Formalizing metrics for the CHAOSS

June 26, 2018

1 Definitions

- A **software system** is a set of**entities** who collaborate together to create or modify **assets** that form a software product.
- **Assets** are files created as part of the software system being developed. These could have source code, documentation, test-code, etc.
- **Events** are (recorded) activities in the software development project.
- **Entities**. Examples of entities are either organizations supporting a project, software developers, etc. Entities could be related to each other. Entities might be only identifiable by inspecting events.
- **Measureable property**. A measurable property of an entity or asset is a feature/property of the entity/asset that can be measured.
- In general a **metric** is a function that converts a **measurable property** of a entity/asset/event (or sets of) into a value. In general, the type of a metric can be any. For example, a single value (categorical –type of a file; discrete–SLOCs of the entire system, continuous –avg size of files in the system, etc.), or a vector/set of values (the entities who reviewed a given pull-request).
- A **snapshot metric** When the metric applies to an asset/entity/event, it is necessary to specify the timestamp when the measurement is to be made. Some events might be immutable (e.g. a commit), in that case the timestamp is irrelevant. In the absence of a timestamp, the metric is performed at of today. Hence a metric is parameterized with a timestamp.

2 Aggregation of metrics

Frequently metrics need to be aggregated. This aggregation happens in two dimensions: assets (e.g. what is the total number of SLOCs in the project), and time-based (how many developers have participated in the project since its conception?)

2.1 Asset based aggregation

Some asset-based aggregation is defined in the metric itself. In this case the metric applies to a set of assets. Otherwise a metric an be defined as an aggregation of other metrics if that is possible. For example, the total number of SLOCs of a project can be defined a an aggregation metric of the sum of the SLOCs of every file in the system.

2.2 Time based aggregation

Timebased aggregation implies that we want to aggregate a metric at a particular interval. This interval is defined by two dates (beginDate, endDate). In this case, the metric is computed at the beginDate, endDate, and a function is applied to the two resulting values.

For example, we can compute the number of commits per release by first identifying the dates of each release of the system, and then counting the commits in the system at each of these dates. The aggregation in this case is simply the number of commits at the endDate (the aggregation function ignores the number of commits at the beginning date and simply returns the number commits at endDate).

We could compute the increase in number of commits by altering the aggregation function such that it returns the difference between the beginDate and endDate.

- The result of timebased aggregation is a time-series, e.g. a ordered list of pairs (date, metric).
- **In the absence of beginDate, endDate**, a time-based aggregation considers the entire life of the project
- **In the absence of beginDate**, a time-based aggregation considers the period that starts when the project begins and ends in the specified endDate

2.2.1 Aggregation of events

When applied to events, timebased aggregation measures events that happened during its time window.

3 Aggregation of aggregated metrics

Some metrics can be further aggregated. For example, if we have computed the aggregated metric: “Number of commits per day”, we can easily compute the “number of commits per week” or “number of commits per month”.