SOFTWARE QUALITY

CPTS 583

Introduction to the course and *software quality*

Outline

- Introduction to this course
    - Logistics
- Introduction to software quality
    - Software engineering
        - Software process
        - Quality focus
    - Software quality
        - What and why

Course topics

- Software quality concepts and process
- Quality factors, value, and costs
- Quality models and metrics
- Process and product quality
- Quality assessment and management
- Software verification and validation
- Software quality assurance (SQA)
- SQA strategies (inspection, review, audits, walk-throughs)
- SQA tools and standards
- Software analysis, testing, debugging

Course objectives

- Define quality goals
- Plan for SQA
- Estimate quality cost
- Model and control SQA process
- Define metrics and assess quality
- Improve quality using SQA techniques and tools
- Use software analysis/testing/debugging techniques

Resources

- Primary: in-class lecture and discussion

- Supplementary
  - Textbooks and other readings
  - Q/A forum (Blackboard Discussion Board)
  - Project topics
  - WSU libraries
  - Internet
  - Office hours (and extra appointments)

Expected coursework

- Participation
  - 5%
- Midterm
  - 15%
- Final exam
  - 20%
- Course project
  - 60%

Exams

- Midterm
  - 7th week
  - Tentatively: February 27th; class meeting time
- Final exam
  - May 6 (8-10am)

Project deliverables

- Submission
  - Code/documentation: to project repository
  - Single PDF to blackboard
- Late policies
  - If past due
    - If noticed instructor in advance
      - 10% penalty for each day after due
    - Else
      - Zero

Course interaction

- In-classroom discussion
  - Ask questions anytime, as soon as possible; don't wait and accumulate confusions
- Offline questions
  - Mostly go to Blackboard Discussion Board
  - Others (e.g., personal notice) go to email
  - Urgent issues via phone call

Policies

- Academic integrity
- Incomplete grade
- Accommodations
- Team collaboration

Introduction to software quality

- What is software quality?
- Why should we study software quality?
- How can we achieve good software quality?

What is software quality?

- Software

Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system. (IEEE definition)

What is software quality?

- Software quality

By IEEE:

(1) The degree to which a system, component, or process meets specified requirements.

(2) The degree to which a system, component, or process meets customer or user needs or expectations.

What is software quality?

- A matter of perspective
  - As end users
    - Functionality
    - Reliability
    - Resilience/robustness
    - Usability
    - …

What if any of these properties is missing or failing?

What is software quality?

- A matter of perspective
  - As operation / deployment engineer
    - Security
    - Performance

What if any of these properties is missing or failing?

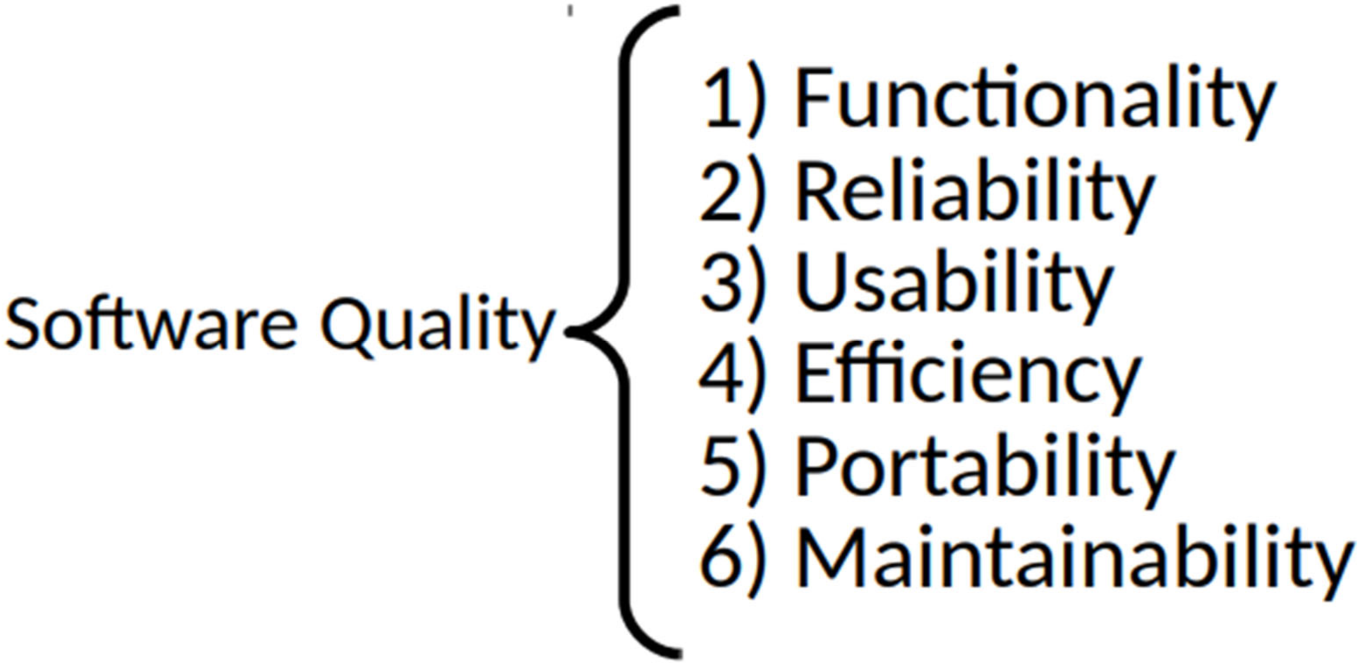
What is software quality?

