Portability testing

Portability testing is the process of determining the degree of ease or difficulty to which a software component or application can be effectively and efficiently transferred from one hardware, software or other operational or usage environment to another.[1] The test results, defined by the individual needs of the system, are some measurement of how easily the component or application will be to integrate into the environment and these results will then be compared to the software system's non-functional requirement of portability[2] for correctness. The levels of correctness are usually measured by the cost to adapt the software to the new environment[3] compared to the cost of redevelopment.[4]

Contents

Use cases

When multiple subsystems share components of a larger system, portability testing can be used to help prevent propagation of errors throughout the system.[5] Changing or upgrading to a newer system, adapting to a new interface or interfacing a new system in an existing environment are all problems that software systems with longevity will face sooner or later and properly testing the environment for portability can save on overall cost throughout the life of the system.[5] A general guideline for portability testing is that it should be done if the software system is designed to move from one hardware platform, operating system, or web browser to another.[6]

Examples

- Software designed to run on Macintosh OS X and Microsoft Windows operating systems.[7]
- Applications developed to be compatible with Google Android and Apple iOS phones.[7]
- Video Games or other graphic intensive software intended to work with OpenGL and DirectX API's.[7]
- Software that should be compatible with Google Chrome and Mozilla Firefox browsers.[7]

Attributes

There are four testing attributes included in portability testing. The ISO 9126 (1991) standard breaks down portability testing attributes[5] as Installability, Compatibility, Adaptability and Replaceability. The ISO 29119 (2013) standard describes Portability with the attributes of Compatibility, Installability, Interoperability and

Localization testing.[8]

- Adaptability testing- Functional test to verify that the software can perform all of its intended behaviors in each of the target environments.[9][10] Using communication standards, such as HTML can help with adaptability. Adaptability may include testing in the following areas: hardware dependency, software dependency, representation dependency, standard language conformance, dependency encapsulation and/or text convertibility.[5]
- Compatibility/ Co-existence- Testing the compatibility of multiple, unrelated software systems to co-exist in the same environment, without effecting each other's behavior.[9][11][12] This is a growing issue with advanced systems, increased functionality and interconnections between systems and subsystems who share components. Components that fail this requirement could have profound effects on a system. For example, if 2 sub-systems share memory or a stack, an error in one could propagate to the other and in some cases cause complete failure of the entire system.[5]
- Installability testing- Installation software is tested on its ability to effectively install the target software in the intended environment.[5][9][13][14] Installability may include tests for: space demand, checking prerequisites, installation procedures, completeness, installation interruption, customization, initialization, and/or deinstallation.[5]
- Interoperability testing- Testing the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units.[1]
- Localization testing- Localization is also known as internationalization. Its purpose is to test if the software can be understood in using the local language where the software is being used.[8]
- Replaceability testing- Testing the capability of one software component to be replaced by another software component within a single system. The system, in regards to the replaced component, should produce the same results that it produced before the replacement.[9][15][16] The issues for adaptability also apply for replaceability, but for replaceability you may also need to test for data load-ability and/or convertibility.[5]

See also

- Porting
- Software portability
- Software system
- Software testing
- Software testability
- Application portability
- Operational Acceptance

References

1. "ISO/IEC/IEEE 29119-4 Software and Systems Engineering - Software Testing -Part 4- Test Techniques" (http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=60245).
2. "Portability Testing" (http://www.opfro.org/index.html?Components/WorkUnits/Activities/Testing/PortabilityTesting.html~Contents). OPEN Process Framework Repository Organization. Retrieved 29 April 2014.
3. Rouse, Margaret. "DEFINITION environment" (http://whatis.techtarget.com/definition/environment). Retrieved 29 April 2014.

4. Mooney, James. "Bringing Portability to the Software Process" (https://web.archive.org/web/20080725004932/http://www.cs.wvu.edu/~jdm/research/portability/reports/TR_97-1.pdf) (PDF). Archived from the original (http://www.cs.wvu.edu/~jdm/research/portability/reports/TR_97-1.pdf) (PDF) on 25 July 2008. Retrieved 29 April 2014.
5. Hass, Anne Mette Jonassen (2008). *Guide to advanced software testing* ([Online-Ausg.] ed.). Boston: Artech House. pp. 271–272. ISBN 978-1596932852.
6. Salonen, Ville. "Automatic Portability Testing" (https://jyx.jyu.fi/dspace/bitstream/handle/123456789/40043/URN%3ANBN%3Afi%3Ajyu-201210212735.pdf?sequence=1) (PDF). Retrieved 29 April 2014.
7. Salonen, Ville (October 17, 2012). "Automatic Portability Testing" (https://jyx.jyu.fi/dspace/bitstream/handle/123456789/40043/URN%3ANBN%3Afi%3Ajyu-201210212735.pdf?sequence=1) (PDF). Ville Salonen. pp. 11–18. Retrieved 15 May 2014.
8. Woods, Anthony (2015). "Operational Acceptance - an application of the ISO 29119 Software Testing standard".
9. "ISTQB Advanced Level Syllabi" (http://www.astqb.org/educational-resources/advanced-characteristics5.php). ASTQB. Retrieved 29 April 2014.
10. Hass, Anne Mette Jonassen (2008). *Guide to advanced software testing* ([Online-Ausg.] ed.). Boston: Artech House. pp. 272–273. ISBN 978-1596932852.
11. "What is Compatibility testing in Software testing?" (http://istqbexamcertification.com/what-is-compatibility-testing-in-software/). Mindstream Theme on Genesis Framework. Retrieved 29 April 2014.
12. Hass, Anne Mette Jonassen (2008). *Guide to advanced software testing* ([Online-Ausg.] ed.). Boston: Artech House. p. 272. ISBN 978-1596932852.
13. "Installability Guidelines" (http://www.testingstandards.co.uk/installability_guidelines.htm). Retrieved 29 April 2014.
14. "What is Portability testing in software?" (http://istqbexamcertification.com/what-is-portability-testing-in-software/). Mindstream Theme. Retrieved 29 April 2014.
15. "Replaceability" (http://maisqual.squoring.com/wiki/index.php/Replaceability). Retrieved 29 April 2014.
16. Hass, Anne Mette Jonassen (2008). *Guide to advanced software testing* ([Online-Ausg.] ed.). Boston: Artech House. p. 273. ISBN 978-1596932852.