Usefulness Determinants

. . . of Open Source Projects

Mikael Svahnberg[1]

2018-11-07

[1] Mikael.Svahnberg@bth.se

Background

How to make decisions successfully for efficiently using software components or services from different sources to develop competitive software-intensive systems in relation to the trade-off between functionality, time to market, cost, quality, legacy and risk?

One year ago the focus was on

- the decision *process*
- decision *attributes*
- decision *environment*

D. Badampudi, K. Wnuk, C. Wohlin, U. Franke, D. Smite, and A. Cicchetti, “A Decision-making Process-line for Selection of Software Asset Origins and Components”, Journal of Systems and Software, Available, 135, pp. 88-104, 2018.

The Developer's Perspective

Goal

Select the best open source component for use in a development project.

Input for Questions

P. Chatzipetrou, E. Alégroth, E. Papatheocharous, M. Borg, T. Gorschek and K. Wnuk, “Component selection in Software Engineering - Which attributes are the most important in the decision process?”, In Proc. of the 44th Euromicro Conference on Software Engineering and Advanced Applications, 2018. *Distinguished paper*

Priority	Attribute
1	Cost
2	Support of the Component
3	Longevity Prediction
4	Level of off-the-shef fit to product
5	API adequacy
6	Code Quality
7	Access to Relevant Documentation
8	Adherence to Standards
9	Programming Language Performance
10	Complexity
11	Size
12	Other

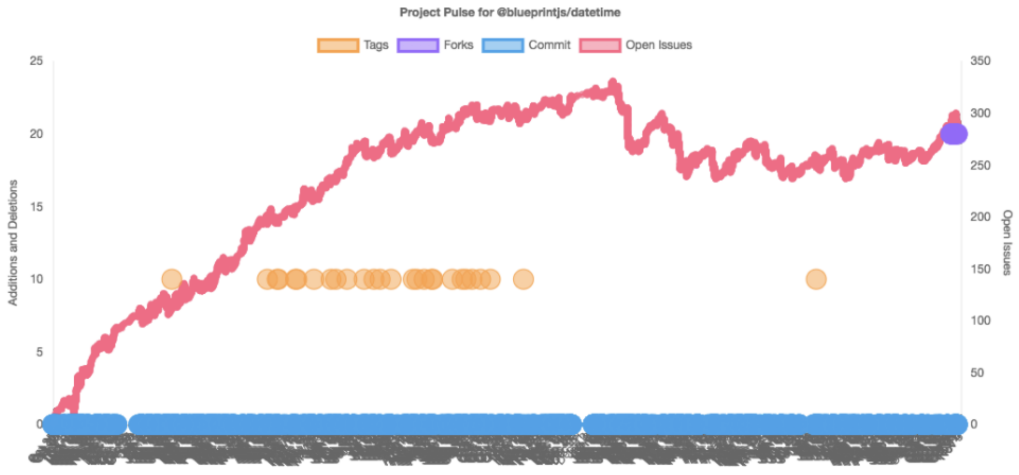
Attributes / Questions

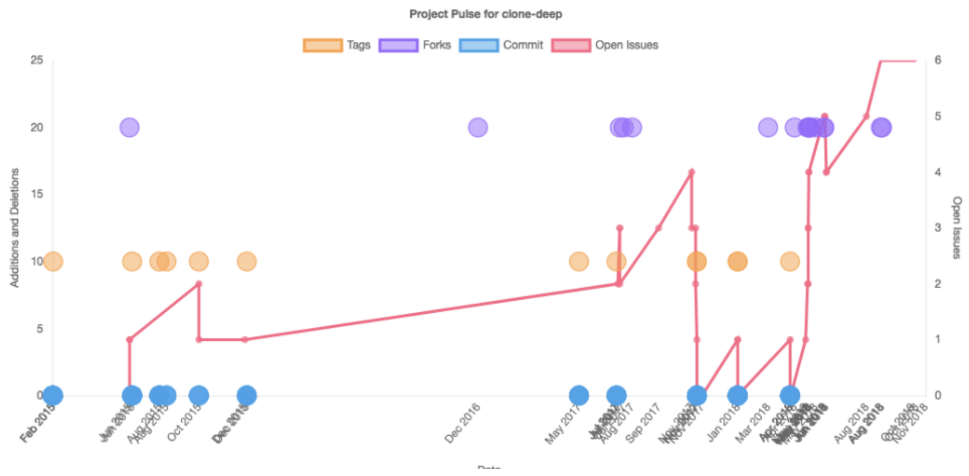
Attribute	Questions
Cost	What is the cost of the component?
Support of the Component	What support channels exist?
Support of the Component	What is the amount of support available?
Support of the Component	How quickly can support be obtained?
Longevity Prediction	How long has the component existed?
Longevity Prediction	How likely is the component to continue existing?
Longevity Prediction	How likely is the component to continue being developed?
Level of off-the-shelf fit to product	How much customisation is required to use the component?
Level of off-the-shelf fit to product	How much of the required functionality is already supported by th
API adequacy	What is the maturity of the external API?
Code Quality	What is the current test status?
Code Quality	What code revew practices are the developers of the component
Access to relevant documentation	What documentation is available?
Adherence to standards	To what extent does the component adhere to relevant standards
Programming Language Performance	What programming language is the component written in?
Complexity	What is the complexity of the code in the component?
Size	What is the code size of the component?
Size	What is the memory footprint of the running component?