- A matter of perspective
  - As developer
    - Adaptability
    - Extensibility
    - Measurability
    - Testability
    - ….

What if any of these properties is missing or failing?

What is software quality?

- Six characteristics (as per ISO/IEC 9126)

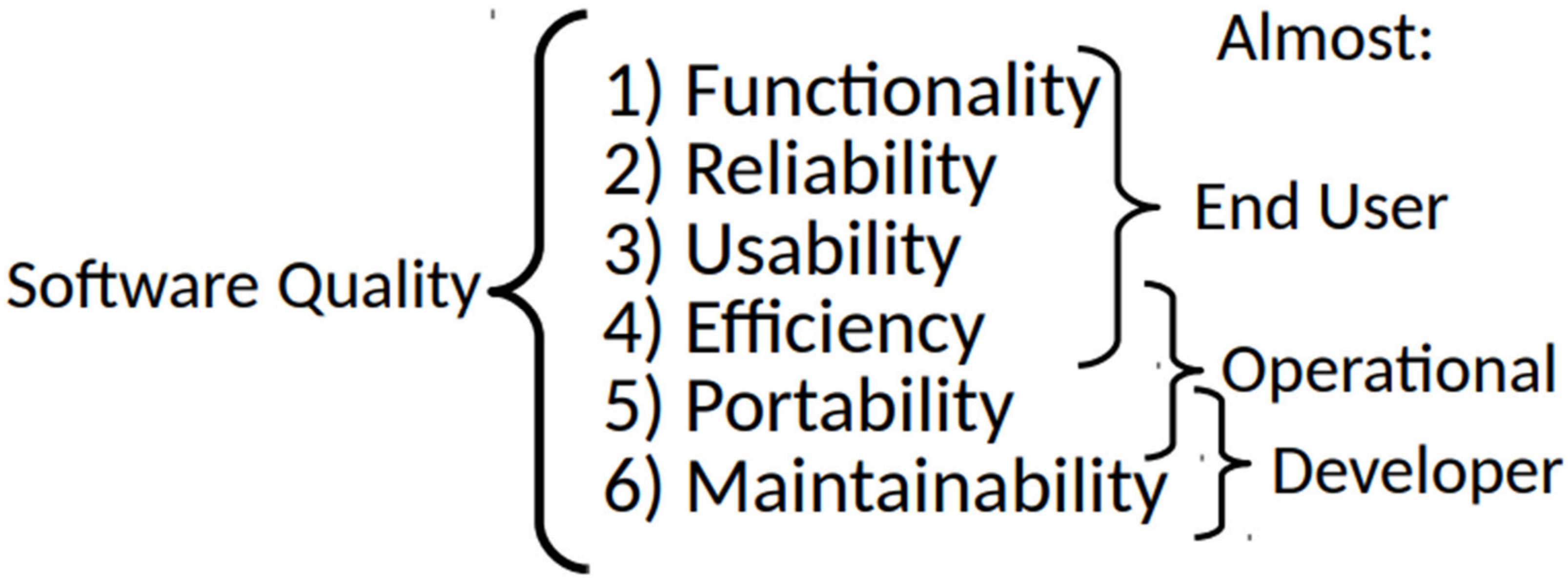


Software Quality

1) Functionality
2) Reliability
3) Usability
4) Efficiency
5) Portability
6) Maintainability

What is software quality?

