AQuA
App Quality Alliance

AQuA Round Table

The Challenges of Mobile App Quality

The outcome of a Round Table with parties interested in app quality, held in London during October 2016.

These discussions were held under the Chatham House Rule, which permits reporting of free debate without attribution, and the topics and ideas covered are summarised below. The objective was to bring interested parties together to freely discuss matters of interest in development and testing of mobile-related applications, to promote information-sharing in the industry where practical and to identify for AQuA the subjects of greatest interest amongst its membership.

Topics covered

- Security testing.
- Managing device test coverage and device / platform updates.
- Medical education apps.
- Crowd Testing.
- QA as a differentiator and essential hygiene element for start-ups.
- New horizons in testing:
    - AR & VR.
    - Testing of connected things.
    - Performance testing.

Security Testing

AQuA has frequently been asked what can be done to address issues relating to security. To progress this, they have been talking to the Open Web Application Security Project (OWASP), an open-source organisation of some 2000 security professionals worldwide, who have a group considering mobile security issues. This group has already created a body of work that is of considerable value, although at a more technical level than would be appropriate for many in the app development and testing community.

As this work has been made freely available under a Creative Commons licence, AQuA is building on these processes and tools to develop a set of Security Testing Criteria. In the first instance they will concentrate on the Android platform rather than iOS; this is because apps for iOS are already subject to rigorous testing, including security aspects, in the journey towards acceptance into the App Store. We have started work from OWASP's existing Top

Ten issue definitions, using the standard AQuA document structure to maximise usability for developers.

Areas of concern have been identified for further examination as the Security Testing Criteria take shape. For example, use of excessive permissions cause significant user concern, and are one of the major reasons for apps being uninstalled; also for dealing with known vulnerabilities in third party 'libraries' or SDKs (etc) the best tactic is thought to be use of the scanning tools and practices defined by OWASP.

OWASP had already warned that there is no such thing as 100% security as any protection can be broken, given sufficient incentive and time. The trick is to make an app difficult enough that an attacker would be likely to move on to an easier target, and AQuA's aim is to help developers avoid the obvious errors that created security weaknesses, and supply information that would help those inclined to look for further protections.

Security has tended to be a specialist area within testing dealt with by separate teams with their own skill sets. Because of this separation, developers are not always aware of security risks outside the obvious, such as regarding handling payment card information, and rarely ask for security testing. AQuA's proposals may make it easier to justify security testing to developers, but cost is still a limitation for many. A pricing model used by some testing firms is "pay per vulnerability", where there is the possibility of low cost if an app is found to be secure – which also helps focus the mind of the developer if their app turns out less secure than expected.

Rather than presenting any security testing criteria as a "complete" solution, the intention is that the AQuA proposals would be more of a tool for an existing test team to be able to identify if problems exist that require more work / specialist investigation.

For test houses, the existence of AQuA Criteria in this area could enable introducing consideration of security for the first time for some clients. But they must carry a warning to the effect that this can only address the low-hanging fruit in security vulnerability terms. Test cycle duration is currently expected to be around half a day, which is expected to be acceptable in most circumstances.

In a world exposing more and more personal data through social networks, the value of security depends on the value of the data – the criminal world can concentrate tremendous resources on acquiring data seen as valuable. There is also the issue of potential damage to trust: apps can be a channel to market, and the potential for damage to an associated brand could be fatal for an organisation with no other revenue stream. There is a difference in attitudes between the US and EU – in the US, people typically trust corporations more than government, whereas in the EU governments tend to be trusted more than corporations.

AQuA hopes to be able to publish a first draft of this document at the end of 2016.

Managing device test coverage and device / platform updates

How many of the devices on the market to cover when testing will depend on the view of the risk involved – whether the greatest risk for an app would come from it having a failure on a popular device or across a larger number of less-popular ones. Analytics suggest that to exhaustively cover the whole European market would take at least 200 devices – this is generally not achievable on a realistic budget, so deciding on the best compromise becomes important.

Some testers prefer to cover a set percentage of the devices currently available; others to cover a set number, usually testing on between 20 and 100 of what they regard as the most relevant devices. Identifying a subset of devices will often depend on the platform or type of audience to be covered. On platform differences, the previous stereotypes of minimal iOS device variation versus Android fragmentation have changed to a form of convergence: iOS now have a number of device size and version variants (with more OS versions extant at a given time), whereas Android devices have become more similar.

