**SANA MODULES**

**Project Document**

Prepared for

SER 515—Software Enterprise

Instructor: Dr. Kevin Gary

Fall 2015

**Team Members:**

Arpit Jaiswal

Devanshi Panu

Pankhi Prasher

Parneet Kaur

Sweta Singhal

**Table of Contents**

Revision History iii

1. Introduction 1

1.1 Overview 1

1.2 Scope 1

1.3 Purpose 1

2. Software Lifecycle Process 1

2.1 Agile Scrum Methodology 1

2.2 Why Agile Scrum? 1

2.3 Tools 2

2.3.1 Taiga 2

2.3.2 Github 2

2.4 Definitions 2

2.5 Progress of Scrum Board 3

2.5.1 Sprint 1 3

2.5.2 Sprint 2 4

2.5.3 Sprint 3 5

2.5.4 Sprint 4 5

2.5.5 Sprint 5 6

3. Feasibility Study 7

4. Future Scope 7

5. Issues and Risk 7

6. Glossary 8

# 

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 20-11-15 | Version 1 | Sweta Singhal | Initial Draft |
| 21-11-15 | Version 2 | Pankhi Prasher | Updated Scrum Progress |
| 21-11-15 | Version 3 | Devanshi Panu | Updated Scrum Progress |
| 21-11-15 | Version 4 | Arpit Jaiswal | Updated Scrum Progress and Feasibility Study |
| 21-11-15 | Version 5 | Parneet Kaur | Updated Scrum Progress |
| 22-11-15 | Final Version | Pankhi Prasher | Final project Document |

# 1. Introduction

## 1.1 Overview

SANA is a cross-disciplinary organization, including clinicians, engineers, policy, public health, and business experts along the entire healthcare value chain. Our approach is to democratize access to quality healthcare through open source technologies, democratize knowledge through the exchange of learning across partners, and to democratize access to global networks of multidisciplinary experts. We believe that geniuses abound in our partner countries, and these geniuses are more likely to develop sustainable and scalable solutions, as they better understand the local problems and environment.

## 1.2 Project Scope

We aim to modify current SANA mobile client so that patients can utilize the current SANA infrastructure on their personal devices. Earlier patients were dependent on nurses to register them and upload their data to OpenMRS. Using the new version of this app, a patient could directly register himself. SANA app is also integrated with PROMIS pain management app for patients suffering from SCD disease. To start with, we have integrated PROMIS Pain management android app with SANA mobile client. With the OpenMRS patient management portal, Doctor would be directly able to send notifications to user on its mobile client.

## 1.3 Purpose

The purpose of this document is to briefly describe the software process followed in the development of this project.

# 2. Software Lifecycle Process

## 2.1 Agile Scrum Methodology

This project was carried out using agile methodology which helped in responding to unpredictability through incremental, iterative work cadences, known as sprints.

## 2.2 Why Agile Scrum?

The SCORE sponsors for the SANA modules were not available initially. The project was aimed to be a feasibility study of the SANA application. Since there was no specific set of requirements, we decided to explore how we can utilize SANA. Absence of frozen requirements was on the reasons we choose to follow agile methodology for our project since agile methodology provides opportunities to assess the direction of a project throughout the development lifecycle.

## 2.3 Tools

Primarily two tools were used to follow agile methodology.

## 2.3.1 Taiga

It is a free, open source project management platform for agile developers. It is like an online scrum board in the agile process.

## 2.3.2 Github

GitHub is a Web-based Git repository hosting service. It offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

## 2.4 Definitions

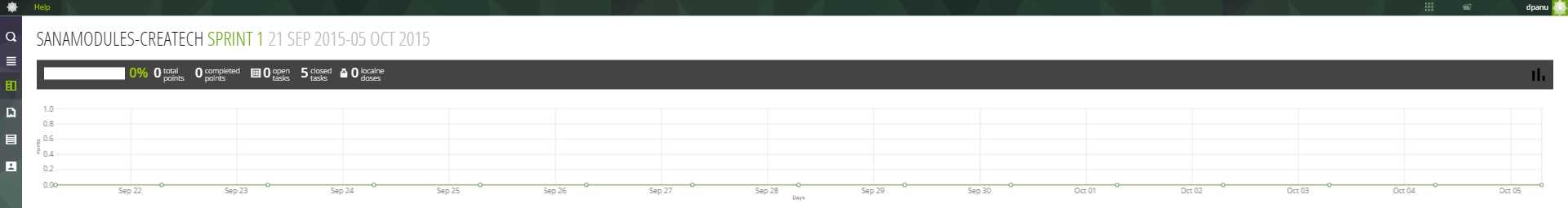
* **Concept** Functional unit of meaning, or, a term which provides context. The concept is the basic element of flexibility in OpenMRS. Concepts are the individual data points collected from a population of patients. Concepts include both questions and answers.
  1. We need the following concepts:
     1. SCD PAIN AGGREGATE SCORE (about the disease)
     2. DOCTOR COMMENTS (Doctor’s comments after looking at SCD pain score of patient)
     3. DOCTOR INFO REQUEST ( Doctor needs more information from patient)
     4. RETAKE SCD SURVEY ( Flag to indicate if doctor wants patient to retake survey)
* **Device** The entity that is used to collect data – e.g. a mobile phone.
  1. subjects mobile phone from where encounter is coming
  2. can be created when user registers
* **Encounter** A collection of one or more observations obtained as the result of an Observer executing a Procedure about a subject
  1. Every time a patient takes a survey an encounter will be created.
  2. Every time a patient follows up survey an encounter will be created.
* **Location** A label for where something of significance occurred.
  1. Required when patient is created in OpenMRS
  2. sample locations can be stored e.g. USA to reduce complexity
* **Observation** The smallest discrete unit of collected data. The collected data may be represented by a character sequence or more complex, file object.
  1. Will contain the survey result information.
* **Observer** Who or what is collecting data-e.g. a community health worker.
  1. In case of SCD every patient will be an observer.
  2. Doctor will also be an observer.
  3. Observer authentication is required to create any type of data in MDS database.
* **Procedure** The executable set of instructions for collecting data.
  1. In case of SCD we need the following procedures:
     1. SCD\_SURVEY => this procedure will store the data of a patient’s survey. We will get information for this from Pain Report App.
     2. SCD\_SURVEY\_FOLLOW\_UP => user can view his notification from doctor and take appropriate action as suggested by the doctor
* **Relationship** Declares a relationship between two concept instances
* **Relationship Category** The nature of the relationship between two concept instances.
* **Person** May be a user or patient. Identified by uuid.
* **Patient** Person whose medical data is entered into OpenMRS as observations in an encounter. Every person has an identifier which in our case is “SCD<SCD Pin>”.

## 2.5 Progress of Scrum Board

## 2.5.1 Sprint 1

* Sprint 1 was used to study the feasibility of SANA with two applications
  + Aspira: A home asthma monitoring platform for monitoring children with asthma.
  + PROMIS for Pain Management: This mobile application implements a survey for collecting pain scores and pain burden reporting for patients with Sickle Cell Disease (SCD).
* Accordingly we created developer stories to research SANA infrastructure and its integration possibilities with other apps.
* After several discussions within the team and with Dr.Gary we finally came to the conclusion that the SCD PROMIS app is best suited for integration with the SANA app.

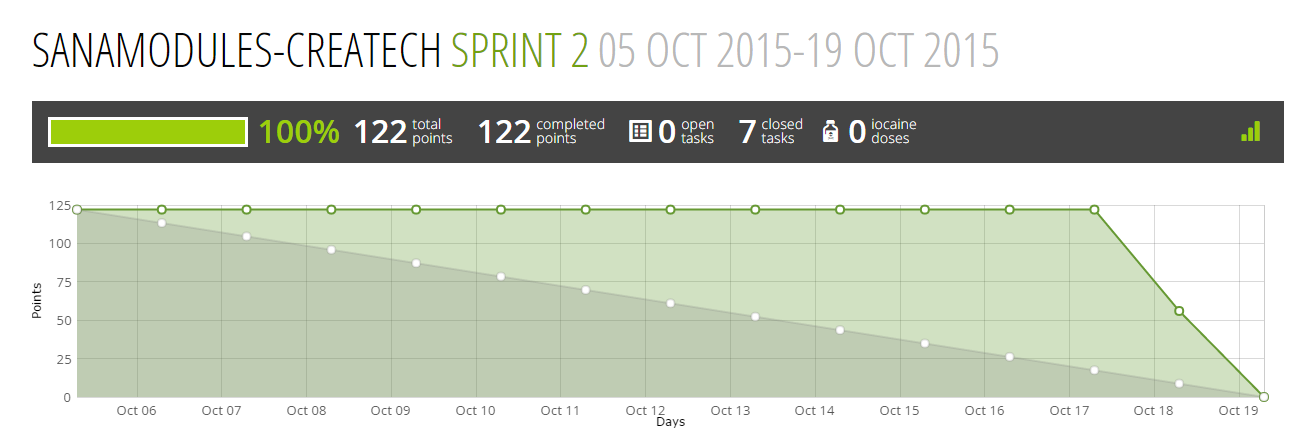
**Burndown Chart**



## 2.5.2 Sprint 2

* Sprint 2 primarily focused on building the framework in mobile client for a patient. Earlier only clinician/ nurse could login into SANA and send patient information. Now the patient himself can register.
* The functionality of navigating from SANA app to PROMIS app to take the SCD survey was also successfully implemented.
* Meanwhile, another part of the team was working on the backend. An Ubuntu machine was set up and MDS was installed, configured and deployed on the local server.
* Many different version of OpenMRS are available, but all aren’t compatible with SANA. A suitable version of OpenMRS was also installed, configured and deployed in this sprint.
* A sample module named SCD was created for OpenMRS and then deployed along with SANA module to understand how OpenMRS functionalities could be utilized and how was SANA encounter queue module is being used currently.

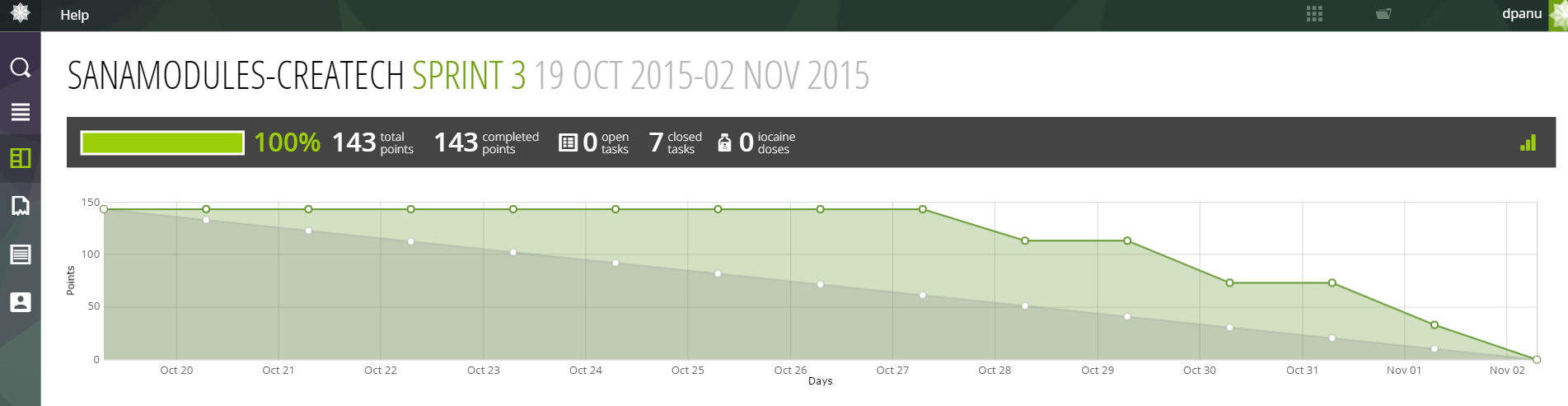
**Burndown Chart**



## 2.5.3 Sprint 3

* The functionality of editing user profile was implemented in this sprint which allowed the patient to edit his details after he registered.
* Interface for a History tab where patient past notification could be viewed was created.
* Another functionality to send an email notification to the doctor was implemented.
* We started working on creating a connection between MDS and OpenMRS and held meetings with SANA developer regarding the things we faced issues with.
* We learned that in order to access the SCD module deployed in OpenMRS, we would require another middle layer. So instead we created procedures (SCD Survey and follow up) in SANA mobile app which could be uploaded with the help of MDS to the Encounter queue module in OpenMRS.
* We created SCD related concepts, patients and encounters in OpenMRS in this sprint.
* Other tasks like validating and maintaining user session and creating databases connection classes were carried out in this sprint.
* This sprint we were able to manage our user stories and tasks, as compared to earlier sprints, in a better manner.

