COURSE NAME

SOFTWARE QUALITY AND TESTING

CSC 4133

(UNDERGRADUATE)

CHAPTER 4

SOFTWARE QUALITY ASSURANCE

QUALITY ASSURANCE

- QA focus on correctness aspect of quality, and dealing with defects
- QA activities can be classified into three generic categories –
  - Defect Prevention: QA activities prevent certain types of faults from being injected into the software since errors are the missing/incorrect human actions that lead to injection of faults into software systems (default values, options to select)
  - Defect Reduction: QA activities detect and remove certain faults once they have been injected into the software systems.
  - Defect Containment: control defect through fault tolerance, failure prevention, or failure impact minimization to assure software reliability and safety (e.g. auto save option)

DEFECT PREVENTION

Defect prevention can be done by two generic ways:

- Error source removal (known error sources)
  - Eliminating certain error sources such as eliminating ambiguities or correcting human misconceptions, which are the root causes for errors and remove through education and training (people-based solution)
- Error Blocking
  - Fault prevention or blocking by directly correcting or blocking these missing or incorrect human actions (e.g. user can not give 0 value to the divisor attribute; data validation in excel)
  - This will result the direct intervention to block errors (fault injections prevented)
  - Systematic defect prevention via process improvements
  - If imprecise designs and implementations, tools and technology that deviate from product specifications or design intensions are the causes for faults, formal methods can help prevent such deviations

DEFECT REDUCTION

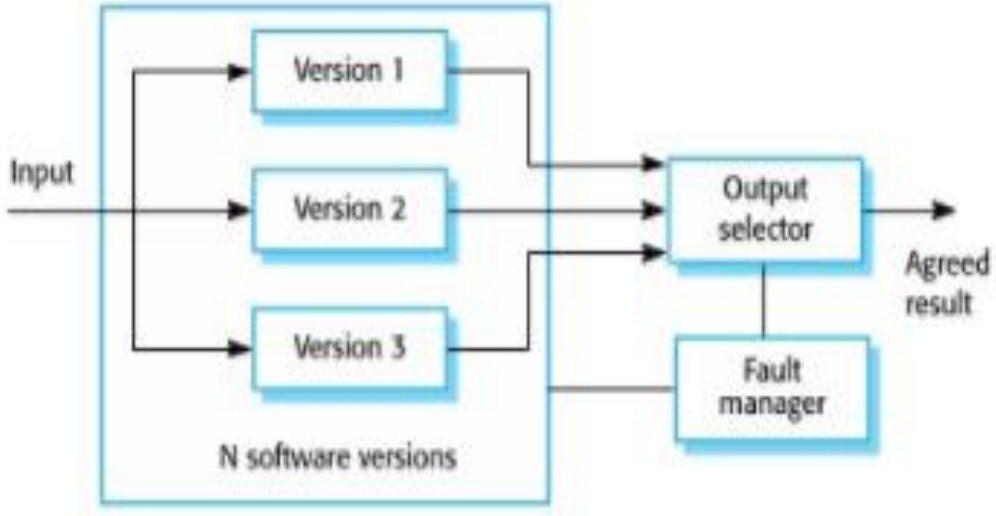
- It is unrealistic to expect the defect prevention activities to be 100% effective in preventing accidental fault injections. Following are the techniques to remove as many faults as possible
- Inspection Method to Reduce Defects
  - The most commonly used static technique uses critical reading and analysis of software code, designs, product specifications, test plans, etc.
  - Formality and structures of Inspections varies in
    - Informal reviews/walkthroughs (self-conducted, independent, pass around, canteen discussion)
    - Formal Inspection are typically conducted by multiple human inspectors through some predefined coordination of process.

DEFECT REDUCTION

- Testing Method to Reduce Defects
  - Involves the execution of software and the observation of the program behavior/outcome
  - If a failure is observed, the execution record is then analyzed to locate & fix the fault(s) that caused the failure.
- Other techniques and Risk identification Method to Reduce Defects
  - Formal model-based analyses
  - Boundary value analysis (e.g. loop condition check)
  - Control flow and data flow analyses (e.g. conditional statements)
  - Simulation and Prototyping (e.g. Autopilot)

DEFECT CONTAINMENT

- Defect reduction activities can only reduce the number of faults to a fairly low level, but not completely eliminate them (because of the large size and high complexity of most software systems in use today as complete testing is not possible)
- The containment measures focus on the failures by either containing them to local areas so that there are no global failures observable to users, or limiting the damage caused by software system failures (e.g. exception handling)
- Defect Containment can be done in two generic ways:
  - Software fault-tolerance
    - Recovery: rollback and redo
    - NVP: N-Version programming (fault blocked)
  - Safety assurance and failure containment



DEFECT CONTAINMENT

- Safety assurance & failure containment
  - Safety : Accident free (e.g. Autopilot safety eliminate pilot error)
  - Accident: Failure with severe consequences
  - Hazard: Pre-condition to accident
  - Safety Assurance: Hazard analysis, hazard elimination/reduction/control, damage control

