Engineering Quality Cycle | Driving Engineering Best Practices

Thu, 08/28/2008 - 09:16 — Chris

Often we focus on quality when our impact is greatly diminished. Trying to improve the quality of widget at the manufacturing phase is only marginally effective considering a large portion of defects are designed in. This is also true for software and services, If we were to consider quality in the upfront design or creation process our decisions would have a greater impact and cost a great deal less than discovering them further down the chain. In this article, I describe the DFQ/QA/QC/CE model (Engineering Quality Cycle) which is used to build quality into the engineering process.

The Engineering Quality Cycle became evident to me when I was working on Microsoft's Basics program. The purpose of this program was to bundle all the topics/features that weren't functionally oriented but just as important to the customer. They were accounted for by each development team and driven from a number of centers of excellence in varying ways. This included areas such as; Reliability, Performance, Application Compatibility, Feedback, Manageability, Serviceability, etc. My role was to consolidate and drive a 'common' way this was implemented and tracked across all of the Windows engineering teams.

We called this model the Engineering Quality Cycle and it includes four dimensions.

- DFQ - Design for Quality,
- QA - Quality Assurance
- QC - Quality Control
- CE - Customer Experience

Design for Quality (DFQ) is focused on declaring up front what is expected in designing a product / service for a specific area. This usually takes the form of design specifications, policies and/or engineering guidelines.

Quality Assurance (QA) is responsible for designing & executing tests, tools, procedures, sign-offs, etc. that reinforce what is declared in the DFQ guidelines.

Quality Control (QC) is focused on putting in control points that can effectively stop the release of the product / service until a quality expectation is satisfied.

Customer Experience (CE) represents the metrics that are measured to validate that the DFQ guidelines are driving the right outcomes.

Examples using a software company paradigm - - -

DFQ: Using the Manageability area as an illustration; a DFQ guideline was created with instructions on how to develop applications so that they can easily be configured from a common console or upgraded using a common upgrade platform.

QA: Continuing the example, QA included test cases to assure ability to manipulate all of the configurable features via a console of the application. The Manageability team also built tools that scanned the code to assure that configurable variables were included in the manifest and exposed to enable greater manageability.

QC: This was a sign-off at a gate review by the product managers accepting the QA results in assuring a manageable product. If there were any Sev 1 manageability bugs, i.e. test results or behaviors that

Copyright © Ceptara Corp.

didn't meet the DFQ guidelines, they were either fixed or sign-off was required from a higher level authority.

CE: This included feedback metrics that gather the usage of the manageability console assuring the customers find the manageability feature useful.

The Engineering Quality Cycle helped us reinforce quality focused behaviors and set a standard for each team responsible for quality helping them assure that they implemented programs which consistently drove a higher quality product.

- Quality Excellence [1]

Source URL: http://ceptara.com/blog/engineering-quality-cycle

Links:
[1] http://ceptara.com/taxonomy/term/3

Copyright © Ceptara Corp.