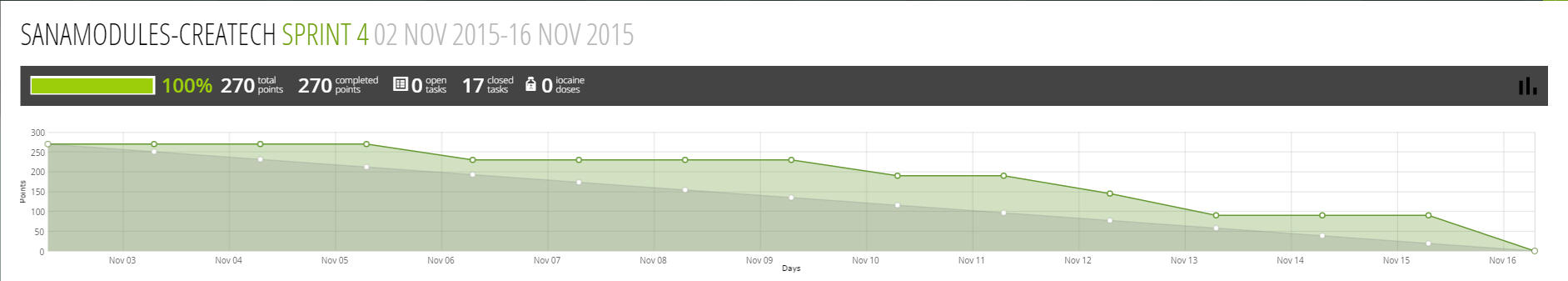
**Burndown Chart**



## 2.5.4 Sprint 4

* After realising that the middle layer of SANA (MDS) contained bugs which prevented us from establishing a connection between MDS and OpenMRS, we tried to debug and find the problem area.
* We decided to work on accessing MDS and OpenMRS through their REST API simultaneously to be prepared to follow both courses.
* After numerous unsuccessful attempts to connect OpenMRS with MDS, we decided to remove the middle layer and access OpenMRS directly. For this we deployed a module in OpenMRS which allowed us to access various resources through its API.
* UI for notification page was also created in this sprint.
* New connection classes were written so that mobile client could access OpenMRS.
* Documentation to assist developers working on mobile client was created which assisted them in mapping OpenMRS fields to the database in mobile client.
* From an agile perspective this sprint, we were able to coordinate tasks and user stories in a much better manner as compared to the previous sprint which is visible from our burn down charts.

