ĐẠI HỌC
BÁCH KHOA

25 YEARS ANNIVERSARY
SOICT

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Software Quality Assurance
Đảm bảo chất lượng phần mềm

Lecture 1: Introduction to SQA

Outline

- Basic concepts and terms
- Definition of Software Quality
- Quality Models
- Software Quality Assurance Process

1.1. Basic concepts

Basic concepts and terms
Các khái niệm và thuật ngữ cơ bản

- Verification vs Validation:
  - Verification (to verify – xác minh): Determining whether the product of a given development phase satisfies the requirements established before the start of this phase.
    - **Verify if the system is built right**
  - Validation (to validate – kiểm chứng): Confirming that a product meets its customer’s expectation
    - **Validate if we built a right system**

Failure– Fault – Error – Defect (Lỗi)

- Failure: is an **event** that the behavior of a system does not conform to that prescribed in the system specification
  - Failure – failed function caused by an error
- Error: is an erroneous **state** of the software
  - Faulty state
- Fault: is the adjudged **cause** of an error
  - Human error: mistake by engineer
  - Fault/Defect/Bug: anomaly in the software
- Debugging – Fault localizing (Sửa lỗi hay xác định lỗi)

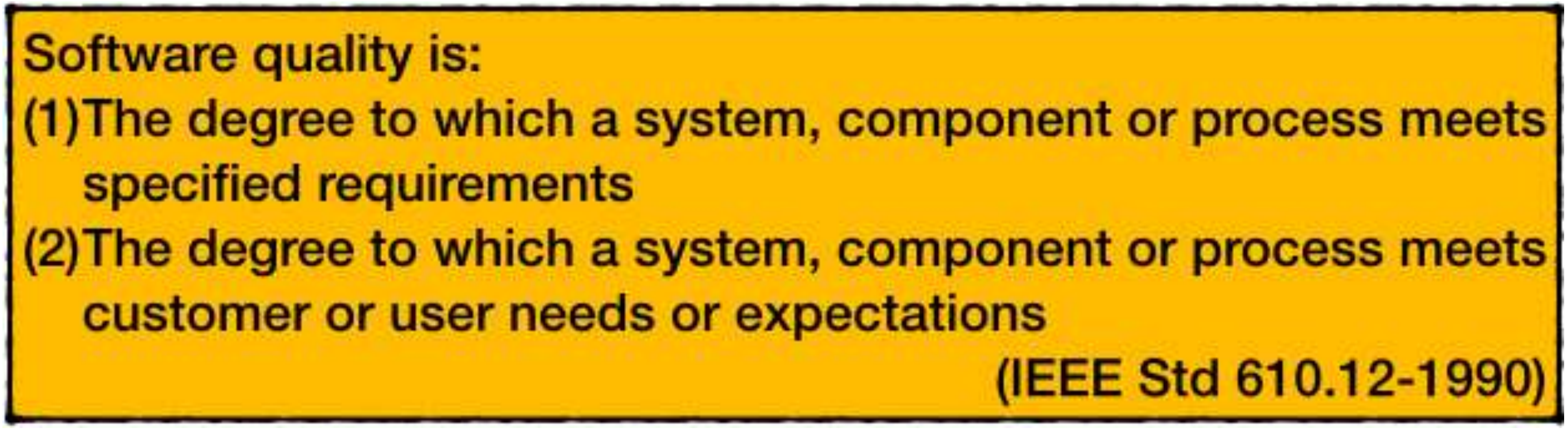
1.2. Software Quality

Five views of Quality
5 quan điểm về chất lượng sản phẩm nói chung

- **Transcendental view (Quan điểm siêu việt về chất lượng)**: quality is something we can recognize but not define
- **User view**: quality is fitness for purpose
- **Manufacturing view**: quality is conformance to specification
- **Product view**: quality is tied to inherent product characteristics
- **Value-based view (Economic)**: quality depends on the amount the customer is willing to pay for it

Software Quality
Chất lượng phần mềm là gì?

- In general, **quality** means that a product should **meet its specifications**



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

(IEEE Std 610.12-1990)

Problems in Software Quality Assurance

- A tension between:
  - Customer quality requirements (efficiency, reliability etc.)
  - Developer quality requirements (reusability, maintainability, etc.)
- Software specifications are usually incomplete and often inconsistent
- Some quality requirements are difficult to specify in an unambiguous way

1.3. Software quality model

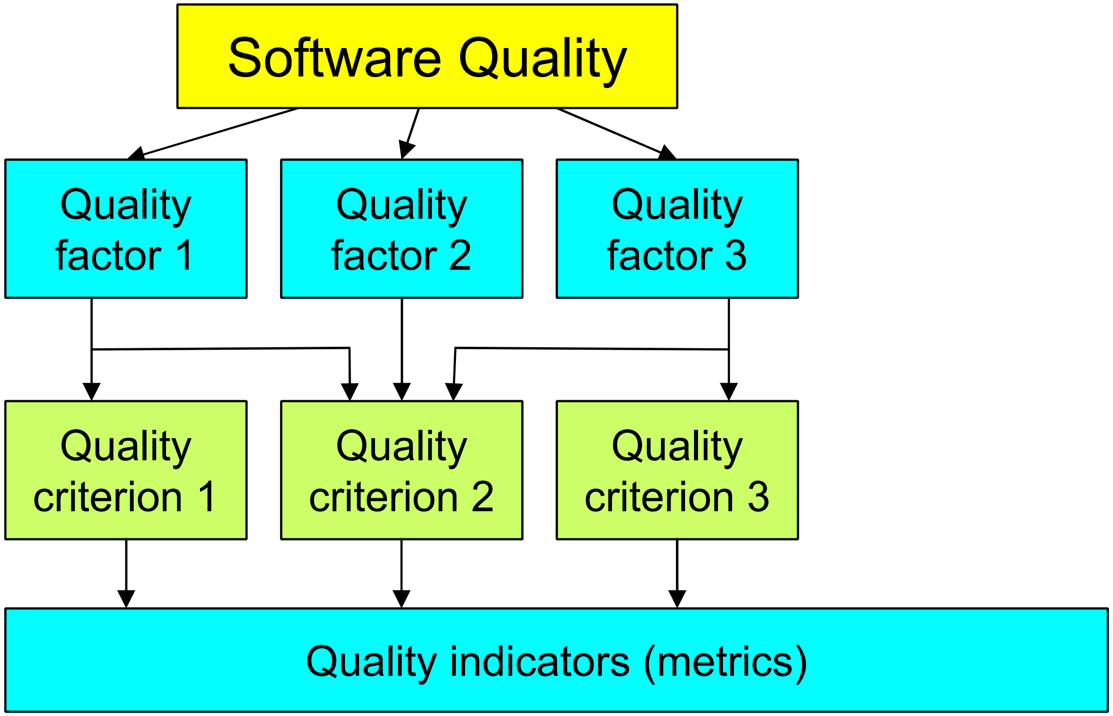
Software Quality Model
Các mô hình chất lượng phần mềm

- Such general definitions of software quality are not sufficient in practice
- Thus, software quality is described by specific quality models
- Two main approaches:
  - Standard Models: McCall, ISO/IEC 9126, ISO/IEC 25000, Boehm, Dromey
  - Application or company specific quality models: FURPS, GQM Approach

Factors – Criteria – Metrics
Nhân tố – Tiêu chí – Độ đo

- Each quality model contains:
  - Factors (to **specify**): describe the external view of the software, as viewed by users
  - Criteria (to **build**): describe the internal view of the software, as seen by developers. Considered as characteristics which define the quality factors
  - Metrics (to **control**): defined and used to provide a scale and method for measurement

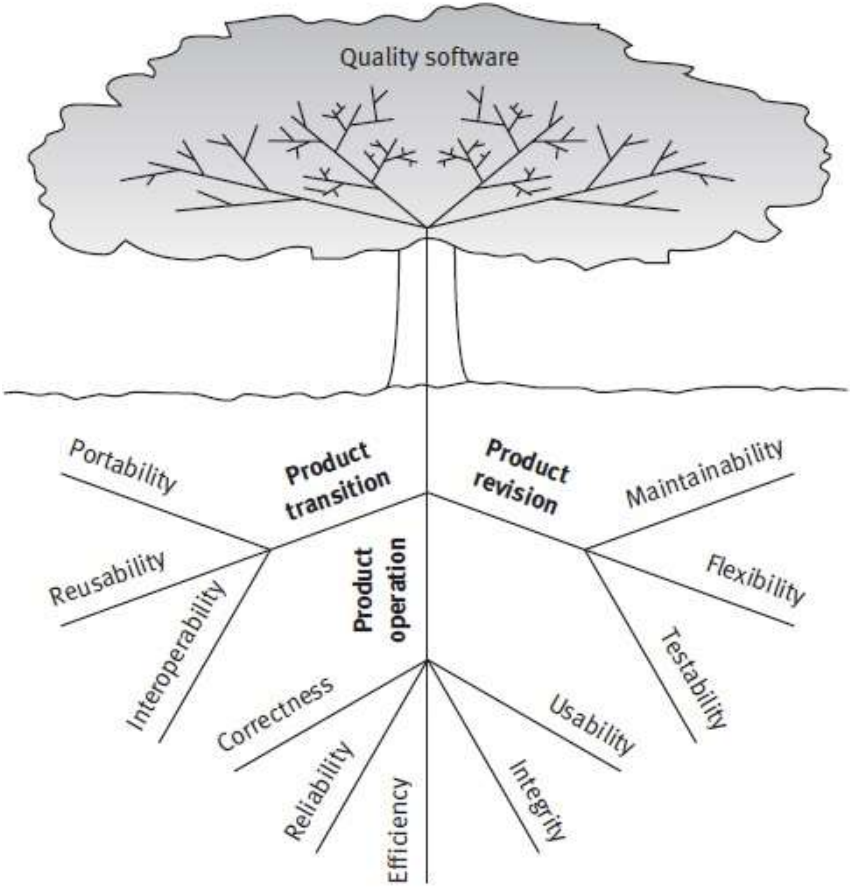
Factors – Criteria – Metrics in Software Quality Model



McCall’s Quality Factors

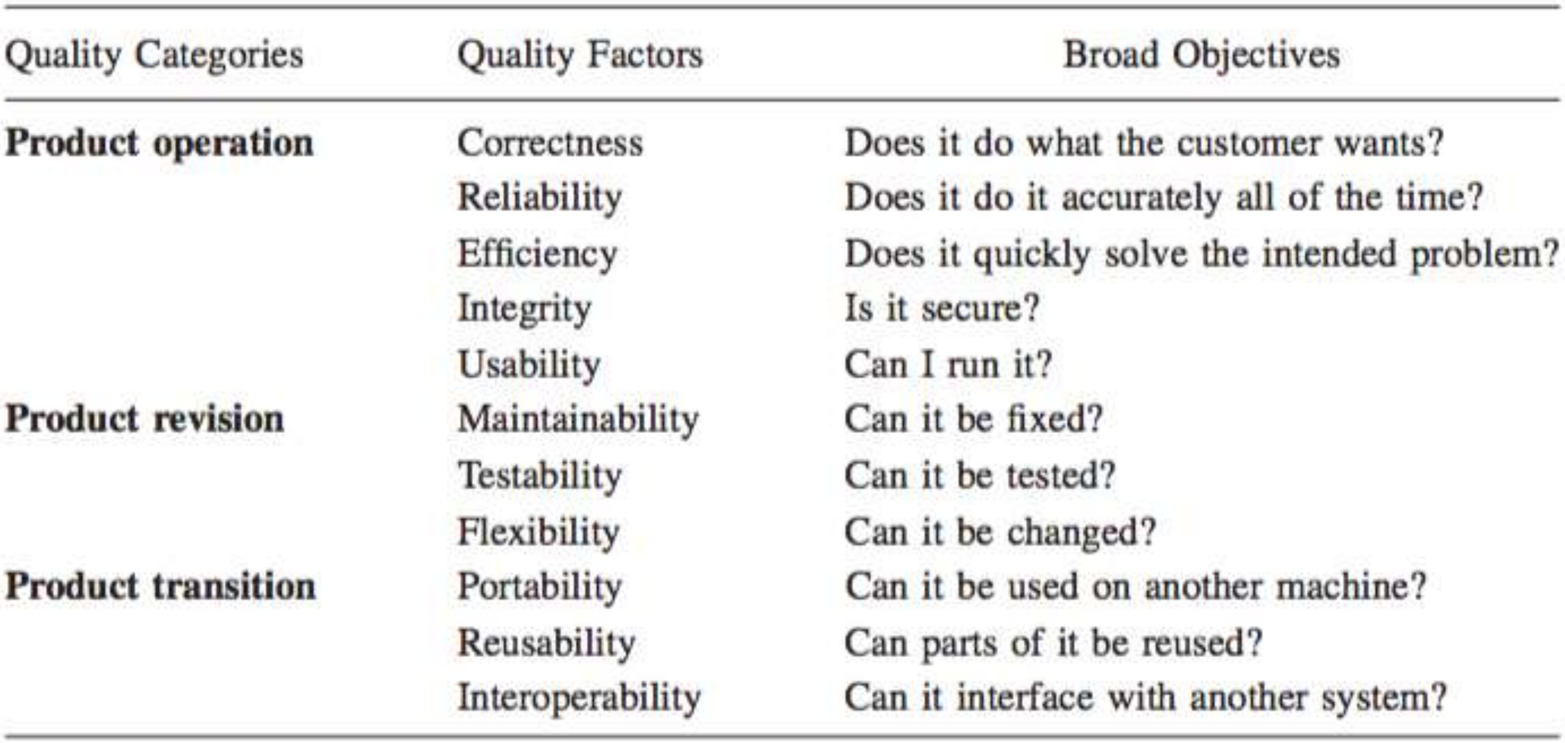
- A **quality factor** represents a behavioral characteristic of the system
- 3 groups
  - Product Operation
  - Product Revision
  - Product Transition
- 11 quality factors