For the latter platform, the preponderance of Samsung devices in many markets means that around 50 devices from the one manufacturer could be enough to cover a worthwhile cross-section of the popular devices in use. So for many testers the target device pool is already well below the number originally suggested by analytics, and in a field like higher education the range of devices could be even further reduced. Analytics used there suggested that between 10 and 15 devices would be sufficient to cover all the popular variations in this market.

A contributor to the reduction of variation on the Android platform is the automatic, non-optional updating of Google Play Services, meaning that even older devices can have the same APIs as the latest ones. For some testers this means, with so many Android devices now being touchscreen slabs with a 14:9 display ratio, that perhaps a dozen devices might cover most variations. Issues are seen as more likely to arise from OEM code additions to the OS than from physical device variations.

The situation in China regarding testing on Android is radically different, however. The regulatory environment there means that all devices had to be certified by official test laboratories before going on sale, meaning that those laboratories have examples of every device on the market available for testing apps. In this case the deciding factor in selecting how many devices to use would be logistical trade-off between risk and the resources available for testing. For example, around 2,000 new devices have come on to the market in China in the last year: an app regarded as having a significant level of risk (e.g. carrying a prestigious brand, handling financial or other sensitive data) might be tested on all 2,000 devices; whereas a more mundane app with limited functions and low risk would be more likely to be tested on a subset of the device range.

There is a degree of consensus that the best return usually comes from concentrating on new and top-of-the-market devices, but this does not hold true for all markets. To cover the majority of the market in India, it would be better to select a wider range of mid-priced devices.

In education there are particular problems with testing: the academic year starts in September when many new devices go on sale, so there can be a race to get preliminary testing done with the latest devices before they appear in the hands of students.

On the iOS platform there may be a need to have more devices available at a time when OS updates are being released, because it is more common now for those updates to be rolled out in slow stages so the effects can be monitored, meaning that there will be greater OS variation in the typical user base.

Shared remote device testing facilities are less popular than in the past, partly because of the need to retain complete control of client IP, and partly because remote device servers tend to provide an inferior experience when testing on a touchscreen OS compared to a physical device in the hands of the tester. They are also viewed as expensive for what they provide, to the extent that the money is often better spent on buying popular devices to perform hands-on testing. Testing on emulators is generally regarded as inferior to testing on real devices.

Automated testing is an important part of functional and regression testing, offering consistency with a valuable saving of time and effort, although some degree of manual checking and analysis of results is still needed. However, it is less easy to automate user experience testing, and most testing companies tend to flexibly mix automated and manual testing, determining on an ad hoc basis which parts of a test cycle could be automated and which needed to be run manually.

Some prefer to do extended testing on a restricted number of the most relevant devices, then a quick pass on a selection of non-core devices, rather than devoting the same amount of effort to all. The size of the test team required to cover a large number of devices and the time required to run the tests could be constraints on device range just as much as the actual cost of acquiring devices.

Medical education apps

A European Commission green paper on mHealth apps estimates that these could help bring about as much as a €90bn saving across EU public healthcare budgets, and mobile devices would be key to the use of such applications.

These applications generally require very specific testing because of the greater risks involved in the use of the subject matter – whilst not as immediately critical as something like aircraft systems software, they are undoubtedly higher risk than a game app would be.

Most users would be medical students, but as the apps are also useful for patient education users could occasionally be members of the general public, and so testing may need to address both audiences.

Protection of sensitive data is paramount in the medical field, requiring testers to sign agreements blocking the use of confidential patient data during testing. Penalties for transgression in this area are substantial, so constant vigilance on this issue is essential.

Different areas of medical education will raise different requirements for recording data – dentistry for example being more data driven, where the number of procedures carried out during education would be important; whereas in other areas of medicine less quantifiable things like people skills might need to be evaluated.

Because of hygiene requirements and the impossibility of providing work surfaces in many medical situations, desktops, laptops and even tablets can all be impractical for the student, who will usually be limited to the a device that they can hold and use mostly one-handed, meaning that the primary app platform is the smartphone.

There are also special challenges in testing medical education apps because the test community (medical students) only exists from the beginning of the academic year, making advance preparation often impossible; and because students are not allowed to access National Health Service (NHS) networks, data transmission can only be done using the limited mobile signal available inside hospitals.

Apps are useful both in supplying information to the students, and in capturing evaluation data and feedback on their performance. Feedback must however be retained and accessible throughout a student's education, yet not be editable by them, which raise issues to be addressed during development and testing.

Rehearsal of procedures is also an important function for apps, and both Virtual Reality (VR) and Augmented Reality (AR) are expected to play a role in future apps involving simulation of procedures, leading to a need to expand testing procedures to address these areas.

Developers need to be aware that the level of regulation increases dramatically as soon as an app crosses the dividing line between education and service delivery in the medical world. Much discussion is taking place on the appropriate level of regulation in each area so as to maintain safety and confidentiality without inhibiting innovation as far as possible. Checks are needed to ensure that the information in datasets is not individually identifiable, for example by the rarity of the medical conditions or other information included, or the use of too small a dataset. There also need to be checks to ensure that the information presented in apps is both credible and consistent as students brought up with the constant presence of mobile devices may be over-ready to rely on the information presented by those devices.

Devices are, as mentioned earlier, something of a challenge in this field. Students generally have better devices than education providers can afford to supply, so hardware provision is now largely replaced by the bring-your-own-device (BYOD) model. To simplify a constantly changing device population, development frameworks like Xamarin are being increasingly used – these can enable a single body of application code to address multiple device platforms.

Trust is often mentioned as an important issue to be identified in medical education, whether it is trust in a student's ability to perform a procedure, or trust in the accuracy of information presented by an app, always with reference to whether the level of risk is high or low for the particular circumstances.

Testing in this area is generally not well optimised – medical specialists are good at identifying test requirements for their specialism, but less so at identifying general usability issues. To ensure consistency and fairness of evaluation and testing, it would be best to separate testing of medical aspects from technical and usability areas, using a layered approach where the latter areas could be addressed by following, for example, the AQuA Baseline Testing Criteria, and the medical specialists could devote their expertise to testing the elements involving medical information and procedures.

Regulatory requirements are also not static, which means maintenance activities are also required to conform with regulatory changes, and testing of that conformance would need to take priority over development of new functionality. Failure to maintain that priority could give rise to legal risk, meaning that a single adverse outcome could bring about dramatic - perhaps even retrospective – changes as a result of regulatory intervention. If medical app developers and testers do not find an efficient way to separate technical testing from evaluation of medical content, there is a risk that a less than optimal scheme might be imposed by regulators at some point, to the detriment of innovation.

There is also a need to improve the terminology used in this field so that has more in common with that used in other development and testing areas, along with a need to start talking in terms of quality when holding conversations with people outside of the testing specialism, as speaking solely of testing and conformance may not always convey the critical importance of the activity to more business-focussed minds.

Crowd Testing

Crowd testing is seen as having suffered from a degree of over-promotion when it was first utilised, but it is now a valid means of evaluating aspects of the user experience, particularly for products that are intended for use worldwide. It can harness the abilities of large quantities of people - all the crowd testing networks taken together number around half a million users – yet can also be narrowed down to a particular locality or demographic, and can be valuable for ensuring that an app remains usable across a wide customer base.

It is however only as good as the management of the crowd. For some types of client it may raise as many issues as it solves, particularly for high value products – how can the client be sure that the tester isn't an employee of a competitor gathering intelligence on their future strategy, or be sure that the app won't be hacked or intellectual property (IP) be inappropriately copied? For this type of client and product the cost of crowd testing becomes much greater because of the overhead of closely managing the participants and their behaviour, and perhaps also meeting very specific requirements regarding the quality of results.

There are some areas where its strengths are particularly useful, for example geo-specific testing like serving particular adverts in appropriate countries, or load-testing with large numbers of users. In these circumstances it can be most useful as an adjunct to in-house formal testing rather than in place of it. It can also uncover aspects of the user journey that might not be seen in a formal testing environment, and uncover issues caused by interaction

with many different apps installed on user devices – again something that is difficult to quantify in formal testing. These aspects are particularly important in the games industry, where stability and usability issues need to be quickly uncovered and fixed before code is committed to production.

As long as it is suited to the client's requirements, crowd testing is a useful tool: Gartner have suggested that up to 20% of testing could be performed using crowd testing.

