OpenTestCenter
Application product specifications
v0.a ECM 10/13/2012

OpenTestCenter Application Purpose

Given the OpenTestCenter open database schema designed to provide tracking of tests and defects as related to requirement and story artifacts, provide a flexible, scalable, open-source application which transparently solves the problem of QA engineers being forced to use expensive, obtuse tools in order to:

- author and track requirements and stories,
- author and track the tests written to cover those stories,
- author and track the bugs that result from testing,
- maintain basic metadata

and derive meaning from metrics that measure

- the relationships and causal relationships between development and testing artifacts,
- tell the story of how a release is shaping up,
- indicate the snapshot and synthesis to-date of the quality of a given build,
- the strong and weak areas of a product over time, and
- custom metrics of interest to the team

OpenTestCenter can and must

(1)   **provide base functionality derived from principles of real testing** that solves the problems testers need solved in order to do their job effectively,
(2)   **be completely transparent to the tester** so that the tester may adjust and tweak the application to their needs, and
(3)   **outperform** existing commercial products.

OpenTestCenter Application Goal

Provide a smart, adaptable, modular application that works with the OpenTestCenter data layer in order to provide logic and programmable interfaces to

- accept programmatic and manual queries and CRUD assertions to artifacts (acceptance criteria, stories, tests, bugs, associations between entities, and membership assertions/changes/removals from relationships)
- accept queries and CRUD manipulation of metrics
- provide clean and clear implementation of a base set of QA metrics with an open interface to allow for the programmatic creation of further metrics as a tester sees fit, including:
  - First-order metrics:
    - Tests passed, failed, not run, and anything in between
    - Requirements/Stories/Acceptance Criteria covered by tests
    - Bugs open, closed, and in between
  - Second-order metrics:
    - Build and Release quality based on weighted bug and test states
    - Weighted heatmap of tests indicating quality issues of the system anatomy
    - Weighted heatmap of tests indicating tests that have captured the most useful bugs vs tests that have found zero or useless bugs
  - Third-order metrics:

- Probabilistic likelihood of completion of release delivery to quality standards over time based on past history of builds and
- QA timeline predictor of time needed to test changes to specific areas of system anatomy based on what is being proposed to change, who is doing the testing, and who is developing the change
- Automated testing analysis to show the most valuable return on investment of automating the most time spent on manual testing on highest risk areas of the product.

Stories (High level)

As a product manager, I want to ensure that my requirements management tool can interface with OpenTestCenter via an open API so that I don't have to manually enter in requirements and stories in two tools.

As a <u>tester</u> writing test scripts, I want the capability to interface with OpenTestCenter programmatically so that my tests can create and modify test status event information without manual intervention.

As a <u>test manager</u>, I want to <u>interact with a fast, beautiful, configurable dashboard</u> of pie charts, graphs, and drill down table data so that I can <u>make sense of a synthesis of how testing is going</u> for various products, where <u>testing bottlenecks and other blocks are now</u>, and interact with <u>a radar screen that warns me</u> where future problems may likely be as a function of past results.

As a <u>product manager</u>, I want to <u>interact with a fast, beautiful, configurable dashboard</u> of pie charts, graphs, and drill down table data so that I can <u>view a snapshot and moving synthesis over time of the quality of my product</u> as a whole and <u>my product's individual components now, in the past, and reasonable projections</u> onto my future product timeline.

Please note: no further updates are being made to this document. Further drill-down on stories and tasks can be found in the Pivotal Tracker public project OpenTestCenter, which can be found at this link:

<u>https://www.pivotaltracker.com/projects/670693/stories</u>#