McCall's Factor Model Tree
Cây nhân tố chất lượng của mô hình McCall



Source: from Internet

11 quality factors of McCall’s Model
11 nhân tố chất lượng của mô hình McCall



Quality Categories	Quality Factors	Broad Objectives
Product operation	Correctness	Does it do what the customer wants?
	Reliability	Does it do it accurately all of the time?
	Efficiency	Does it quickly solve the intended problem?
	Integrity	Is it secure?
	Usability	Can I run it?
Product revision	Maintainability	Can it be fixed?
	Testability	Can it be tested?
	Flexibility	Can it be changed?
Product transition	Portability	Can it be used on another machine?
	Reusability	Can parts of it be reused?
	Interoperability	Can it interface with another system?

Source: Software
Testing and Quality Assurance
Theory and Practice

McCall’s Quality Criteria
Các tiêu chí chất lượng của mô hình McCall

- A **quality criterion** is an attribute of a **quality factor** that is related to software development
  - Ex: Modularity is a quality criterion related to the quality factor maintainability
  - Traceability is a quality criterion related to the correctness of a system
- There are 23 quality criteria of McCall’s Model

23 quality criteria of McCall
23 tiêu chí chất lượng đưa ra bởi McCall

Quality Criteria	Definition
Access audit (kiểm tra truy nhập)	Ease with which software and data can be checked for compliance with standards or other requirements
Access control (kiểm soát truy nhập)	Provisions for control and protection of the software and data
Accuracy (Tính chính xác)	Precision of computations and output
Communication commonality (Độ tương đồng giao tiếp)	Degree to which standard protocols and interfaces are used
Completeness (Tính đầy đủ)	Degree to which a full implementation of the required functionalities has been achieved

23 quality criteria of McCall
23 tiêu chí chất lượng đưa ra bởi McCall

Quality Criteria	Definition
Communicativeness (Sự sẵn sàng giao tiếp)	Ease with which inputs and outputs can be assimilated
Conciseness (Tính xúc tích)	Compactness of the source code, in terms of lines of code
Consistency (Tính nhất quán)	Use of uniform design and implementation techniques and notation throughout a project
Data commonality (Độ tương đồng dữ liệu)	Use of standard data representations
Error tolerance (Độ dung thứ lỗi)	Degree to which continuity of operation is ensured under adverse conditions
Execution efficiency	Run time efficiency of the software

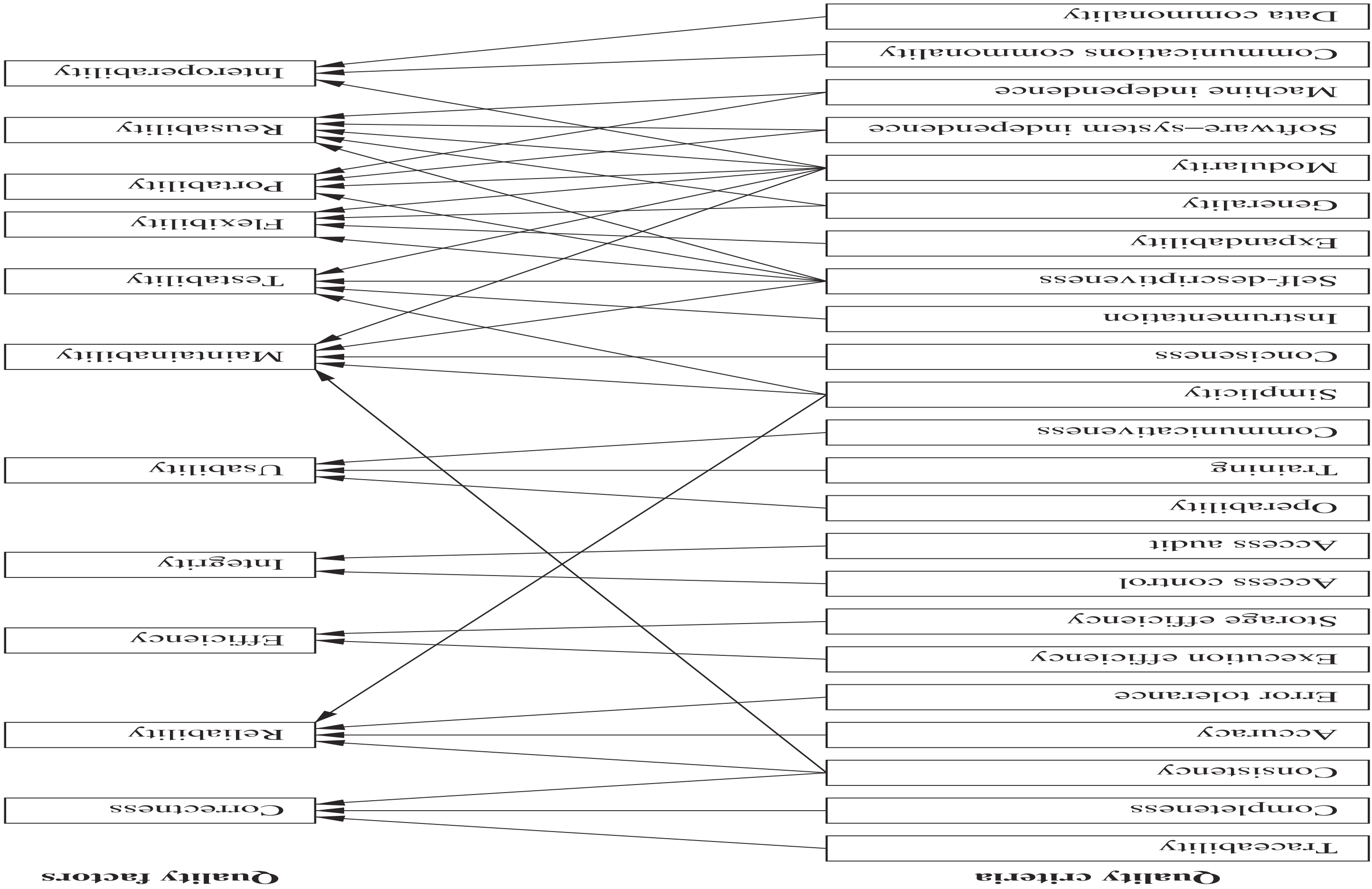
23 quality criteria of McCall
23 tiêu chí chất lượng đưa ra bởi McCall