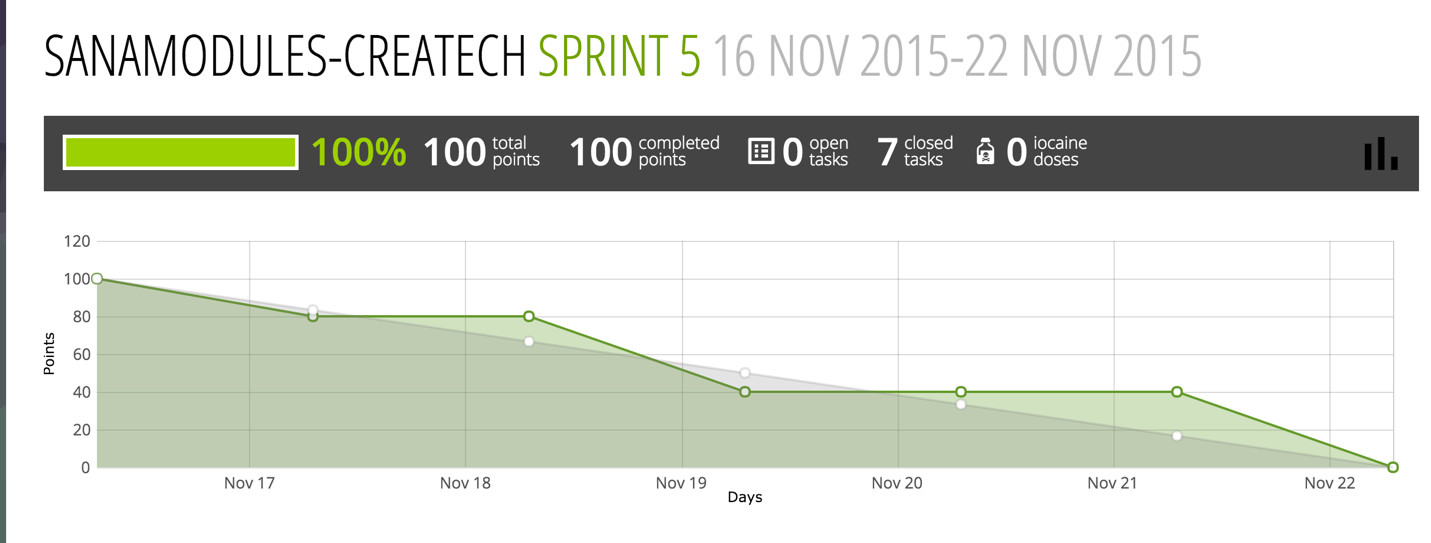
**Burndown Chart**



## 2.5.5 Sprint 5

* OpenMRS was deployed to an EC2 server instance so it can be accessed from anywhere and PROMIS server could push data to it. All the SCD related data was recreated on this new instance of OpenMRS.
* Simultaneously, connection classes, databases were updated in mobile client to accommodate and view data on the history and notification tab in the desired format.
* Integration testing was also carried out in this sprint.
* Documentation was updated for the submission.

**Burndown Chart**

****

# 3. Feasibility Study

Feasibility is the likelihood that the system will be useful to the organization and it is an important outcome of the preliminary investigation. This project was intended to implement a mobile health application in which the patients who will be using the SANA application can fill the PROMIS SCD survey and the survey reports can be sent to Open MRS so that doctor’s appropriate observation can be notified to the patient for the pain aggregate score, this will increase the scope of the survey to a large amount of area and awareness for this disease and its proper medically precautionary measures to be taken.

# 4. Future Scope

There are a few things that could not covered within this limited span of time for the project. Some of them are:

* The patient’s pain aggregate score from the PROMIS SCD application is not being calculated and displayed on the screen as soon as he fills out the survey.
* There are still improvisations possible in SANA, as there is only android platform designed till now, but IOS platform and Web Based Platform for this service is still to be done.
* The patient’s login details are stored only for his/her own device, and this cannot be accessed globally from any other mobile device.
* In this SANA application, we are giving the patient an option to go to the SCD application and fill out the survey, but going ahead if we have two or more available survey options, then we can provide a drop down for the patient to choose which survey to fill out specifically.
* Presently there is a bug in the middle layer MDS, due to which data sent through this cannot be reflected either in Open MRS or mobile client which is hindering the ease of access of necessary details.

# 5. Issues and Risk

There are always some existing issues and flaws within a system that have to be resolved once we start working on all its functionalities. There were a few issues that we faced, like:

* As this is a feasibility study so the biggest issue was that we had no sponsor who assigned us with a specific set of requirements, the team had to come up with requirements from the starting till the end.
* As SANA is an open source, so this application has not been launched in the market yet and this leads to the biggest drawback that is the code had bugs in all three layers, i.e. front end, back end and the middle layer.
* In this project, our main concern is whether SCD survey application can be integrated with the SANA application, which was the main motive for our project.
* The main compliance issue which is prevailing for the survey is the biggest risk because according to those survey details only the whole long process from notification being generated and till it reaches patient will not be authentic then.

# 6. Glossary

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Term (word or phrase)** | **Full form or Explanation** |
| 1. | MDS | Mobile Dispatcher Server |
| 2. | SCD | Sickle Cell Disease |
| 3. | PROMIS | Patient Reported Outcomes Measurement Information System |
| 4. | MRS | Medical Record System |
|  |  |  |