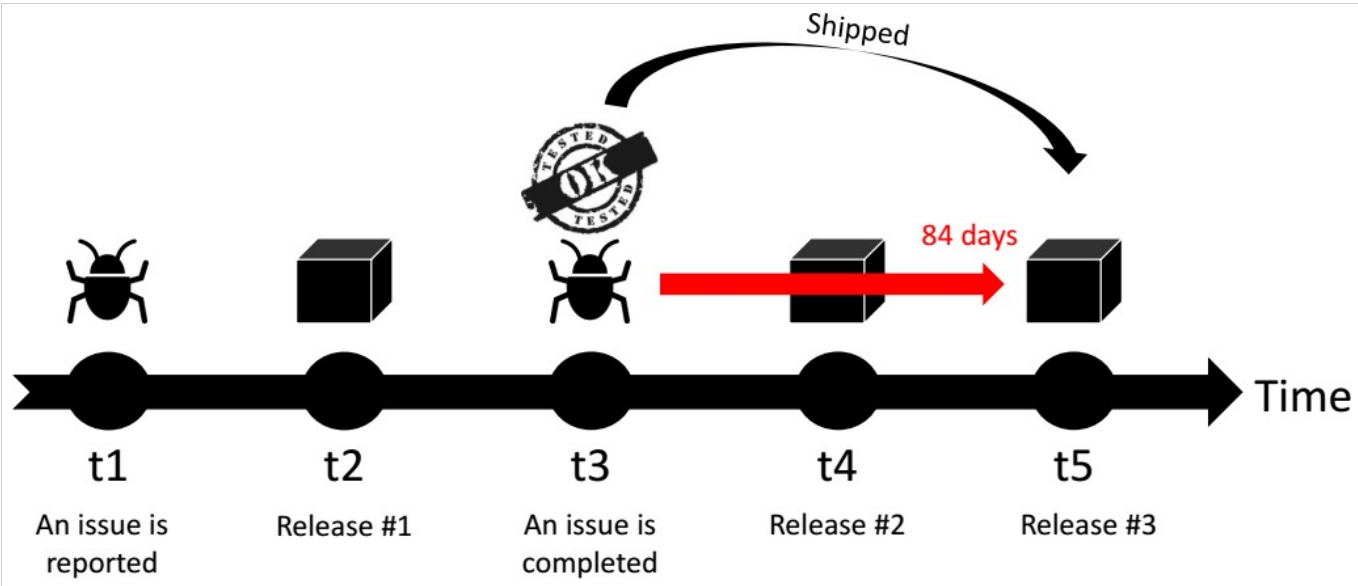


# How do we compare rapid vs traditional releases?

In this page, we explain how we measure the data that is shown in page 6 of our survey. You can find the concepts that are necessary to understand the data collection process below.



**Delivery delay** measures how long it takes for a system functionality (i.e., an issue) to be delivered to the end user from the time at which the issue was completed (i.e., implemented and tested).

**Delivery delay in terms of days** is the number of days for an issue to be officially shipped after it is completed. The figure above illustrates an issue that is completed at time  $t_3$ . This issue takes 84 days to be shipped at  $t_5$ .

By "official release" we mean a release that is intended to be used by the entire user base of the project. For example, in a pipelining release strategy (e.g., as in the Firefox project), in which a release is stabilized through several channels, an official release is the final product of the process, i.e., the release that is to be published to every user from the *release channel*.

In the figure below, we show the **delivery delay in terms of days** for the completed issues in the Firefox project. We collected data from traditional releases (major and minor releases from version 1.0 to 4.0) and from rapid releases (releases in the *release channel* from version 10 to 27). The Figure shows a **beanplot** for each release strategy. The vertical curves of beanplots compare the distributions in traditional and rapid releases. The higher the frequency of data within a particular value, the thicker the bean is plotted at that particular value on the y axis. Finally, the black horizontal line represents the median value of each distribution. We observe that the median number of days to deliver is significantly higher with the rapid release cycle, but there is much less variation.

