Contents

1 Overview

1.1 Purpose

When development and initial testing for a transaction or project is complete, it must pass quality assurance testing before it can be delivered to a customer. Only items built according to the approved build procedures can be delivered to QA.

This document describes how to perform the QA process.

1.2 Terms, acronyms, and definitions

Term/Acronym	Definition
AM	Activity monitor. Companion dashboard application to TicketSystem.
Build	A compiled piece of software.
CRF	Change request form. A TicketSystem ticket type.
Delivery ticket	A CRF in TicketSystem that groups project task tickets (transactions, label designs, etc.) together.
ERP system	Enterprise resource planning system. "A type of software system that helps organizations automate and manage core business processes for optimal performance." Used by customers to manage and keep track of inventory.
QA	Quality assurance.
QA ticket	A CRF in TicketSystem that is created by QA testers during the testing process. Assigned to consultants who worked on the transaction.
Released build	A build that has passed QA and has been placed in a customer's Client Release folder.
Regression testing	The process re-running functional and non-functional tests to ensure that previously developed and tested software still performs as expected after a change.
SOW	Statement of work. A document that outlines the software specifications for a customer's project.
Task ticket	A CRF in TicketSystem for a specific transaction or task in a customer project. Assigns development work and QA work and is often linked to a delivery ticket.
TicketSystem	Internal customer issue/ticket tracking application.
RDP	A proprietary protocol developed by Microsoft which provides a user with a graphical interface to connect to another computer over a network connection.

1.3 Referenced documents

- QA Complete – email template
- QA Complete with Issues – email template
- QA Failed – email template
- QA Update – email template
- Release to QA – email template

1.4 Tools

- Android Studio – Windows software
- BareTail – Windows software
- Activity Monitor – website
- Microsoft Outlook – Windows software
- Microsoft Word – Windows software
- Pleasant Password Portal – website
- PuTTY – Windows software
- SERVER6 – file storage server
- TicketSystem – website
- Remote Desktop Connection – Windows software

2 Work instructions

2.1 QA notification of work

QA is notified of work to complete by one (or more) of three ways:

1. Project management

   QA attends a weekly meeting with project management over Microsoft Teams and discusses ongoing customer deliveries as well as new projects scheduled for QA.

   > During this meeting, the project manager updates the QA schedule as appropriate.

2. Consultant email

   When work has been completed by a consultant, the consultant will send an email to the QA distribution list (qagroup@company.com) using the “Release to QA” email template.

   > The email will serve as notification that the transaction is developed and tested to the written specification in the customer SOW.

3. AM dashboard/TicketSystem task tickets

   The GR QA Tasks dashboard in AM is used by project management and displays necessary information about pending QA tasks—due dates, delivery dates, ticket summaries, the consultant who completed the work, and the QA tester responsible for testing.

   > All task tickets that are set to “QA Ready” status appear in this dashboard and are considered ready to test.

2.2 First QA cycle

The QA tester will ensure the released transaction specified is properly completed by the consultant.

Using the information supplied in the corresponding task ticket(s) and/or "Release to QA" email sent by the consultant, the QA tester will:

1. Locate the customer's folder on SERVER6 using File Explorer, identify the correct build in the customer's Builds folder, and create a local copy for testing.

   EXAMPLE DIRECTORY

   ```
   \\SERVER6\$ERP\Clients\$CustomerName\$Builds
   ```

2. Navigate to the customer's SOW folder and open the appropriate SOW version in Microsoft Word.

   EXAMPLE DIRECTORY

   ```
   \\SERVER6\$ERP\Clients\$CustomerName\$SOW
   ```

3. Ensure the build's major and minor version numbers (v0.0.0) correctly match the SOW's major and minor version numbers (SOW v0.0).

4. Ensure the build's patch number (v0.0.0) is appropriate.

   EXAMPLE

   If build v1.2.2 has been delivered to the customer and more work for SOW v1.2 is being done, the next build to arrive in QA should be v1.2.3.

