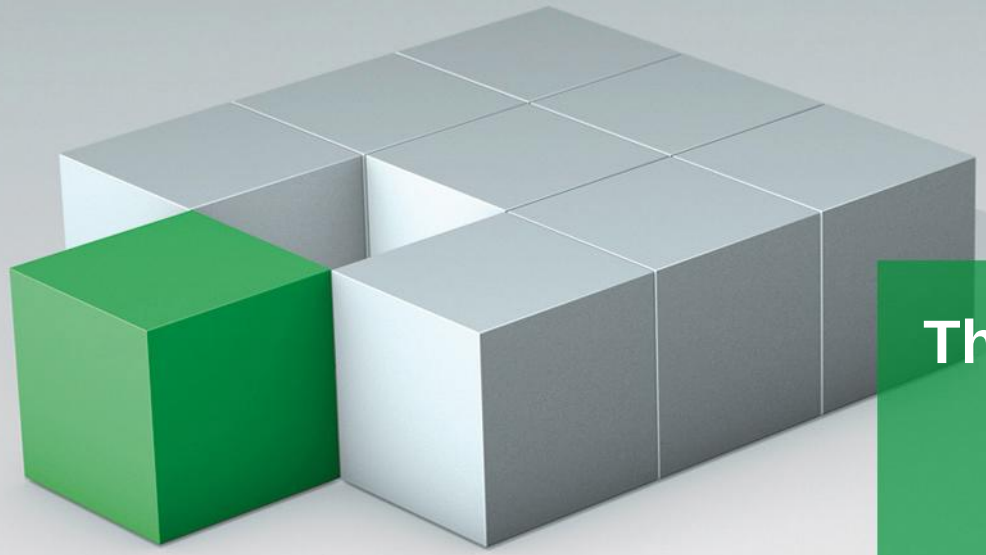
SQS. Excellence through Independence



The Quality of Software

Kim Johnston
August 2011

SQS South Africa

Agenda

1. What defines quality in software?
2. What impacts software quality?
3. How do we ensure software quality?
4. Hints and tips on improving software quality.

1. What defines quality in software ?

Quality software meets a client's needs.

1. What defines quality in software ?

Client's needs can be broken down into:

- Known requirements
- Unknown requirements
- Non-functional requirements

2. What impacts software quality?

- Lack of conformance to the requirement is lack of quality
  - Often caused by lack of understanding
- Specific and clear standards for all stages in the development methodology not defined.
- Implicit requirements not defined or met.
  - Often the non-functional requirements.

3. How do we ensure software quality?

- Define what software quality should look like
- Determine how to test for quality
- Implement quality checks throughout the development lifecycle.

4. Hints and tips on improving software quality.

1.Assist in determining the client's needs.

Clients often don't know exactly what they want and how they want their software to work, but they know very quickly that the delivered product does not meet their quality needs.

Some ideas on this:

-Start small – step by step, the most successful projects often use an iterative development model.

-Ensure all project goals are achievable and understood

- Conduct walkthroughs at the start of the project

-According to Tom Peters – the key to successful innovation is to "Test it now, at least a piece of it, in the real world.."

- Testing starts at project initiation – use test techniques early on in project to test requirements

