

The outcome of a Round Table with parties interested in app quality, held in London in November 2015.

Agenda

- Welcome & introductions.
- Automated Testing vs. Manual Testing.
- Test Planning in an Agile Environment.
- Exploratory Testing.
- What are the new challenges beyond Mobile apps?
- mHealth.
- Wrap up & close.

Welcome & Introductions

Martin Wrigley opened the proceedings on behalf of AQuA and welcomed everyone. He explained how it was being held under the Chatham House Rule, which permits reporting of free debate without attribution. After introductions, he opened the first topic for discussion.

Automated Testing vs. Manual Testing

There is a strong movement towards automating as much testing as possible, which is doing a great deal to improve the quality of software. However, non-functional testing related to the user experience is not well suited to being automated so will still need to have a manual component, and extensive automation may not be cost-effective where only a few develop-and-test cycles need to be run.

Automated testing may also be particularly useful in environments where regulatory requirements mean software behaviour post-launch must be closely monitored, or where significant platform updates could bring the need for corresponding app updates to maintain functionality and user satisfaction. Preserving test scripts can be essential, so they are ready to be used to re-check results in a changed environment, or if any aspect of app behaviour is called into question after release. Some developers found their existing scripts particularly valuable for identifying issues with platform changes, as the old script will generate new errors on an amended platform, and these new errors can be used to quickly home in on software areas affected and inform creation of updated scripts for the new environment.

The general consensus was that for most environments around 70% of testing can be usefully automated, with the remaining 30% staying manual. Knowledge of both the industry

and the context in which an app is used was necessary to identify elements that needed to be manually tested. However, it was suggested that for products with very short development lifecycles testing automation might actually be counterproductive – some attendees reckoned automation achieved maximum efficiencies with a development lifecycle greater than three months, and where the quantity of change in each release was around 15% of the application or less.

Crowd-sourced testing (which is essentially manual) was reckoned by many to be a valuable add-on that provided a good degree of real-world usage experience, but was best used as an adjunct to systematic formal testing, and for apps with a degree of complexity it was unlikely to uncover all the significant issues by itself.

Because of the higher overall costs associated with manual testing, it was reckoned essential to identify the key devices for the market or markets being targeted. Testing on those devices was seen as essential for uncovering real-world issues which could occur, for managing client expectations, and for meeting objectives as comprehensively as possible.

Prototyping – and getting client feedback from it – was indicated by a number of speakers to save them wasted effort early in development, by helping to clarify user requirements. It was also felt beneficial in that it could help shorten the development cycle and reduce the quantity of formal testing required to achieve client satisfaction.

AQuA's Testing Criteria were recognised as capturing the knowledge and experience of software engineers and testers, and presenting it in a format that was accessible to all of today's developers – who may have varying backgrounds, both technical and non-technical. This concentration of knowledge was often most beneficial in pre-release or user acceptance activity, but was used by many to inform all stages of the development process – designs and prototypes could be put through usability testing with this information to identify issues and minimise rework later in the development process.

Test Planning in an Agile Environment

Many felt that the Agile process gave better ownership of the product and better access to the people who made the decisions that would affect the viability of the product. With regard to the development techniques used, Test Driven Design (TDD) needed to be planned into a development from the start but could definitely simplify code design and maintenance. Behaviour Driven Design (BDD) was especially liked by smaller developers as it was flexible and changes were easy to track.

Big developments with frequent deployments were generally agreed to rely heavily on automated testing to maintain speed of delivery, whereas smaller developers were more likely to be using a strategy involving more manual testing, and initial deployment of beta versions to judge user reactions before committing resources to development of additional features – however one drawback with the latter strategy was that not all app store feedback was regarded as realistic, so care needed to be taken to avoid using it as the sole driver of feature development.

Exploratory Testing

Exploratory testing was agreed to have value during development, but it was regarded as important to set limits to avoid testing areas in depth which were not feature complete, or spending too much time searching for esoteric issues which were unlikely to occur in general usage. It was regarded as essential to have people of varied capabilities and differing focus doing exploratory testing to ensure as good a match as possible with real-world experience.

For some developers, brief exploratory testing as soon as a build was available was seen as valuable to confirm the basic user experience before committing resources to running formal test scripts. Where the environment lent itself to rapid formal testing, exploratory testing was alternatively used by some as a follow-up after the formal testing phase, to act as a sanity check and ensure there were no unintended gaps in the formal tests.

Exploratory testing results needed to be well documented so that they could be used to inform future testing, as the insights gained would have value for both automated and manual testing procedures.

Security testing was also mentioned as an area which benefited from having people with the exploratory testing mindset, who could think of an app's functionality in ways that differed from the concept of the original designer. It was agreed that there was a definite need for guidance to establish what the typical developer needed to do, in the way of best practice and testing, to ensure compliance with relevant security and privacy standards.

What are the new challenges beyond Mobile Apps?

Wearable technology was seen as a growing area of change and disruption, but there were also concerns that a lack of vision regarding how it could be used might restrict its potential. Types of technology mentioned were smartwatches, fit bands, Google Glass and smart clothing, but there were possibilities beyond these that were not yet well defined.

Medical teaching aids, body-mounted sensors, smart tattoos, apps for fitness tracking, medication compliance, self-monitoring, safety and home monitoring were all mentioned as new areas that had opened up relatively recently for most developers. The emerging markets were being driven by innovation, much of it from small-scale providers.

