

Title of Paper

Testable Requirements: What are the Standards we need to adopt to make sure Requirements are Testable?

Presenter

Tom Gilb

Abstract

Ten principles for testable requirements

- Some basic requirements specification standards to make requirements testable
 - numeric exit and entry levels as a quality level
 - two levels of quality: clear and relevant
 - defined 'rules' to teach and measure requirements
 - well defined concepts – like 'requirement'
 - templates to guide requirements specs

Ten Principles for Testable Requirements: Based on use of 'Planguage'; a requirements specification language.

1. The requirements must themselves be clear, complete, and consistent
 2. The requirements must follow our current standards, measurably
 3. Performance, including quality requirements must be specified quantitatively. A 'Scale' and a future point on the scale must be defined.
 4. Requirements must be correctly "Typed" (Function, Quality, Constraint etc.)
 5. The requirement level priorities must be specified (Catastrophe, Fail, Goal, Stretch)
 6. One or more high level 'Meter' specifications should have been agreed and integrated into the specification initially.
 7. Pointers to corresponding Test Plans, Test Scripts, Test Results, and Test Responsible people should be integrated into the requirement specification itself by the test planners.
 8. Version Control for each requirement change must be fed to responsible test planners (automatically, with note it is done)
 9. The Total set of conditions for a requirement will be specified in one or more [Qualifier] statements.
 10. The Priority Information about a requirement will be suitable to help test planners understand the priority of test quality and timeliness.
-

Biography

Tom Gilb was born in Pasadena, California in 1940; emigrated to London, England 1956, and to Norway 1958, where he joined IBM for 5 years, and where he resides.

He has worked within the software engineering community, but since 1983 with Corporate Top Management problems, and since 1988 with large-scale systems engineering (Aircraft, Telecoms and Electronics).

He is an independent teacher, consultant and writer. He has published 9 books, including the early coining of the term "Software Metrics" (1976, CMM L4 Foundation); "Principles of Software Engineering Management" (1988) and "Software Inspection" (1993); 'Competitive Engineering: A Handbook for Systems Engineering, Requirements Engineering, and Software Engineering Using PLanguage' (2005).

He is recognized as the founder or major driver of several technical disciplines such as software metrics and evolutionary project management, as well as being an innovative pioneer in Inspections, and the inventor of the planning language PLanguage.

He is directly recognized as the idea source for parts of the Agile and Extreme programming methods (primarily the incremental cycles).

His methods are widely adopted by organizations such as IBM, Nokia, Ericsson, HP, Intel, Citigroup, Symbian, Philips Medical Systems, Boeing - and other large and small organizations.

See www.gilb.com, for papers and downloads.

Full Contact information of the Presenter

Name: Tom

Surname: Gilb

Full Postal Address:

E-mail Address: Tom@Gilb.com

Phone Numbers: 47 920 66 705