5. Connect to the customer's network via their VPN.

   PLEASANT PASSWORD

   Credentials are stored in customer-specific folders in Pleasant Password Portal:
   `Company > Enterprise Dev > Customer Remote Acecss > $CustomerName`

6. Create an RDP session with the Remote Desktop Connection application (or an SSH/telnet session with PuTTY if the customer is running Unix) and connect to the specified environment.

> Most customers have two environments: production and test. Some customers may have more. Unless directed otherwise, all Enterprise testing and development should be performed in the customer's test environment.

7. Locate the appropriate instance of the customer's ERP system, start a session, and log in.

> Logging into the ERP system will allow the QA tester to search for valid test data. Enterprise aims to automate the customer's process flow; understanding what transactions are being automated and what data is required to perform them will assist with testing.

8. Locate the directory where Enterprise has been installed and navigate to the test BaseDev Logs folder for monitoring.

EXAMPLE DIRECTORY

```
D:\Enterprise\DC\Test\Services\BaseDevs\Logs
```

BareTail is present most on customers' environments in the Utilities folder where Enterprise is installed. It will be used to open and follow logs during testing.

EXAMPLE DIRECTORY

```
D: \Enterprise\DC\Utilities
```

9. Set up the build.

If testing the Android version:

   a. Start the Android Studio application.

   b. Navigate to the Virtual Device Manager and launch an emulator.

   c. Delete the previous `.db3` file in the emulator's Downloads folder (if applicable).

d. Open the build’s folder and navigate to the Android Test folder.

EXAMPLE DIRECTORY

```
C:\QA\TestBuilds\v1.2.3.0_201906181320\Android\Test
```

e. Drag the build’s test `.db3` file from the build’s folder onto the emulator’s viewport. This will install it.

f. Launch the application on the emulator.

If testing the PC version:

a. Right-click and copy the build.

b. Navigate to the directory where Enterprise is installed on the RDP session and go to the client Testing folder.

EXAMPLE DIRECTORY

```
D:\ Enterprise\DC\Clients\Testing
```

c. Paste the build.

d. Open the build’s folder, navigate to the PC Test folder, and launch the `.exe` file.

EXAMPLE DIRECTORY

```
D:\ Enterprise\DC\Clients\Testing\v1.2.3.0_201906181320\PC2\Test\
```

10. Log in to the Enterprise mobile client.

The QA tester will then test the build according to what is detailed in the customer’s SOW—and, in the case of a customer support issue, any supplemental information provided by the support team and/or consultant.

2.2.1 Testing a build

Testing a build involves:

1. Stepping through the transaction(s) manually, attempting as many scenarios as possible based on the criteria outlined in the customer's SOW.

> Scenarios are often varied, as every customer's project is uniquely suited to their needs. Depending on the customer's ERP system, there may be scenarios for lot-controlled parts, serialized parts, certain locations, or certain materials.

2. Reviewing all prompts and text written in red on the SOW transaction page, as this indicates new changes for that SOW version.
3. Ensuring that the transaction's behavior matches what is described in the SOW. Each transaction must accurately reflect:
    a. the prompt list
    b. the written text
    c. the transaction flowchart
4. Checking for cosmetic flaws such as misspellings, UI inconsistencies, display problems, and various prompt, error, or message texts that do not match the SOW.
5. Checking for difficulties in usability—if the transaction is easy to use, if the flow makes sense, if any factors complicate the process.
6. Confirming that the customer's ERP system, if applicable, has been correctly updated by the transaction(s) and that there is no missing data.

REGRESSION TESTING

Testing a build may or may not include regression testing. This is dependent on the customer and what new changes, configurations, or functionality have been added to their project. Some changes may affect past functionality while others may not.

Transactions in a project are often treated as individual pieces of a whole; a change in one transaction generally does not affect another transaction. However, if a change was made to a message that multiple transactions use (for example, a specific query or validation), then those transactions must undergo regression testing.