Quality Criteria	Definition
Expandability (Khả năng mở rộng)	Degree to which storage requirements or software functions can be expanded
Generality (Tính khái quát hoá)	Breadth of the potential application of software components
Hardware independence (Độc lập phần cứng)	Degree to which the software is dependent on the underlying hardware
Instrumentation (Thiết bị đo đạc)	Degree to which the software provides for measurement of its use or identification of errors
Modularity (Tính đầy đủ)	Provision of highly independent modules
Operability (Khả năng vận hành)	Ease of operation of the software

23 quality criteria of McCall
23 tiêu chí chất lượng đưa ra bởi McCall

Quality Criteria	Definition
Self-documentation (Tự đặc tả tài liệu)	Provision of in-line documentation that explains implementation of components
Simplicity (Tính đơn giản)	Ease with which the software can be understood
Software system independence (Độc lập phần mềm)	Degree to which the software is independent of its software environment—nonstandard language constructs, operating system, libraries, database management system, etc.
Software efficiency (Hiệu năng phần mềm)	Run time storage requirements of the software
Traceability (Khả năng theo vết)	Ability to link software components to requirements
Training (Đào tạo)	Ease with which new users can use the system

Relationship between Quality factors and criteria
Mối quan hệ giữa nhân tố và tiêu chí chất lượng



Source: Software
Testing and Quality Assurance
Theory and Practice

Relationship between Quality factors and criteria Mối quan hệ giữa nhân tố và tiêu chí chất lượng

- 2 characteristics of relationship:
  - If an effort is made to improve one quality factor, another quality factor may be degraded
    - Make software testable ➔ the efficiency of the software is likely to go down
    - Make the product portable ➔reduce the efficiency
  - Some quality factors positively impacts others
    - an effort to enhance the correctness of a system will increase its reliability
    - an effort to enhance the testability of a system will improve its maintainability

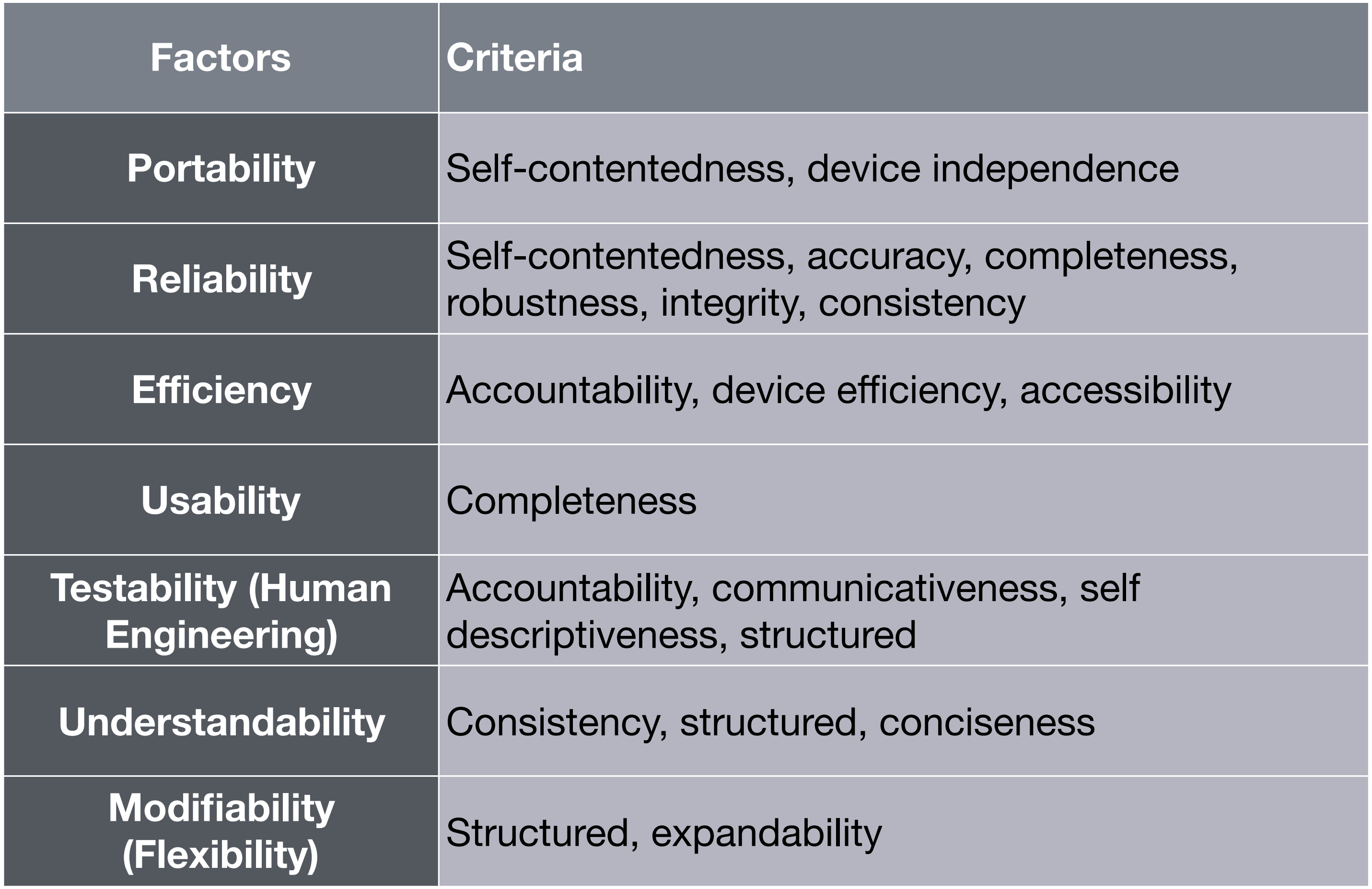
Quality Metrics
Độ đo chất lượng

- A **quality metric** is a measure that captures some aspects of a **quality criterion**
- The metrics can be derived as:
  - Formulate a set of relevant questions concerning the quality criteria and seek a **yes** or **no** answer for each question
  - Divide the number of yes answers by the number of questions to obtain a value between 0 and 1
- Example: **self-descriptiveness**
  - *Is all documentation written clearly and simply such that procedures, functions and algorithms can be easily understood?*
  - *Is the design rationale behind a module clearly understood?*

Boehm's Quality Model
Mô hình chất lượng của Boehm

- Attempts to automatically and qualitatively evaluate the quality of software
- High-level characteristics address three classifications:
  - Utility
  - Maintainability
  - Portability
- Intermediate-level characteristics address 7 quality factors

7 quality factors of Boehm’s Model



Factors	Criteria
Portability	Self-contentedness, device independence
Reliability	Self-contentedness, accuracy, completeness, robustness, integrity, consistency
Efficiency	Accountability, device efficiency, accessibility
Usability	Completeness
Testability (Human Engineering)	Accountability, communicativeness, self descriptiveness, structured
Understandability	Consistency, structured, conciseness
Modifiability (Flexibility)	Structured, expandability

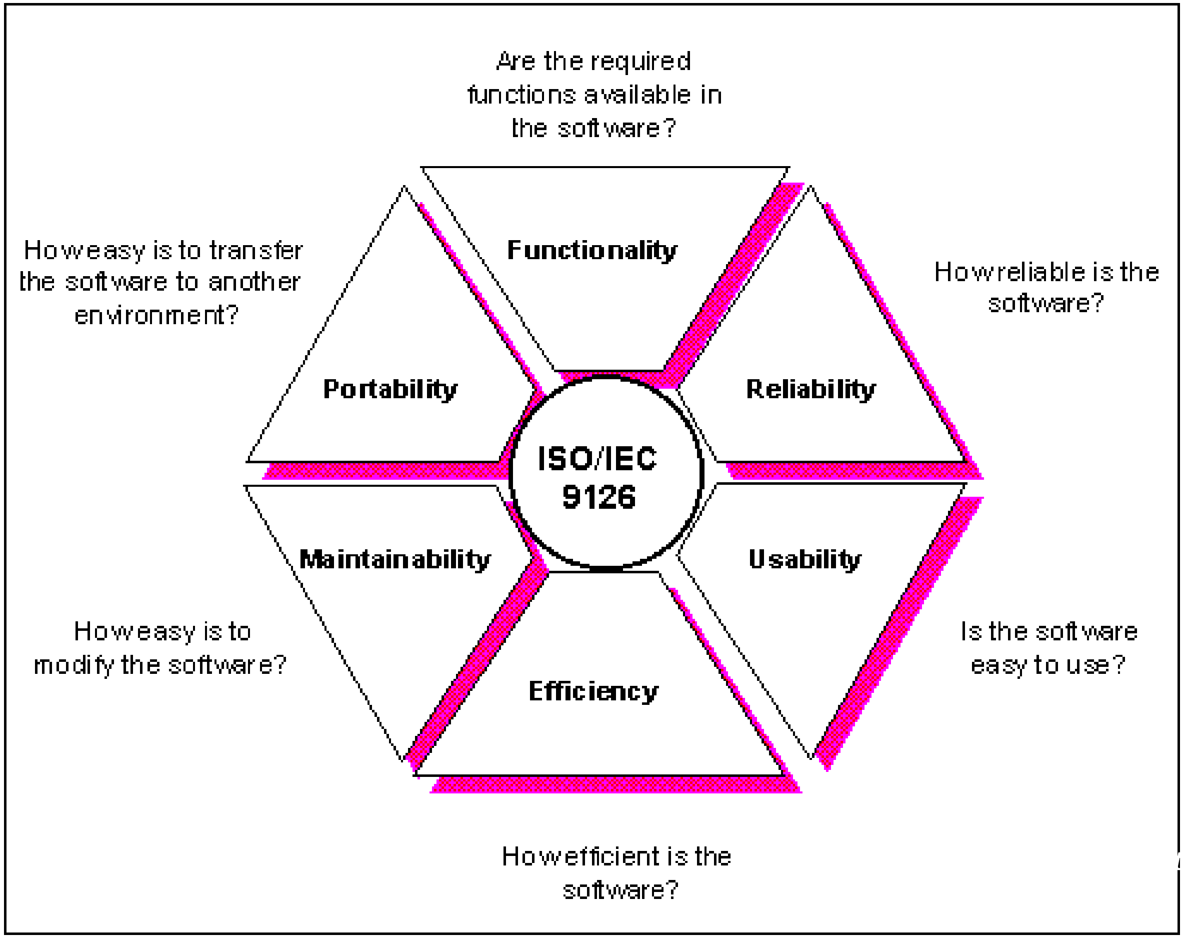
Dromey's Quality Model

- Dromey quality model proposed a framework to evaluate requirement, design and implementation phases
- High-level product properties for the implementation quality model include:
  - Correctness
  - Internal
  - Contextual
  - Descriptive

