App Stores & ISO/IEC 25000: Product Certification at Last?

John Estdale

IT Architecture Specialists Ltd,
The Springers, Broadlayings, Woolton Hill, NEWBURY
john.estdale@bcs.org

Abstract

The idea that software products could usefully be assessed and certified by an independent 3rd party organisation has been around for many years. Back in the 1990s, the European Union's ESPRIT II program proposed investigating the feasibility of a programme for the European certification of software quality, to support certification schemes for industries such as rail and motor manufacture. Unfortunately commercial success seems to have eluded them.

Second party certification is more common. Many supplier organisations go to Microsoft to endorse devices as "Certified for Windows", and software as "Windows *n* Compatible" or "Certified for Windows Server yyyy".

But the world has moved on. According to one survey there are now 81 different smartphone App Stores, with varying kinds of 'quality promise'. This paper compares the requirements of the Apple App Store, and the Microsoft Windows Store, with the software quality model in the relevant International Standard series ISO/IEC 25000, for Systems and software Quality Requirements and Evaluation (SQuaRE), and uses them to identify some strengths and weaknesses of this quality model.

Keywords: Software certification, software product quality, Quality Model, ISO 25000, SQuaRE, App Stores

1.0 Introduction

Software products are initially considered for purchase based on their expected usefulness: their user-visible functions and features. Inevitably they will have practical limitations, in reliability, capacity, performance, maintainability, etc, informally summarised as their Non-functional attributes or 'quality'.

The idea of independent assessment and certification to assist acquirers is attractive. The European Union's ESPRIT II program considered it sufficiently important to fund as a research initiative [1] and it has been successfully applied to IT technology, eg computer language compilers/interpreters, Posix and OSI.

There has been less achieved with business and general-purpose applications. The two key problems have been the lack of accepted sets of objectively verifiable functional and non-functional requirements against which to assess, although industry analysts such as Gartner, provide a valuable service at a much higher, conceptual level.

Many academic and industry commentators have complained that much software is of poor quality, and that this needs to be improved. This led to the publication in 1991 of ISO/IEC 9126: *Software product evaluation – Quality characteristics and guidelines for their use* [2], whose purpose was to provide "the quality related measurement instruments that would allow ... the engineering of quality throughout the entire software product lifecycle." [3]. While working on its revision, ISO 9126-1 [4], ISO/IEC WG6 of the Software Engineering Subcommittee (SC7) recognised various limitations and decided to develop a second generation of standards [3]. Work started in May 2000 on the ISO/IEC 25000 series. There are some dozen standards currently, with more expected. There is a useful introduction and explanation in Esaki et al [5] and to the quality model in particular in Dominguez-Mayo et al [6].

2.0 Scope

One survey found 81 different smartphone App Stores [7], but there are perhaps only a handful with real significance. This paper examines the published acceptance processes of a leading App Store from Apple, and that with the longest history, Microsoft's Windows Store.

2.1 Apple App Store

By June 2015, there had been 100 billion applications downloaded from the Apple App Store, from a choice of 1.5 million [8]. Apple state [9]: "All apps submitted to the App Store … are reviewed to ensure they are reliable, perform as expected, and are free of offensive material." The focus of this paper is on the software quality requirements and hence on the App Store Review, the Review Guidelines [10], design guidelines [11, 12, 13] and some relevant implications arising from practice in Testing, Marketing and Distribution.

2.2 Microsoft Windows Store

It is hard to know how many of the 2 billion PCs shipped to date are still in use, but 300 million are sold every year, and Windows still runs on over 90% of the world's desktops [14]. Microsoft's various certification schemes [15, 16, 17] go back at least to 1995. Windows Marketplace was launched in 2004, providing a platform for on-line, unmediated sales: an early App Store, since replaced by the Microsoft Store and then the Windows Store.

On the phone side, Nokia's original Ovi Store from 2009 [18] became the Nokia Store, and was then taken over by Microsoft to become the Windows Phone Store, which is now being merged with the Windows Store [19] and the Universal Windows Platform.

