**Reviewer(s): Daniel Martinez, Michael Degan, Cody Curry, and Chris Wilkins Date(s): 9-23-14**

Inspection Checklist for Software Requirements Specifications (SRS)

The inspection/review of the requirements is a formal scrutiny of the list of requirements to ascertain different criteria including that the list is complete and consistent, and each requirement is unambiguous, and can be tested. When examining the requirements the reviewer might use the following questions to see that the document is adequate.

## Organization

Is the implementation priority of each requirement included (*with descriptions of priorities provided*)?

Are all internal cross-references to other requirements correct?

* The truncate command needs to be fixed with respect to this issue

**Completeness**

Have algorithms intrinsic to the functional requirements been defined?

Are all requirements written at a consistent and appropriate level of detail?

Does the SRS include all known customer or system needs? *Are all the tasks the user want being specified?*

Is any necessary information missing from a requirement? If so, is it identified as TBD?

* Some requirements are missing info, and we have yet to identify it as a TBD.

Is the expected behavior documented for all anticipated error conditions?

## Correctness

Do any requirements conflict with or duplicate other requirements?

* No requirements conflict although there is some duplicate information, which only helps enforce and clarify necessary requirements.

Is each requirement written in clear, concise, unambiguous language?

* For the most part, the document is clear and unambiguous.

Is each requirement verifiable by testing, demonstration, review, or analysis?

Is each requirement in scope for the project?

Is each requirement free from content and grammatical errors?

* No there were some grammatical errors which were logged by the inspector and stored in the groups e-portfolio (GitHub).

Can all of the requirements be implemented within known constraints?

Are any specified error messages unique and meaningful?

## Quality Attributes (Non functional requirements)

Are performance requirements (such as response time, data storage requirements) stated?

* NA

Are all security and safety considerations properly specified?

* NA

Are other pertinent quality attribute goals explicitly documented and quantified, with the acceptable tradeoffs specified?

* NA

## Traceability

Are requirement dependencies being identified?

* They are defined only implicitly.

Is each requirement uniquely and correctly identified?

* No, we need to added bulleted list to better identify requirements.

Is each requirement traceable to its source?

## Special Issues

Do the requirements provide an adequate basis for design?

Are there requirements that contain an unnecessary level of design detail?

Are all requirements actually requirements, not design or implementation solutions?

Is each requirement testable?

Have internationalization issues been adequately addressed?

* NA

Have localization issues been adequately addressed?

* NA

If the requirements involve complex decision chains, are they expressed in a form that facilitates comprehension (i.e., decision tables, decision trees, etc.)?

* There are no complex decision chains but a picture demonstrating flow would have help assisted with understanding.

Have requirements for performing software upgrades been specified?

* NA

Have the real-time constraints been specified in sufficient detail?

* NA

Has the precision and accuracy of calculations been specified?

Is it possible to develop a thorough set of tests based on the information contained in the SRS? If not, what information is missing?

Have Assumptions and Dependencies been clearly stated?