Software Quality

Philipp Roebrock

Institute for Photonics and ICP (IPI)

September 13, 2019

Fachhochschule Graubünden
University of Applied Sciences

Contents I

Intro

Intro

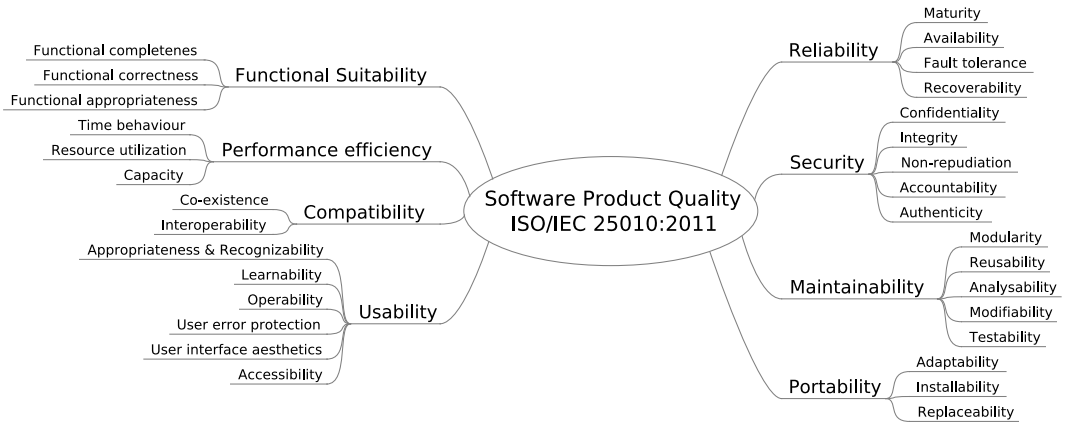
Video

Learning Objective

- Know different categories and types of software product quality
- Be able to tell those types apart in examples.

Definition: Software product quality

Software product quality defined by ISO/IEC25010



Functional suitability (1/8)

Functional Completeness: Degree to which the set of functions covers all the specified tasks and user objectives

Functional Correctness: Degree to which the functions provides the correct results with the needed degree of precision.

Functional Appropriateness: Degree to which the functions facilitate the accomplishment of specified tasks and objectives

Performance & Efficiency (2/8)

Time-behavior: Degree to which the response and processing times and throughput rates of a product or system, when performing its functions, meet requirements

Resource Utilization: Degree to which the amounts and types of resources used by a product or system, when performing its functions, meet requirements.

Capacity: Degree to which the maximum limits of the product or system, parameter meet requirements

Compatibility (3/8)

Co-existence: Degree to which a product can perform its required functions efficiently while sharing a common environment and resources with other products, without detrimental impact on any other product

Interoperability: Degree to which two or more systems, products or components can exchange information and use the information that has been exchanged

Usability (4/8)

Appropriateness & Recognisability: Degree to which users can recognize whether a product or system is appropriate for their needs

Learnability: Degree to which a product or system enables the user to learn how to use it with effectiveness, efficiency in emergency situations

Operability: Degree to which a product or system is easy to operate, control and appropriate to use

User error protection: Degree to which a product or system protects users against making errors

User interface aesthetics: Degree to which a user interface enables pleasing and satisfying interaction for the user

Accessibility: Degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use

Reliability (5/8)

Maturity: Degree to which a system, product or component meets needs for reliability under normal operation

Availability: Degree to which a product or system is operational and accessible when required for use

Fault tolerance: Degree to which a system, product or component operates as intended despite the presence of hardware or software faults

Recoverability: Degree to which, in the event of an interruption or a failure, a product or system can recover the data directly affected and re-establish the desired state of the system

Security (6/8)

Confidentiality: Degree to which the prototype ensures that data are accessible only to those authorized to have access

Integrity: Degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data

Non-repudiation: Degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later

Accountability: Degree to which the actions of an entity can be traced uniquely to the entity

Authenticity: Degree to which the identity of a subject or resource can be proved to be the one claimed

Maintainability (7/8)

Modularity: Degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components

Reusability: Degree to which an asset can be used in more than one system, or in building other assets

Analyzability: Degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified

Modifiability: Degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality

Testability: Degree of effectiveness and efficiency with which test criteria can be established for a system, product or component and tests can be performed to determine whether those criteria have been met

Portability (8/8)

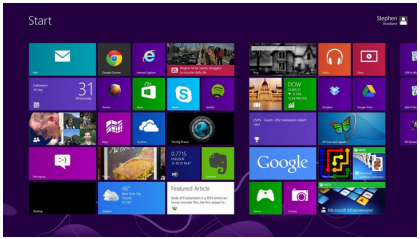
Adaptability: Degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments

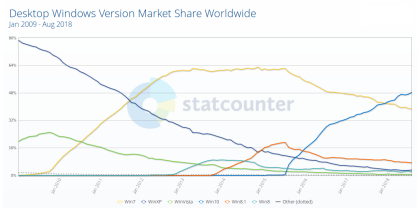
Installability: Degree of effectiveness and efficiency in which a product or system can be successfully installed and/or uninstalled in a specified environment

Replaceability: Degree to which a product can replace another specified software product for the same purpose in the same environment

Famous examples of insufficient software quality

Example 1/6: User Interface of Windows 8





CNET, Nov. 2012

Quotes:

- Microsoft launched Windows 8 late last month. The operating system is a major departure from previous versions of the software, ditching the traditional start button and design for a tile-based system.
- Many people who used the software, however, have criticized it for a steep learning curve that impacts both novices and experienced PC users.