Privacy and security were seen as highly relevant – what data to share with whom. Risks, including cyber attacks on medical devices, all involved areas that the industry had not had to consider to the same degree previously. Usability, interoperability and durability were all areas of concern in these new initiatives, and there was a lack of understanding of the regulatory landscape where it applied to apps and devices being used for regulated purposes. There was concern that there could be a risk of proprietary closed standards becoming dominant and limiting the potential expansion of the marketplace in these new areas. Wearables could also raise new areas of usability concern as well as possible biological compatibility issues.

The BSI and NHS were looking to develop guidelines regarding issues like the benefits and disadvantages of constant vs. intermittent monitoring in the health and wellness field – this sort of guidance was definitely needed by developers.

In the Internet of Things (IoT) development field, connectivity and compatibility were seen as important issues. Devices like beacons also needed very long battery life to be viable, or else alternative sources of power – mention was made of beacons incorporated into light fittings to provide a power source.

It was generally regarded as a demanding environment for developers because each new area required significant R&D work in topics needing specialist knowledge before it could be part of a viable project. For health apps, compatibility with hardware and other apps was often a significant factor for the user, and thus might need more extensive testing than was common for most developments, in order to build user trust.

There were definite concerns about the difficulty of expressing the value of testing to clients in these new areas: their lack of previous experience of software development could mean that they failed to understand the importance of testing, and could be very resistant to budgeting sufficiently for it.

The introduction of app platforms in cars meant that new testing strategies would need to be established, to reflect the issues involved which could be significantly different to conventional mobile software platforms.

At present, most of the bodies working on standards and best practices in these new areas tended to have a narrow focus. There was a need for wider dialogue regarding recommendations to make sure that they did not unintentionally restrict interoperability, or promote the growth of closed proprietary ecosystems.

There was also a concern that new apps might be launched in these areas without a good business case because the potential use cases are not well defined. This could produce uncertainty about whether all the new applications being launched would turn out to have genuine value to the user. At the moment this type of development was in a highly experimental phase and would take some time to reach maturity.

mHealth

One of the significant issues with developing software for the UK's National Health Service (NHS) was that it is not a single organisation, rather an umbrella term for over 9,000 separate health-related organisations.

Traditionally healthcare software development had been driven by healthcare professionals' requirements, but the introduction of mobile platforms meant that now much of the development was being driven by user requirements. This meant that professional and regulatory understanding could be behind the curve at times as new applications were being constantly created. It was also significant that app store data indicated that many of the apps being launched were never downloaded, or never used if downloaded, meaning that there has been a failure to identify functions of genuine value to users.

Where health applications have to interact with existing systems, interoperability becomes of vital importance as there are so many different systems already in use in healthcare.

Regardless of context, security, privacy, safety and cost-effectiveness will always be areas of concern to medical practitioners, so apps have to deliver well in all of those areas. Apps that are used to deliver medical intervention have to be tightly regulated and few make it through the trials and approval processes that can take between five years and a decade to complete. Few investors and app commissioners without healthcare experience will appreciate the timescales and effort required to bring this type of application to fruition. There is also the issue that finding a satisfactory revenue stream can be challenging, as within the NHS advertising is not permitted, removing the commonest solution used to fund apps in other areas. Discussion would be needed between healthcare specialists and app developers to understand each other's limitations and see if acceptable solutions could be identified.

However, apps used for medical training are more liberally regulated, and the commissioning environment is much more flexible, which makes it an easier area for developers to address.

The BSI's initiative in developing PAS 277, which seeks to develop a code of practice for developing quality apps in healthcare and wellness, was regarded as a valuable initiative that could help developers and app commissioners better understand the environment in which apps must function, and the measures of quality which would apply there. There are many areas in which the development community and medical specialists have very different timescales, expectations, priorities and terminology, and this evolving standard was agreed to offer an opportunity to improve understanding – particularly of the constraints of developing in this highly specialised and regulated field. Developers will particularly need guidance that helps them understand the risk-averse nature of the medical profession, which will be quite different to what they may have experienced in more generalised commercial fields.

It was agreed there that would be value in assessing the mechanical fitness and quality of apps separately from their medical fitness, to reduce the amount of specialist testing required, and there needs to be agreement across the industry on how this can best be done. One developer is already using the AQuA Testing Criteria to determine technical fitness before sending an app for medical evaluation. AQuA already has a process and a quality badge for identifying this type of technical fitness in apps, so this could be of value for the non-medical evaluation of apps.

Devices like Google Glass were seen as having value for training and recording purposes in the medical field, and there was likely to be useful to know how Google had approached the usage of such devices in safety-related contexts. Wearables in general were thought to be highly relevant to healthcare.

Wrap up & close

Some of the points that came out of the day's discussion (and also during AQuA's attendance at the Apps World London event in the preceding two days) were:

- Manual testing is not dead, contrary to reports – it is still valid in appropriate areas.

- A lot of support for AQuA's work was expressed (both at the Round Table and at the Apps World event).
- Exploratory testing was confirmed as useful throughout the development process.
- Different tools were needed for different jobs – developers and testers needed to ensure they were using the most appropriate ones and not “going down irrelevant rabbit holes”.
- Both the mHealth industry and the IoT space are still not mature despite how they are presented in the press – it will be interesting to see how these will look a year from now.

Martin Wrigley brought matters to a close by expressing thanks to everyone who attended for their contributions, and to the App Developers' Alliance for accommodating the event at their premises.