# Testing Approach

## Testing Phases

*Define what types of tests will be conducted on the digital CRVS system. Included below are a list of recommended test phases.*

|  |  |  |
| --- | --- | --- |
| Testing Phase | Description | Responsible Tester |
| Component & Assembly Test | When software components are tested in isolation and together. | Developer  |
| Product Test | When the software is tested to confirm fulfilment of functional requirements.  | Developer (observed by Testing Team) |
| System Integration Test | When the software is tested with other systems to confirm fulfilment of integration requirements. | Testing Team |
| User Acceptance Test | When end-users test defined use-cases and scenarios to prove that the system is fit for purpose. | End-Users |
| Performance Test | When the software is tested for speed and/or effectiveness as defined in non-functional requirements. | Developer (observed by Testing Team) |
| Stress Test | When the system is tested to breaking point to assess characteristics of system failure. | Developer (observed by Testing Team) |

## Test Environments

A number of different test environments are required throughout the testing phase. *Define what test environments you need for each testing activity. A minimum set of suggested environments is included below:*

|  |  |  |
| --- | --- | --- |
| Environment Type | Description | Tests completed in environment  |
| *Development Environment* | *Contains everything needed by a team to build and deploy software, including the processes and programming tools used to create the software product.*  | *Component**Assembly* |
| *Test Environment* | *Environment in which the new system is tested by testers and end-users.*  | *Product**UAT**Integration* |
| *Training Environment* | *Environment in which real end-users can simulate production tasks, testing the software product with dummy data that looks and feels like the real thing.*  |  |
| *Production Environment* | *Real-live environment that hosts the software product that will be used and managed by real end-users with real data.*  | *Performance**Stress* |

## Phase Containment

*Define a clear phase containment policy, including the maximum number and severity of defects that can be carried through to the next testing phase.*

## Acceptance Criteria

*Define acceptance criteria using the target use cases and scenarios. These criteria will act as the final check before signing off the system.*

## Defect Management and Resolution

*Identify where defects will be tracked and managed. At a minimum, defects should be logged with the following details:*

* *Unique number*
* *Title*
* *Defect Description incl. What the user did, what was expected to happen, what did happen.*
* *Tester Name (who identified the bug)*
* *Testing Date (when was the bud identified?*
* *Test Case*
* *Test Data*
* *Severity*
* *Status*
* *Software version of fix*