**Reviewer(s): Michael Degan, Chris Wilkins, Daniel Martinez, and Cody Curry Date(s): 9/24-9/25/14**

Inspection Checklist for Software Design Specifications (SDS)

Reviews for software design focus on data design, architectural design, and procedural design. In general, two types of design reviews are conducted. The preliminary design review assesses the translation of requirements to design of data and architecture. The second review, often called design walkthrough, concentrates on the procedural correctness of algorithms as they are implemented within program modules. Below you can find two checklists, one for each kind of design review.

## (1) Preliminary Design Review

**Traceability**

Did you ensure traceability of the design back to the systems specification and statement of requirements? See RTM.

* Could have been improved

Is a scheme used for naming of modules, data, and interfaces?

Are all modules, data, and interfaces uniquely identified?

**Consistency**

Is the data structure consistent with the information domain?

Is the data structure consistent with software requirements?

Is a standard design representation used?

Is a standard data usage representation used?

* NA

**Completeness**

Are software requirements reflected in the software architecture?

Are all referenced data defined?

Are all defined data used?

* No.

Are all referenced modules defined?

Are all defined modules used?

* No.

Are interfaces defined for modules and external elements?

Has maintainability been considered?

* No, it is assumed it dies on the vine.

Have quality factors been explicitly assessed?

* No.

**Efficiency**

Are data grouped for efficient processing?

* Not considered.

Are storage requirements allocated to design?

* Not considered.

Is effective modularity achieved? Are modules functionally independent?

**(2) Design Walkthrough**

Does the algorithm accomplish the desired function?

Is the algorithm logically correct?

Is the interface consistent with the architectural design?

Is the logical complexity reasonable?

Has error handling been specified?

Are local data structures properly defined?

Are structured programming constructs used throughout?

Is design detail amenable to implementation language?

Has maintainability been considered?

* No, it is assumed to die on the vine.

Are all conditions and processing defined for each decision point?

* Was not adequately examined but group agree: probably not.

Do all defined and referenced calling sequence parameters agree?

Appendix A – Requirements Traceability Matrix (RTM)

The following table is an example of what may be utilized for a Requirements Traceability Matrix (RTM). This may be customized to fit the stakeholder’s needs.

* For this assignment, we have decided to leave this section in. It is clear that we did not utilize a RTM but we have acknowledged it and as such will include it with our Design Checklist.

| ID | System Requirement | Use Cases | Design Elements | Test Cases |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |