Appian ui

|  | |
| --- | --- |
| Project Summary | *This document guides the user about the Appian UI Phase work from QA standpoint and all the related documents for the 302 Suite of projects.* |
| QA Director / Manager | *Lisa Kats / Bill Wesel* |
| Author’s | *Jaya Santosh Alekya Pilla* |
| Start Date | *February 10 2018* |
| Anticipated Deployment Date | *April 6 2018* |
| Revision History | *V0.3 – Final draft* |

[1.0 Introduction 3](#_Toc491787294)

[1.1 Purpose 3](#_Toc491787295)

[1.2 What is nextgen 3](#_Toc491787296)

[1.3 What is appian cit 3](#_Toc491787301)

[2.0 agile qa process](#_Toc491787294) 4

[2.1 Manual test Approach](#_Toc491787295) 6

[2.2 Automation Test Approach](#_Toc491787296) 6

[2.3 QA Activities](#_Toc491787296) 8

[3.0 PATHS& URL’s](#_Toc491787294) 9

[4.0 APPENDIX](#_Toc491787328) 10

# [Introduction](#_Introduction)

## [Purpose](#_Purpose)

The purpose of this document is to make the user can understand Appian UI Phase2 work from QA standpoint and all the related documents for the 302 Suite of projects.

## [What is NextGen?](#_What_is_X-ray?)

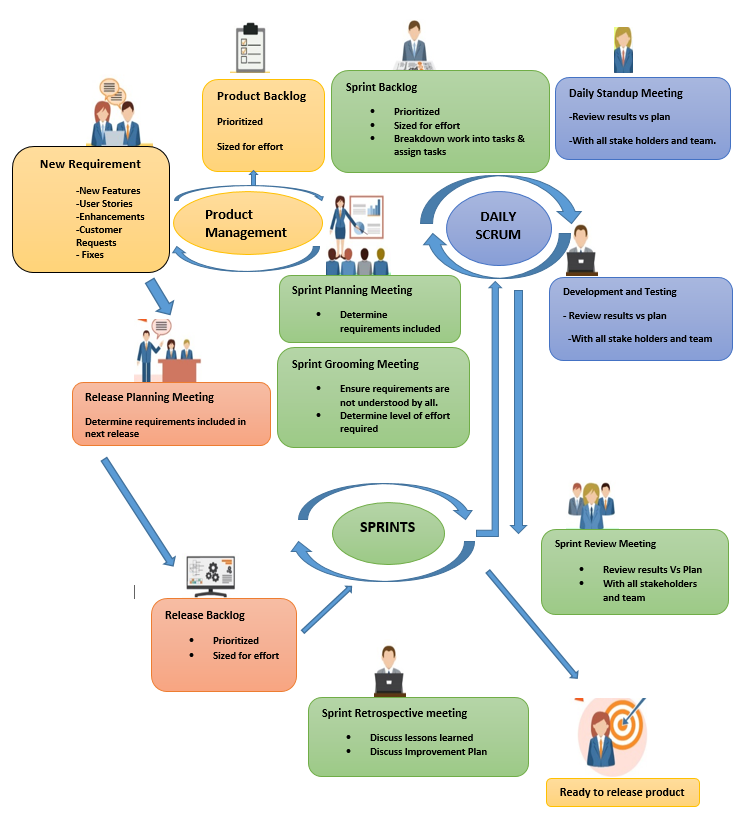
NextGen is a joint business and technology project. It includes the establishment of a common platform onto which we can rapidly build and release products into the market twice as fast as AIM currently do. It introduces a variety of new open source, cloud and analytics technologies. NextGen program is focused on 6 strategic imperatives- Product& Platform Modernization, New Product Innovation, Plug-N-Play architecture, NextGen Data, NextGen Infrastructure, Resource and cultural excellence.

## [What is Appian CIT?](#_Why_X-ray?)

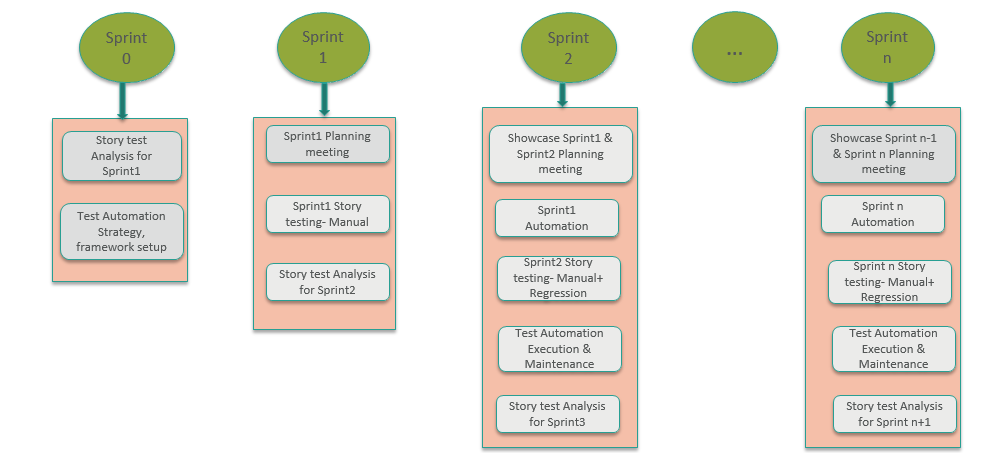
Appian provides the User Interface (UI) for the 302 project through which the user will be able to access/ handle the cases, views, Intake, Process and other functionalities. There are several roles involved- Global Admin, Manager, CIT User, RS, RN, MD.

# 2.0 [Agile](#_Introduction) QA Process

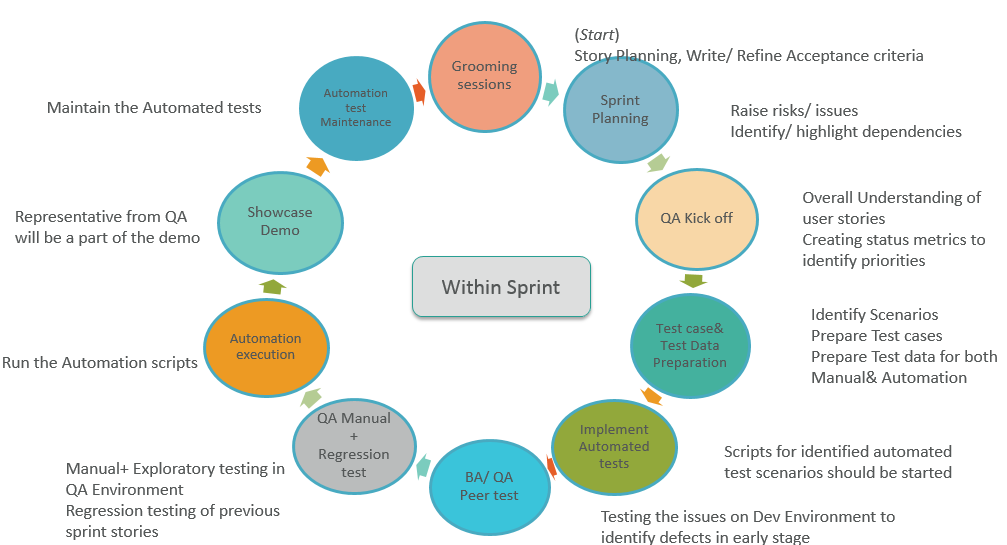
302 Agile Testing Process



Agile Test Process- Sprint 0 to Sprint n



Agile Test Approach for every Sprint



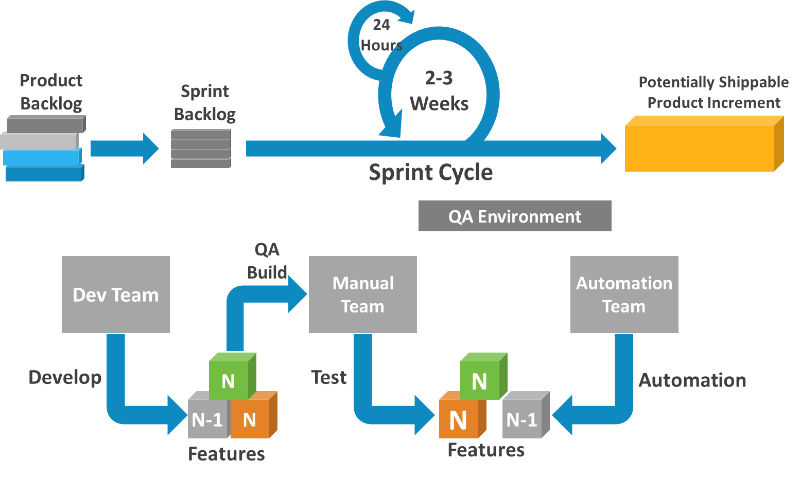
## 2.1 [Manual](#_Purpose) Test Approach

Manual Testing would ensure that the application is error free and it is working in conformance to the specified functional requirements.

* Sprint usually starts on Wednesday and ends on alternate Tuesday
* Sprint duration is 2 weeks
* To start off with a sprint in Agile, all the user stories for the sprint are pulled from Product backlog to Sprint backlog.
* The Dev team starts developing the features for the user stories in the Sprint backlog and passes the build to QA Environment.
* The Manual team from the QA Environment starts testing the features for the user stories in the sprint backlog
* The stories and velocity of Sprint n+1 are usually finalized by Sprint n last day.
* QA starts working on active sprint test cases and finishes it by initial three days of sprint
* Active sprint stories are assigned to QA usually from day2
* QA starts testing the stories based on Acceptance Criteria and test cases.
* Once the testing is completed, QA attaches the captured test results and already written test cases to the respective stories in JIRA
* QA would give approval to the story and pass it on to PO approval for final confirmation
* If QA identifies defects, then QA creates Bugs and attaches screenshots in JIRA, links it to the related story and follows up with Dev
* Once the bug is fixed, QA retests the bug and closes it.

## 2.2 Automation Test Approach

The approach to quality focuses on various automated tests that will be defined and created starting at early phases of the project. The layers of tests across the team will pave the way of ensuring accepted quality.



* The Automation team usually starts writing scripts/ automates the previous sprint user stories test cases and executes them
* The Cycle repeats in the Sprint, where Manual team will always be one sprint ahead and the automation team is always one sprint behind.

Manual team = N features in build (current functionalities)

Automation team = N-1 features in build (starts with regression functionalities)

Automation tools& Framework used

* **Selenium (IntelliJ- Java)**
* Selenium WebDriver would be used to automate the testing of a web application to verify that it works as expected.
* **Cucumber :**
* Cucumber with Selenium WebDriver in Eclipse would be used as Behavior Driven Development framework to create test scripts.
* Cucumber is a collaboration tool, which lets non-technical people write executable specifications using a simple grammar defined by a language called Gherkin.
* Cucumber can be used to “test” code written in Java.
* **Gradle:**
* Consistent Build with other project
* Manage the dependencies

* **Extent Reporting :**
* Excellent execution reports using HTML reporting library for Selenium WebDriver
* Reporting to team in such a manner that they can easily understand the execution result

## 2.3 QA Activities

QA performs several activities in a particular day

* Day starts with a QA scrum call- In this meeting, QA mentions about work done yesterday, work to be done today and any impediments. Only QA’s are present in the meeting
* Appian scrum call - In this meeting, QA mentions about work done yesterday, work to be done today and any impediments. Entire team belonging to that particular domain will be present including PO, TPO, Scrum master, Dev, Ux,
* QA performs functional testing on active sprint and automation on previous sprint
* QA writes all the possible test cases for active sprint, Capture test results
* Before starting the previous sprint automation, QA identifies, what can be automated and what cannot be automated and updates it in test case document
* QA makes sure Regression testing is performed on first day of each sprint
* Once the Automation scripts are completed, QA pushes the code to Gitlab
* End of the day, QA uploads all the required documents to shared drive and updates the daily status.

# [Paths](#_Introduction) and URL’s

|  |  |
| --- | --- |
| Details | Paths |
| Phase 2 Shared drive path | Z:\Automation - QTP\302 Project -QA\Phase 2\Appian |
| Test Cases | Z:\Automation - QTP\302 Project -QA\Phase 2\Appian\Test Cases |
| Test results | Z:\Automation - QTP\302 Project -QA\Phase 2\Appian\Test Results |
| Test Reports | Z:\Automation - QTP\302 Project -QA\Phase 2\Appian\Test reports |
| Automation Status | Z:\Automation - QTP\302 Project -QA\Phase 2\Appian\Automation |
| Automation code | nextgen / cit-ui / phase2-testing / appian-ui-automation Z:\Automation - QTP\302 Project -QA\Phase 2\Appian\Automation\Master |
| Installation documents | Z:\Automation - QTP\302 Project -QA\Knowledge Management\PlayBook\Tool\FrameWork |
| QA Status update | Z:\Automation - QTP\302 Project -QA\Tracker\Daily\_Status\_Report |
| Git Lab Repository | [git@gitlab.aimspecialtyhealth.com:nextgen/cit-ui/phase2-testing/appian-ui-automation.git](mailto:git@gitlab.aimspecialtyhealth.com:nextgen/cit-ui/phase2-testing/appian-ui-automation.git) |
| Jira Board Name | NCP Appian Phase2  <https://jira.aimspecialtyhealth.com/secure/RapidBoard.jspa?rapidView=76&view=detail&selectedIssue> |

# 4.0 Appendix