2.3 ISO/IEC 25000 (SQuaRE)

The ISO/IEC 25000 to ISO/IEC 25099 series of International Standards is entitled *Systems and software engineering -- Systems and software Quality Requirements and Evaluation*, hence the acronym: SQuaRE. The guide to the series, now in its 2nd edition [20] states that "the general goal … was to ... [cover] two main processes: software quality requirements specification and system and software quality evaluation; supported by a system and software quality measurement process. The purpose … is to assist those developing and acquiring systems and software products with the specification and evaluation of quality requirements."

The traditional ISO 9001 position was that quality concerned "conformance to specified requirements". This has been broadened to relate instead to "satisfy stated and implied needs". As the universe of such needs is not well-defined and classified, evaluation of quality is ultimately purchaser-dependant. App Stores generally sort applications into domain-based categories and sub-categories, and provide various other selection facilities.

The SQuaRE Quality Model has simplified that in ISO 9126, and now divides characteristics in two: Quality in Use: "the degree to which a product or system can be used by specific users to meet their needs to achieve specific goals ... in specific contexts of use" and Product Quality: "characteristics … that relate to static properties of software and dynamic properties of the computer system". Given the previously quoted focus on needs, one might ask why the second group, but ISO 25000 explains this as providing targets to drive development and verification, and to predict Quality in Use before delivery [20].

3.0 App Store Certification

App Stores are generally commercial undertakings, intended to provide users of smartphones (or similar) with easy access to the widest possible range of pre-qualified applications, enabling potential customers to try them out, knowing that the important risks – to their finances, data privacy, device integrity and children, have been firmly addressed by trustworthy organizations, to mitigate the risk of dealing with product developers of whom they have never heard.

Apple and Microsoft do not produce their own measures of 'goodness' for the public to use in selecting between alternative products. The purpose of their rules is to arrive at a simple accept or reject decision for their App Stores, assuring the market that the product is 'satisfactory' in some sense. It is left to buyers to decide whether they want to try the app, based on the supplier's description, etc.

"It's often said that people spend no more than a minute or two evaluating a new app." [21] Presumably that refers to consideration before installation, where the information available is limited [22]. However, the prices are much lower than for traditional PC applications, and indeed, many are free, so it becomes more cost-effective for potential customers to trial an application whose 'safety' has been certified, and discard it if unsatisfactory, rather than spend a lot of time hunting for documentation, reviews etc.

4.0 Applying the ISO/IEC 25010: 2011 Quality Model

Although ISO 9126 has been frequently used as a Quality Model in academic papers, many alternative quality models have been published over the years. Oriol et al [23] have compared 47 quality models for web services from 65 papers with ISO 25010 and found little consistency. Unlike a physical object, with clearly independent dimensions and well-defined measures (length, mass, time, electric current, thermodynamic temperature, luminous intensity, etc), concepts such as *compatibility* are somewhat nebulous, and indeed have been redefined and reorganised as part of the ISO 25000 work (see Table 1).

Biscoglio and Marchetti [24] found similar difficulties in applying ISO 25000, which they described as "a conceptual framework and not a ready-to-use solution". Corral et al [25] examined the developer guidelines from six App Stores, and sorted them into the 42 sub-characteristics of the then draft stage (FDIS) of ISO/IEC 25010. This was used to derive the most important characteristics, to guide developers.

The two App Stores selected have many similarities, but this paper is not intended to compare them. Examples are chosen to demonstrate the way that current App Store rules could be mapped to the terms of ISO/IEC 25000.

The rest of this section follows the structure of the standard [26], looking at each defined characteristic in turn, and going down to sub-characteristic level where this is reflected in App Store practice. Italics are used throughout the paper to denote the 13 characteristics and 40 sub-characteristics defined by the ISO 25010 model.