- Six characteristics (as per ISO/IEC 9126)



Software Quality

1) Functionality
2) Reliability
3) Usability
4) Efficiency
5) Portability
6) Maintainability

Almost:

End User

Operational

Developer

Uniqueness of software quality

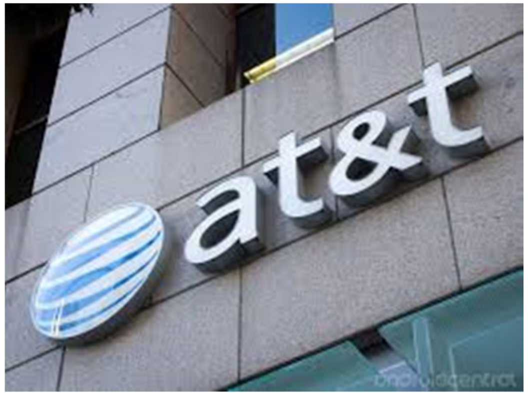
- “Software” versus other industrial product
  - Higher complexity
  - Invisibility
  - Difficulty to detect defects

Why software quality?

- Good versus bad software

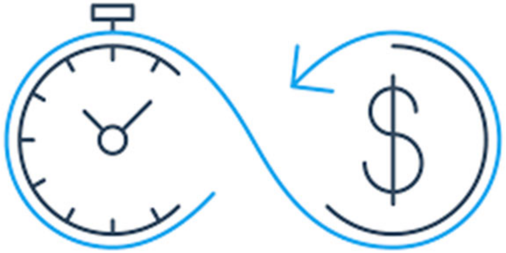
Why software quality?

- Failure in software quality →
  - Airbus 320 shooting down
  - Patriot missile failure
  - London Ambulance deaths
  - AT&T service breakdown
  - ……



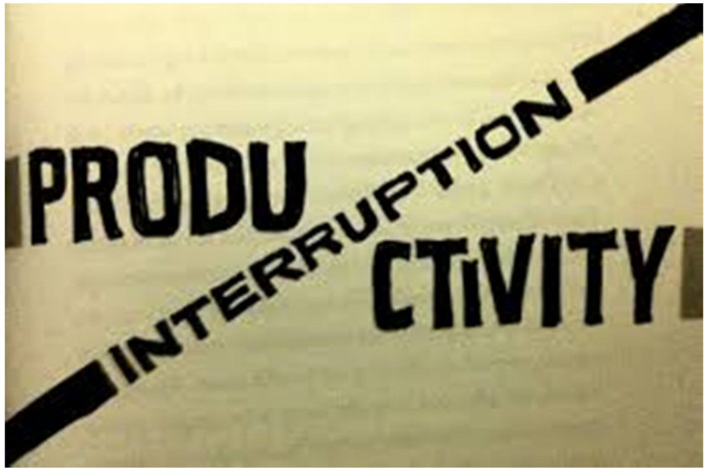
Why software quality?

- Good software quality →
  - Save cost
  - Save time
  - Attract and keep customer
  - Future value
  - ……



Why software quality?