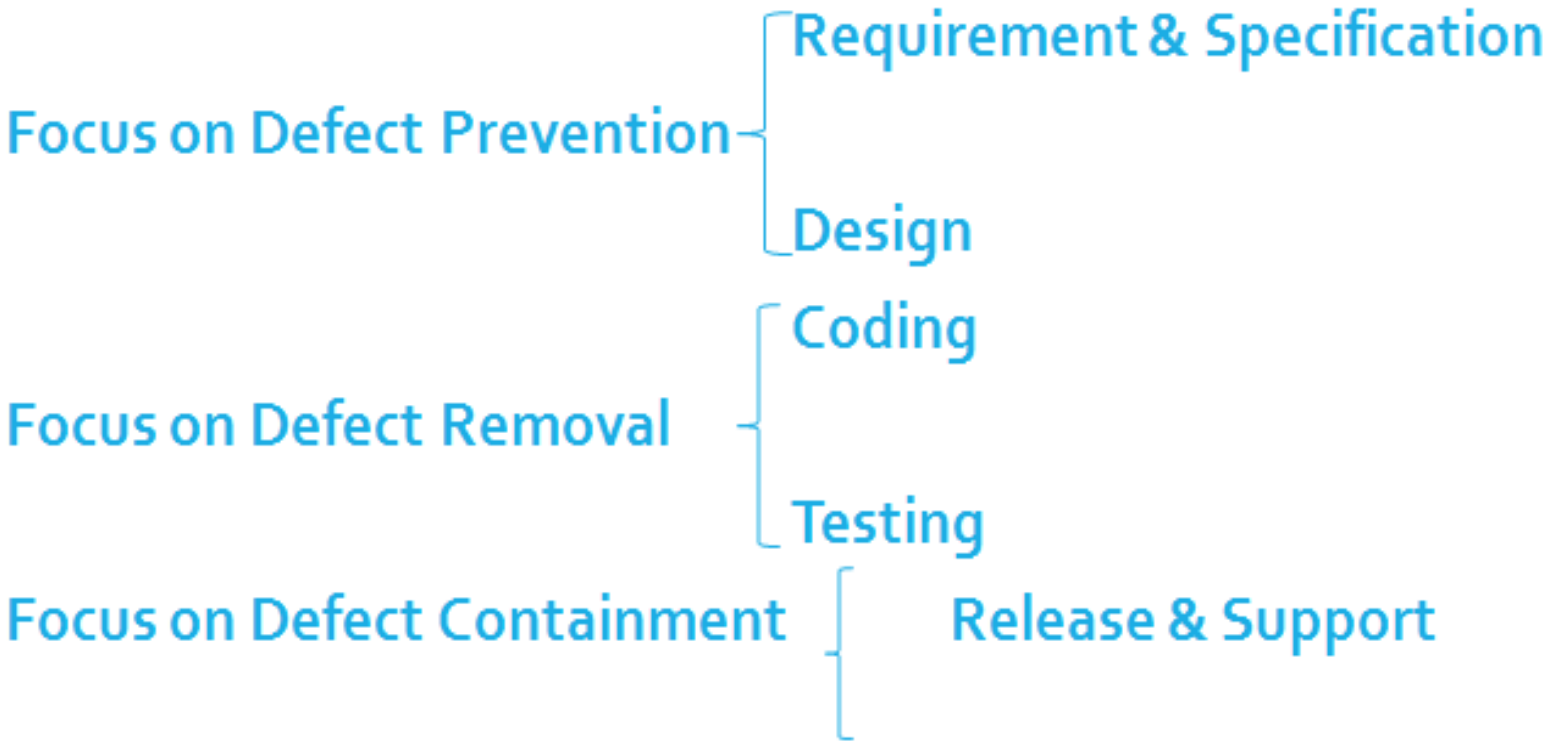
QA IN SOFTWARE PROCESS

- Mega-process
  - Initiation
  - Development
  - Maintenance
  - Termination

- Development components
  - Requirement
  - Specification
  - Design
  - Coding
  - Testing
  - Release

- Process variations
  - Waterfall development process
  - Iterative development process (QA in increments)
  - Spiral development process (QA & risk management)
  - Agile development process (XP- test driven development pair programming)
  - Maintenance processes (focus on defect handling)

QA IN WATERFALL SOFTWARE PROCESS



QA ACTIVITIES: MAPPING FROM **DC VIEW** TO **V&V VIEW**

DC (defect-centered) view	**QA activity**	**Validation(external focus) &Verification (internal focus) view**
Defect prevention		Both, mostly indirectly
	Requirement-related	Validation, indirectly
	Other defect prevention (Project Plan)	Verification indirectly
	Formal specification	Validation, indirectly
Defect Reduction	Testing type	Both, but mostly verification
	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

REFERENCES

- Software Testing And Quality Assurance – Theory and Practice - Kshirasagar Naik & Priyadarshi Tripathy
- Software Quality Engineering: Testing, Quality Assurance and Quantifiable Improvement - Jeff Tian