**Master Test Plan**Team Decided - Raft Consensus Library

## Testing Strategy

### Test Driven Development (TDD)

Test driven development will be utilised in this project; this methodology focuses on testing the design specifications is accurately implemented in the code, rather than testing just the implementation of code itself. If all of these test are written and pass, then it can reliability be confirmed that the code is accurate to the specification, no more no less.

### Unit testing (UT)

These tests are directly related to the use of TDD, and confirm that code is meeting the requirements of each unit’s functionality. Unit testing is the set of atomic tests on the implementation of each object’s public contract to ensure they meet requirements. If all of these tests are written and pass, then it can be reliably be confirmed that each unit of code is correctly performing it’s functionality.

### Integration Testing (IT)

These tests are directly related to the use of TDD, and confirm that code is meeting the requirements of the design’s use cases. These IT are simply groups of UTs which form together to make a use case. If all of these tests are written and pass, then it can be reliably be confirmed that the code is performing all the use cases to specification.

### Code review (CR)

During development, strict adherence to coding guidelines greatly improve maintainability with a flow on effect of code reliability. To perform code review, a developer writes or makes changes to the code base, and then a separate developer reviews and audits line-by-line those changes, ensuring quality and implementation ideas match design. This reduces, and ideally removes, obvious logic errors, code smells, improper implementation and other various code issues.

### Prototype (PT)

The prototype will include a minimal functional feature set of the library which can be used to confirm successful implementation of the consensus algorithm. This prototype will also be confirmed by the TDD life cycle.

### Demo (DM)

The demo is the full implementation of all code library features. It is continual development of the prototype. This demo can be used to show correct executing behaviour in a production environment. The demo will pull it’s consensus library code through Nuget, and be developed separately from the code library to simulate real world production usage.

### Developer Evidence (DE)

Some functionality is not so black/white where it can be confirmed through software tests, for this we have the professional work of developers confirming functionality. For example, minimal resource usage cannot be unit tested, however a developer can give their professional results confirming the matter.

## Tests to be Conducted

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Feature/functionality** | **Testing environment** | **Acceptance criteria** | **Role** | **Planned Stage** |
| 1 | Can read/write to debug log | Dev. Evidence | Evidence of prototype completing action | Joshua | Prototype |
| 2 | Can read/write to distributed log | Prototype | Evidence of prototype completing action | Sean | Prototype |
| 3 | Can send and receive messages from other nodes | Prototype | Test 2 has pass result | Joshua | Prototype |
| 4 | Can communicate using encrypted messages | Dev. Evidence | Reasonable developer evidence provided | Sean | Prototype |
| 5 | Node authenticates using zero knowledge password proof | Dev. Evidence | Reasonable developer evidence provided | Joshua | Prototype |
| 6 | Does library call UAS Start/Stop | Prototype | Evidence of prototype completing action | Sean | Prototype |
| 7 | Can hold successful election | Prototype | Evidence of prototype completing action | Joshua | Prototype |
| 8 | Bring node log up to date | Prototype | Evidence of prototype completing action | Sean | Prototype |
| 9 | Falls to follower when detecting newer leader | Dev. Evidence | Reasonable developer evidence provided | Joshua | Prototype |
| 10 | Can maintain service during node failure/loss | Prototype | Evidence of prototype completing action | Sean | Prototype |
| 11 | Can recover from node failure/loss | Prototype | Evidence of prototype completing action | Joshua | Prototype |
| 12 | Consensus between distributed systems | Dev. Evidence | Test 2 has pass result | Sean | Prototype |
| 13 | Fault tolerant distributed service | Dev. Evidence | Test 2,10, and 11 have pass results | Joshua | Prototype |
| 14 | Security - All comms encrypted | Dev. Evidence | Test 4 and 5 have pass results | Sean | Prototype |
| 15 | Privacy - Joining securely to cluster | Dev. Evidence | Test 14 has pass results | Joshua | Prototype |
| 16 | Cross Platform - Dev. language | Dev. Evidence | Reasonable developer evidence provided | Sean | Prototype |
| 17 | Mitigate project abandonment | Dev. Evidence | Reasonable developer evidence provided | Joshua | Prototype |
| 18 | Compatibility - Dev. language popularity | Dev. Evidence | Reasonable developer evidence provided | Sean | Prototype |
| 19 | Troubleshooting - Logging | Dev. Evidence | Reasonable developer evidence provided | Joshua | Prototype |
| 20 | Reliability | Dev. Evidence | Test 13 has pass result | Sean | Prototype |
| 21 | Minimal overhead/impact to service performance | Dev. Evidence | Reasonable developer evidence provided | Joshua | Prototype |
| 22 | Usability - Minimalistic public interface | Dev. Evidence | Reasonable developer evidence provided | Sean | Prototype |
| 23 | Availability - Can run locally and over internet | Dev. Evidence | Reasonable developer evidence provided | Joshua | Prototype |
| 24 | Confirm library is one-click integratable from Nuget | Dev. Evidence | Reasonable developer evidence provided | Sean | Prototype |
| 25 | Full project code review | Dev. Evidence | Reasonable developer evidence provided | Joshua | Version 1.0 |
| 26 | Redesigned unit testing suite for extended verification of existing functionality | Dev. Evidence | Reasonable developer evidence provided | Sean | Version 1.0 |
| 27 | Cluster can grow upon new node | Demo | Evidence of demo completing action | Joshua | Version 1.0 |
| 28 | Cluster can shrink upon losing node | Demo | Evidence of demo completing action | Sean | Version 1.0 |
| 29 | UAS can attempt to change leader of cluster | Demo | Evidence of demo completing action | Joshua | Version 1.0 |
| 30 | Persistent Log implementation  (“Log compaction”) | Demo | Evidence of demo completing action | Sean | Version 1.0 |
| 31 | **(Optional feature)** Support for upgrade path | Demo | Evidence of demo completing action | Joshua | Version 1.0 |
| 32 | **(Optional feature)** Completed performance analysis/optimization of code | Dev. Evidence | Reasonable developer evidence provided | Sean | Version 1.0 |
| 33 | Quality - Full code coverage unit testing | Dev. Evidence | Reasonable developer evidence provided | Joshua | Final |
| 34 | Documentation - Examples/XML comments | Dev. Evidence | Reasonable developer evidence provided | Sean | Final |
| 35 | Testability - Open sourcing unit tests | Dev. Evidence | Reasonable developer evidence provided | Joshua | Final |
| 36 | Extendability - Open sourcing code | Dev. Evidence | Reasonable developer evidence provided | Sean | Final |
| 37 | Auditability - Open source and logging | Dev. Evidence | Reasonable developer evidence provided | Joshua | Final |
| 38 | Code library available through Nuget | Dev. Evidence | Reasonable developer evidence provided | Sean | Final |