Metrics

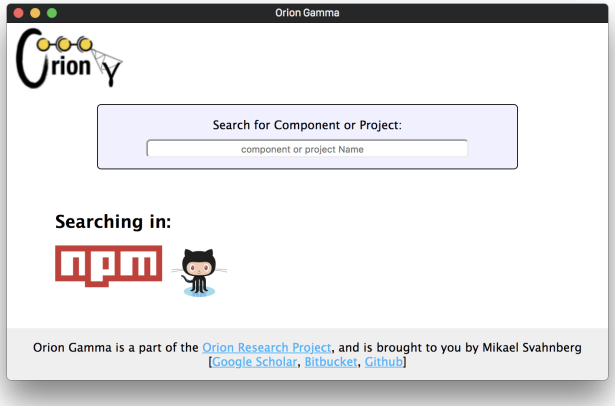
Questions	Metrics
What support channels exist?	Activity on Github Issue Tracker (last 6 months)
What is the amount of support available?	Ratio of opened vs closed issues last 6 months
	Amount of closed issues last 6 months
How quickly can support be obtained?	Average closing time of issues last 6 months
How long has the component existed?	Project creation date
How likely is the component to continue existing?	COMPOUND: Commits && Closed issues && creat
How likely is the component to continue being developed?	COMPOUND: commits && closed issues

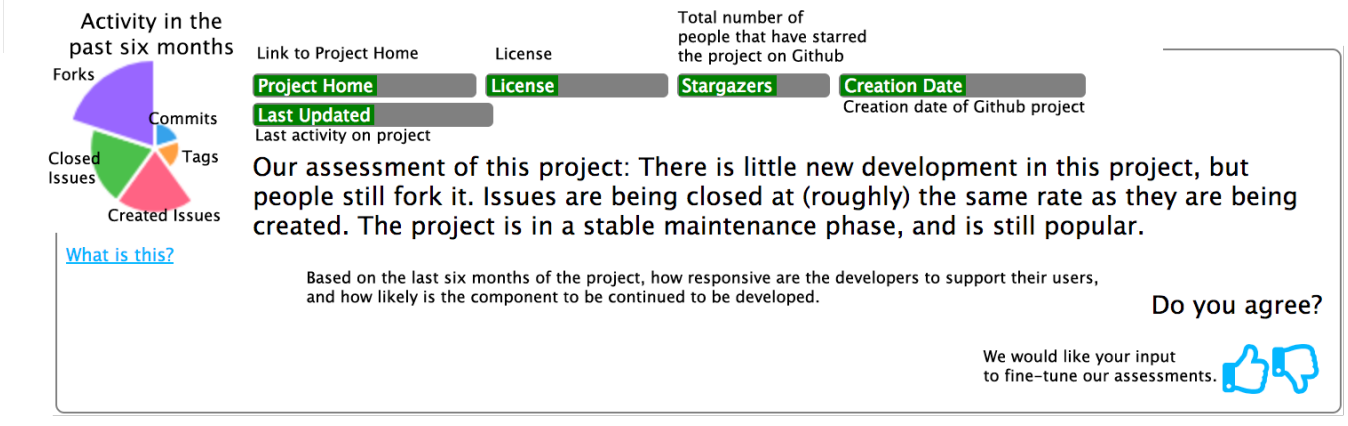
Two Example Projects





Search tool


Orion Gamma
Search for Component or Project:
component or project Name
Searching in:
Orion Gamma is a part of the Orion Research Project, and is brought to you by Mikael Svahnberg
[Google Scholar, Bitbucket, Github]


Activity in the past six months
Forks
Commits
Tags
Closed Issues
Created Issues
What is this?
Link to Project Home
License
Total number of people that have starred the project on Github
Project Home
License
Stargazers
Creation Date
Creation date of Github project
Last Updated
Last activity on project
Our assessment of this project: There is little new development in this project, but people still fork it. Issues are being closed at (roughly) the same rate as they are being created. The project is in a stable maintenance phase, and is still popular.
Based on the last six months of the project, how responsive are the developers to support their users, and how likely is the component to be continued to be developed.
Do you agree?
We would like your input to fine-tune our assessments.

Collecting Feedback

- “Our assessment of this project” is based on classification into a project archetype.
- We may be wrong, we collect input to refine our classification:

Live Demo

`http://msv-nuc00.dap.bth.se:8088/`

Next Steps

- ☐ Pilot Static Validation
- ☐ Industry Static Validation
- ☐ Industry Validation
- ☐ Determine frequency of different project archetypes.
- ☐ Investigate other metrics, *easily harvested*, that can enable software developers to make more informed decisions.
  - Wash, rinse, repeat.