|  |  |
| --- | --- |
| Module | Design Principles |
| Sub-Module |  |
| Author | D Twaddell |
| Date | 20 Aug 17 |
| Reviewer |  |
| Review Date |  |
| Review Status |  |
| Review Comments |  |

## Design Log

# rDQ Design Principles

Principles are there to make sure we deliver to expectations, and all modules will comply to these principles, although exemptions can be applied for in certain circumstances.

## Overarching Design Principles

1. **Integration** - All functions available via UI and also by direct function call
2. **Scalability** - All functions must be able to operate over large datasets (1 million rows to start. Adding big data capabilities later)
3. **Style** - All code com plies with style guides (Google R, and internally agreed DB standards)
4. **Ease of implementation** - Entire application delivered as a downloadable package
5. **Parameterisation** - all parameters must be configurable by the user, not hard-coded
6. **Usability** – all UI modules to comply with UI Skeleton. All functions can be run directly from the console, or integrated into the rDQ UI (or a different UI)

## Database Standards

The data model will be developed and maintained in mySQL, and shared via Git. The actual database will be implemented initially in SQLite, but later releases will allow user to select a database, and will support mySQL and HIVE for ‘big data’.

### Database naming standards

Table naming standard: first three chars lower case and unique to the table. Remain characters are meaningful word describing the table contents

Attribute naming standard: first three chars lower case and same as the table prefix. For a non-association attribute, i.e. an attribute that doesn’t link to another table, the remaining characters are a meaningful word that describes the attribute. For an attribute that links to another table, i.e. a foreign key, the format is:

<table prefix>\_<relationship name>\_<foreign attribute name>

For example: par\_parent\_modId