2.2.2 Reporting issues or defects

If issues/defects are discovered while testing, for each specific issue the QA tester will:

1. Create a new CRF-type ticket under the appropriate TicketSystem customer using the “New Change Request” button. This will be a QA ticket.

    a. The “Assigned To” field will be set to the consultant who worked on the transaction.

    b. The “State” field will be “New”.

    c. The “Category” field will be set to “Internal QA Problem”.

    d. The “Est Hrs” field will be set to 2.

    > This is a default value to ensure the QA ticket is calculated into the consultant’s weekly hours in AM and appears on their dashboard. The consultant may take more or less time to address the issue, depending on its complexity.

    e. The “Target Date” field will be set to two days after the current day’s date.

    f. The “QA Assigned To” field will be set to the QA tester’s name.

    g. The “Primary Product” field will be set to “Bar Code Data Collection”.

    h. The ticket will be given a name in the “Summary” field in the following format:

    ```
    QA - $TransactionName - $BriefDescription
    ```

    i. The “Description” field will state specifically how the transaction is not to the specification.

    j. Data used to (re)produce the issue will be entered into the “Testing/Installation Notes” field.

2. Save the ticket.

3. Upload relevant or supporting screenshots.

    a. On the right-hand sidebar in the “Attachments” section, click the plus button.

   b. Select a file to upload by left-clicking on the folder icon and navigating to the image in File Explorer.

   c. Click “Upload”.

4. Link the QA ticket to the project’s delivery ticket.

   a. On the righthand sidebar in the “Related Tickets” section, click the small link button.

   b. Enter the delivery ticket’s number into the “Ticket To Link To” field.

   c. Click “Save”.

   > Linking QA tickets to delivery tickets ensures that the QA tickets properly appear on the consultant’s AM dashboard.

After the QA ticket has been saved, the assigned consultant will get an email notification that a CRF has been created and they can start working to fix the problem.

2.2.3 Transaction testing completed with found issues

When testing is finished for the specified transaction and issues have been reported in TicketSystem, the QA tester will:

1. Create a “QA Failed” email addressed to the assigned consultant and to the QA distribution list. The email subject will be in the following format:

   ```
   QA Failed - $CustomerName - $TransactionName - $0 Issues
   ```

2. Paste the link of each QA ticket created in TicketSystem for the transaction into the body of the email using the “QA Failed” email template.

3. Send the email.

This email will serve as notification that the transaction will not be reviewed again until a new build is made and delivered accordingly.

2.2.4 Transaction testing completed with no found issues

If there are no problems discovered while testing the transaction, the QA tester will:

1. Create a “QA Complete” email to the consultant and the QA distribution list. The email subject will be in the following format:

```
QA Complete - $CustomerName - $TransactionName - 0 Issues
```

2. State in the body of the email that QA is finished and the transaction is ready for delivery using the “QA Complete” email template.
3. Send the email.
4. Set the “Status” field of the transaction’s task ticket to “QA Complete”.

2.3 Development between QA cycles

The consultant may begin development work on newly created QA tickets while the first cycle of QA is still taking place.

2.3.1 QA ticket development complete

When development is completed for a QA ticket:

1. The consultant will add an entry to the “Current Status” field and change the “Status” field to “QA Ready”.
2. The QA tester assigned will receive an email notification from TicketSystem that this issue is ready to be reviewed in the next QA cycle.

2.3.2 Transaction development complete

When development is completed for all QA tickets related to the transaction:

1. The consultant will create a new build and test it accordingly.
2. After testing, the consultant will send an email using the “Release to QA” email template to the QA distribution list stating the transaction is ready to test in the next QA cycle.

2.3.3 Inapplicable QA tickets

If a QA ticket is not applicable or does not conform to the SOW:

1. The consultant will add a timestamped entry in the “Current Status” field stating the reason.

   > A consultant may challenge any QA ticket by providing documentation in the form of the SOW or ERP-specific documentation that refutes the point. ERP-specific information includes, but is not limited to, technical specifications or help documentation provided by the ERP vendor.

2. The consultant may contact the QA tester for discussion or further explanation.
3. The QA tester or the consultant will set the QA ticket’s “Status” field to “Closed”.