Quality in use	Product quality	Product quality (cont.)
• Effectiveness • Efficiency • Satisfaction - Usefulness - Trust - Pleasure - Comfort • Freedom from risk - Economic risk mitigation - Health and safety risk mitigation - Environmental risk mitigation • Context coverage - Context completeness - Flexibility	• Functional suitability - Functional completeness - Functional correctness - Functional appropriateness • Performance efficiency - Time behaviour - Resource utilization - Capacity • Compatibility - Co-existence - Interoperability • Usability - Appropriateness recognizability - Learnability - Operability - User error protection - User interface aesthetics - Accessibility	• Reliability - Maturity - Availability - Fault tolerance - Recoverability • Security - Confidentiality - Integrity - Non-repudiation - Accountability - Authenticity • Maintainability - Modularity - Reusability - Analysability - Modifiability - Testability • Portability - Adaptability - Installability - Replaceability

Table 1: ISO/IEC 25010: 2011: Characteristics and sub-characteristics

4.1 Quality in Use

4.1.1 Achievement of needs

In general App Stores do not seek to direct the user functionality of applications submitted. Indeed they encourage new ideas. Much of the content of App Stores consists of games, or items for entertainment. Apple's functional requirement is: "If your App doesn't do something useful, unique or provide some form of lasting entertainment, or if your app is plain creepy, it may not be accepted." [10] However Apple does attempt to restrict the silly, witness their statement: "We don't need any more Fart apps" [27], (which raises the question of how they determine whether a new one provides a significant advantage over the many already in store!)

This is somewhat of a stretch from SQuaRE's more solemn discussion of "stated and implied needs", and one should bear this broader view of applications in mind when interpreting the standard.

The App Stores have no general rules about meeting users' functional needs and the overall *effectiveness*, *efficiency*, *satisfaction* and completeness (called *context coverage*) with which they are achieved. Apparently this is a matter for the potential user to evaluate for themselves, as only they know their needs in detail.

4.1.2 Freedom from risk

The draft ISO/IEC 25022 [28] discusses potential metrics for this characteristic, but focusses on safety issues arising from the users' physical interactions with the smartphone. Smartphone apps do not normally engage with physical hazards and should not be safety-critical.

Apps in the Windows Phone Store are subject to a content policy, which exists to guide app developers, and to facilitate restriction or banning of certain content [29]. Examples of restricted or banned content include pornography, promotion of violence, discrimination, hate, or the use of drugs, alcohol and tobacco. Suggestions or depictions of prostitution, sexual fetishes, or generally anything that "a reasonable person would consider to be adult or borderline adult content" will be forbidden from the marketplace [30].

Both App Stores have controls on in-app purchases, to protect the bill payer, and restrict opportunities for gambling and money laundering.

4.2 Product quality

4.2.1 Functional suitability

This quality property is again relative to stated and implied needs. It is left to potential customers to do their own assessment and selection. Many producers adopt a 'soft marketing' approach by providing a free basic or sample package, with important functionality requiring a payment to be made.

4.2.1 Performance efficiency

Performance relative to the resources used is not addressed in the App Store rules, perhaps because apps run on a dedicated and personal platform, so efficiency is not normally a significant issue.

4.2.2 Compatibility

This is divided into *interoperability* between applications – the exchange and use of information, and *co-existence* – the impact on other products sharing the same platform. Thus it does not include compatibility with a specific platform or platforms.

Required operating environments

This is actually a critical matter to all commercially-minded smartphone vendors, as they want to ensure that they can design new and improved platforms in the future, whilst minimising the impact on existing applications. Note that the platform is not only the hardware and firmware of the computer processor concerned, but its devices, operating system, and any other services available to applications. It is desirable to be able to revise or replace any of these. Thus the platform supported by an App Store is not a single phone model on which the application can be tested and simply work or fail, but is a set of specifications of what services and facilities may be used, and usually some specific exclusions on what may not, primarily device- and implementation- specific idiosyncrasies: a purely 'virtual platform', that cannot be exactly physically implemented for testing on.

Both Apple and Microsoft specify a range of smartphones and tablets on which approved applications must run. Apple prohibits the use of non-public APIs, accessing outside the designated container, and using background services for unsupported purposes [10].

4.2.3 Usability

Appropriateness recognizability is described as "the degree to which users can recognize whether a product or system is appropriate for their needs". The notes mention associated documentation. For App Store purchases, this will normally be information offered to customers by the App Store, some authored by the vendors, but perhaps including reviews and comments left by earlier users.

Apple's UI design advice [13, 12] addresses *user interface aesthetics*, *accessibility* (to people with disabilities), *user error protection* and consistency of UI features, supporting *operability* and *learnability*. As Apple says: "Consistency [within the iOS environment] lets people transfer their knowledge and skills from one part of an app's UI to another and from one app to another app." [13]

Apple expects that applications should not be simply ported to its platform, but should be reconsidered in the iOS environment, and should adopt its paradigm (or "Themes"), its UIKit framework of common UI elements, etc [12], further refining the virtual platform specification.

4.2.4 Reliability

The main sub-characteristics of *availability* and *recoverability* are more obviously relevant to continuous real-time systems. However, with many smartphone applications expected to continue processing in the background [21], they should be applied.

Fault tolerance is addressed by "the app must continue to run and remain responsive to user input after the exception is handled." [29] Microsoft also include a requirement for graceful shutdown and will fail an application that closes unexpectedly.

4.2.5 Security

Much of the security surrounding an app is provided by the operating environment, with the applications and settings chosen by the device owner.

"iOS is designed and built to … accept and install software that has been approved by Apple and run through the App Store. As such Apple has pretty much guaranteed that you won't encounter any malicious software on your iOS device." [31]

Applications are required to conform to the virtual platform specification, which includes rules against loading more code, and controls on stored and real-time personal data detected by the device (such as location and audio or video input).

If personal data is collected, the Microsoft Windows Store requires the vendor to publish a Privacy Policy.

4.2.6 Maintainability

Maintainability (and *portability*) are normally supplier-side issues and would not appear to be relevant to the normal App Store purchase of an application, to run on the device in hand. However, ISO 25010's description of *maintainability* includes modifications carried out by "business or operational staff, or end users". App Stores do have rules about user settings eg notifications, permissions, etc, which could be assigned to this characteristic, or perhaps to *usability*, under *operability*.

Analysability includes the ability to "diagnose … causes of failures". Apple requires that diagnostic messages are correct, and also disallows the installation or launch of other executable code, probably to provide some pragmatic limit on the functionality that has to be reviewed.

Testability: the Microsoft evaluation process requires the provision of any login credentials, gift cards and server access needed, with appropriate written advice. Many of its traditional certifications provide mandatory test suites [32, 16].

4.2.7 Portability

Portability is concerned with transfer "from one … environment to another", so is presumably intended to cover activities beyond mere download and installation on a known supported platform. Traditionally this is an activity for developers, working on source code, and is not normally an advertised feature of App Store applications. However, *portability* in this standard includes adaptation for "different … operational or usage environments" and by end users, so is partly addressed by the ability to run on any instance of the specified virtual platform, as discussed in 4.2.2.

The notes mention report formats, so *portability* should perhaps include rotating the device to move between portrait and landscape display orientation.

Replaceability of one product by another is not generally addressed by the App Stores. However, again the small print of the standard broadens the concept, to include the ease with which a new version of a product can replace an earlier one.

4.3 Other

There are a number of requirements in the App Store rules that do not seem to be directly addressed in SQuaRE. We now examine the most important of these.

4.3.1 Software faults

"Apps that crash … [or] that exhibit bugs will be rejected." [10] Arguably these are covered by SQuaRE's *functional correctness*, but Apple's wording gives clear acceptance criteria. Although their App Store Review includes a certain amount of testing, there is no indication that this goes beyond 'exploratory testing'. Nor does it seem that user error reports result in immediate disqualification.

4.3.2 Honesty and Openness

There are general rules around open and honest communication – both for the customer contemplating a purchase, and for the user employing the app.
Reasons for rejection include [10]:

- "Apps that do not perform as advertised by the developer"
- "[Apps] that include undocumented or hidden features inconsistent with the description"
- "Apps that are intended to provide trick or fake functionality that are not clearly marked as such"

More drastically, "if you attempt to cheat the system … you will be expelled" [10]. Misleading documentation does not seem to be envisaged in the SQuaRE quality model.

4.3.3 Access

Apps should be usable by all potential users (and not arbitrarily restricted) [10]. This can be put under SQuaRE's *context completeness* within the scope of the acquirer's specified contexts of use, and under *flexibility* beyond.

4.3.4 Intuitive (and simple)

Whilst the advent of the home PC massively extended the market for applications, the smartphone revolution is an order of magnitude bigger, bringing in a whole range of users who don't need to know about files, folders, etc. [33] This aspect can be covered under SQuaRE's *accessibility*, or perhaps *user error protection* or *operability*.

4.3.5 Readiness

Applications should be presented as fully supported production versions and not "cobbled together in a few days" [10].

5.0 Analysis

The two App Stores studied here are a highly successful answer to a consumer need for certification. They have apparently not followed the SQuaRE standards. Reviewing their product requirements against SQuaRE, the 'fit' is not straightforward, and requires interpretation of the characteristics and sub-characteristics offered. Some of ISO 25010's defined characteristics are rather abstract, combining several concepts, not all immediately evident. Where alternative allocations exist, it is likely that users of SQuaRE will differ, reducing the comparability of product reports.

If Apple or Microsoft had started with SQuaRE, it seems unlikely that they would have generated the less obvious requirements discussed in section 4. It could be helpful to list the more obscure interpretations for each of the (sub-)characteristics in a future version of the standard.

An important extension to the SQuaRE set is ISO/IEC 25051 [34], which deals with RUSP: “ready-to-use Software Product”. It mentions smartphones though not App Stores, and is intended to support certification schemes. It requires very substantial vendor documentation, including “verifiable evidence of compliance” of each software quality characteristic declared according to the ISO 25010 quality model, so is unlikely to be adopted for the usual limited functionality and supporting materials of a smartphone app.

6.0 Conclusions

The SQuaRE series of standards provides a more in-depth treatment of software quality models than any previous standards, and should become the new reference. Nevertheless, it will be hard to create an accepted set of objective metrics, as the characteristics will need substantial further refinement into quantifiable measures.

The App Stores reviewed here have built up their assessment rules and certified over 2 million applications against them. The fees for suppliers are cheap, less than a day’s consultancy. Clearly product certification is here, although without the functional assessment: not what researchers were thinking of, 25 years ago.

7.0 References

1 Esprit, Esprit - 1988 Annual Report. CEC 1989, ISBN 9282608921

2 ISO/IEC 9126: 1991, *Software product evaluation – Quality characteristics and guidelines for their use*

3 Suryn W, Abran A, April A, Hamza M H, ISO/IEC SQuaRE: The Second Generation of Standards for Software Product Quality. Proceedings of IASTED, 7th International Conference on Software Engineering and Applications, pp807-814, California, Nov 2003

4 ISO/IEC 9126-1: 2001 *Software engineering -- Product quality -- Part 1: Quality model*

5 Esaki K, Azuma M, Komiyama T, Introduction of Quality Requirement and Evaluation Based on ISO/IEC SQuaRE Series of Standards. Proceedings of ISCTCS 2012, International standard conference on trustworthy computing and services, pp94-101, Beijing, May 2012

6 Dominguez-Mayo F, Escalona M, Mejias M, Ross M, Staples G (2012). Quality evaluation for Model-Driven Web Engineering methodologies, Information and Software Technology, 54, pp1265-82

7 Slinger J, Bloemendal E, Defining App Stores: The role of curated marketplaces in software ecosystems. Lecture Notes in Business Information Processing vol 150, 2013, pp195-206, Elsevier 2014, ISBN 9783642393358

8 Statista Inc (2016). Cumulative number of apps downloaded from the Apple App Store from July 2008 to June 2015 (in billions). Retrieved 28th January 2016, from Statista: http://www.statista.com/statistics/263794/number-of-downloads-from-the-apple-app-store/

9 Apple Inc (2016). Distributing on the App Store. Retrieved 25th January 2016, from Apple: https://developer.apple.com/app-store/

