AUT UNIVERSITY

Redefining the Use of Augmented Reality

Quality Assurance Plan

Version 1.0
20 April 2015

Jason Gerbes	Joshua Son	Paul Lee	Sean Young
1274664	1388288	1264218	1302108

Contents

0.0 Version History

Version 1.0

Version 1.0 is the original version of the Quality Assurance Plan Document. This version of the document was created as part of the Project Proposal Version 1.1.

1.0 Introduction

The quality of documentation, research and development will follow standard practices outlined in this section. Each project team member will be expected to adhere to these practices, and a quality assurance review will take place during each weekly team meeting.

2.0 Documentation Quality Assurance

All documentation will be produced in Pages for OS X. These files can be edited and viewed using Pages for OS X, Pages for iOS or Pages for iCloud. A PDF exported version of the document can be viewed by any appropriate PDF reader, but cannot be edited.

2.1 Consistency

All documentation should follow a consistent style. All text should use ‘Helvetica Neue’ font, except in special circumstances. The AUT logo should be present in a corner of the document (usually top-right), and the name and page number of the document should appear in the footer (except on the title page). Font and line weights should be thinner than regular (e.g., 0.75pt line weight and ‘Light’ font weight), to give the document a ‘lightweight’ overall look and feel.

2.2 Proofreading

All documents must be proofread by at least two team members (including the Quality Control team member — see Team Member Roles) before being presented to the client or any AUT staff.

2.3 Exporting

Document files may be exported for viewing on non-supported devices or when read-only viewing of a document is required. Files must be exported in PDF format, using the “Best” quality setting for image quality. Exported files must have an appropriate, unambiguous name which directly encompasses the contents of the document.

2.4 Sharing & Editing

Sharing of the document files should be done using Pages for iCloud wherever possible to ensure that only a single, fully updated version of the document exists. It is foreseeable that documents may need to be shared in offline form (where incompatibilities may exist), therefore any changes that need to be made to the offline document must be made to the online master copy.

2.5 Images

Images must be of a resolution and size appropriate for the intended purpose — they must not appear blurry or otherwise visually unappealing. Text in images must be clearly legible. Resized images must match the proportions of the original image — as to not ‘skew’ the image in any way.

2.6 References

All direct and indirect quotes, and images from external sources must be referenced using APA 6th referencing. A reference list is to be maintained near the end of the document (usually before the disclaimer).

2.7 Printing

Documents that are to be printed must fit cleanly within the margins of the intended page size. All text should be clearly legible, all images must be clear, and (where possible) the document should be professionally bound. Pages with colour images or text should be printed in colour, rather than black and white.

3.0 Research Quality Assurance

We will undertake research throughout all three stages of this project. Our first stage is entirely research-based and will reference a well formed and regarded research methodology; Action research. Action research is a cyclic process, which means that it is flexible and responsive. While we research and develop an understanding we are ensuring that in each cycle process our answers become more precise.

The following stage, stage two, requires the group, supervisor and client to come to a decision on which development path will be followed for the remained of the project. This decision will be heavily influenced by the outcome of the research in stage one. It is essential that all findings during research are correctly recorded, and this valuable data will influence the decision in stage two.

The final stage of the project, stage three, is our development stage. Extreme Programming development is an iterative process. Each iteration has an evaluation stage that will heavily rely on research to correctly evaluate the outcome. It is also foreseeable that research will need to be conducted during the planning and execution of each phase as Augmented Reality technology is a new field for the group.

The skills and knowledge involved in this project (see Project Plan) do not necessarily match the current skill-set of the group. The group will need to research new technologies in order to gain the skills required to complete the project. High quality research practices are essential in this practice.

3.1 Scholarly sources

Research consisting of the analysis of scholarly publications, which have been peer reviewed. These sources provide credible depth into the field of interest and enhance credibility in our own research. All external sources of information must be referenced both inline and within a maintained reference list; following the APA 6th standard of referencing.

3.2 Sharing findings

The findings of all research should be discussed with the other team members. This can be achieved formally (e.g., during our weekly meetings), or less formally (e.g., through FaceBook posts, emails, txt messages, etc.)

