Test Plan Document

(e-Care Clinicals Application)

by:

Aishwarya Kashid (001819145)

Huiwen Gan (001836779)

Hongxi Li (001893090)

(04-18-2019)

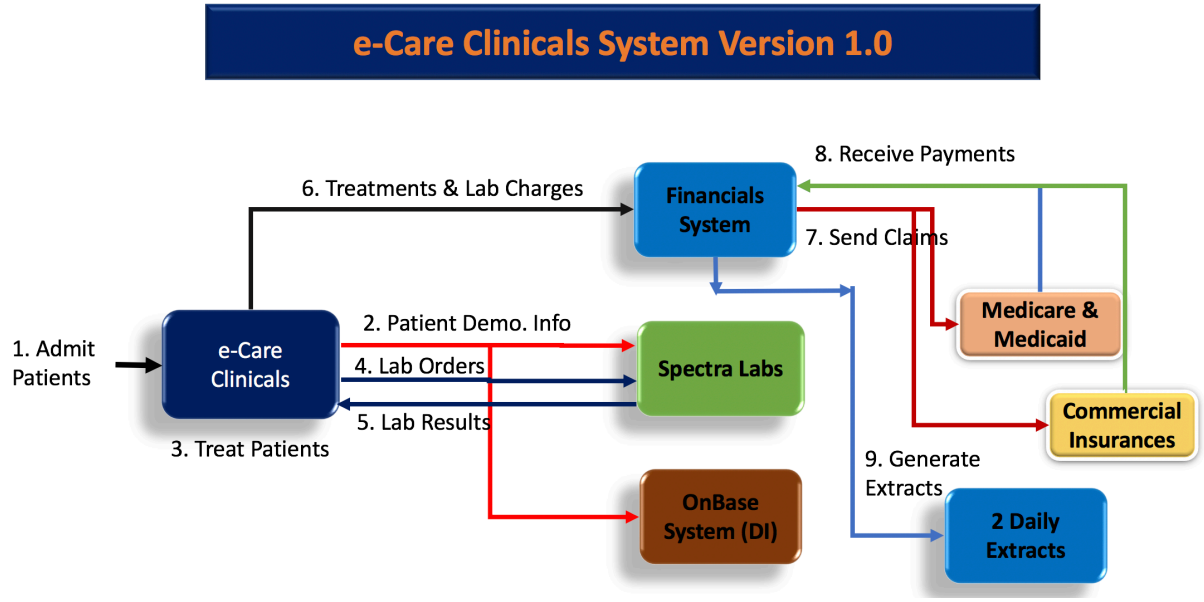
Table Of Contents

1. INTRODUCTION

The Test plan is designed to prescribe the scope, approach, resource, and schedule of all testing activities of the e-Care clinicals application. The plan identify the items to be tested, the features to be tested, the types of testing to be performed, the personnel responsible for testing, the resources and schedule required to complete testing, and the risks associated with plan.

A new e-Care Clinicals system is replacing the legacy Pro-Care application. In the e-Care clinicals system patients will get admitted and treated based on their Financial and Clinical status and their types of insurances, Medicare/Medicaid or Commercial Insurance. All their labs will be sent to the Spectra Labs to ensure that they receive the best care.

The e-Care Clinicals system will send all treatment and lab charges to the Financials system and the Financials System will create claims with the insurance companies and gets paid accordingly. The clinicals systems also produces two reports for the nurses daily.



2. OBJECTIVES AND TASKS

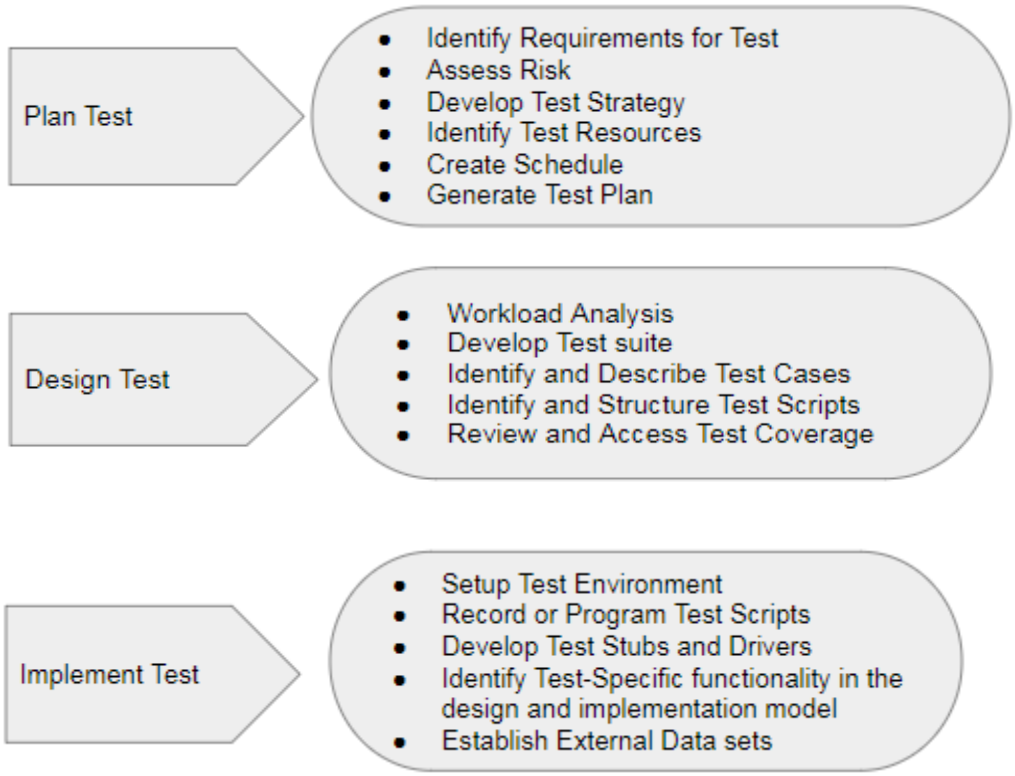
2.1. Objectives

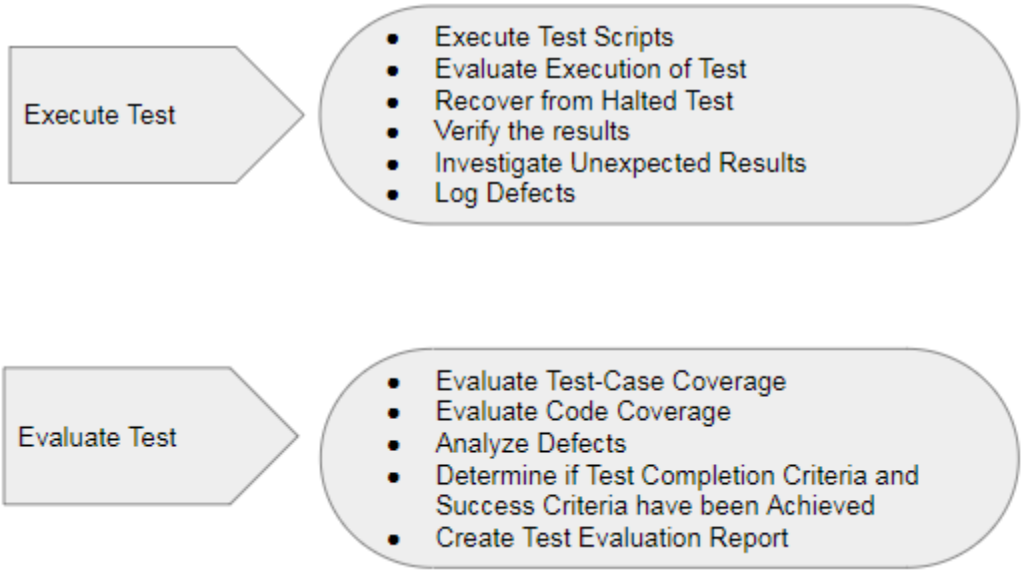
The objective of the test is to verify that the functionality of E-CARE CLINICALS SYSTEM works according to the specifications.
The test will execute and verify the test scripts, identify, fix and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects for future fixing via CR.
The final product of the test is two fold:

- A production-ready software
- A set of stable test scripts that can be reused for Functional and UAT test execution

2.2. Tasks

Here is the tasks assigned to the testing team:





3. SCOPE

In scope of the test plan, the test scenarios or test objectives are mentioned that will be validated in the process. Here is the list of test scenarios which should be validated:

- Validate the financial clearance for new and existing patient
- Verify the demographic information is sent to DI
- Check clinical clearance for treatment
- Check for insurance type
- Lab orders to Spectra Lab and result send back
- Validate the financial claim status
- Check Reports

4. TESTING STRATEGY

The test strategy presents the recommended approach to the testing of the software applications. The previous section on scope described *what* will be tested; this describes *how* it will be tested.

- The main considerations for the test strategy are the techniques to be used and the criterion for knowing when the testing is completed.
- In addition to the considerations provided for each test below, testing should only be executed using known, controlled databases, in secured environments.

- The following test strategy is generic in nature and is meant to apply to the scope listed in Section 3 of this document.

4.1. Alpha Testing (Unit Testing)

Unit testing will be done by developer. He will check the basic functionalities like all the functional requirements are met and the application should be up and running.

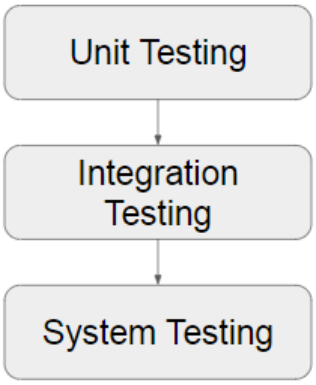
Example: Developer will check if DI of the same customer is present in the user details when added.

4.2. System and Integration Testing

Integration Testing will be done by the tester (Aishwarya) and she will check every part of the application from start. For example, if the financial clearance flag part is done by the developer then she will check all the test cases against it and when next part is developed by the developer then she will check both the parts again as it can have dependency.

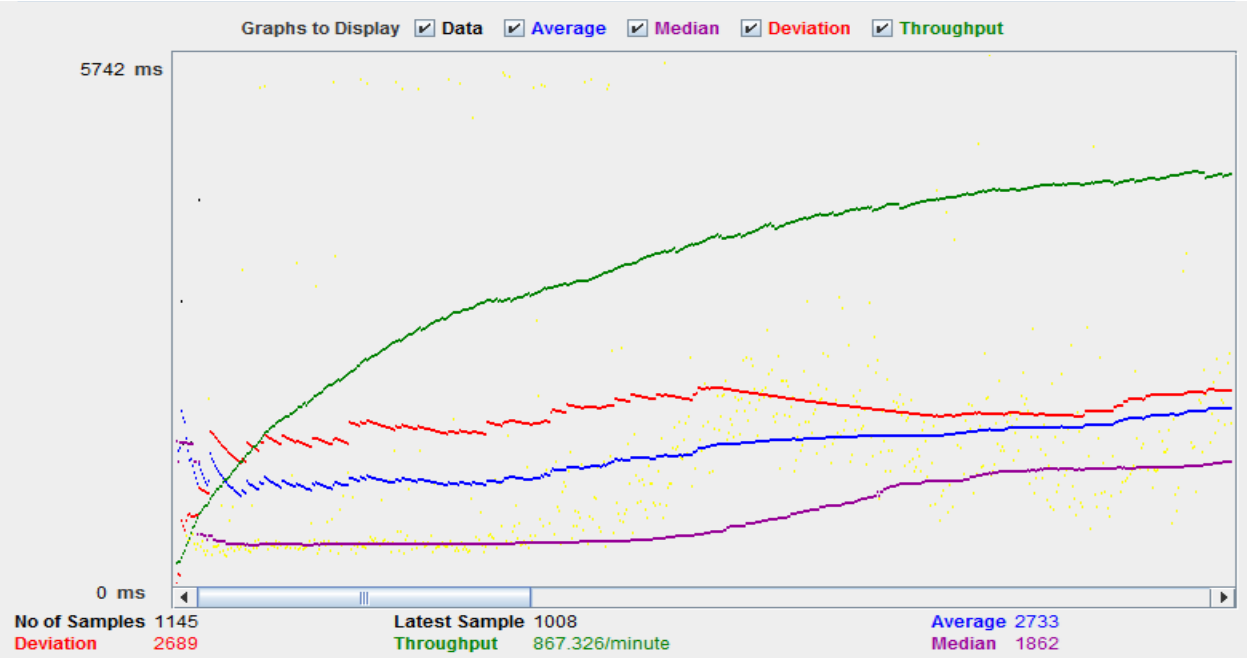
Example: Tester will test if financial flag is cleared then the details should go to Spectra Labs and labs can access the patient data.

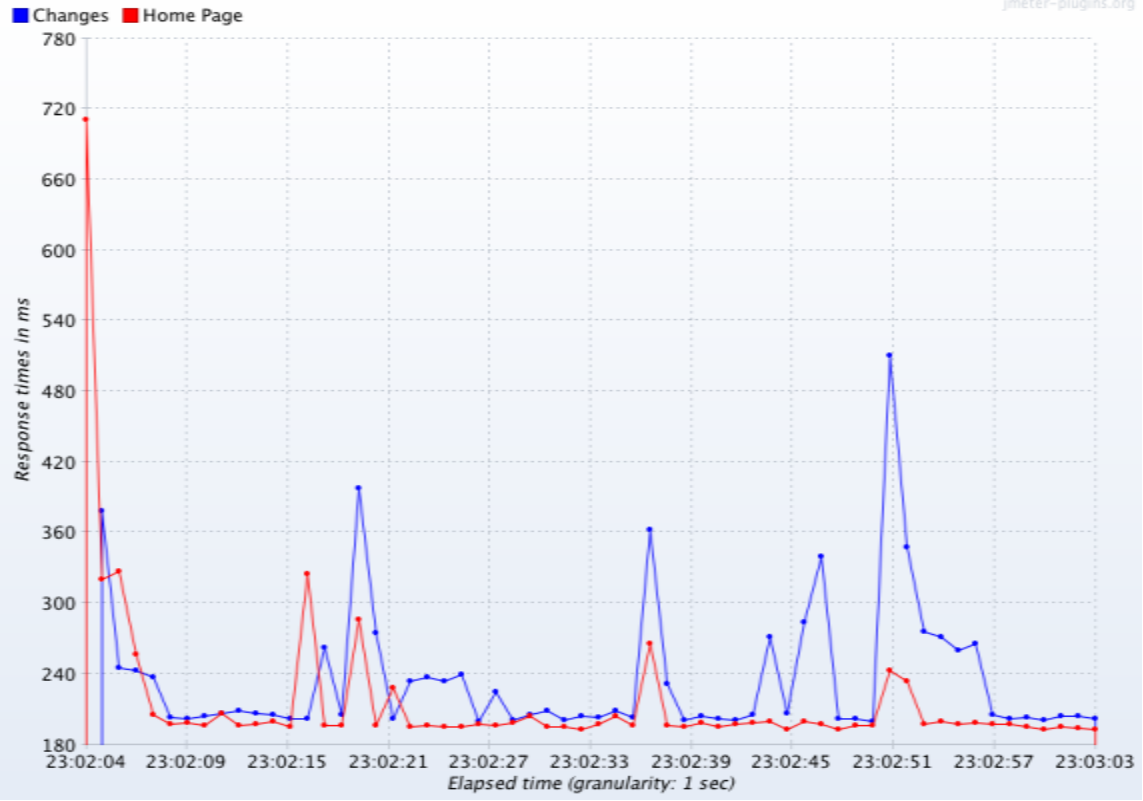
System testing will be done by the tester (Huiwen) and she will check the application meticulously from start to end and will check for each test case.



4.3. Performance and Stress Testing

In the performance testing the tester (Hongxi) will test the application performance using tools like Apache JMeter. Then he will do load and stress testing by bombarding the application with 1000 users using the tool Apache JMeter.





Response Metrics

- Average Response Time
- Peak Response Time
- Error Rate

Volume Measurements

- Concurrent Users
- Requests per Second
- Throughput

4.4. User Acceptance Testing

In this section we will be creating an UAT environment and will be deploying the application in UAT. We will give access to the business user to test the application and to check the key functionalities if they are working or any change is needed in the application. If user accepts the case, the application will be then deployed in the production. User acceptance Testing will be done by Huiwen Gan.
Every team member involved in the UAT process should have a clear understanding of what his/her responsibility is.
Here is the example result of UAT:

Test Case/ Script ID	Test Case/Script Description	Date Tested	Pass/Fail	Comments
TS_eCare_001	Verification of Admitting a new Patient into e-Care with the Financial Clearance set to Yes and the the demographic information is sent to DI and Spectra	04/18/2019	Pass	New Patient admission done successfully

4.5. Batch Testing

- After the QA team receives the approved package (Package contains procedures, JCL, Control Cards, Modules, etc.), the tester will preview and retrieve the contents into PDS as required.
- Convert the production JCL or Development JCL into QA JCL otherwise called JOB SETUP.
- Copying production file and preparing test files.
- For every functionality, there will be a job sequence defined. (As explained in the example in Methodology in Mainframe section). The jobs should be submitted using the SUB command with the test data files.
- Check the intermediate file in order to identify the reasons for missing or error-out data
- Check the final output file, database and the Spool to validate the test results.
- If the job fails, the spool will have the reason for the job failure. Address the error and resubmit the job.

4.6. Automated Regression Testing

- Takes an inventory of the workflow present in the application
- Transforms the workflow into test scripts
- Maintains and version-controls the test scripts as software assets
- Automates the operation of test scripts based on changes to the application
- Uses the test results as inputs to management's business risk analysis

4.7. Beta Testing

After UAT is done and business accepts the application, we will launch the beta version of the application and will give access to the application to the specific users to run the application and track the report of the application if there is any issue, the user will report it and we will solve that bug.

5. TEST SCHEDULE

Milestone	Effort (Person Hour)	Start Date	End Date
Test Planning	25 Person hour	04/15/2019	04/21/2019
Test Design	20 Person hour	04/22/2019	04/30/2019
Test Development	10 Person hour	05/17/2019	06/18/2019
Test Execution	15 Person hour	06/01/2019	07/05/2019
Test Evaluation	10 Person hour	07/06/2019	08/01/2019

6. CONTROL PROCEDURES

Problem Reporting:

As tester test the application, he/she may find the defect in the application. After the defect is detected the incident will be automatically raised and will be assigned to the developer and the incident will be opened till. The application used for the problem reporting is HP ALM.

Change Requests:

Suppose there is a bug in the application and to solve that bug the application needs to be turned off that means there is a downtime in the application and it may affect the business. Since, there will be an effect on business we need to raise the change. Here is the procedure of change requests:

- Developer will raise the change in the incident reporting system.
- Change manager will review the change and collect the dependencies from respective teams

- ➢ After gathering the information, change manager take the change to the CAB meeting where the change will be reviewed by the business and the application SME's.
- ➢ Depending upon the severity, change will be approved
- ➢ Once the change is approved, on the scheduled the day the change will be executed
- ➢ After the execution, application team will check the application and will inform business about it. If there are any issues in the application, the changes will be rolled back immediately.
- ➢ After the change is successfully executed, the change manager will sign off.

Change Manager: Medi Servattalab

7. FEATURES TO BE TESTED

The interfaces between the following subsystems will be tested:

1. Patient Admission
2. Treatments System (Spectra Labs)
3. OnBase System
4. Financial System
5. Report System

The external interfaces to the following devices will be tested:

1. Local PC
2. Remote PCs
3. The connections between e-Care Clinicals and commercial insurance and Medicare insurance must be linked all the time.

The most critical performance measures to test are:

1. Response time for Patient Admission
2. Response time to access the Treatments System.
3. Response time to access the Financial System.
4. Patient response time when system loaded with 20000 existing patients.
5. System response time when 5000 simultaneous accesses to the Financial System.

8. FEATURES NOT TO BE TESTED

- Spectra Labs: The internal working of spectra lab system where the lab orders are generated like if the lab order is assigned to patient will not be tested
- Navigation: The flow of the application is not tested like particular screen should appear after particular click.

9. RESOURCES/ROLES & RESPONSIBILITIES

No.	Member	Tasks
1.	Test Manager: Medi Servattlab	**Manage** the whole project
		Define Project **directions**
		Acquire appropriate resources
2.	Tester (s): Aishwarya Kashid Hongxi Li Huiwen Gan	Identifying and describing appropriate test techniques/tools/automation architecture
		Verify and assess the Test Approach
		Execute the tests, **Log** results, **Report** the defects.
		Tester could be in-sourced or out-sourced members,based on the project budget.
		For the task which required **low** skill, I recommend you choose **outsourced** members to **save** project cost.

3.	Developer in Test	**Implement** the test cases, test program, test suite etc.
4.	Test Administrator	Builds up and ensures Test Environment and assets are managed and maintained
		Support Tester to use the test environment for test execution
5.	SQA members	Take in charge of quality assurance.
		Check to confirm whether the testing process is meeting specified requirements

10. SCHEDULES

Project Tasks	Start Date	End Date
Design & functional requirements review	04/15/2019	04/30/2019
Code Development	04/20/2019	05/15/2019
Test Plan Review	05/15/2019	05/16/2019
Test Cases Review	05/17/2019	05/21/2019
QA Testing	05/25/2019	06/18/2019
Performance Testing	06/01/2019	06/10/2019
UAT	06/25/2019	07/05/2019
Pilot Release	07/06/2019	07/15/2019
Full Deployment	07/16/2019	08/01/2019

11. DEPENDENCIES

A task dependency is a relationship between two tasks in which one task depends on the finish of another task to begin. Dependencies can be created between two or more tasks, tasks and tasks groups or between two or more task groups.

In our case, it the task group would be **Finish to Start (FS)** i.e. The first task must complete before the second task can start. For example, we cannot proceed if the financial flag is not cleared.

Other types of dependencies are:
Finish to Finish (FF), Start to Start (SS) and Start to Finish (SF)

12. RISKS/ASSUMPTIONS

- Risk is future uncertain events with a probability of occurrence and a potential for loss.
- Risk identification and management are the main concerns in every software project.
- Effective analysis of software risks will help to effective planning and assignments of work.
- Each Release Plan will include the following basic assumptions. The scope and content of a Release may dictate additional assumptions.

Schedule Risks

- Wrong time estimation.
- Resources are not tracked properly. All resources like staff, systems, skills of individuals etc.
- Failure to identify complex functionalities and time required to develop those functionalities.
- Unexpected project scope expansions.

Budget Risks

- Wrong budget estimation.
- Cost overruns.

Technical Risks

- ➢ Continuous changing requirements
- ➢ No advanced technology available or the existing technology is in initial stages.
- ➢ The product is complex to implement.
- ➢ Difficult project modules integration.

Assumptions

- ➢ The Test environment described in Section 5 will be available by the start date given in the schedule for executing the test scripts, and sign-off has been forwarded to the Test Manager.
- ➢ The object code will be fully unit and integration tested and made available on the Test environment by the date(s) given in the schedule for executing the test scripts and sign-off has been forwarded to the Test Manager.
- ➢ The application environment and content components will have successfully completed the SDLC Application Environment and Content review process as outlined in the Acceptance Process section of the Project Plan and the appropriate sign-offs have been forwarded to the Test Manager.
- ➢ The various specifications related to the Release will be the basis for testing the functionality of each release of E-care Clinicals. These documents will be available to the Test Team Leader by the start date given in the schedule for preparing test scripts for each release.
- ➢ All approved Decision Requests and Change Requests will be forwarded upon approval to the Test Team Leader to ensure that the test scripts are modified and the function re-tested, where necessary.

13. TOOLS

Automation Tools:

- We will be using UFT for automated testing

Bug Tracking:

- For bug tracking, we will be using HP ALM

Performance Testing Tool:

- Loadrunner will be used

14. APPROVALS

Name	Role	Signature	Date
Medi Servattlab (Test Manager)	Test Manager		04/18/2019
Aishwarya Kashid (QA)	QA		04/18/2019
Hongxi Li (QA)	QA		04/18/2019
Huiwen Gan (QA)	QA		04/18/2019