**APPROVAL SIGNATURES:**

|  |
| --- |
| *The REDCap System Test Plan document has been reviewed and approved. I understand that this document may change over the course of the project. This is the original project team approval for this document.* |
| **Name** | **Signature** | **Title** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| *The REDCap System Test Plan document has been reviewed and approved. I understand that this document may change over the course of the project. This is the original sponsor approval for this document.* |
| **Name** | **Signature** | **Title** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**DOCUMENT HISTORY:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Revision** | **Author** | **Date reviewed/revised** | **Changes** | **Reason for changes** |
|  |  |  |  |  |

1. **Introduction**

This document details the Testing Plan and methodology for the <ProjectName> conducted using the REDCap electronic data capture system hosted by Partners HealthCare.

1. Document purpose

The intent of this document is to document how System Testing of the <ProjectName> REDCap project is conducted by XXXX (name of your study team).

1. Document scope

The scope of this document covers all aspects of the testing process that are necessary for the verification and release of the <ProjectName> REDCap project.

1. Referenced documents
* <ProjectName> Project Specifications
* <add additional as needed>
1. **Overview**

Testing and verification of the database is conducted to test critical functionalities of the <ProjectName> REDCap project as follows:

* Entry of Mock Data: Entering test data for all variables in the project and running exports/reports to confirm that the data is in the expected format.
* User rights/roles: Verifying that all user roles have been properly configured and the access rights function properly.
* General configuration: Ensuring that the system settings accurately mirror the specifications established.
* Data Quality Rule and Other Testing: Any Data Quality Rules, Branching logic, piping, Action Tags, etc. are tested to confirm expected functionality.

The executed test scripts and issue log are compiled, and the results summarized at the end of the test process. All identified issues are categorized, and resolutions are documented.

Note: This is not a substitute for the testing of the REDCap system itself, which has already occurred at Partners HealthCare.

1. Risks

The level of testing is determined by a risk assessment. Testing of the project configuration and data quality rule programming is performed by XXXX.

This Testing Plan, therefore, focuses on verifying that XXXX’s procedures for conducting a project using the REDCap electronic data capture system can be performed as expected, and that user roles and rights are configured properly.

1. Assumptions

It is assumed that the project instructions for implementation have been completed according to the following specifications and documents:

* (eCRF) Specifications (if applicable)
* Data Quality Rule Specification document
* REDCap Project Specifications
1. Limitations

The project used for testing does not contain the volume of data that will be contained during production.

In addition, only a subset of the out-of-the-box functions of REDCap are tested. The subset is defined by the expected outcomes of the Data Quality Rule test scripts.

This document is based on the status of the REDCap system at the time of testing. Any changes to REDCap version X.X.X post-deployment is documented through XXXX’s change control process and regression testing of any changes are documented.

1. **Test Execution Plan**

Testing is performed to verify a critical set of activities required to conduct the <ProjectName> project is behaving as expected.

The test scenarios below are executed to ensure that testing assumptions are made correctly:

* Initial instrument testing to confirm variable/response programming
* Entry of mock data
* Data Quality testing
* <Add additional scenarios and tests as needed>

User role testing and system configuration testing is performed in accordance with the <ProjectName> Project Specifications.

1. Prerequisites

The following prerequisites must be confirmed prior to executing project testing:

* The project must be configured for the <ProjectName> protocol.
* Users are given appropriate access to the development/test project.
* Test users have received instructions for executing testing.
* <Add additional scenarios and tests as needed>
1. Resources

The Study Team are responsible for the various tasks and responsibilities required to conduct the testing.

1. Test Approach

Testing scenarios are devised to test pre-determined functionality separately. Each scenario may consist of one or more test scripts depending on the functions being tested. Test scripts for each of these scenarios are written by a Study Team member in order to correspond to the workflow functions that an end user might employ during the conduct of the project.

1. Test Methods

System testing takes place in stages. The initial round of testing is performed upon completion of instrument development. The tester verifies that all data points are enterable and that question text and predefined responses are provided as defined in the eCRF Specification.

The next round of testing consists of a “dry run” and is used to finalize the formal test data and scripts. This round includes a test of all defined data quality rules along with any other project functionality (randomization, double-data entry, etc).

1. Assessment of Issues

Once all tests are executed, the consolidated issues log, containing the findings, is jointly reviewed by the Study Team.

Any issues that remain after testing are reviewed by the Study Team and decisions are made as to the critical nature of the issue(s) to warrant further action before the project is formally moved into production.

1. **Test Results and Summary Report**

The <ProjectName> testing documentation binders contain the signed, dated, executed test scripts, test data, screenshots, and issues log. Each of these is version-controlled (if applicable) and accurately labeled.