

TEST AUTOMATION METRICS – MASHING UP NON-TEST DATA

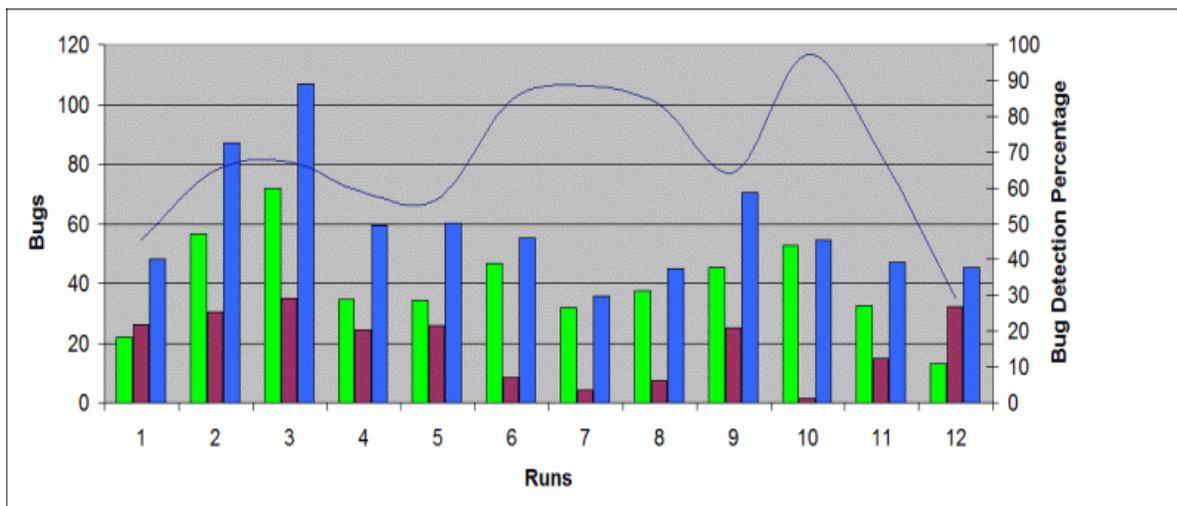
In my [previous post](#) I wrote about reporting on test automation and one of the main things I said in there was: use the vast amount of data you have. In this post I will elaborate a bit more on that, in fact, I am going to go one step beyond and say: don't *just* use your own data! Generally in an organisation there is more data at hand than just the data you create with test automation. Some very powerful pieces of data are release-boards and bug-metrics.

Bug metrics

The bug-metrics are quite often a logical item to look at. However, keep in mind that generally speaking, test automation will not have a very high bug-detection-percentage. The bug-detection-percentage curiously enough is something you do not often read about, but it is being measured all over the place actively; it is the amount of bugs found in production set off as a percentage of the amount of bug found before production and

in production, in other words:
$$BDP = \left(\frac{preProductionBugs}{preProductionBugs + ProductionBugs} \right) * 100$$
 An

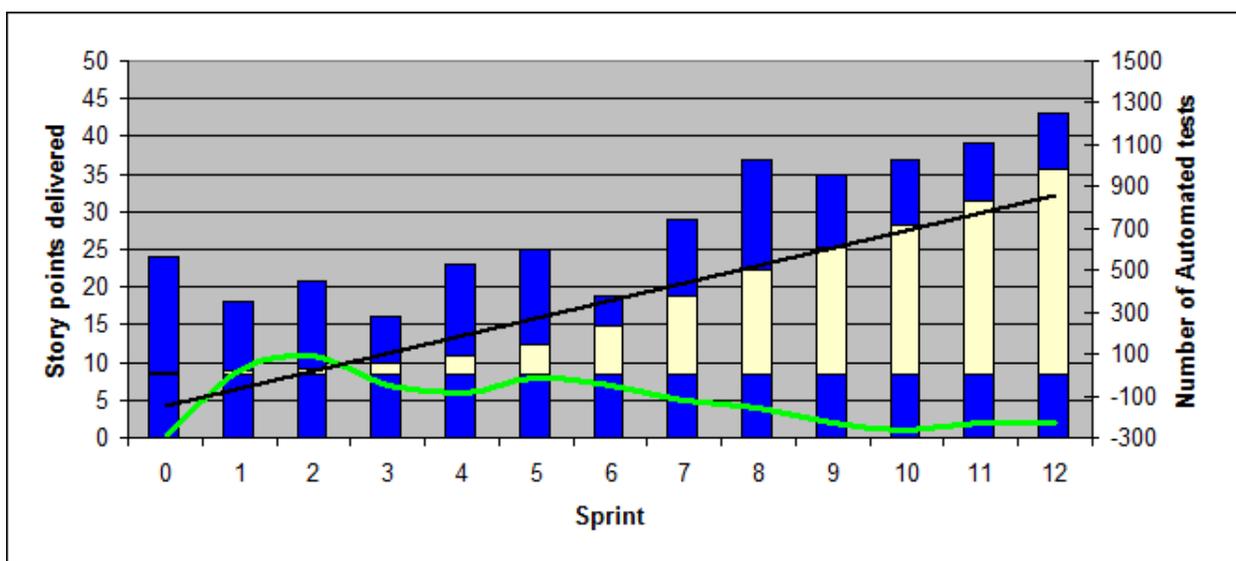
example of a BDP graph, with Preproduction found bugs, bugs found in production and the grand total of bugs found per run (*please keep in mind, this is simply sample data and not from an actual project*)



The problem I see in using the BDP to show the effectiveness of test automation is that the objective of test automation is generally *notto* find new bugs, but instead to ensure all existing functionality, on a new release, functions as expected and thus as it was on previous builds or releases. So in short, test automation is excellent at running a regression test, but not at finding new problems. Of course it's possible to then see if you can define a bug detection percentage for test automation specifically, e.g. make the numbers solely based on test automation. This however will not really give any insights into the effectiveness of test automation considering the low amount of bugs found by test automation in general. What can we do with bug data and test automation? Frankly, I am convinced that generally speaking test automation will not benefit from setting metrics based on bugs. Use other data instead, such as new features introduced, lines of code changed, number of tasks or user stories completed and set that off against your test set showing growth.

Test automation and User Stories

In agile environments it is generally considered (and thankfully so) a good thing to automate the crap out of everything being built. Generally there is quite some data available on the amount of stories or tasks coming out of a sprint, or on the story-points having been committed to per sprint. A graph setting off the story-points (or complexity points if you will) against the amount of tests having been automated can give a nice insight into the “coverage” of your automation. Add to this set the amount of bugs appearing in production and you have a sweet showcase of how your agile effort is performing over all.



A graph like this contains a lot of information so it will need a bit of explanation when showing to the team and especially to managers. On a per sprint basis this graph shows:

- Blue bars: the amount of story points delivered per sprint (the actual delivery, not the committed amount!) set off against the left vertical axis
- White bars: the cumulative amount of automated tests run on these stories and run as regression set off against the right vertical axis.
- Black line: a trend line of the overall amount of automated tests
- Green line: the actual amount of production issue occurrences after releasing a sprint worth of code to production.

A graph like this is relatively easy to setup and maintain and it will show a very strong picture. It clearly shows the progress of the team effort (growing amount of story points taken up, less production issues over time) and at the same time gives a very clear insight into the added value of test automation within the team by showing the rising coverage against the declining production issues.

Source : <http://martijndevrieze.net/2012/06/08/test-automation-metrics-mashing-up-non-test-data/>