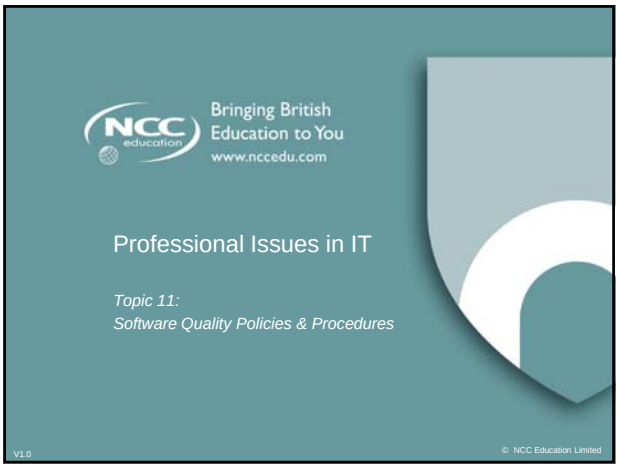
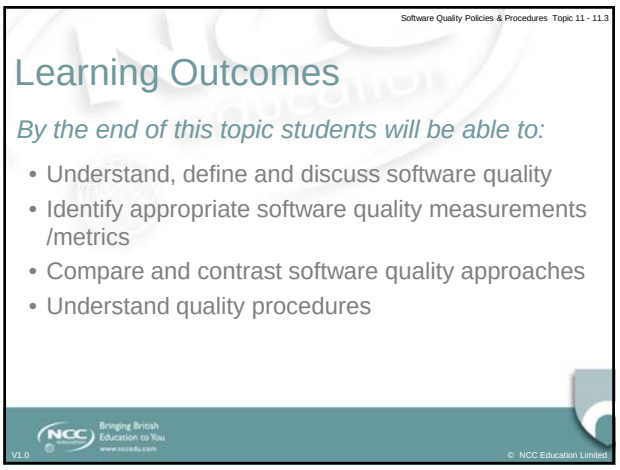

Bringing British
Education to You
www.nccedu.com
Professional Issues in IT
Topic 11:
Software Quality Policies & Procedures
V1.0
© NCC Education Limited


Software Quality Policies & Procedures Topic 11 - 11.2
Scope and Coverage
This topic will cover:
• Software quality
• Approaches
V1.0
© NCC Education Limited


Software Quality Policies & Procedures Topic 11 - 11.3
Learning Outcomes
By the end of this topic students will be able to:
• Understand, define and discuss software quality
• Identify appropriate software quality measurements /metrics
• Compare and contrast software quality approaches
• Understand quality procedures
V1.0
© NCC Education Limited

What is Software Quality?

- Software quality is:

"conformance to customer requirements"

(Cadle & Yeates 2001:203)

Software Quality - 1

- No enforced industry standard
- General industry concern over software quality
- Software is embedded in organisations
  - Huge strategic impacts
- Reliability
- Advanced specification
- Perspective
  - Technical
  - Human

Software Quality - 2

- When considering software quality, we must compare factors.
  - Time vs. detail
  - Costs vs. benefits
  - Features vs. faults
- Quality is ultimately achieved through testing.

Software Quality Factors - 1

- Disseminated from the overall concept of quality
- Many quality factors identified
- Correctness
- Efficient
- Testable
- Portable
- Reliable

Software Quality Factors - 2

- Reusable
- Secure
- Connectability
- Usable
- Maintainable
- Consistent

Software Quality Factor Measurement

- Measurement choice is as varied as definitions of quality!
- Be prepared to have your own opinion
- Software is complex
- Quantitative vs. qualitative

Software Measurements

- Quantitative
  - Numeric, a 'yes/no' measure
  - Faults per lines of code
  - Program load time
  - Program execution time
- Qualitative
  - Judgement based
  - Subjective
  - The interpretation of quantitative data

Fault Measurement

- Is software with fewer faults of greater quality than software with many faults?
- Discuss this question with the person next to you and then share your opinion.

Cost of Quality Factors (COQ)

- Encompasses all quality process costs.
- Represented as cost of conformance and cost of non-conformance.
- 3 classifications:
  - Prevention
  - Appraisal
  - Failure: Internal and External

Maturity Models - 1

- Frameworks
- Iterative approach for software quality development
- Benchmarks
  - Improved performance
- Assess current position
  - Internal or external
- Identify potential for improvement
  - Performance action

Maturity Models - 2

- Many maturity models exist.
- The majority take a 5 layered approach:
  - Beginner
  - Developing
  - Practising
  - Exploiting
  - Optimising
- Examples
  - Software Quality Function Deployment model (SQFD)
  - Capability Maturity Model Integration (CMMI)

SQFD Model

- Client requirement and technical necessity focused
- Enable software planning
  - Design phase
- Objectives
  - Identify client expectations
  - Fulfil client requirements
  - Identify and fulfil software priorities
  - Improve process efficiency
- Measurable graphical approach

CMMI - 1

- Software Engineering Institute (SEI)
- Replaced Capability Maturity Model (CMM)
- Version 1.3 launched 2010
- Enable software product development
- Aim
  - Performance improvement
- Two versions
  - Staged and continuous

CMMI - 2

- The staged approach used 5 classifications:
  1. Initial
  2. Managed
  3. Defined
  4. Quantitatively managed
  5. Optimising
- The continuous approach:
  - 4 groups
  - 24 processes
  - Each organisation process rated

Total Quality Management (TQM) - 1

- TQM is the European term
- QM (quality management) is the US term
- W. Edwards Deming is credited a key contributer to TQM.
- Philosophy is to provide many tools and techniques
- People-orientated
- Must be organisation wide
  - Requires management backing

TQM - 2

- Techniques
  - Benchmarking
  - Quality cost
  - Quality function deployment
- Tools
  - Pareto
  - Flowcharts
  - Cause and effect
  - Scatter diagram
  - Histogram

Pareto

- Interchangeably known as Pareto chart/principle/ analysis
- Based on 80:20 rule
  - 20% of processes account for 80% of problems
- It is represented as a histogram/bar chart
- Identifies where to invest efforts
- Before/after comparison

Cause and Effect

- Interchangeably known as the cause and effect diagram, the Fishbone diagram and the Ishikawa diagram
- Problem/opportunity identification
- Enables wider picture
- Sequential
- Complex

TQM Issues

- Management underestimation
- Quest for certification
- Organisational separation
- Lack of trust

Six Sigma - 1

- Created by Motorola
- Launched 1986
- Replaced TQM?
- Number orientated approach
- Not intuitive
- Series of ‘belts’

Six Sigma - 2

- Three levels:
  - Metric
  - Methodology
  - Philosophy
- Two methodologies:
  - DMAIC
  - DMADV

DMAIC - 1

- Define Measure Analyse Improve Control (DMAIC)
- Data driven
- Process improvement
- Iterative
- Existing processes/software
- Define
  - Problem/opportunity

DMAIC - 2

- Measure
  - Develop, collect, compare data
- Analyse
  - Identify opportunities
    - Collected data
- Improve
  - Formulate solutions
- Control
  - Solution implementation

DMADV - 1

- Define Measure Analyse Design Verify (DMADV)
- Data driven
- Iterative
- New process/software
- Define
  - Goals/client requirements
- Measure
  - Specification

DMADV - 2

- Analyse
  - Available processes
- Design
  - Produce processes
- Verify
  - Performance

Software Testing - 1

- Essential investment process
- Validate software
  - Fulfils business requirements
  - Meets technical requirements
  - Operates as expected
- Ideally built in throughout development process
  - Based on development methodology
- Cannot expect to identify all faults
  - Like risk

Software Testing - 2

- Dependent on:
  - Software criticality
  - Client acceptance criteria
  - Approach/method chosen
- Generates:
  - Test plans
  - Test scripts
  - Test data

Software Testing Approaches

- Many different approaches
  - Black box
  - White box
  - Unit
  - Integration
  - Regression
  - System
  - Acceptance
  - Accessibility
  - Stability
  - Usability
  - Security

Black Box Testing

- Functionality testing
  - Looks at requirements
- Method included in various approaches
- Non technical
- Functional and non-functional
- Test case based
  - Blindfold approach

White Box Testing

- Opposite of black box
- Tests internal coding and structure
- Internal knowledge and technical skills
- Usually within unit testing
- High error detection rates
- Lacking requirement awareness

Regression Testing

- Undertaken after coding alterations
  - Client requirement changes
  - Failure to fulfil specification
- Identify returning faults
- Common in upgrades and patches
  - Non conformance of new and old code
- Reliant on original testing
  - Process documentation

Integration Testing

- Combines sections of code to validate
  - Performance
  - Reliability
  - Functionality
- Incorporates black box
- Integration test plan
  - Test cases

Usability Testing

- User testing
- Measures users' ability to interact with software
- Data collection
- Four criteria
  - Efficiency
  - Accuracy
  - Recall
  - Emotion

Accessibility Testing

- Higher priority
- Equal access
  - Functionality
- Written into specification
- Software application programme interfaces (API)
  - Assist technology product interaction

Software Quality Plan - 1

- Created for each software project
- Contain all relevant information
  - Processes, techniques, standards
    - Requirements
    - Methodology/philosophy
    - Planning
    - Measurements/metrics
    - Standards/ frameworks
    - Testing and validation
    - Documentation

Software Quality Plan - 2

- Multi-chapter report
- Planning phase
- Updatable

References - 1

- Cadle, J & Yeates, D. (2001*). Project Management for Information Systems*. FT Prentice Hall.
- Dalcher, D & Brodie, L.(2007). *Successful IT Projects*. Thomson.
- Hall, P & Fernandez-Ramil, J.(2007). *Managing the Software Enterprise*. Thomson.
- Hughes, B. & Cotterell, M. (1999). *Software Project Management*. McGraw Hill.
- Schwalbe, K. (2005). *Information Technology Project Management*. Course Technology

References - 2

- www.forbes.com
- www.isixsigma.com
- www.thecqi.org

Topic 11 – Software Quality Policies & Procedures

Any Questions?

Bringing British Education to You
www.nccedu.com