2.4 Second QA cycle

The QA tester will:

1. Test the build again according to what is outlined in the customer's SOW.
2. Create a new QA ticket for each new issue discovered while testing, following the procedure as described in the first cycle.
3. Add an "Observation" type Activity to each existing QA ticket if the issue has not been correctly resolved.
    a. The "Description" field will state specifically how the transaction continues to function incorrectly and specifically how it should function/is expected to function according to the SOW.
    b. The data used to (re)produce the issue will be updated in the "Testing/Installation Notes" field if necessary.
    c. The "Status" field of the QA ticket will be set to "QA Rejected".
4. Edit and close each QA ticket that has been correctly resolved.
    a. A new timestamped entry will be added into the QA ticket's "Current Status" field stating that the modifications are working in the specific build that was tested.

    EXAMPLE

    07-Dec-2022 10:06 - CCofer: Working in v1.2.3.

    b. The QA ticket's "Status" field will be updated to "Closed".

2.4.1 Outstanding QA tickets

If there are open QA tickets at the end of the second QA cycle, the QA tester will:

1. Create an email addressed to the assigned consultant and to the QA distribution list with the "QA Failed" email template.
2. List the links of all the QA tickets that are outstanding, like testing completed with found issues in the first cycle.

This email will serve as notification that the transaction will not be looked at again until a new build is made.

2.5 Subsequent QA cycles

Subsequent cycles of QA will be necessary until all QA tickets are closed and no further problems are discovered.

Ideally, the QA process for a transaction should not progress beyond the second cycle.

2.6 QA cycle(s) complete with known issues

Occasionally, there are situations where an issue discovered during testing cannot be fixed in time for delivery. This may be due to a problem with the ERP system APIs or other unexpected behavior.

If there are open QA tickets for a transaction at the end of the final QA cycle that cannot be addressed before delivery, the QA tester will:

1. Create an email addressed to the assigned consultant and to the QA distribution list. The email subject will be in the following format:

```
QA Complete with Issues - $CustomerName - $TransactionName - $0 Issue(s)
```

2. Paste the link of each QA ticket created in TicketSystem for the transaction into the body of the email using the with the “QA Complete with Issues” email template.

3. Send the email.

This email will serve as notification that the transaction is ready to be delivered with existing issues that will be addressed in the next build.

2.7 QA cycle(s) complete

When no QA tickets remain and all transactions in the delivery have been tested, the QA tester will release the build so the consultant can deliver it to the customer.

2.7.1 Releasing a build

To release a build:

1. Navigate to the customer's folder on SERVER6 using File Explorer and find their Builds folder.

2. Right-click and copy the specific build that passed QA.

3. If there is a shortcut in the Builds folder that leads to the customer's Client Release folder on SERVER6, follow it.

   Otherwise, navigate to the general Client Releases folder, choose the folder for the customer's ERP system, and then select the customer's folder.

   EXAMPLE DIRECTORY

   ```
   \\SERVER6\Release\Client Releases\$ERP\$CustomerName\
   ```

4. Paste the build into the folder.

5. When the build is finished copying over, right-click the folder and select "Rename".

6. Delete the numbers trailing after the build's patch number (v0.0.0) to rename the build.

   EXAMPLE

   v1.2.3.0_201906181320 → v1.2.3

7. Return to the Builds folder where the released build was first retrieved and delete all the builds that were generated by the consultant(s) during development prior to release.

Once this process is complete, the build is ready to be delivered to the customer.

2.7.2 QA update for delivery

After the build is released, the QA tester will:

1. Create a "QA Update" email to the consultant and QA distribution list. The email subject will be in the following format, with the version number being the specific build version that was released:

   ```
   QA Update - $CustomerName - v1.2.3
   ```

2. Using the "QA Update" email template, state in the body of the email that QA is finished and that the project is ready for delivery.

3. List the applicable transactions that are ready for delivery, ready for delivery with issues, or not ready for delivery.

4. List the build version number and a link to the released build on SERVER6 in customer's Client Release folder.

5. Send the email.

This email will serve as a notification to both the consultant and to project management that the build is ready for delivery.