

Development - QA Test Plan Template

12/29/2023 4:25 pm EST

Overview

- We will be testing in the following areas
 - Bug specific testing
 - New development testing
 - Regression testing
- Goals will be to test and pass changes and new development without introducing new bugs or errors
- We will use pre-set VMs with pre-set data sets for consistent data testing. Test plans will include data specifics from the pre-set data.
- Have a multiple VM setup in place
- Each VM or series of VM will be used to test specific areas
- Each VM will have a snapshot done once loaded with the pre-set data for roll back
- Each VM will have a secondary snapshot done when new development is done and merged to master
- Each VM (with the exception of one regression VM) will start on the latest released patch
- QA will work within a 4 week patch release cycle

Quality Risks

- It is impossible to test every possible data set. This could result in some scenarios that would leave some existing bugs in place and possibly introduce new bugs.

VM Structure

- Core process structure – 2 servers
 - Used to test core processes such as Appserver, Broker, Signal Handler
 - Used to test load balancing and maxing
 - Set up automated testing that includes rapid login/logoff
 - Set up automated create/save/delete basic customer
 - Use the FEP in demo mode and the Auto Client in demo mode to load test the FEP and Signal Handler
 - Establish replication between the core processes servers
 - Used for bug testing any bugs that have core process changes
 - Used for license testing
- Reports – 1 server
 - Used for testing new reports
 - Used for testing scheduled reports

- Set up automated running of each report (as is currently done on QANIRVANA)
- Used for testing report load balancing
- Can connect to the core process servers
- Manitou Web Client/VCC – 1 server
 - Used for testing bug fixes to the Manitou Web Client
 - Will use its own set of core services
 - Will have the licensing for all VCC drivers
- Previous Patch – 1 server
 - Used for comparison of possible regressions or existing bugs
- Regression testing – 1 server
 - Has complete core services
 - Has all peripherals installed (snap reporter, location services, notify me, auto dispatch, etc)
 - A snap shot will be taken prior to pre-release patch placement
 - Install the pre-release patch
 - Use regression testing test cases to test all the areas
 - Roll back to snapshot if necessary, for re-testing, additional bug fixes, bugs cherry picked into the release

Testing Process

- 1-2 QA will be assigned to working on bug fixes/support issues
 - QA will work with the developer assigned to the bug to set up the test environment
 - QA will create a pre-outlined test case for the bug
 - QA will step through the test case with the pre-set data on the appropriate VM to test the bug
 - Development will make QA aware of any effected areas that need to be tested
 - Additional testing around the areas of development changes will be done to cut back on the time for regression testing
- 1-2 QA will be assigned to working on new development
 - QA will review any documentation that the developer has put together in regards to the new development
 - QA will start to work through the documents to install and set up the new development
 - The developers will work directly with QA testing the new functionality/development
 - During the process of setting up and testing QA will create permanent test cases to use for future testing
- All QA will assist in regression testing for patch release

Bug and regression tracking

- If a bug being tested is not working as expected or causes unexpected behaviors in the same area of development, QA will work with the developer until all issues are resolved before closing out the bug.
 - If it is found that the issue (not specifically the bug that was fixed) was there in previous versions it will be

decided if a new bug should be entered and close the existing bug and open a new bug for the new found issue.

- When working in new development QA will work with the developer through each piece of the development and close out the children of the feature/epic as they are tested.
 - Regressions and bugs found around the area of development will be entered in as new bugs and tied to the feature or the epic
 - The feature and or epic will not be merged with master until all the children below are closed
- Bugs found during regression testing
 - QA will first see if the bug existed in a previous patch
 - If the bug existed in a previous patch QA will see if there is an existing bug for it and if not enter a bug
 - The QA team and management will decide if the bug fix needs to be included in the patch
 - If the bug is a true regression a bug will be entered and handed off to development to correct
 - When the bug has been fixed QA will verify the fix and cherry pick the bug into the patch release branch
 - QA will write a test case specific to the regression to insure that the particular area is looked at more closely with the specific data sets for the next patch

Development/QA testing cycle

- Patch release will occur every 4 weeks
- Week 1 of the Development cycle will be the week QA is regression testing for patch release
 - This will give time for Development to set up their coding plans and make the code changes
 - By doing more testing around areas of code changes we can reduce the time needed for regression testing
 - One week of all QA regression testing using automation and regression test cases on a single regression server
 - Utilizing the Core Services VM to test load balancing and automation

Week 2-4 QA will work with the developers assigned to their tickets (either bugs/support, or new development) to test and complete and merge changes to master for the next patch release.