The comments and suggestions received by the group are important in project development and progress. The feedback received assists in assessing and assuring the quality of work that has been undertaken by other team members.

3.3 Peer review

Peer review is similar to sharing findings, but research is evaluated by qualified members within a field of profession. Articles are sent to scholars in the same field of study to get their confirmation (or disapproval) on the quality and validity of the work.

3.4 Monitoring and oversight

The client and AUT staff (particularly the group supervisor) will act as the overseer of the research being undertaken by the group.

We will provide the client with periodic status reports to ensure the project is progressing as planned. “Such processes can help in keeping projects on track, and in resolving issues and problems as they arise” (Gray, 2010).

3.5 Documenting Findings

It is essential that all valuable findings discovered during the undertaking of research are properly documented. Findings that are shared internally within the group should be conveyed via the communication means outlined in the internal communication plan (see Communication Plan).

Any findings that are to be shared with the client should be formed into a brief report-like structure or presentation form. These materials should be treated as formal communication means.

3.6 Interviews

To ensure that our interviews are of the highest possible quality, a recording of the interview should be taken using high quality recording equipment. The audio should then be transcribed and made available to the group for cross-analysis.

The interviewee should be encouraged to talk, and should not be disrupted while speaking. The interview should be taken in a quiet, non-public place to ensure the quality of the audio recording and to limit possible distractions (Amanda, 2014).

3.7 Surveys

To ensure high quality surveys we should have a specific goal for the survey that is clear cut and unambiguous. The surveys should set out to answer a specific question, or series of questions. Alternatives to surveys should always be considered to ensure that the best form of research is being utilised.

Advantages of surveys include:

- A large sample size is possible.
- They are relatively easy to administer.
- They can cover a vast range of information.
- They are economical.

Disadvantages of surveys include:

- They are extremely subject dependant.
- Validity issues may arise.
- Errors may occur due to non-response.
- They are limited by response choices.

A good survey should:

- Select samples that represent the population to be studied
- Use designs that balance costs with errors
- Clearly define topics, concepts and content (attention to question wording and order, attention to survey length and format)
- Pretest questionnaires and procedures to identify problems prior to survey. (ensure that the participants understand the questions)
- Use statistical & analytical report techniques appropriate to the data collected (data analysis and interpretation should be competent and clear, findings should be easy to understand)
- Carefully develop and fulfil pledges of confidentiality to respondents
- Disclose all methods of the survey to permit evaluation and replication (description of population and sampling frame used, purpose of study with specific objectives)
  (Johnson, 2011).

4.0 Development Quality Assurance

Extreme Programming (XP) Development leverages a number of quality assurance methods to enforce good programming practices. Being an agile development practice, it is essential that quality assurance is taken into consideration at all phases of development — as to enforce an expected of deliverables (Balkanski, 2003).

4.1 Unit Tests

According to Osherove (2014), a unit test is an automated piece of code that invokes a unit of work in the system and then checks a single assumption about the behaviour of that unit of work. A unit of work is a single logical, functional use case in the system that can be invoked by some public interface (in most cases). A unit of work can span a single method, a whole class or multiple classes working together to achieve one single logical purpose that can be verified.

A good unit test:

- Is able to be fully automated
- Has full control over all the pieces running (use mocks or stubs to achieve this isolation when needed)
- Can be run in any order if part of many other tests
- Runs in memory (no DB or File access, for example)
- Consistently returns the same result (You always run the same test, so no random numbers, for example. save those for integration or range tests)
- Runs fast
- Tests a single logical concept in the system
- Is readable
- Is maintainable
- Is trustworthy (when you see its result, you don't need to debug the code just to be sure)

Under XP Development, all code must be bound by appropriate unit tests. All relevant tests must pass before the code can be release. If any bugs are found in the system, tests must be created to fix it. The unit tests are run often, and the score is published.

4.2 Test-First Development

XP utilises a test-first development practice. Unit tests are created first, before the code has been written. The code is then written with the intention of passing the unit tests. Creating a unit test helps a developer to really consider what needs to be done. Requirements are nailed down firmly by tests. Specifications cannot be misunderstood written in the form of executable code (Wells, 2000).

