SOFTWARE QUALITY

CPTS 583

Concepts of software quality

Software quality - IEEE definition

Software quality is:

(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.

ISO Definition of Quality

ISO 9216 Model:

Quality characteristics

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

ISO 8402 definition of QUALITY:

The totality of features and characteristics of a product or a service that bear on its ability to satisfy stated or implied needs.

Outline

- Quality perspectives
- Quality frameworks
- Software correctness
- Quality focus
- Quality measurement
- Quality process
- Quality assurance

Software quality perspectives

- Different roles

Consumer

Third party/indirect users

Generalized users

Software quality perspectives

Various perspectives:

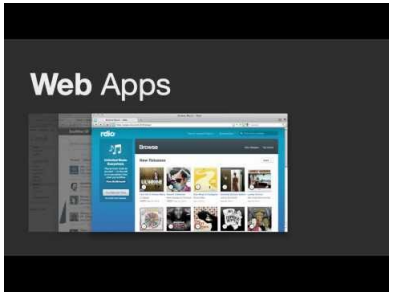
- **Customer** : Complete requirements (Functional and non functional)

- **Project manager**: Cost and schedule
- **Maintenance engineer**: Detection and correction times

Quality perspective

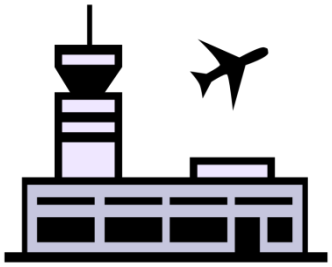
- Vary with different software

General-purpose system
(functionality)



End-user / Web / GUI applications
(usability)

Embedded software
(interoperability)



Safety-critical systems
(safety, reliability)

Quality frameworks

- ISO 9126 quality characteristics:
  - Functionality: what is needed?
  - Reliability: function correctly.
  - Usability: effort to use.
  - Efficiency: resource needed.
  - Maintainability: correct/improve/adapt.
  - Portability: one environment to another.

Quality frameworks

- Adaptation of ISO-9126:
  - ▷ customized for companies
    – e.g., IBM's CUPRIMDSO.
  - ▷ adapted to application domains
    – reliability, usability, security for Web

Quality frameworks

- Other quality frameworks/mega-models
  - ▷ McCall: factors, criteria, and metrics
  - ▷ Basili: GQM (goal-question-metric)
  - ▷ SEI/CMM: process focus/levels
  - ▷ Dromey: component reflects Q-attributes
  - ▷ Defect-based view: common in industry
    - cost of defect: by Boehm, NIST, etc.

Software correctness

- High quality -> few defect
- Defect
  - Error – Fault – Failure

  Error—a quality problem found *before* the software is released to end users

  Fault/Failure—a quality problem found only *after* the software has been released to end-users
- Correctness: an attribute of Quality

Software correctness

- Error – Fault – Failure

Release

```
Int sum (int a[ ], int n)
{
    int sum = 0;
    for (int i=0; I <= n; i++)
        sum += a[i];
    return sum;
}
```

