



جامعة الحمدانية

كلية التربية للعلوم الصرفة

قسم علوم الحاسوب

المرحلة الثالثة

هندسة البرمجيات

Software Engineering

Software Quality Management

Software Quality

What is Software Quality?

- **Software Quality is:** [IEEE Definition]
 - The degree to which a system, component, or process meets specified requirements
[based on Philip B. Crosby's definition, 1979]
 - The degree to which a system, component or process meets customer or user needs or expectations. [based on Joseph M. Juran, 1988].

■ جودة البرنامج هي:

■ الدرجة التي يلبي بها النظام أو المكون أو العملية المتطلبات المحددة [استنادًا إلى تعريف فيليب ب. كروسبي، 1979]

■ الدرجة التي يلبي بها النظام أو المكون أو العملية احتياجات أو توقعات العميل أو المستخدم. [استنادًا إلى جوزيف م.

جوران، 1988].

Software Quality Assurance

■ Software quality assurance is:

- A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.
- A set of activities designed to evaluate the process by which the products are developed or manufactured.

■ ضمان جودة البرمجيات هو:

■ نمط مخطط ومنهجي لجميع الإجراءات اللازمة لتوفير الثقة الكافية بأن العنصر أو المنتج يتوافق مع المتطلبات الفنية المحددة.

■ مجموعة من الأنشطة المصممة لتقييم العملية التي يتم من خلالها تطوير أو تصنيع المنتجات.

When should software quality assurance be done?

- At **every stage** in the software **process** !

What to do in SQA?

- Monitoring **processes** and **products** throughout the software development life cycle
- To ensure the **quality** of the delivered product(s)

■ مراقبة العمليات والمنتجات طوال دورة حياة تطوير البرمجيات

■ لضمان جودة المنتج (المنتجات) المسلمة

Standards and Disciplines

- Because of the importance of quality in software production and the relatively bad track record of the past, many standards and disciplines have been developed to enforce quality practice
- Examples are:
 - Total Quality Management (TQM),
 - Capability Maturity Model (CMM),
 - Personal Software Process (PSP),
 - Microsoft Shared Development Process (SDP),
 - Rational Unified Process (RUP),
 - and ISO 9000

McCall's Factor Model

- McCall proposed 11 quality factors; Grouped into 3 categories.
 - 1977; others have added, but this still prevail.
- **Product operation factors:** Factors that deal with requirements that directly affect the daily operation of the software.
 - How well does it run & ease of use?
 - Correctness, reliability, efficiency, integrity, and usability

■ عوامل تشغيل المنتج: العوامل التي تتناول المتطلبات والتي تؤثر بشكل مباشر على التشغيل اليومي للبرنامج.

■ ما مدى جودة تشغيله وسهولة استخدامه؟

■ الصحة والموثوقية والكفاءة والسلامة وسهولة الاستخدام

- **Product revision factors:** Factors that deal with requirements that affect software maintenance.
 - How well it can be changed, tested, and redeployed?
 - Maintainability; flexibility; testability
- **Product transition factors:** Factors that deal with requirements that affect the adaptation of software to other platforms, environments, interaction with other software.
 - How well it can be moved to different platforms and interface with other systems?
 - Portability; Reusability; Interoperability

■ عوامل مراجعة المنتج: العوامل التي تتناول المتطلبات التي تؤثر على صيانة البرمجيات.

■ ما مدى إمكانية تغييرها واختبارها وإعادة نشرها؟

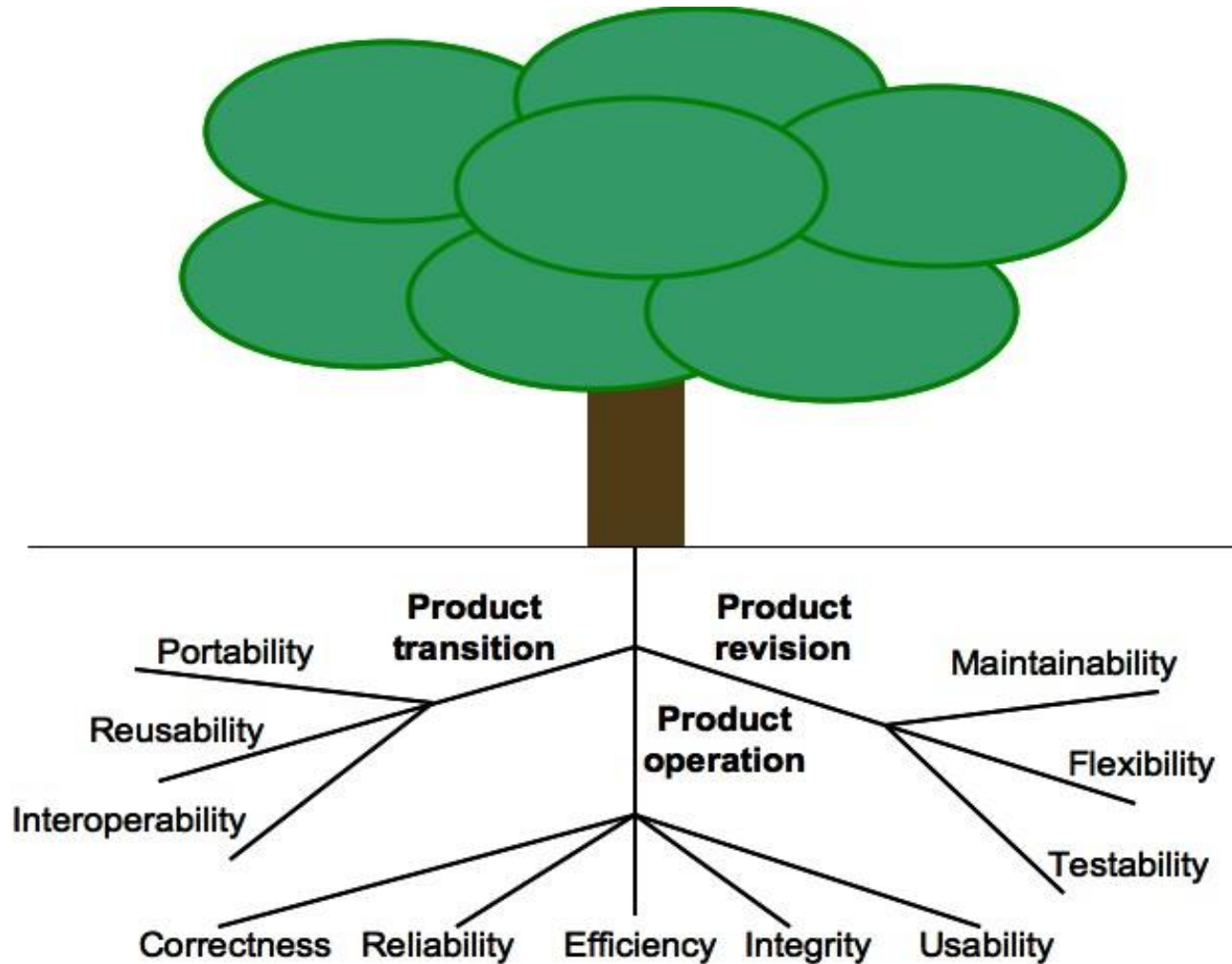
■ قابلية الصيانة؛ المرونة؛ قابلية الاختبار

■ عوامل انتقال المنتج: العوامل التي تتعامل مع المتطلبات التي تؤثر على تكيف البرامج مع الأنظمة الأساسية والبيئات الأخرى والتفاعل مع البرامج الأخرى.

■ ما مدى إمكانية نقلها إلى منصات مختلفة والتفاعل مع الأنظمة الأخرى؟

■ قابلية التنقل؛ إعادة الاستخدام؛ التوافقية

McCall's Factor Model Tree



Product Operation Factors

- **Correctness** - the presence and appropriateness of a set of functions for specified tasks; lack of bugs and defects
 - measured in terms of defect rate (# bugs per line of code)
- Software system's required outputs should be correct (such as a query display of a customer's balance, sales invoice printout and red alarms when temperature rises above 250°F)

- الصحة - وجود وملاءمة مجموعة من الوظائف لمهام محددة؛ عدم وجود الأخطاء والعيوب
- يتم قياسه من حيث معدل العيوب (# أخطاء لكل سطر من التعليمات البرمجية)
- يجب أن تكون المخرجات المطلوبة للنظام البرمجي صحيحة (مثل عرض الاستعلام عن رصيد العميل وطباعة فاتورة المبيعات والإنذارات الحمراء عند ارتفاع درجة الحرارة فوق 250 درجة فهرنهايت)

- **Reliability** - capability to maintain its level of performance under stated conditions for a stated period of time; attribute that bears on the frequency of failure by faults in the software; does not fail or crash often
 - measured in terms of failure rate (#failures per hour), Mean time between failures (MTBF); Mean time to failure (MTTF)
- Reliability requirements deal with failures to provide service.
- They determine the maximum allowed software system failure rate, and can refer to the entire system or to one or more of its separate functions.