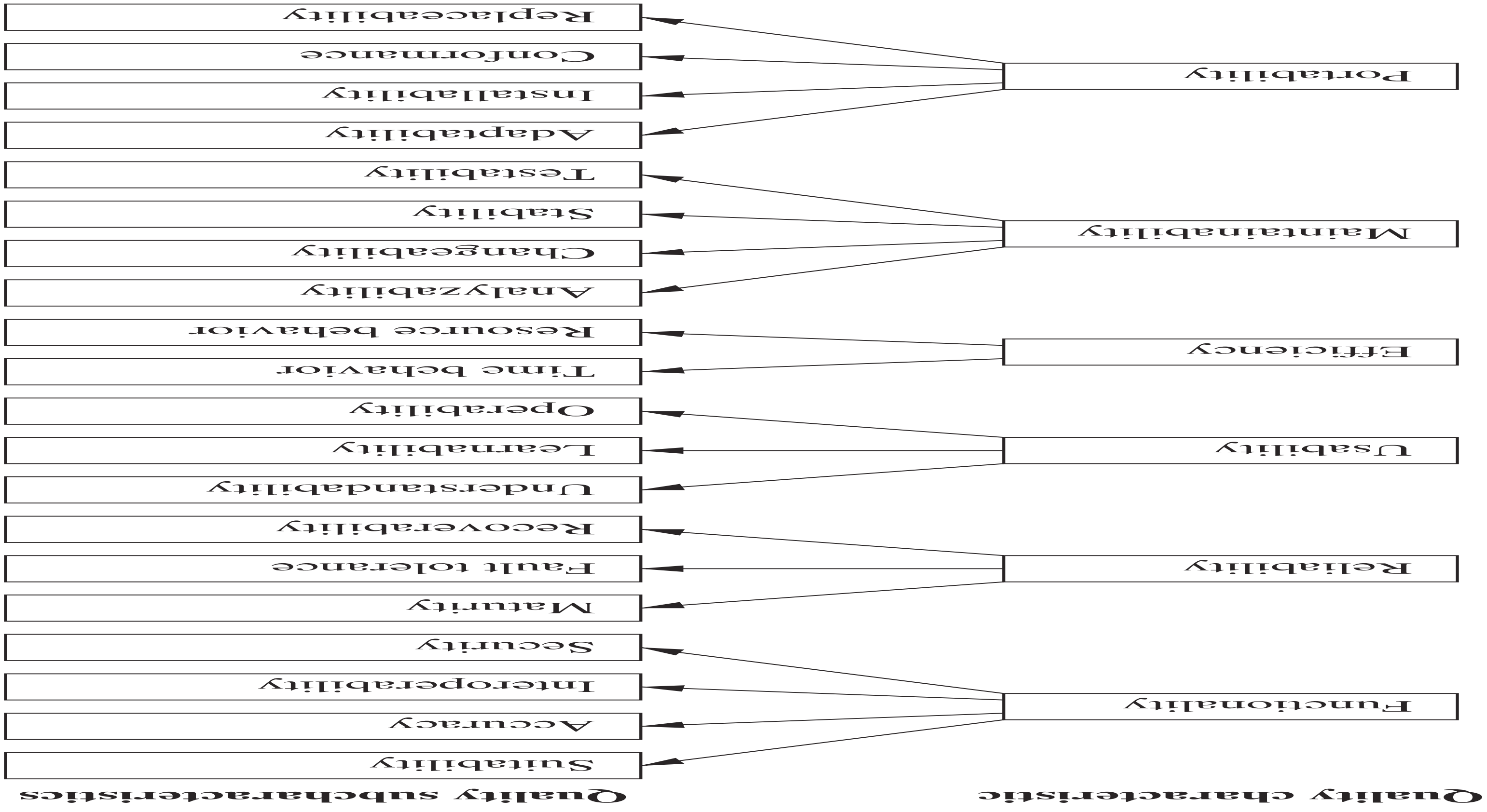
Dromey's quality factors and criteria
Các nhân tố và tiêu chí chất lượng của mô hình Dromey

Factors	Criteria
Correctness	Functionality, Reliability
Internal	Maintainability, Efficiency, Reliability
Contextual	Maintainability, Reusability, Portability, Reliability
Descriptive	Maintainability, Efficiency, Reliability, Usability

ISO/IEC 9126 Quality Model
Mô hình chất lượng ISO/IEC 9126


Are the required functions available in the software?
How easy is to transfer the software to another environment?
Functionality
How reliable is the software?
Portability
ISO/IEC 9126
Reliability
Maintainability
Usability
How easy is to modify the software?
Is the software easy to use?
Efficiency
How efficient is the software?

20 quality sub characteristics of ISO/IEC 9126



Comparison of McCall and ISO/IEC 9126 So sánh hai mô hình (sự tương đồng)

- Quality factor of McCall model is in fact quality characteristic in ISO/IEC model
- Quality criterion of McCall is sub-quality-characteristic of ISO/IEC model
- Some high-level quality factors/characteristics are found in both models
    - reliability
    - usability
    - efficiency
    - maintainability
    - portability

Comparison of McCall and ISO/IEC 9126 So sánh hai mô hình (sự khác nhau)

- ISO 9126 model emphasizes **characteristics visible to users** whereas McCall model considers **internal qualities**
  - Ex: Developpers produce reusable components whereas its impact is not perceived by customers
- In McCall's model, one quality criterion can impact several quality factors whereas in the ISO 9126 model, one subcharacteristic impacts exactly one quality characteristic
- A high-level **quality factor**, **testability**, in the McCall model is a low-level **subcharacteristic** of **maintainability** in the ISO 9126 model

Hewlett Packard: F.U.R.P.S
Mô hình FURPS của HP

- Result of a statistical project survey at Hewlett Packard 1987 to improve its products:
- Factors:
  - **F**unctionality: functions it performs, their generality and security
  - **U**sability: consistency, documentation
  - **R**eliability: frequency and severity of failure, accuracy of output
  - **P**erformance: response time, resource consumption
  - **S**upportability: can it be extended, adapted, corrected?
- FURPS is originally a company specific quality model

