**Sprint Zero Artifacts**

1. The Contractor shall produce, review and written agreement from the State for all Sprint Zero artifacts prior to commencing delivery sprints.
2. The Contractor shall produce a User Story Definition and Acceptance Criteria Format.
3. The Contractor shall produce Coding Standards (including style guidelines) and Commenting Standards, including Peer Review checklist.
4. The Contractor shall produce a Sprint-level Definition of Done that includes the following concepts:
5. Code produced (all ‘to do’ items in code completed)
6. Code commented, checked in and run against current version in source control
7. Peer reviewed (or produced with pair programming) and meeting development standards
8. Builds without errors
9. Unit tests written and passing
10. Deployed to system test environment and passed system tests
11. Passed Product Owner Acceptance Testing
12. Any build/deployment/configuration changes Implemented/documented/communicated
13. Relevant documentation produced/updated (e.g., how the API services supports service module needs and user stories, user stories, sketches, wireframes, clickable prototypes)
14. Remaining hours for task set to zero and task closed
15. The Contractor shall produce a Release-level Definition of Done that includes the following concepts:
16. Release Notes Prepared
17. Deployed to staging environment and integration, performance and load tests run
18. Relevant documentation/diagrams produced and/or updated

**Sprint Planning and Execution**

1. The Contractor shall use an Agile Sprint Planning and User Story Approval process for each Sprint. The Agile Sprint Planning process includes the following activities: Product Backlog refinement, User Story creation, estimation and commitment.
2. The Contractor shall demonstrate that each User Story has met the definition of done so that the State Product Owner can approve each User Story as it is completed.
3. The Contractor shall utilize scrum-based agile processes in sprint execution activities (e.g., User Story development, Product Backlog maintenance, User Story acceptance, Retrospective, and Product Review).
4. The Contractor shall revise Sprint Zero artifacts during each Sprint Retrospective process, as appropriate.
5. The Contractor shall generate documentation within the code itself and within the version control system (e.g., through proper use of descriptive commit messages, issue tracking, pull requests, etc.).
6. The Contractor shall use Pivotal Tracker to manage the Product Backlog, User Story acceptance, and maintain a scrum board.
7. The Contractor shall use Slack as the primary mechanism for project-related communication and real-time messaging, archiving and search for all project teams.
8. The Contractor shall provide a report at the conclusion of each sprint that documents the planned user stories, accepted user stories, open impediments, and technical debt.
9. The Contractor shall adhere to Twelve-Factor Application design constraints (see [http://12factor.net/](http:///h)).

**Modularity**

1. The Contractor shall design the application architecture to ensure a separation of concerns and a reasonable degree of modularity between systems.
2. The Contractor shall adhere to the DRY (Don’t Repeat Yourself) principle to ensure that the codebase remains flexible.

**Code Style**

1. The Contractor shall ensure that all code will be written to a language specific code-style guideline (e.g., PEP8 for Python).
2. The Contractor shall use an automated tool to evaluate the codebase and ensure compliance with the code-style guideline (e.g., if the Contractor uses Python, PyLint may be used).

**Version Control System**

1. The Contractor shall manage all assets (e.g., source code, automated tests, user stories, configuration files, knowledge transfer material, etc.) using GitHub.

**Code Review**

1. The Contractor shall ensure all code written by one developer is reviewed by another developer before merging into the mainline codebase.
2. The Contractor shall follow industry standard code review practices (e.g., <http://blog.fogcreek.com/increase-defect-detection-with-our-code-review-checklist-example/>).

**Automated Testing**

1. The Contractor shall create and execute automated unit testing.
2. The Contractor shall create and execute automated system tests to verify all Features of the software module.
3. The Contractor shall create and execute automated Product Owner Acceptance testing to verify all user facing functionality.
4. The Contractor shall run tests automatically on code merged into version control.
5. The Contractor shall use an automated tool that measures the amount of the codebase that is covered by tests (e.g., RCov may be used to measure test coverage of Ruby code).
6. The Contractor shall create and execute automated integration testing with other contractor developed modules.
7. The Contractor shall make the bugs identified during testing available to view real-time and on a historical basis.

**Load Tests**

1. The Contractor shall create and execute load and performance tests at regular intervals, and at each release.
2. The Contractor shall provide a summary of all load and performance test results.

**Accessibility**

1. The Contractor shall incorporate and test accessibility throughout the design and development processes (see section 508 Amendment to the Rehabilitation Act of 1973).
2. The Contractor shall use an automated accessibility testing tool (e.g., Pa11y).

**Issue Tracking**

1. The Contractor shall use GitHub to keep track of all bugs and application issues.

**Mobile Friendly**

1. The Contractor shall design the User Interface (UI) using responsive design.

**Logging and Monitoring**

1. The Contractor shall implement centralized and continuous monitoring.
2. The Contractor shall implement centralized system logging.
3. The Contractor shall implement auditing.

**Security**

1. The Contractor shall use an automated black/white box security scanning tool (e.g., HP Fortify, or [https://hakiri.io](https://hakiri.io/)) to ensure a minimal baseline of security at regular intervals, and at each release.
2. The Contractor shall provide the results of the security scans to the State.
3. The Contractor must adhere to the HTTPS-Only Standard as outlined in <https://https.cio.gov/>.
4. The Contractor shall adhere to the baseline moderate tailored NIST 800-53 (see Attachment).
5. The Contractor shall ensure adequate security controls using penetration testing, red teaming, etc.

## User Authentication and System Administration

1. The Contractor shall ensure that the module integrates with the existing authentication protocols, which involves Active Directory and IBM’s RACF technology.
2. The Contractor shall enable external service authentication and authorization using a modern single sign-on tool (e.g., Shibboleth and OAuth 2.0).

**Build and Deployment**

1. The Contractor shall provide continuous integration of source code into the source code version control system.
2. The Contractor shall use a continuous source code build tool that enables continuous deployment of all applications into testing and staging environments.
3. The Contractor shall include mock test data that should be publicly accessible for development by other module Contractors and not include personally identifiable information (PII).
4. The Contractor shall use at least one of the following methods to deploy code changes to a staging environment under the control of the Contractor with the issuance of a single command:
5. Containerization (e.g., Docker Engine, Rkt, and Warden)
6. Configuration Management tools (e.g., Chef, Puppet, Salt, and Ansible)