- الموثوقية - القدرة على الحفاظ على مستوى أدائها في ظل ظروف محددة لفترة زمنية محددة؛ السمة التي تؤثر على تكرار الفشل بسبب أخطاء في البرنامج؛ لا تفشل أو تتعطل في كثير من الأحيان
- يتم قياسه من حيث معدل الفشل (عدد مرات الفشل في الساعة)، ومتوسط الوقت بين حالات الفشل ((MTBF)؛ متوسط الوقت حتى الفشل ((MTTF))
- تتعامل متطلبات الموثوقية مع الفشل في تقديم الخدمة.
- وهي تحدد الحد الأقصى المسموح به لمعدل فشل نظام البرمجيات، ويمكن أن تشير إلى النظام بأكمله أو إلى واحدة أو أكثر من وظائفه المنفصلة.

Example:

- The failure frequency of a heart-monitoring unit that will operate in a hospital's intensive care ward is required to be less than one in 20 years.
- Its heart attack detection function is required to have a failure rate of less than one per million cases.
- **Downtime** for a system will not be more than ten minutes per month

مثال:

- ومن المفترض أن يكون معدل تكرار فشل وحدة مراقبة القلب التي ستعمل في جناح العناية المركزة في المستشفى أقل من واحد كل عشرين عامًا.
- ومن المطلوب أن تكون وظيفة الكشف عن النوبات القلبية لديها معدل فشل أقل من حالة واحدة لكل مليون حالة.
- لن تزيد مدة توقف النظام عن عشر دقائق شهريًا

Efficiency: relationship between the level of performance of the software and the amount of resources used, under stated conditions; is fast and small enough.

Integrity: Integrity requirements deal with the software system security that prevent unauthorized persons access.

Usability: effort needed for use; is sufficiently convenient for the intended users. Usability requirement deals with the scope of staff resources needed to train a new employee to operate the software system.

Maintainability: effort needed to make modifications, including corrections, improvements or adaptation of software to changes in environment, requirements, and functional specifications; is easy to change and adapt to new requirements

- **Flexibility:** The capabilities and efforts required to adapt a software package to changes in the environment/different types of customers that use the software perhaps a little differently.

Example:

- TSS (teacher support software) deals with the documentation of student achievements, the calculation of final grades, the printing of term grade documents, and the automatic printing of warning letters to parents of failing students.
- The software specifications included the following flexibility requirements:
 - The software should be suitable for teachers of all subjects and all school levels (elementary, junior and high schools).

المرونة: القدرات والجهود المطلوبة لتكييف حزمة البرامج مع التغيرات في البيئة/الأنواع المختلفة من العملاء الذين يستخدمون البرنامج ربما بشكل مختلف قليلاً.

مثال:

يتعامل (TSS برنامج دعم المعلم) مع توثيق إنجازات الطلاب، وحساب الدرجات النهائية، وطباعة وثائق درجات الفصل الدراسي، والطباعة التلقائية لرسائل التحذير لأولياء أمور الطلاب الراسبين.

تضمنت مواصفات البرنامج متطلبات المرونة التالية:

يجب أن يكون البرنامج مناسباً لمعلمي جميع المواد وجميع المراحل الدراسية (الابتدائية والإعدادية والثانوية).

- **Testability:** Testability requirements for the ease of testing are related to special features in the programs that help the tester.
- **Portability:** ability to transfer the software from one organization or hardware or software environment to another.
- **Reusability:** Reusability requirements deal with the use of software modules originally designed for one project in a new software project currently being developed.
- **Interoperability:** software's ability to interact with other specified systems. Interoperability requirements focus on creating interfaces with other software systems.

McCall's Factor Model & Other alternative models

| No. | Software quality factor | McCall's classic model | Alternative factor models | |
|-----|-------------------------|------------------------|---------------------------|--------------------------|
| | | | Evans and Marciniak model | Deutsch and Willis model |
| 1 | Correctness | + | + | + |
| 2 | Reliability | + | + | + |
| 3 | Efficiency | + | + | + |
| 4 | Integrity | + | + | + |
| 5 | Usability | + | + | + |
| 6 | Maintainability | + | + | + |
| 7 | Flexibility | + | + | + |
| 8 | Testability | + | | |
| 9 | Portability | + | + | + |
| 10 | Reusability | + | + | + |
| 11 | Interoperability | + | + | + |
| 12 | Verifiability | | + | + |
| 13 | Expandability | | + | + |
| 14 | Safety | | | + |
| 15 | Manageability | | | + |
| 16 | Survivability | | | + |