Error

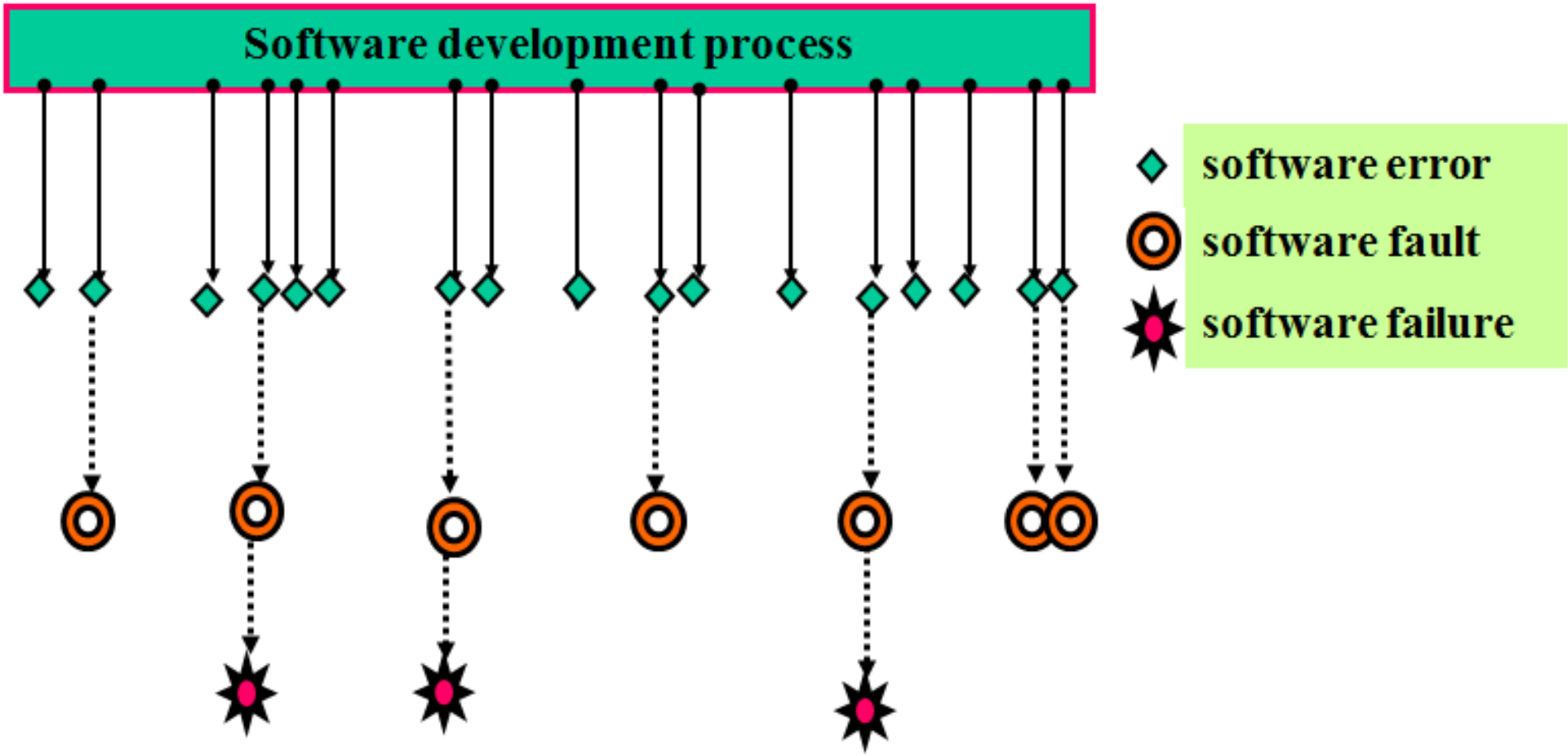
Fault

```
Int sum (int a[ ], int n)
{
    int sum = 0;
    for (int i=0; I <= n; i++)
        sum += a[i];
    return sum;
}
```

Input: {1,2,3}
Output: core dumped…

Failure

Software correctness



Software correctness

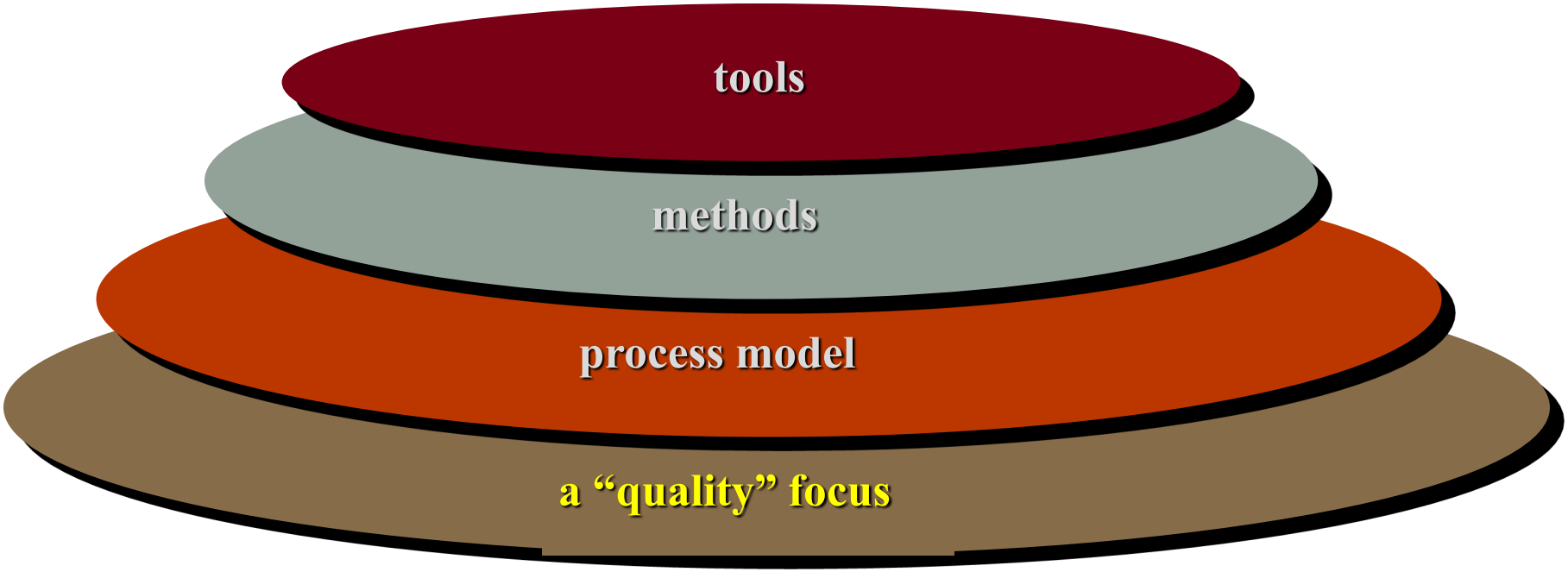
The nine causes of software errors are:

1. **Faulty requirements definition**
2. **Client-developer communication failures**
3. **Deliberate deviations from software requirements**
4. **Logical design errors**
5. **Coding errors**
6. **Non-compliance with documentation and coding instructions**
7. **Shortcomings of the testing process**
8. **User interface and procedure errors**
9. **Documentation errors**

Quality focus

- Quality is the focus of an entire software engineering process

A layered overview of SE



Quality measurement

Planning

- ✓ Decide what criteria are most important
- ✓ Form a plan to assess them, directly or indirectly

Quality measurement

Procedure

- ✓Code reviews help to improve maintainability & reduce bugs
- ✓Regular testing catches issues early

Quality measurement

Measuring

- ✓ Show that the software meets the agreed requirements
- ✓ Observe defects and other issues

Quality process

- Software development as a process
  - (requirement) analysis – design – construction – maintenance

- Software quality as a process

The quality of software (**product**) comes from the quality of the **process** used to build it.

Quality assurance

- Quality assurance (QA)
  - Focus - correctness
  - What - Dealing with defects
  - When - Earlier the QA, lower the cost
    - Post-release versus pre-release

SQA - IEEE definition

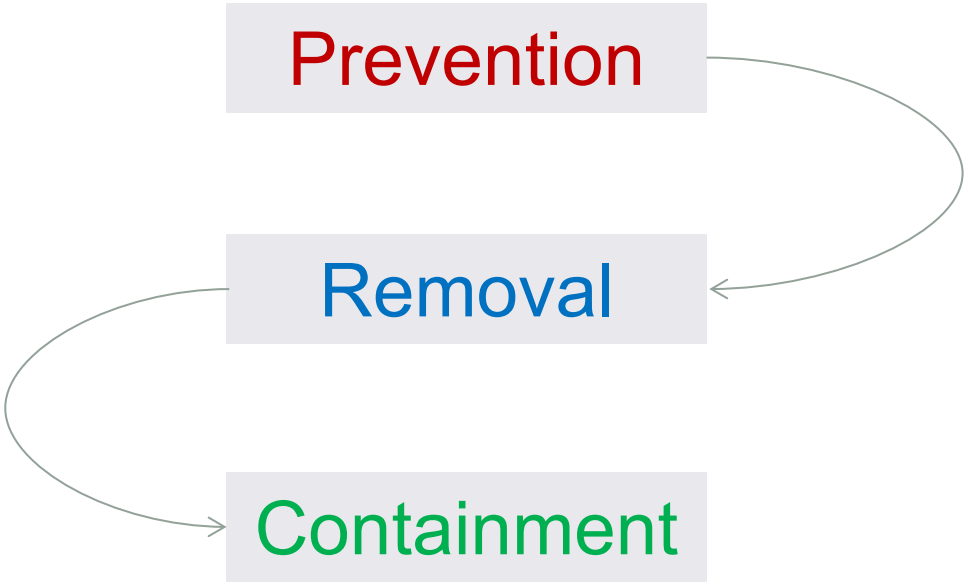
SQA **is:**

(1). *A* planned and systematic pattern of all actions necessary to **provide adequate confidence** that an item or product conforms to established technical requirements.