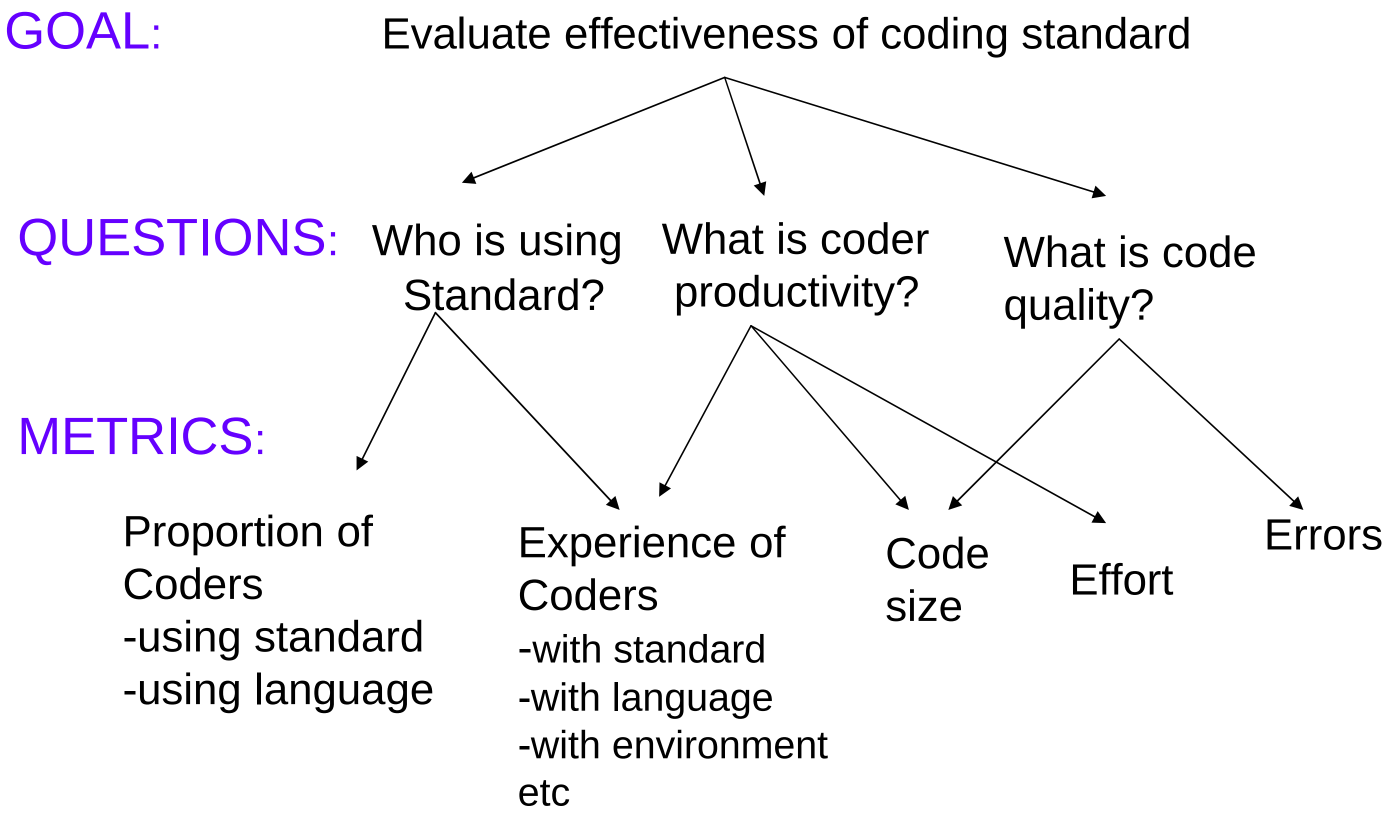
Factors and Criteria of FURPS

Factors	Criteria
Functionality	Capability, security
Usability	Consistency, user documentation, training materials
Reliability	Frequency and security of failure, recoverability, predictability, accuracy, mean time between failure
Performance	Speed efficiency, availability, accuracy, throughput, response time, recovery time, resource usage
Supportability	Testability, extensibility, adaptability, maintainability, compatibility, configurability, serviceability, install ability, localizability

GQM: Goal-Question-Metric
Mô hình GQM

- A measurement program can be more successful if designed with the goals in mind
- GQM approach provides a framework with 3 steps:
  - List the major goals of the development/maintenance project
  - Derive from each goal the questions that must be answered to determine if the goals are being met
  - Decide what must be measured to answer the questions adequately

Example of GQM



GOAL: Evaluate effectiveness of coding standard

QUESTIONS: Who is using Standard? What is coder productivity? What is code quality?

METRICS:

Proportion of Coders
-using standard
-using language

Experience of Coders
-with standard
-with language
-with environment
etc

Code size

Effort

Errors

GQM Discussion
Ưu điểm và nhược điểm của mô hình GQM

- Benefits:
  - generates only those measures relevant to the goal
  - several measurements may be needed to answer a single question
  - a single measurement may apply to more than one question
  - the goal provides the purpose for collecting the data
- Not evident from the GQM
  - The model needed to combine the measurement in a sensible way so that the question can be answered
  - It must be supplemented by one or more models that express the relationship among metrics
- Disadvantages:
  - Additional efforts to derive the goals and metrics
  - Error prone compared to standard models

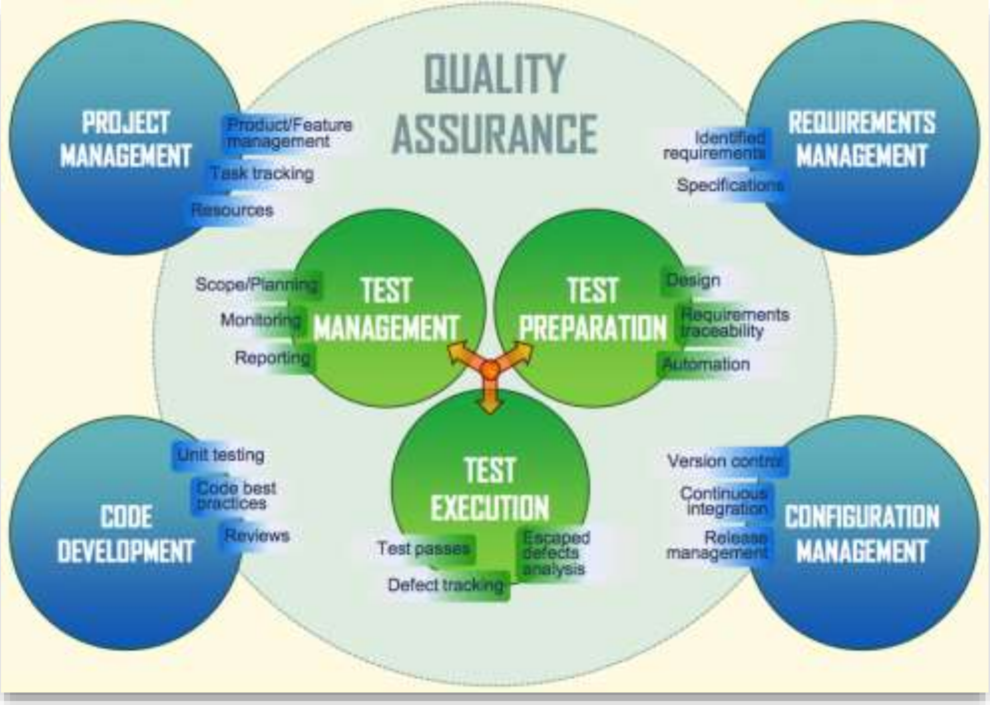
1.4. Quality Assurance Process

Quy trình đảm bảo chất lượng phần mềm

"Quality Assurance is any systematic process of checking to see whether a product or service being developed is meeting its specified requirements"

