods is also shown for comparison.

Contents

1 Project overview

The report looks at activities across the AIMA-PYTHON community during 15-01-18-06 (2018-03-06 to 2018-06-05), comparing it to previous period of analysis.

Data source	Activity last quarter	Change (wrt to prev. quarter)
github_issues	0 Closed tickets	-100%
github_issues	0 Opened tickets	-100%
github_prs	0 Closed reviews	-100%
github_prs	2 Submitted reviews	-250%
git	0 Commits	-100%

Table 1: Activity during the last period of analysis and its evolution

Table 1 shows development activity for each of the analyzed data sources. The activity column displays information about the net activity numbers, while the Change column displays information about the relative difference with respect to the previous period of analysis.

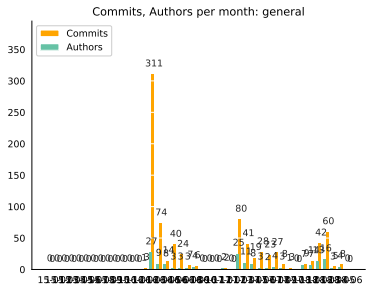
The rest of the document is divided into three sections with information from the last periods:

- Activity: focused on contributions.
- Community: focused on contributors.
- Process: focused on efficiency and timing.

2 Activity

This section covers contributions in the different data sources.

The bar chart below shows the evolution of the number of commits and authors in Git through time, grouped by quarters.

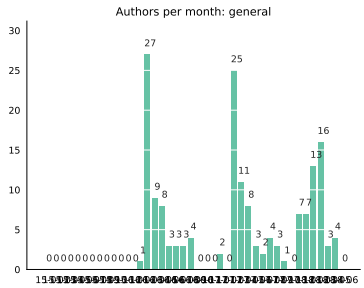


Period	Commits	Authors
15-01	0	0
15-02	0	0
15-03	0	0
15-04	0	0
15-05	0	0
15-06	0	0
15-07	0	0
15-08	0	0
15-09	0	0
15-10	0	0
15-11	0	0
15-12	0	0
16-01	0	0
16-02	3	1
16-03	311	27
16-04	74	9
16-05	14	8
16-06	40	3
16-07	24	3
16-08	7	3
16-09	6	4
16-10	0	0
16-11	0	0
16-12	0	0
17-01	2	2
17-02	0	0
17-03	80	25
17-04	41	11
17-05	19	8
17-06	28	3
17-07	23	2
17-08	27	4
17-09	8	3
17-10	3	1
17-11	0	0
17-12	9	7
18-01	14	7
18-02	42	13
18-03	60	16
18-04	5	3
18-05	8	4
18-06	0	0

3 Community

This section tries to help us to understand the evolution of AIMA-PYTHON community by looking at active contributors and organizations in the last period of analysis, compared to previous ones.

Number of active authors in Git is shown below, giving us a quick look of contributors evolution in the last quarter compared to previous ones.



Period	Active Authors
15-01	0
15-02	0
15-03	0
15-04	0
15-05	0
15-06	0
15-07	0
15-08	0
15-09	0
15-10	0
15-11	0
15-12	0
16-01	0
16-02	1
16-03	27
16-04	9
16-05	8
16-06	3
16-07	3
16-08	3
16-09	4
16-10	0
16-11	0
16-12	0
17-01	2
17-02	0
17-03	25
17-04	11
17-05	8
17-06	3
17-07	2
17-08	4
17-09	3
17-10	1
17-11	0
17-12	7
18-01	7
18-02	13
18-03	16
18-04	3
18-05	4
18-06	0

In addition, table below offers a quick glance to the most active authors in Git in the whole period of time shown in the bar chart above.

Author	**Commit (s)**
Aman Deep Singh	5
AdityaDaflapurkar	1
DKE	1
tbcdebug	1

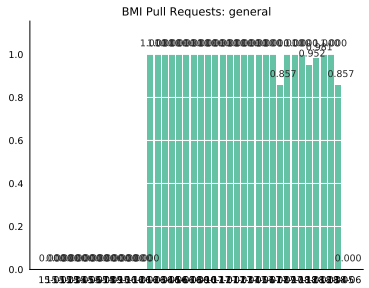
In a similar way, table below shows the same information groped by organization instead of author.

Organization	**Commit (s)**
Unknown	8

4 Process

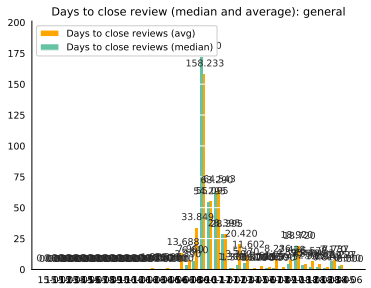
This section intends to show the evolution of efficiency and timing when dealing with tasks related with code review processes.

To measure the efficiency of the process of software development in GitHub first we focus on BMI, that was defined in section 1. Chart below shows evolution of BMI by quarters, allowing us to check how pull requests are being managed by the community in the last quarter compared to previous ones.



Period	Closed/Subm.
15-01	0
15-02	0
15-03	0
15-04	0
15-05	0
15-06	0
15-07	0
15-08	0
15-09	0
15-10	0
15-11	0
15-12	0
16-01	0
16-02	0
16-03	1.00
16-04	1.00
16-05	1.00
16-06	1.00
16-07	1.00
16-08	1.00
16-09	1.00
16-10	1.00
16-11	1.00
16-12	1.00
17-01	1.00
17-02	1.00
17-03	1.00
17-04	1.00
17-05	1.00
17-06	1.00
17-07	1.00
17-08	1.00
17-09	0.86
17-10	1.00
17-11	1.00
17-12	1.00
18-01	0.95
18-02	0.98
18-03	1.00
18-04	1.00
18-05	0.86
18-06	0.00

Besides, next chart deals with timing. It shows mean and median times to merge pull requests in GitHub (TTM, defined in section 1)–in days–for last quarter compared to previous ones.



Period	Median	Mean
15-01	NA	NA
15-02	NA	NA
15-03	NA	NA
15-04	NA	NA
15-05	NA	NA
15-06	NA	NA
15-07	NA	NA
15-08	NA	NA
15-09	NA	NA
15-10	NA	NA
15-11	NA	NA
15-12	NA	NA
16-01	NA	NA
16-02	NA	NA
16-03	0.46	1.38
16-04	0.20	0.28
16-05	0.60	1.39
16-06	0.00	0.00
16-07	0.00	13.69
16-08	3.39	7.96
16-09	6.61	33.85
16-10	172.06	158.23
16-11	54.29	55.10
16-12	63.29	64.54
17-01	28.40	28.40
17-02	1.30	1.30
17-03	3.39	20.42
17-04	5.37	11.60
17-05	0.60	1.14
17-06	0.70	2.61
17-07	1.02	1.86
17-08	0.90	8.24
17-09	0.59	1.74
17-10	4.60	7.44
17-11	18.92	18.72
17-12	3.85	4.42
18-01	2.36	6.58
18-02	1.68	4.48
18-03	0.81	2.15
18-04	7.17	7.74
18-05	2.75	3.23
18-06	NA	NA

A Metrics Definitions

- Commit: this is defined as the action(s) that performes a change in the source code. Bots, merges and other type of automatic activity is removed from the records. In addition, when aggregating several git repositories, this metric only counts unique revisions (unique hashes found in the git repositories). In addition, all branches are aggregated to the analysis.

- Submitted changesets: a code review is the process of peer reviewing source code changes. A submitted code is not merged to the master code of a given project till this is approved. A submitted code review is defined as any changeset submitted to the Gerrit system.

- Authors: a developer is defined as author if she is the owner of the patchset sent for reviewing and this is merged into the source code. As previously indicated, automatic commits such bot's are removed from this analysis.

- Efficiency closing changesets: this metric is a derivation of the Backlog Management Index as it is named as Review efficiency index (REI). As similarly used in the BMI index, this metrics measures the number of closed changesets out of the total number of new changesets in a given period.

- Time to Merge: this time consists of the time between the first upload of the first changeset till the last iteration of the code review process is merged into the code. This metric is provided in number of days.

- Developer per period: average of developers per period ignoring bots and merges.

- Emails sent: number of emails sent by people to the several mailing lists. Bots are not registered.

- People sending emails: number of people sending those emails ignoring bots.