QA as a differentiator and essential hygiene element for start-ups

Startups can sometimes have an over-optimistic view of their product's readiness for market, and persuading them to run proper QA and testing can be a chance to avert the risk of a disastrous launch. Evaluating the issues that have been found in code that was supposed to be market-ready emphasises that the cost of quality failures is much greater than the cost of the testing that finds them. Implementing a decent testing regime as early as possible in the development cycle is a worthwhile investment, and the AQuA Baseline Testing Criteria address exactly the sort of issues that can be overlooked because of pressure to get to market.

Given that the introduction of testing is often a difficult conversation to have with startups, testing companies can do a number of things to make their services more attractive to this type of customer. The testing environment should be simple to enter with the minimum of paperwork consistent with good legal protections, and it may be more productive to talk in terms of the service being a "health check" than laying emphasis on the formality and rigour of testing.

Professional testers should be ready to explain to clients the importance identifying a minimum viable quality for their product, and the risks created by omitting this. Sometimes where the client has unrealistic expectations of an absence of bugs in their code, the best tactic may be to offer some testing on a "no fault, no fee" basis – as bugs will almost certainly be found, the financial risk may not be great, and the sobering effect on the client's overconfidence may be a better way of demonstrating the value of professional testing than any presentation. New terminology can sometimes help engage the mind of a client – the games industry invented the term "playability" to make clear the relevance of QA and user experience (UX) issues.

New horizons in testing

AR & VR

Virtual Reality (VR) gives an immersive, detached experience, whereas Augmented Reality (AR) recognises items in the camera's view and augments the display with an overlay. Both have great potential in the app field, but will present new challenges in testing. One issue that is likely to arise is keeping content current with reality – the real world changes continuously, and therefore testing needs to ensure that where an app is visually tied to a real-world view, the presentation in the app remains an accurate representation of the reality. Effectively this raises a similar possibility to that discussed for Medical Education apps – the possibility that ongoing maintenance will need to be planned into the development process to remain currency.

The most important aspects of testing in these new fields are still likely to relate to key elements of the AQuA Baseline Testing Criteria – that actions are safe, logical and consistent. Other important issues are likely to relate to battery drain and device heating, viewing area size, detail and refresh during movement. In many cases it may be necessary to compare against other similar applications to try to establish what is currently reasonable for the type of app on present hardware. It is likely that technical capabilities and user expectations will see significant changes over the next year, so it is difficult at present to identify specifics for testing beyond those already suggested.

Testing of connected things

The growth of the Internet of Things (IoT) has introduced a need to test apps distributed across a wide range of devices, meaning that the scope of what needs to be tested on each device, and the functionality to be covered for each implementation, may have to be defined in an ad hoc fashion. Because this field is still so new, commonalities and best practice are still emerging. An example is testing with beacons – there may be a need to test in the presence of competing products to verify that inappropriate connections do not happen, and to test the effects of poor or non-existent GPS signal for location purposes.

Testing connected things need not be significantly different to other types of testing if the system is well designed, but there are some areas that are likely to be of greater importance in a connected things environment, such as configuration and installation (given the current lack of standardisation), physical performance, and the handling of signal interference or overload.

Because of the lack of standardisation, it may be necessary to check every part of a connectivity implementation during testing, in contrast to mobile phone based app testing, where network connections can generally be treated as a standardised element that has already been evaluated outside of the app testing. The best technique is likely to be dividing the implementation into manageable subsystems that can be separately tested, before progressing to an integration test for the overall system, avoiding trying to test everything at once.

The market will again need to mature considerably before it will be possible to develop any standardised testing criteria – competition and evolution of products will be likely to determine what sort of testing will be needed in future.

Performance testing

AT&T's Application Resource Optimizer (ARO) tool has made it possible to understand the low level resource usage of an app and how it can be improved – there are many examples of practical benefits from its use.

It's important to clarify customer expectations for app performance early on, and ensure that developer intentions are well aligned to them so as to limit misunderstandings and identify important areas. There is value in identifying how an app handles API and web services, handovers and radio functions, in comparison with any competing products and against client and user expectations. Identifying underlying activities initiated per click can help identify bottlenecks, and design choices such as moving activities earlier or later in the flow can help make an app feel much more responsive, even if the overall activity takes the same amount of time. This is where performance testing, and the exposure of underlying radio handling and data caching can bring significant benefits.

Conclusion

AQuA thanks all attendees for the depth and detail of the discussions. These will be extremely valuable in helping understand the issues that are of most concern to the industry, and how we should move forwards with the various projects in hand over the coming months.