Example 2/6: Trading firm's computer system out of control

Knight Capital Says Trading Glitch Cost It $440 Million

BY NATHANIEL POPPER AUGUST 2, 2012 9:07 AM 356

Runaway Trades Spread Turmoil Across Wall St.

Errant trades from the Knight Capital Group began hitting the New York Stock Exchange almost as soon as the opening bell rang on Wednesday. Brendan McDermid/Reuters

1 of 4

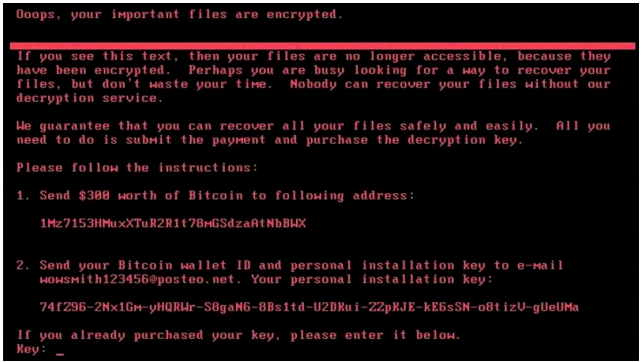
New York Times, 02 Aug 2012

Quotes:

- The problem on Wednesday led the firm's computers to rapidly buy and sell millions of shares in over a hundred stocks for about 45 minutes after the markets opened.
- $10 million [loss] a minute
- The company said the problems happened because of new trading software that had been installed.

Example 3/6: 2017 cyberattacks on Ukraine

Different vulnerabilities in Microsoft's operating systems allowed attacks on computer systems of major companies:



WIRED, Sep 2018

Losses

- Merck: US$870 million
- FedEx: US$400 million
- Saint-Gobain: US$384 million
- Maersk: US$300 million
- Estimated total damage: US$10 billion

Example 4/6: Losing an astronomy satellite

Software error doomed Japanese Hitomi spacecraft

Space agency declares the astronomy satellite a loss.

Alexandra Witze

28 April 2016

PDF | Rights & Permissions

STR/AFP/Getty

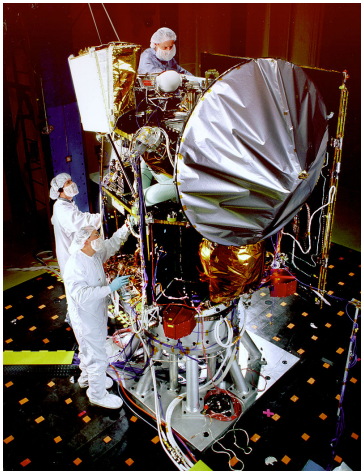
The Japan Aerospace Exploration Agency is investigating the factors that led to Hitomi's demise.

Nature 533, 18-19 (05 May 2016)

Quotes:

- On 28 April, the Japan Aerospace Exploration Agency (JAXA) declared the satellite, on which it had spent ¥31 billion (US$286 million), lost.
- After a failure the spacecraft then automatically switched into a safe mode and, at about 4:10 a.m., fired thrusters to try to stop the rotation. But because the wrong command had been uploaded, the firing caused the spacecraft to accelerate further. (The improper command had been uploaded to the satellite weeks earlier without proper testing)

Example 5/6: Mars climate orbiter



Wikipedia

- The Mars Climate Orbiter was a robotic space probe launched by NASA on December 11, 1998.
- The spacecraft encountered Mars on a trajectory that brought it too close to the planet, causing it to pass through the upper atmosphere and disintegrate.
- The primary cause of this discrepancy was that one piece of ground software supplied by Lockheed Martin produced results in a United States customary unit, contrary to its Software Interface Specification (SIS), while a second system, supplied by NASA, expected those results to be in SI units, in accordance with the SIS.
- The cost of the mission was US$327.6 million

Example 6/6: FBI Virtual Case File project

Wikipedia

- Virtual Case File (or VCF) was a software application developed by the United States Federal Bureau of Investigation (FBI) between 2000 and 2005.
- The project was officially abandoned in April 2005, while still in development stage and cost the federal government nearly $170 million.
- The software was incomplete, inadequate and so poorly designed that it would be essentially unusable under real-world conditions. Even in rudimentary tests, the system did not comply with basic requirements.
- Reasons for failure (among others): Repeated changes in specification, Repeated turnover of management, which contributed to the specification problem, Scope creep as requirements were continually added to the system even as it was falling behind schedule

Conclusions

Conclusions

- Software products have an increasing impact on our daily life.
- Therefore software product quality is an important matter.
- Software product quality has many aspects.
- Now we **know** different types and subtypes of software product quality and we can **tell them apart**.

The end

Questions? Remarks?