-Make use of pilot projects

4. Hints and tips on improving software quality (cont.)

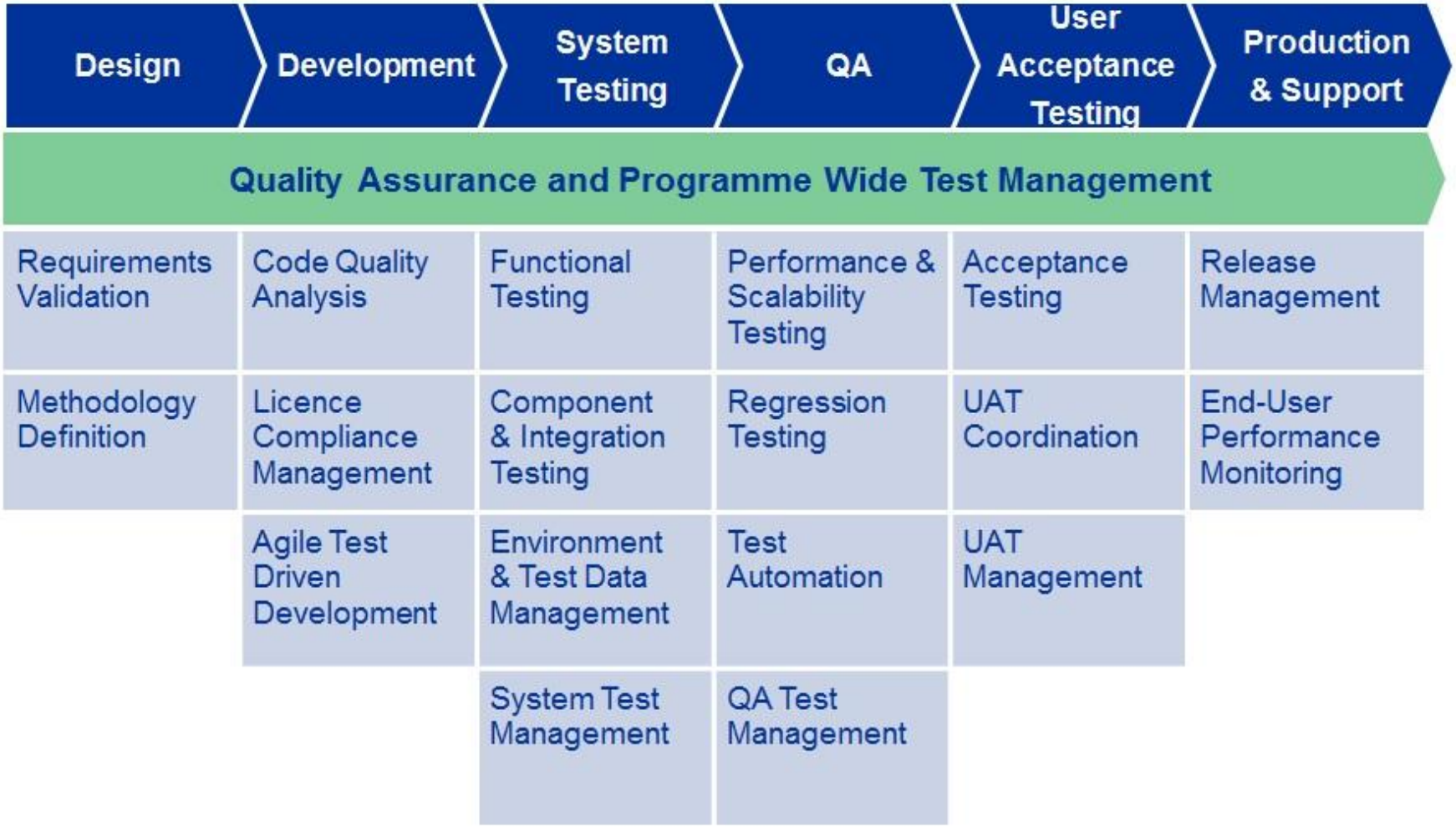
2. Understand and accept that the task of improving software quality is a management task – not a technical task.

- Management to define a methodology that encourages continual improvement
- Objective goals, broken down into small steps, have a much greater chance of success than large exciting stretch goals.
  - Quality demands constant attention.

3. Realise that "correctness" does not equate to quality.

- Quality is not static – what is correct today might not be correct tomorrow.
  - Test constantly throughout the lifecycle

4. Hints and tips on improving software quality. Focus on Testing Early and Continuously

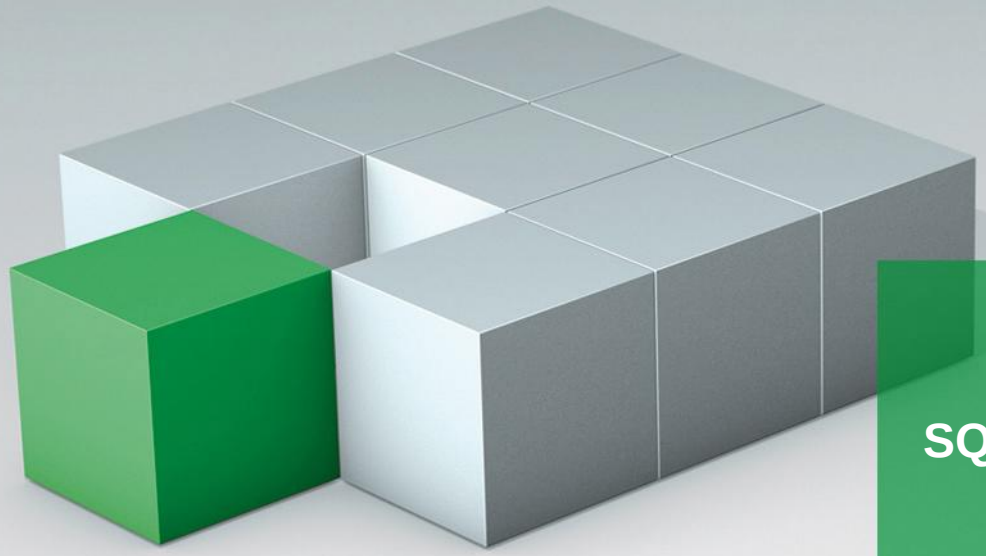


Wrap up on software quality

To wrap up I would like to use my Pen analogy to assist in explaining some of the concepts mentioned :

Classification	User requirement	Test case
Known requirements	The pen must not be too heavy	Test to ensure that the pen weighs between 10 and 12 grams
	It must be able to be used to sign legal documents	Test to ensure that the ink is black
Unknown requirement (expected)	The pen must not leak	Test that the ink does not come out of pen if pen upside down and shaken
	The pen must look nice	Test that out of a sample population of 100 people at least 80 choose this pen
Non-functional requirement	The pen must work for a reasonable time period	Test that the pen can be used to draw a line of 5 km before a refill is required

SQS. Excellence through Independence



SQS South Africa

Tel.: +27 (0) 31 279 2400 | Fax: +27 (0) 31 266 7976
E-Mail: kim.johnston@sqs.com
+27-82-4557920
Internet: www.sqs-sa.com | www.sqs-group.com

Thank you for your attention