10 Apple Inc (2016). App Store Review Guidelines. Retrieved 25th January 2016, from Apple: https://developer.apple.com/app-store/review/guidelines/
11 Apple Inc (2016), UI Design Dos and Don'ts. Retrieved 25th January 2016, from Apple: https://developer.apple.com/design/tips/
12 Apple Inc (2016). iOS Human Interface Guidelines. Retrieved 25th January 2016, from Apple: https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/
13 Apple Inc (2016). Design Principles. Retrieved 25th January 2016, from Apple: https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/Principles.html#//apple_ref/doc/uid/TP40006556-CH4-SW1
14 NetMarketShare. Desktop Top Operating System Share Trend. Retrieved 28th January 2016, from Net Applications.com: https://netmarketshare.com/
15 Microsoft Corporation (2016). Windows hardware certification. Retrieved 28th January 2016, from Microsoft: https://msdn.microsoft.com/en-us/windows/hardware/gg463010
16 Microsoft Corporation (2016). Enterprise Cloud Application Logo Certification Program. Retrieved 28th January 2016, from Microsoft: https://msdn.microsoft.com/enterprisecloudcertified
17 Microsoft Corporation (2016). Certification requirements for Windows desktop apps, version 10, July 29, 2015. Retrieved 28th January 2016, from Microsoft: https://msdn.microsoft.com/en-us/library/windows/desktop/hh749939.aspx
18 Eddy N, (Feb 16, 2009). MWC: Nokia Announces 'Smart Store' Ovi. Retrieved 26th January 2016, from eWEEK.com: http://www.eweek.com/c/a/Mobile-and-Wireless/MWC-Nokia-Announces-Smart-Store-Ovi
19 Whitney L (July 16, 2015). Microsoft offers all-inclusive app store for PCs, tablets and phones. Retrieved 26th January 2016, from CNET Software: http://www.cnet.com/news/microsoft-opens-all-inclusive-app-store-for-pcs-tablets-and-phones/
20 ISO/IEC 25000: 2014, *Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE.*
21 Apple Inc (2016). Starting and Stopping. Retrieved 25th January 2016, from Apple: https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/StartingStopping.html#//apple_ref/doc/uid/TP40006556-CH52-SW1
22 Microsoft Corporation (2016). Write a great app description. Retrieved 26th January 2016, from Microsoft: https://msdn.microsoft.com/en-gb/library/windows/apps/mt148564.aspx
23 Oriol M, Marco J, Franch X (2014). Quality models for web services: A systematic mapping. Information and Software Technology, 56, pp1167-82
24 Biscoglio I, Marchetti E, Definition of Software Quality Evaluation and Measurement Plans, Proceedings of ICSOFT, 9th International Joint Conference on Software Technologies pp63-80, Vienna, August 2014 in Communications in computer and information science. Number 555. Springer 2015, ISBN: 978-3-319-25578-1

25 Corral L, Sillitti A, Succis G, Defining Relevant Software Quality Characteristics from Publishing Policies of Mobile App Stores, Proc of 11th International Conference on Mobile Web Information Systems, MobiWis 2014, Barcelona, in LNCS 8640, Springer 2014, ISBN 978-3-319-10358-7
26 ISO/IEC 25010: 2011, *Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models*
27 McCann T, The Art of the App Store, Wiley 2012, ISBN 9780470952788
28 ISO/IEC DIS 25022: 2004, Draft for public comment – Measurement of quality in use
29 Microsoft Corporation (2016), Windows Store Policies, version 6.7, 3 December 2015. Retrieved 26th January 2016, from Microsoft: https://msdn.microsoft.com/en-us/library/windows/apps/dn764944.aspx
30 David K (June 19, 2010), WM6.5 and WP7 App Content Policies Compared – Pretty Damn Similar. Retrieved 29th January 2016, from Mobile Technology News: http://mobilitydigest.com/wm6-5-and-wp7-app-content-policies-compared-pretty-damn-similar/
31 Haslam K, (22 Sep 15), iOS security risks: After the XcodeGhost exploit is Apple's iOS really safer than Android? Plus: What security apps do you need for iPad & iPhone. Retrieved 15th January 2016, from MacWorld: http://www.macworld.co.uk/feature/iosapps/is-ipad-iphone-ios-safe-xcodeghost-what-security-software-need-3453938/#antivirus
32 Microsoft Corporation (2016), Windows App Certification Kit tests for Windows Store apps. Retrieved 26th January 2016, from Microsoft: https://msdn.microsoft.com/en-us/library/windows/apps/jj657973.aspx
33 Apple Inc (2016), Integrating with iOS. Retrieved 29th January 2016, from Apple: https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/Integration.html#//apple_ref/doc/uid/TP40006556-CH61-SW1
34 ISO/IEC 25051: 2014, *Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) – Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing.*