- Good versus bad software
- Lower quality = more rework



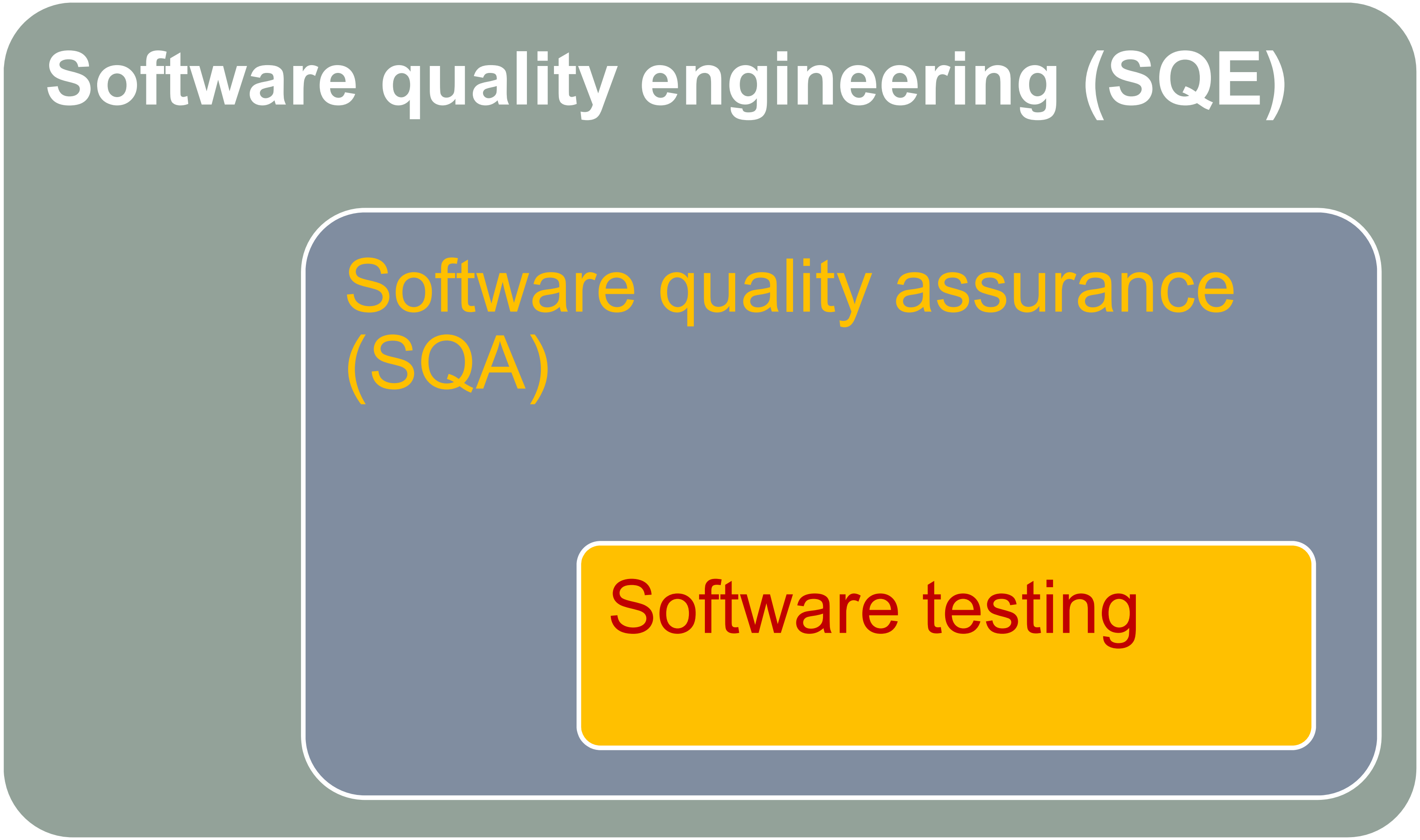
How to achieve software quality?

- Principled approach to software process
  - Define clear goals and metrics
  - Assess and manage quality throughout entire software process
  - Quality assurance (techniques and tools)
  - Validation and verification techniques
  - Quality improvement
  - ……

Challenges to software quality

- Difficult to achieve good quality
    - Size
    - Complexity
    - Environmental constraints
    - Flexibility/adaptability expected

Software quality: ***scope***



Software quality engineering (SQE)

Software quality assurance (SQA)

Software testing

Summary: what we have learned?

- Definitions of Software and Software Quality
- Characteristics of Software Quality
  - Different perspectives
  - Different roles
- The motivation for studying Software Quality
  - Consequences of bad quality
  - Benefits of good quality
- Approaches to good quality for software
- Scope of our study