QA in Context

Course Code: CSC4133

Course Title: Software Quality and Testing

Dept. of Computer Science
Faculty of Science and Technology

Lecturer No:	6	**Week No:**	3	**Semester:**	
Lecturer:	*Prof. Dr. kamruddin Nur,* *kamruddin@aiub.edu*				

Lecture Outline

- Defect Measurement & Analysis
- QA in Software Processes
  - Waterfall, Iterative & Incremental, Spiral, Agile
- Two views of QA
  - V&V(verification & validation) view
  - DC (defect-centered) view
- Mapping from V&V view and DC view

Objectives and Outcomes

- **Objectives**: To understand the defect handling, measurement and resolution procedure, to understand QA activities in the different types of software processes, to understand mapping between V&V view and DC view.

- **Outcomes**: Students are expected to be able to explain the activities for defect measurement and resolution; be able to explain the different QA activities in software processes; be able to explain the mapping of V&V view and DC view.

QA in Context

- Defect handling is an integral part of QA activities, and different QA alternatives & related activities can be viewed as a concerted effort to ensure software quality. These activities can be integrated into software development & maintenance processes as an integral part of the overall process activities, typically in the following fashion –
  - **Testing** is an integral part of any development process, forming an important link in the overall development chain.
  - Quality **reviews/inspections** often accompany the transition from one phase to another.
  - Various **defect prevention activities** are typically carried out in the **early stages**.
  - **Defect containment activities** typically focus on the **later**, operational part of the development process (although their planning & implementation need to be carried out throughout the development process).

QA in Context

- QA and the overall development context
  - Defect handling/resolution
  - Activities in process
  - Alternative perspectives: **Verification** & **Validation** view
- Defect handling/resolution
  - Status & tracking
  - Causal (root-cause) analysis
  - Resolution: defect removal/etc.
  - Improvement: break causal chain

Defect Measurement & Analysis

- **Defect Measurement:**
  - Parallel to defect handling
  - Where injected/found?
  - Type/severity/impact?
  - More detailed classification possible?
  - Consistent interpretation
  - Timely defect reporting
- **Defect analyses/quality models**
  - As follow up to defect handling
  - Data & historical baselines
  - Goal: assessment/prediction/improvement
  - Causal /risk/reliability/etc. analyses

QA in Software Processes

- **Mega-process**: initiation, development, maintenance, termination
- **Development components:** **R**equirement, **S**pecification, **D**esign, **C**oding, **T**esting, **R**elease
- **Process variations:**
  - **Waterfall** development process
  - **Iterative** development process
  - ***Spiral*** development process
  - **Lightweight** /**agile** development process, e.g. XP, SCRUM
  - Maintenance process too
  - Mixed/Synthesized /customized processes

QA in Waterfall Process

Focus on **defect Prevention**

- Requirement & Specification
- Design
- Coding

Focus on **defect Removal**

- Coding
- Testing

Focus on **defect Containment**

- Release & Support

QA in Waterfall Process

- QA throughout process:
  - Defect prevention in early phases
  - Focused defect removal in testing phase
  - Defect containment in late phases
  - Phase transition: **Inspection/Review/etc**.

QA in Software Processes

- **Process variation(not Waterfall) and QA:**
  - **Iterative**: QA in iterative/increments
  - **Spiral**: QA & risk management
  - **XP**: test-driven development
- **QA in maintenance processes:**
  - Focus on defect handling
  - Some defect containment activities for critical or highly-dependable systems
  - Data for future QA activities
- **QA scattered through all processes**

V&V (Verification & Validation)

- Core QA activities grouped into **V&V**
- **Validation**: w. r. t. requirement (**what**?)
  - – Appropriate/fit-for-use/ ”doing right things”?
  - – Scenario & usage inspection/testing
  - – System/integration/ acceptance testing
  - – Beta testing & operational support
- **Verification**: w. r. t. specification/design (**how**?)
  - – Correct/ “doing things right”?
  - – Design as specification for components
  - – Structural & functional testing
  - – Inspections and formal verification

V&V vs. DC View

- **Two views of QA:**
  - **V&V**(verification & validation) **view**
  - **DC** (defect-centered) **view**
  - Interconnected: mapping possible?
- Mapping between **V&V** and **DC** view:
  - V&V after commitment

    (defect injected directly)➔ defect removal & containment focus
  - Verification: more internal focus
  - Validation: more external focus
  - In **V-model**: closer to user (near top) or developer(near bottom)?

Mapping from **DC view** to **V&V view**

DC- view	**QA activity**	**V&V view**
Defect prevention		Both, mostly indirectly
	Requirement-related	Validation, indirectly
	Other defect prevention	Verification indirectly
	Formal specification	Validation, indirectly
	Formal verification	Verification
Defect Reduction		Both, but mostly verification
	Testing type-	
	Unit	Verification
	integration	Both, more verification
	system	Both
	acceptance	Both, more validation
	beta	Validation
Defect Containment		Both, but mostly validation
	Operation	Validation
	Design & implementation	Both, but mostly verification

V&V vs. DC View

DC-view	QA activity	V&V view
Defect prevention		Both, mostly indirectly
	Requirement-related	Validation, indirectly
	Other defect prevention	Verification indirectly
	Formal specification	Validation, indirectly
	Formal verification	Verification
	Design & implementation	Both, but mostly verification

V&V vs. DC View

DC-view	QA activity	V&V view
Defect Reduction		Both, but mostly verification
	Testing type-	
	Unit	Verification
	integration	Both, more verification
	system	Both
	acceptance	Both, more validation
	beta	Validation

V&V vs. DC View

DC-view	QA activity	V&V view
Defect Containment		Both, but mostly validation
	Operation	Validation
	Design & implementation	Both, but mostly verification

Books

- *Software Quality Engineering: Testing, Quality Assurance and Quantifiable Improvement*, by Jeff Tian

References

1. *Software Testing and Quality Assurance: Theory and Practice*, by Kshirasagar Naik, Priyadarshi Tripathy
2. *Software Quality Assurance: From Theory to Implementation*, by Daniel Galin
3. *Software Testing and Continuous Quality Improvement*, by William E. Lewis
4. *The Art of Software Testing*, by Glenford J. Myers, Corey Sandler and Tom Badgett
5. *Software Testing Fundamentals: Methods and Metrics* by Marnie L. Hutcheson