

0.1 Architectural Requirements Review

Software quality requirements

- **Reliability:** The switch statement's (`getThreadAppraisal.js`) default case was not specified. If an error occurred, it wouldn't be traceable since there is no error handling. If "garbage" values were passed through this switch statement, the error's result could be a system failure.
- **Maintainability:** There is code duplication within the conditional statements. Many statistical operations exist and should someone wish to add them, it would be a daunting task. A function, instead of duplicated code, would simplify the modifications.
These two files (`importThread.js` and `importThreadAppraisal.js`) also have duplicate code, the effects are similar to the ones mentioned earlier.
- **Test-ability:** Unit tests were included for each function. This reduces single-point-of-failure, because errors in smaller functions can be addressed without affecting the whole system.
- **Audit-ability:** (`getThreadStats.js`) has a few entries for auditing (ParentID, Author, Timestamp, Content, Status). These are not adequate for auditing, especially since there's a single timestamp (which was ambiguous).

Architectural components

- **Framework(s):** Node.js was used, as specified. Node-aop and Broadway plugin framework were included in "package.json" but they both appear unused.
- **Technologies:** JSON (JavaScript Object Notation) was used to store "thread objects". HTML5 was used, however, this was for testing the low-level implementation. Electrolyte was included in "package.json" but there were no signs of it's use for the actual implementation.