Assignment 12: Quality Assurance Plan

17-654: Analysis of Software Artifacts
Jonathan Aldrich (`jonathan.aldrich@cs.cmu.edu`)

100 points total
This is an Master's (17-654) only assignment

The goal of the quality assurance plan assignment is to provide MSE/MSIT students with a practical opportunity to develop a quality assurance plan for their studio, practicum, or other project.

Groups. Students may work on this project in groups of up to 6. Group projects will be given a single grade. You are free to choose your own groups, and should generally form groups that correspond to existing studio or practicum projects. Students not part of a studio or practicum group may join another group, or develop a QA plan for some other current, past, or future project in which they are involved. Expectations for the level of depth in the plan will be scaled to the size of the group; as a rule of thumb, your QA plan should be about 2 pages (12-point single spaced) per group member.

Collaboration Policy. Since different groups will be working on different projects, the only collaboration policy is that your work must be your own (as always).

1   Quality Assurance Plan (100 points)

Your quality assurance plan is a comprehensive description of how you intend to assure a quality product from your project. It generalizes the usual notion of a test plan to describe strategies for using reviews, static analysis and possibly other techniques in addition to testing.

You may choose any appropriate organization for your quality assurance plan document (if you are stuck, you might consider standards such as IEEE

730-2002, but this is not required). However, the document must discuss all of the following:

Header and Versioning. Your QA plan should include the name of your group or project and the names of the group members. It should include a version number and date. There should be a section describing any changes made to the document after consultation with a mentor that you may have.

Context. Explain enough of the context of your project for reviewers to understand what you are doing and judge if the quality assurance plan is appropriate. This section may be a short summary combined with a pointer to more information (which should be bundled with your QA plan). Define at a high level what quality means to your team, and how you will measure it in your product.

Goals. Define more specifically the quality goals you intend to achieve. The goals should be clearly defined, measurable, and quantitative where possible (e.g. %, # defects/kLOC, limits on # clients for stress testing, etc.). Cover both functional requirements and quality attribute requirements. Describe the scope of your plan; what will you assure, and what will you *not* assure?

Strategy. Begin with a 1-paragraph high-level summary of your quality assurance strategy. Then continue with:

What tools and techniques will you use to assure the quality of each functional requirement and each quality attribute requirement? Consider static analysis, testing, reviews, and other techniques if applicable. Different approaches are likely to be useful for different quality attributes. You may find you want to assure a quality attribute using more than one approach. In this case, explain how the different approaches you chose complement one another.

In this QA document, you must include the reasons behind the strategic choices you make. You should not feel you have to use every technique, but you should justify why you chose the technique you did and why you did not choose other techniques.

If you chose static analysis, why? Which tool will you use, or (perhaps) will you develop a new analysis? What benefits will this give you over other techniques? If you did not include static analysis, why not?

Every plan will probably include some kind of testing. What approaches will you use for testing–e.g., white box, black box, unit, acceptance? Are

there design considerations that will enable testing, such as test scaffolding? How will you approach integration testing and/or regression testing? What is your criteria for completeness of testing?

What documents or parts of the code will you review? What checklists will you use for each relevant document, and for the code? What kind of review will be done in each case (note this may vary for different documents or for different sections of the code)? What process will you use in the reviews–e.g. will you tailor or leave out some of the process elements of the review techniques discussed in class, and if so, why?

How will you assure quality of artifacts other than code?

Organization. What is the total amount of resources available for your project, and what proportion do you intend to devote to quality assurance? How will you organize your quality assurance team? (Note: the complexity of your answer will of course depend on team size)

How will you manage artifacts related to quality assurance, such as version control for test cases, a test automation framework, documentation of tests, review results, and bug databases? What process will you use to ensure that defects found via testing, reviews, or static analysis are properly prioritized and addressed?

How does your QA strategy break down into tasks, how much effort will you devote to each task? When will each task begin and end, i.e. what is the schedule for QA tasks? For each task, what is the exit criterion (i.e. how will you know when you are done?) and what are the deliverables? Note that tasks like testing may be integrated into phases like coding, especially for unit tests in teams using agile processes. If you are using an agile process, you may not have a dedicated test team, but you should still estimate the amount of resources (e.g. person-hours) you will be using to support various aspects of quality assurance on your project.

What will you measure about your quality assurance processes (this may be different for e.g. testing and reviews), and what approach will you use to improve your QA processes as you go along?

References. Throughout your quality assurance plan, refer to other relevant documentation (if applicable) such as requirements, design, or architecture documents for your project. Turn in copies of these documents for review along with your quality assurance plan, as a zip file, to Blackboard. If a particular document would be useful to your reviewers but has not been written yet, say so and provide substitute information if possible.

Mentor Interaction. In the past, enthusiastic students have produced gold-plated QA plans that cover every conceivable aspect of the course, but somehow don't get fully implemented under the deadline pressure of MSE/MSIT studio or practicum projects. We want your QA plan to be useful. **Therefore, we ask you to produce a draft of your QA plan at least 24 hours before the assignment is due and email it to your MSE/MSIT studio or practicum mentor, with a request for comments.** Include a copy of the email, any response you get, and any changes you made in response to that email, along with your QA plan submission. If you do not have a mentor, you may contact the course instructor and TAs.

Grading. The quality assurance plan will be graded according to the criteria listed in the quality assurance plan checklist given below. We expect you to have contacted your mentor 24 hours before submitting the assignment, and we will deduct points if we feel your QA plan is not realistic. We may ask your mentor for feedback on the submitted version of the QA plan, so be sure to respond to their feedback if they give it to you.

2   Review Checklist

Format and Presentation.

- Are all elements of the required header and versioning information present?

- Is the document clearly written, easy to understand, and with correct usage of English?

- Is the project context given sufficient for you to evaluate the QA plan?

- Are necessary reference documents included, such as requirements, architecture, or design documents? Be understanding if they have not completed these documents yet, but ask yourself if they could have provided substitutes.

Goals and Strategy.

- Are the quality assurance goals clearly defined? Are they measurable? Are the goals quantitative where possible?

- Is the scope appropriate given the project?

- Are any major functional requirements or quality attribute requirements missing?

- Is the overview of strategy clear and appropriate to the project?

Approach.

- Is a quality assurance strategy specified for each of the goals given above?

- Is the strategy chosen for each goal appropriate for that functional requirement or quality attribute? Is the strategy explicitly and convincingly justified?

- Does the plan consider (not necessarily choose) aspects of testing, reviews, and static analysis? Does it combine multiple approaches in appropriate ways?

- Does the plan consider quality assurance at early phases of the software lifecycle?

Testing.

- Does the QA plan clearly define the testing approaches to be used?

- Does the QA plan define criteria for when testing is complete?

- Does the plan describe strategic considerations such as test scaffolding, integration testing, or regression testing?

Reviews.

- Does the plan define what will be reviewed, considering (but not necessarily choosing to review) both documents and code?

- Does the plan define the checklists to be used in the review? Are the checklists appropriate?

- Does the plan describe the kind of review and review process to be used?

Organization.

- Does the document describe the quality assurance process and how QA activities will be organized?

- Is the resource allocation reasonable with respect to both available resources and the quality assurance goals of the team? Are estimates of how much resources each task will take realistic?

- Is the management process adequate for ensuring that issues will be addressed? Are appropriate tools and processes chosen to coordinate various aspects of QA tasks?

- Are the deliverables and exit criteria clearly defined for each quality assurance task?

- Is the quality assurance process improvement plan, including applicable measurements, adequate?

- Are all processes and techniques proposed relevant and specific to the project?