

# Requirements-Centric or Change-Centric Development?

Ran Lavi, Methoda

[ranl@methoda.com](mailto:ranl@methoda.com)

Over the several decades of consulting IT and High-Tech organizations, we have realized that organizations deal with product development and maintenance using two main approaches:

**Change-Centric Development** – the organization manages the changes and additions to the product's capabilities (Feature View). There is no updated baseline of the product's latest capabilities. From our experience, this approach is the most common.

**Requirements-Centric Development** – the organization defines and maintains a requirements tree that specifies all of the product's capabilities (Product View), up-to-date product architecture, and a set of test scenarios to test (verify) all the product's latest requirements. This approach is rarely found in small to mid-sized organizations (but more common in development of medical devices and aerospace industry, probably due to regulatory requirements).

Development managers tend to think that change-centric development is easy, simple, and effective, both in the short term and the long term. They tend to think that the requirements-centric approach involves too many unwarranted overheads. Indeed, change-centric development does appear good and simple, but quite quickly (usually after releasing a few versions) the organization realizes that it is faced with a dead end: there is no updated and clear list of all the product's capabilities, and it is difficult to characterize, design and test new changes/additions.

In our lecture, we present the recommended methodology for an organization in relation to these issues, and how the two approaches can be combined efficiently and effectively. We'll also present a practical example of how to implement this methodology using modern tools (using Atlassian tools: Jira & Confluence).

We recommend that the academy will incorporate this subject into software engineering courses, ensure that software engineering students understand the pros and cons of change centric vs. requirements centric approaches, and offer students to implement those approaches in software engineering workshops as part of their curriculum.

We estimate that academy students and the academy itself will gain important benefits from such a move: most of the activities assigned on a new employee are related to changing an existing product, and it is very important that students will implement those changes correctly, with full understanding (and experience) of how to combine change-centric and requirements-centric approaches.

*Ran Lavi is a certified quality engineer (CQE) by the Israeli Society for Quality (ISQ), holds a BSc degree in Mathematics and Computer science and MBA, both from Tel-Aviv University.*

*Ran is a senior consultant for quality management and software engineering in Methoda Computers, with over 20 years of experience in Project Management, Quality Management, Software Engineering, Risk Management, Requirements Engineering, Lesson Learning, ISO 9001 and Software Quality Metrics.*

*Some of the companies/organizations that Ran has consulted to are: NICE Systems, HP Indigo, Israeli Defense Forces, BrightSource Energy, Advanced Vision Technology, and many more.*