QA Process

- Need to be integrated with any other processes in the development life cycle of the software system
- Project management process
- Code Development process
- Requirements Management process
- Configuration Management process



QA Dependencies – Project Management Vai trò của QA trong quá trình quản lý dự án

- QA should be considered in planning stage
  - Define quality goals & criteria
  - Identify applicable QA/testing procedures
  - Pre-assign necessary resources (including personnel, hardware and infrastructure environment)

QA Dependencies - Requirements Management Vai trò của QA trong quản lý yêu cầu phần mềm

- Keep product requirements in a well organized way
    - To identify all important aspects of functionality and receive customer's confirmation
    - To define restrictions and exceptions
    - To establish common understanding across the entire project team
    - To provide basis for project estimates, code development and validation processes
    - To be a starting point for effective communication and change management during product evolution process

QA Dependencies - Configuration Management QA đối với quy trình quản lý cấu hình

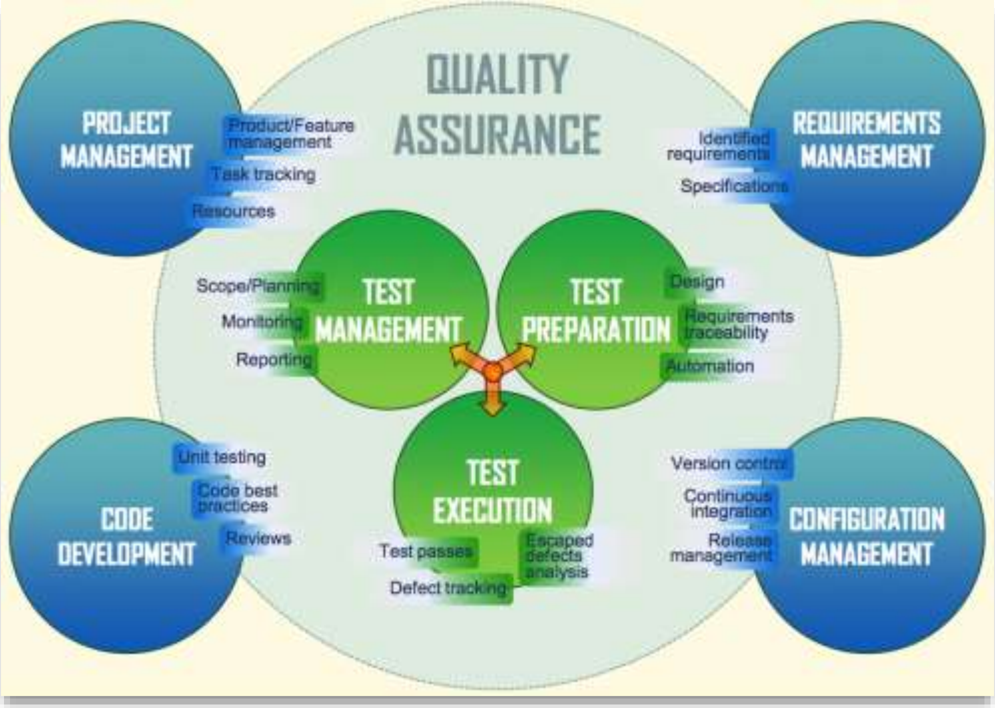
- Configuration Management presents on any software project
  - To allow distributed teams simultaneously to work on the same code
  - To keep the code baseline free from breakages introduced by bug fixing
  - To know code version where new error appeared for the first time and to identify what change had introduced it
  - To realize all advantages of automated testing on each and every code version published

QA Dependencies - Code Development
Vai trò của QA đối với quy trình phát triển mã nguồn

- Benefits of applying a Code Development Process
  - Unified coding style suitable for further product maintenance and updates without single team member lock-in
  - Earliest breakage detection and the most effective cost on fixing bug
  - Test-oriented code design
  - Localize errors and minimize impacts on dependent product components
  - Share knowledge between team members and Improve personal skills

QA Core Components - Testing Process
Quy trình kiểm thử là nền tảng của quy trình QA

- Testing is implemented in separate and/or overlapped test phase/types
  - Testing on different product levels: unit, feature, features interaction, system
  - Testing during product maintenance: smoke, sanity, regression
  - Testing of special aspects: bring-up, stress, stability, performance, interoperability, compliance



Test Management
Quản lý kiểm thử trong quy trình QA

- Perform from the very beginning through the end of a project to achieve goals of improved product quality
- Major Objects of Test Management
  - Testing scope / strategy
  - Approaches and Constraints
  - Features / Components / Functional Areas to be or not to be tested
  - Test Item Pass / Fail and Test Cycle Suspension / Resumption Criteria
  - Test Tasks and Deliverables, Environmental Needs
  - Responsibilities, Staffing and Training Needs / Collaboration with other project groups
  - Schedule, Risk and Contingencies
  - Reporting and Quality Metrics

Test Preparation
Chuẩn bị kiểm thử trong quy trình QA

- Define tests and test environment required for each test phase/type according to test plans
- Basic test preparation activities
  - Analysis of functional requirements, specifications and functional areas's
  - Development of Requirements Traceability Matrix (RTM)
  - Design and Development of the test cases for all planned phases/types
  - Using approaches and techniques to ensure effective and optimal test suites
    - Various test approaches (Black Box, White Box)
    - Effective test techniques: Functionality, Equivalent classes, Boundary, Negative/Positive Combinatorial..
  - Development of additional tools/utilities and Test automation
  - Development and setup of specific test environment

Test Execution
Thực thi kiểm thử trong quy trình QA

- Straightforward apply test definitions elaborated during the Test Preparation Phase to the software product
- Usual Test Execution activities:
  - Preparation of developed test environment
  - Execution of test cases
  - Deep analysis of execution results
  - Collecting/Recording of test execution results and failures found
  - Support development team to reproduce and to analyze failures
  - Validation of fixes
  - Analysis of escaped defects
  - Update test cases/test environment according to the results of Test Execution/Escaped Defect Analysis

Summary
Tóm lược lại nội dung bài học

- Principal terms of Quality Assurance
- IEEE’s definition of Software Quality
- Quality models: McCall, Boehm, Dromey, ISO/IEC 9126, FURPS, GQM
- QA process and main activities of:
  - Test Preparation
  - Test Management
  - Test Execution

25 SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you
for your
attention!!!

soict.hust.edu.vn/ fb.com/groups/soict