Unit tests give the developer immediate feedback as they work. It is often not clear when a developer has finished all the necessary functionality. Scope creep can occur as extensions and error conditions are considered. If we create our unit tests first then we know when we are done; the unit tests all run.

A test-first approach also benefits system design. It can be very difficult to unit test some software systems. These systems are typically built code first and testing second, often by a different team entirely. By creating tests first, the design will be influenced by a desire to test everything of value to the client. The design will reflect this by being easier to test.

4.3 Peer Programming

All code sent into production in XP development is created by two people working together at a single computer. Peer programming increases software quality without impacting time to deliver. Peer programming assures that the entire source code is reviewed all the time.

Wray (2010) explained peer programming using the metaphor of one programmer being the “driver” and the other the “navigator.” In this metaphor, the driver controls the keyboard and focuses on the immediate task of coding, and the navigator acts as a reviewer, observing and thinking about more strategic architectural issues.

4.4 Code Integration

Only one pair integrates code at a time. (because of parallel integration there is a combination of source code which may not have been tested together before. This is likely to lead to problems. Strictly sequential (or single threaded) integration by developers themselves is a simple solution to this problem. All new code is released to the source code repository by taking turns. That is, only one development pair integrates, tests and commits changes at any given moment. Single threaded integration allows a latest version to be consistently identified.)

Developers should be integrating and committing code into the repository every few hours, whenever possible. Continuous integration often avoids diverging or fragmented development efforts, where developers are not communicating with each other about what can be re-used, or what could be shared. Everyone needs to work with the latest version. Changes should not be made to obsolete code causing integration headaches.

4.5 Code Standards & Refactoring

All code must be written to agreed standards (camel casing etc). Committing to use agreed standards allows for consistent code that is easy for the entire team to read and refactor. A refactoring practice is agreed and adhered to, where duplicated code is removed, code integration is increased and the mixture of the code is reduced.

4.6 Collective Ownership

Collective ownership encourages everyone to contribute new ideas to all segments of the project. Any developer can change any line of code to add functionality, fix bugs, improve designs or refactor. No one person becomes a bottle neck for changes. To do this, developers are to create unit tests for their code as it is developed. All code that is released into the source code repository includes unit tests that run 100%. Code that is added, bugs as they are fixed and old functionality as it changed will be covered by automated testing.

4.7 Client Interaction

The client is always available during development for consultation. XP development utilises an iterative development practice where deliverables are produced during each iteration. These deliverables are communicated to the client, and the client can provide valuable feedback.

The client is accessible by all members of the team, and frequent communication can be expected. Regular meetings are planned, and communication via other means (e.g. email) will be used for questions and concerns mid-development.

5.0 References

Amanda. (2014). *How to Ensure High Quality Recording of Research Interviews*. *GMR Transcription Blog*. Retrieved 19 April 2015, from http://blog.gmrtranscription.com/how-to-ensure-high-quality-recording-of-research-interviews/

Balkanski, P. (2003). *Quality Assurance in Extreme Programming*. Retrieved from http://www.foibg.com/ijita/vol10/ijita10-1-p17.pdf

Gray, C. (2010). *Quality Assurance and Assessment of Scholarly Research.* Retrieved 14 April 2015 from www.rin.ac.uk/quality-assurance

Johnson, K. (2011). *Best Practices & Considerations When Conducting Survey Research*. Presentation, Penn State - Survey Research Center.

Osherove, R. (2014). *Unit Testing Lessons in Ruby, Java and .NET.* Retrieved 14 April 2015 from http://artofunittesting.com/definition-of-a-unit-test/

Wells, D. (2000). *Test First.* Retrieved 14 April 2015 from http://www.extremeprogramming.org/rules/testfirst.html

Wray, S. (2010). *How Pair Programming Really Works*. Software, IEEE , vol.27, no.1, pp.50,55, Jan.-Feb. 2010, doi: 10.1109/MS.2009.199