Software quality means doing every aspect well

- To do Release/Acceptance Testing well
  - Done requirements and specifications well enough
  - Otherwise the development team won't know what the goals are
- To do Unit testing well
  - Clear specifications and model
- To do Coding well
  - Agreement on coding standards

The software quality team aim to make sure :

- A good SE process is chosen for the project
- All aspects of that SE process are all 'done well'

3 principle concerns of SQM

- Defining standards and procedures for the whole company
- Check that projects conform to company standards, so that the company doesn't release low quality products
- Help create project plans, so that the **quality goals** are set

This team is often call the Quality Assurance (QA) Team

- Ideally this team should separate from development teams
- Report to the managers above the project manager

QA : Four things

- Planning for quality
- Agreeing/Using standards
- Inspecting for quality
- Measuring quality

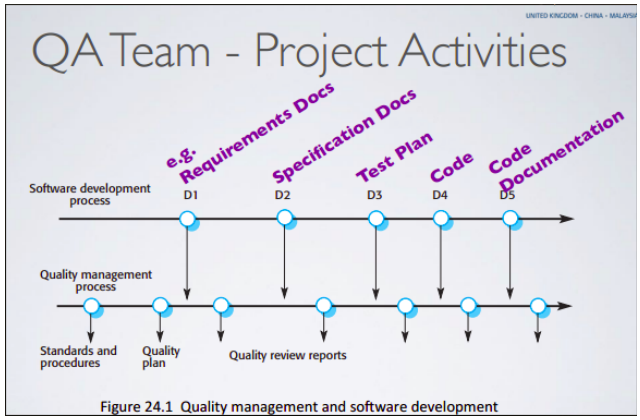

Figure 24.1 Quality management and software development

Quality planning

Sets the bar for the team to achieve. It means everyone knows what will be acceptable at the end. 'It therefore defines what `high-quality` software actually means for a particular system'

Key sections of a quality plan document for a project.

- Product introduction (intended market,etc.)
- Product plans (critical release dates, maintenance, etc.)
- Product description (what process should be followed for this project)
- **Quality goals** (critical quality attributes for final product)
- Risks & Risk management (expected key risk areas)

Takeaway

- Identify standards that must be met
- Recommended quality-promoting processes for the project
- Identify what will count as good quality for that project, so that everyone knows what the **target quality** is

Agreeing/Using standards

- **Product Standards** -e.g. documentation standards, coding conventions, class structure, etc.
- **Process standards** – e.g. when reviews/ testing etc. is done. Defined good practices at each stage of SE processes

Product standards	Process standards
Design review form	Design review conduct
Requirements document structure	Submission of new code for system building
Method header format	Version release process
Java programming style	Project plan approval process
Project plan format	Change control process
Change request form	Test recording process

ISO 9000 standards on Quality Management Processes - its a bit meta - how to do quality standards

Inspections

A group of 3-7 people examine a concrete SE output

- specific reviewers from QA team
- project manager, senior designer
- 1/2 people who led/built the thing that's being reviewed

Should last around 2 hours and focus on :

- Finding problems and non-conformance to standards
- Checking for completeness
- Finding missing or incorrect logical steps
- Seeing if code segments, diagrams, tests, requirements, specifications make sense?

Finally produce documentation as an evidence of quality for client acceptance

Roles in formal inspections

Basic

- Review leader
- Recorder
- Reader
- 1 or more Reviewers

More specific roles

- Moderator
- Designer
- Code implementer
- Tester

Stages of inspection

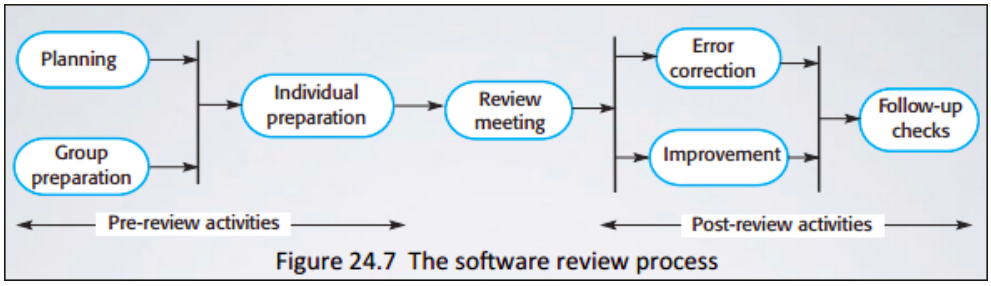

Figure 24.7 The software review process

- Ideally people independently examine the doc in advance and are familiar with it
- The review should take place : a person reads trough the documentation/code focusing on finding. Not solving the problems.
- Output is a list of problems that need fixing
  - Should later be fixed and the fixes audited for completion.

Inspection process

- Introduction to what is being inspected, why, and by whom
- Inspector/Reader step by step reviews the document/code
- People announce/query problems that they spot
  - People judge the severity and importance of the problem, but **do not** discuss how to solve it
- Scribe – tabulates the problem, severity and importance
- At the end : strategies for resolving problems are proposed.

Common inspection problems

- Criticising the person who built the document instead of discussing how to make the document better. Moderator should make sure this doesn't happen
- People worry that their performance will be judged. Could avoid inviting people's line managers to the inspection
- People don't properly prepare before the inspection. Make sure people have carefully read the document first.
- People try to discuss every problem as they find, this leads to very little being inspected.

Measuring software quality

- Have code and document standards have been followed
- Has the software been properly tested
- Is the software sufficiently dependable/ reliable
- Is the performance acceptable
- Is the software usable
- Is the software well structured and understandable

Measure types

Control/Process measures

- Measuring the success of your SE process
- Help decide whether to improve processes later

Code/Predictor Measures

- Help you to judge aspects of code quality
- Are only **predictors** of quality

What do they mean

- We can only make limited assumptions about code measures
- What a metric predicts can depend on the language used
- It's still relatively 'to be proven' whether each of these metrics are good
- Metrics can only be used as **predictors**
  - Once they have been correlated with pas project success
  - Companies develop metrics over time that work for them

Summary

- Quality Assurance is a lot more than release/acceptance testing
- Quality Assurance - is about planning for high quality goals
- Quality Assurance - is about everyone aiming for high quality
- Quality Assurance - is about inspecting for quality at each stage
- Quality Assurance - is about taking / learning from metrics - of software quality - and process quality