

MEASURING SOFTWARE PROJECT HEALTH DURING QUALIFICATION PHASE IN 4 STEPS

By Eric Mariacher

Abstract

"Modern" software development companies use a change management methodology and an associated tool to manage change requests or bugs between developers and testers.

In CMMI V1.2, this is addressed in Process and Product Quality Assurance (PPQA) process area, specifically to "SP 2.1 Communicate and Ensure Resolution of Noncompliance Issues" where typical work products are "Evaluation reports" and "Quality trends".

To get a feeling of where a software project is heading in terms of schedule, quality and performance, while in testing phase, an obvious practice is to look at how many bugs are opened. But what you get is a snapshot, you don't know if the project is in a stabilization phase or if it is out of control.

I am going to concretely present a 4 steps method of gathering, presenting and interpreting data extracted from any bug tracking tool.

This method will help to answer the following questions:

- When will we be able to release a software with a good level of quality?
- Are we on the right track to complete software qualification on time?

step 1: Prerequisites

A bug has 3 states: open, under_verification, closed

To get an efficient view of a software project health through bug statuses, it is a good practice to match bug states with current owners. Basically:

- open bugs are currently under engineering/development responsibility.
- under_verification bugs have been corrected by engineering and are currently owned by testers who perform some validation of delivered fixes.
- closed bugs do not belong to anyone: they are just inactive.

Read one of my other papers on how to design an efficient change request flow.

Both testers and developers shall use change request tracking tool to manage all bugs

Change request tracking tool shall be used not as the sole mean of communication between testers and developers but mainly as a tracking tool in order for all stakeholders (developers, testers and management) to have a clear view of who owns what.

The method, which is going to be presented gives satisfactory information provided, there are at least more than 30 bugs managed during project's testing phase.

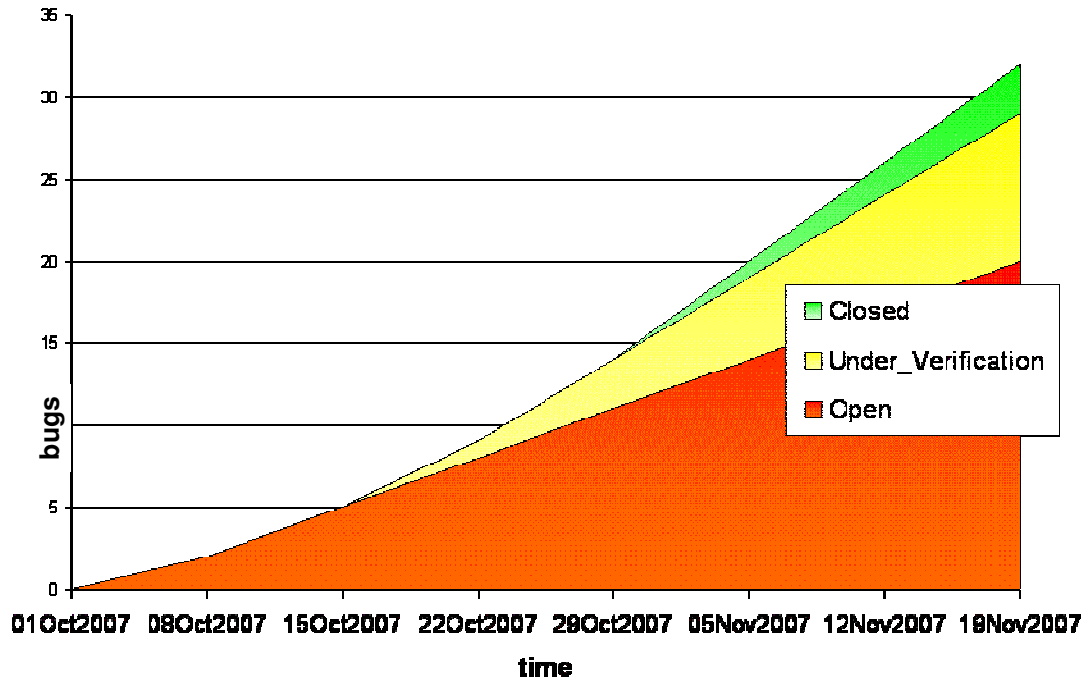
step 2: Every week build an excel sheet with the data presented in the following way

date	Open	Under_Verification	Closed
01Oct2007	0	0	0
08Oct2007	2	0	0
....			
28Jan2008	6	8	30
01Feb2008	0	0	44

Each row contains the date and current bugs status in 3 columns: open, under_verification and closed. The table should grow over time with an additional row added every week.

step 3: Build the curve

Use stacked curve graphics. The graphic should look like the figure below.

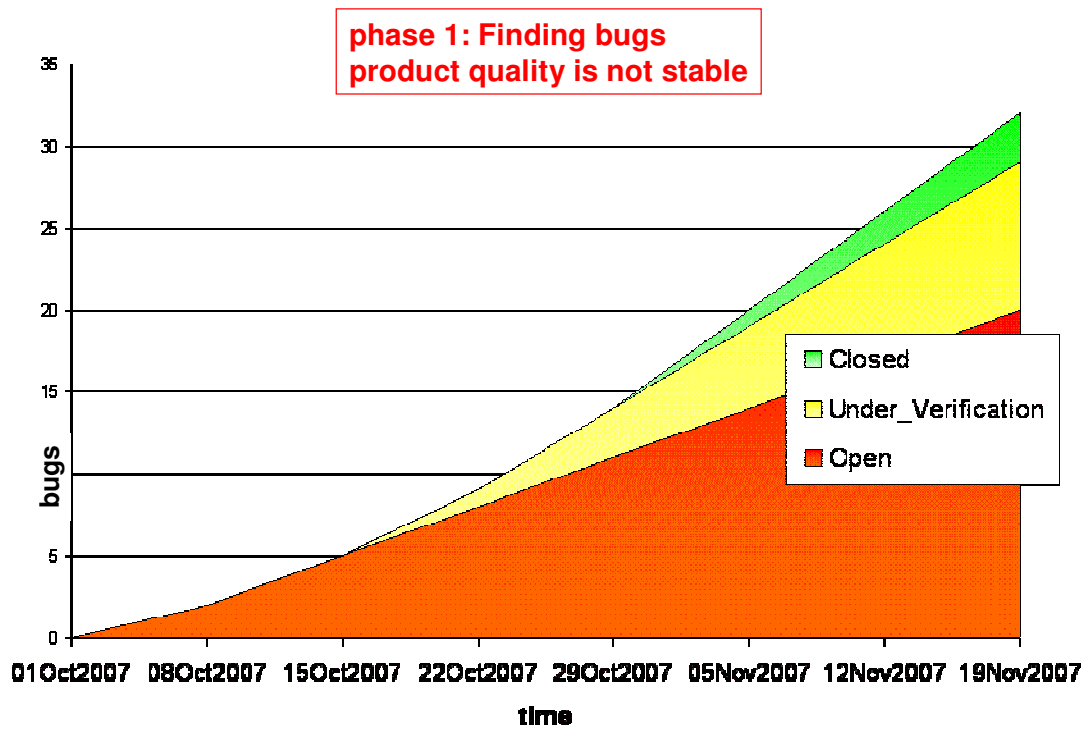


step 4: Interpret trends shown by the curve

Interpretation cannot start before having at least 3 weeks of project qualification history. Trends will be presented as they will be discovered through time with data building up week by week.

The overall bug count trend (sum of open, under_verification and closed bug) over time will be interpreted because it is the main predictor of the project health during qualification phase.

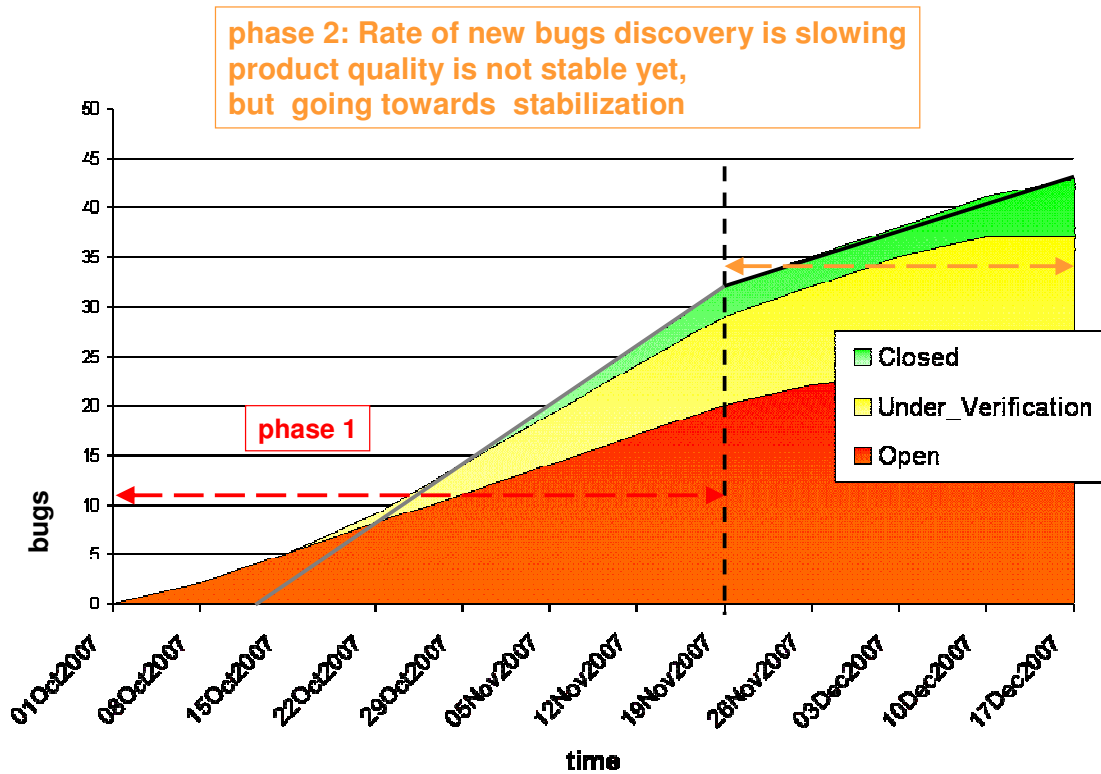
phase 1: Finding bugs - product quality is not predictable



Product just entered test phase. Every week new bugs are regularly found and logged by testers. Developers start to solve bugs found by testers. Developers released software to testers for bug regression checking.

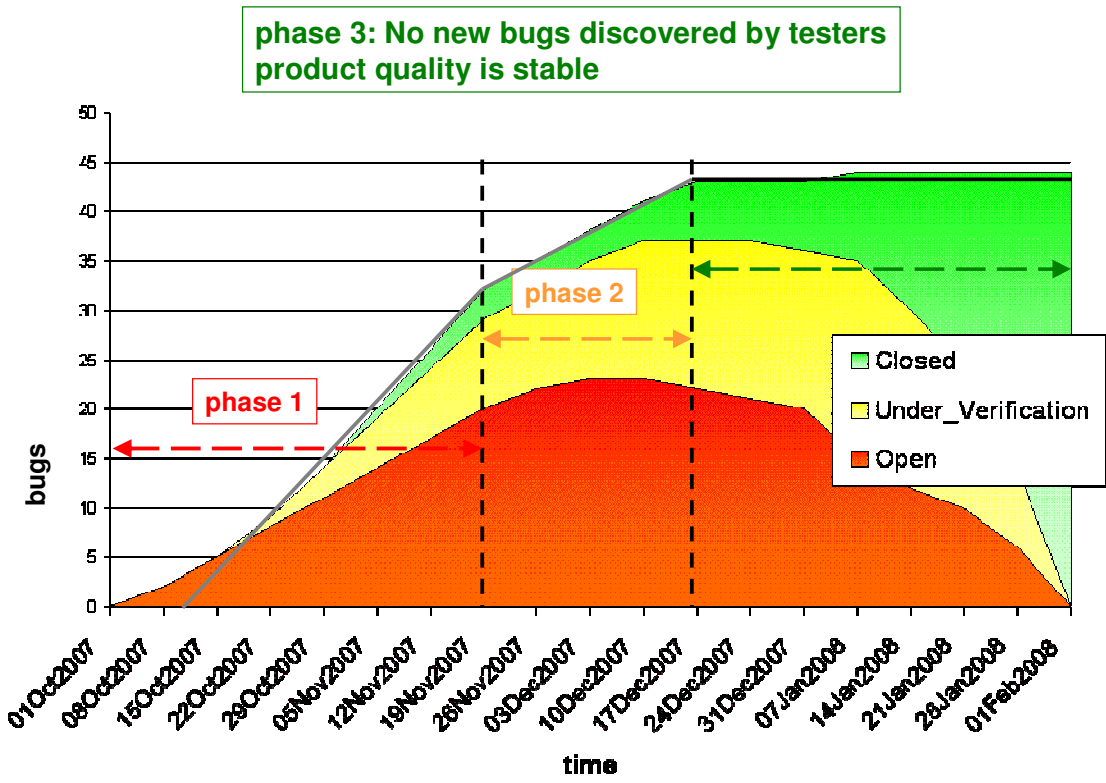
Over time new bugs discovery is steady or even increasing. The total number found on this product cannot be evaluated during this phase: product quality is not predictable.

phase 2: Rate of new bugs discovery is diminishing - product quality is not stable yet, but going towards stabilization



Now testers are finding less bugs every week. If this trend is sustained over time, it is then possible over time to get a good idea of the overall number of bugs associated with this project i.e. how many bugs will be found during qualification phase.

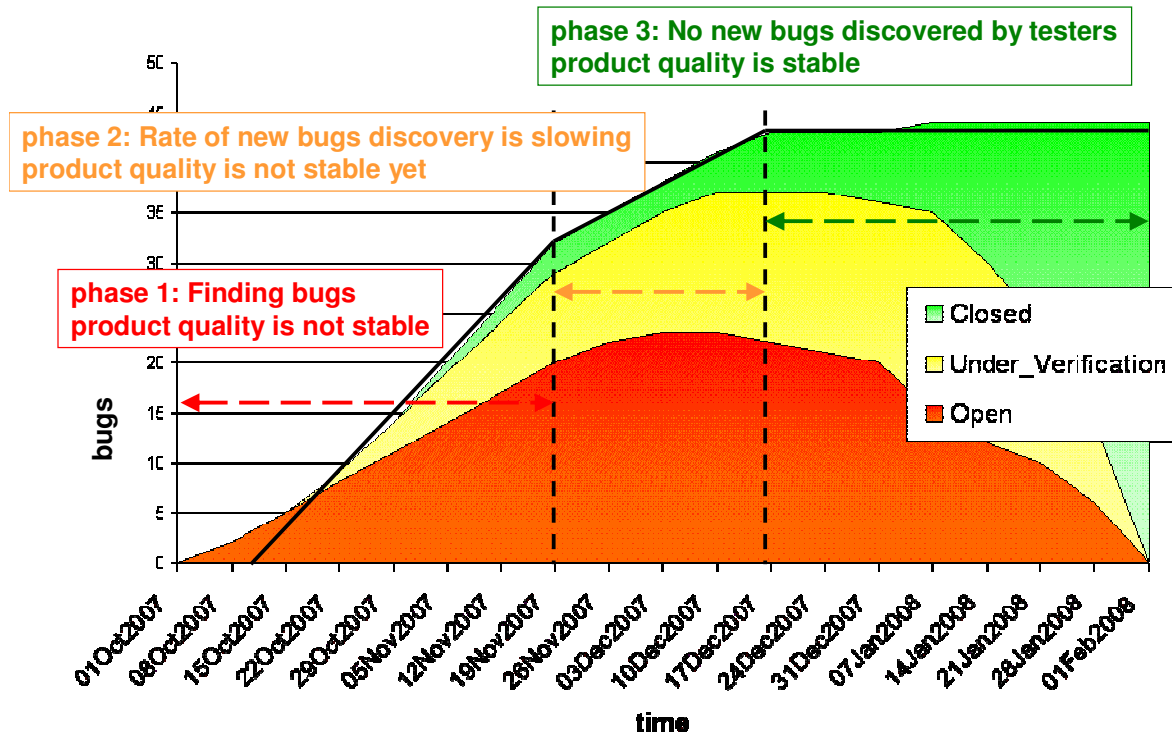
phase 3: No new bugs discovered by testers. - product quality is predictable



All bugs have been discovered/identified. Developers are now correcting bugs.

conclusion

Software projects goes through 3 phases during qualification.



As written previously: the main predictor is the overall bug count trend. open and under_verification sub-curves are just enabling teams management not project management. By "enabling teams management", I mean that we can see if, for instance, testers are not checking bug fix as fast as they are corrected by development team.

A good sub-practice is to split open and/or under_verification data per bug severity, to discriminate between important and nice to have bugs.

About the Author

Eric Mariacher has held various positions in embedded software development, coder, architect, project manager at IBM and now functional manager for wireless keyboards, mice and receivers at Logitech. These projects are of different sizes beginning with 8Kbytes of code running in 8bits microcontrollers coded by one programmer to projects requiring tens of developers, several megabytes of code and thousands of files on 32bits microcontrollers and PC computers.

Eric Mariacher main centers of interest are to set-up the right environment for software developers to deliver products, and also to give management the best visibility on software development process from requirement phase to field support phase. Setting up the right software development often requires reengineering software development processes. Eric Mariacher enjoys driving these software development processes reengineering measured following the CMMI best practice.

Eric Mariacher can be reached at eric.mariacher@gmail.com or eric_mariacher@logitech.com .