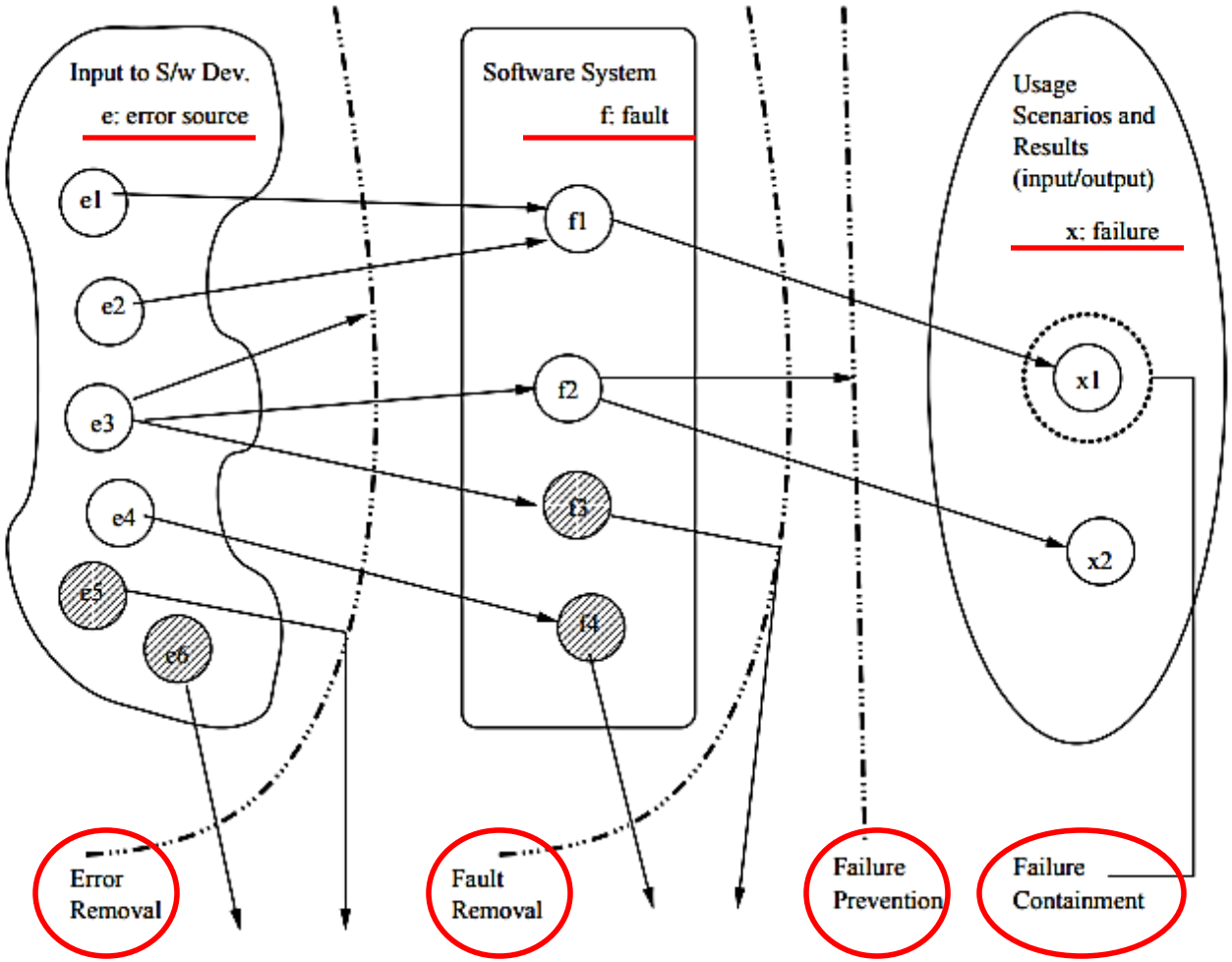
(2). *A* set of activities designed to **evaluate the process** by which the products are developed or manufactured. Contrast with ***quality control***.

Quality assurance

- How?
  - How to deal with defects?



Quality assurance



Quality assurance

- Defect prevention
  - Error blocking
  - Error source removal
  - Systematic prevention
    - Through *process improvement*

- Inspection
  - Informal (reviews)
  - Formal (inspections)

- Fault tolerance
  - Motivation
  - Techniques

- Failure tolerance
  - Concepts
  - Techniques

Quality assurance

- Defect prevention
  - Error blocking
  - Error source removal
  - Systematic prevention
    - Through *process improvement*

- Inspection
  - Informal (reviews)
  - Formal (inspections)

- Fault tolerance
  - Motivation
  - Techniques

- Failure tolerance
  - Concepts
  - Techniques

Quality assurance

- Defect prevention
  - Error blocking
  - Error source removal
  - Systematic prevention
    - Through *process improvement*

- Inspection
  - Informal (reviews)
  - Formal (inspections)

- Fault tolerance
  - Motivation
  - Techniques

- Failure tolerance
  - Concepts
  - Techniques

Quality assurance

- Defect prevention
  - Error blocking
  - Error source removal
  - Systematic prevention
    - Through *process improvement*

- Inspection
  - Informal (reviews)
  - Formal (inspections)

- Fault tolerance
  - Motivation
  - Techniques

- Failure tolerance
  - Concepts
  - Techniques

Quality assurance

- Defect prevention
  - Error blocking
  - Error source removal
  - Systematic prevention
    - Through *process improvement*

- Inspection
  - Informal (reviews)
  - Formal (inspections)

- Fault tolerance
  - Motivation
  - Techniques

- Failure tolerance
  - Concepts
  - Techniques

Summary: what we have learned?

- Quality perspectives
  - Varying with categories of roles
  - Varying with types of software
- Quality frameworks
  - ISO and alternative ones
- Correctness
  - Error – fault – failure
- Quality as a process
- SQA
  - Definition
  - Related concepts