Tissue Analytics Systems Development Life Cycle (SDLC) Document

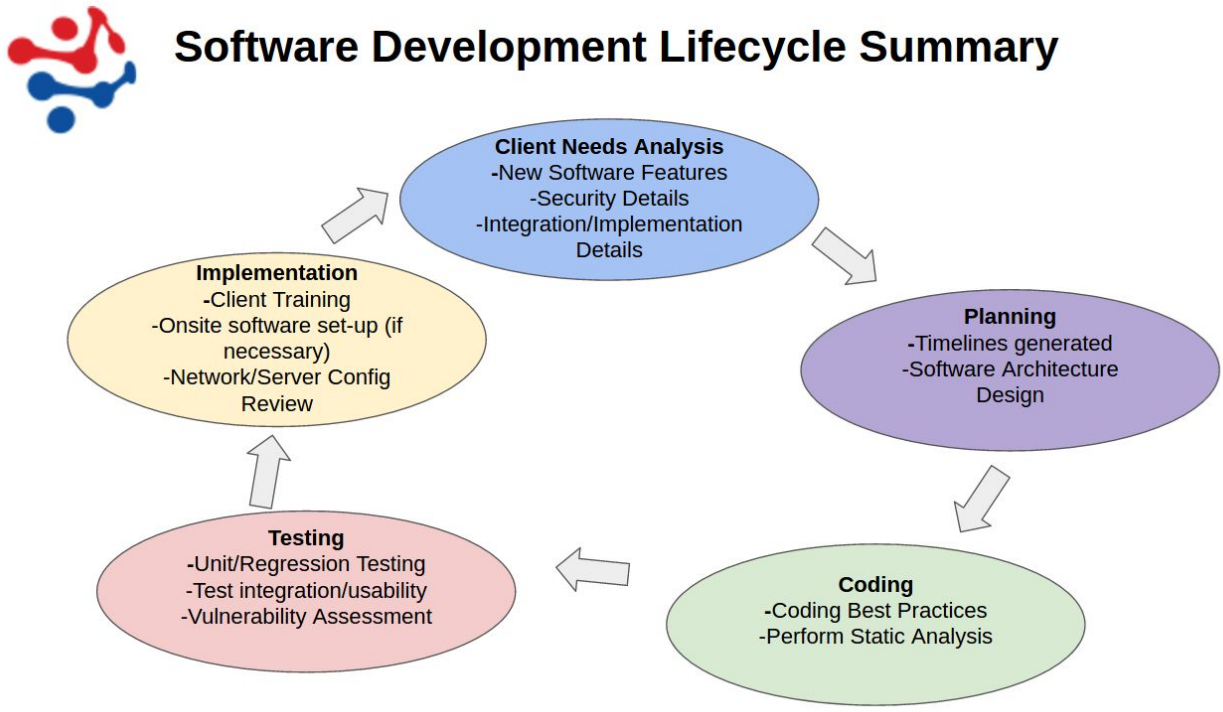
Author: Joshua Budman, CTO
Date Published: 5/31/17
Version: 1.0

Purpose

Tissue Analytics has developed the following document in order to detail the company's systems development life cycle (SDLC). This document discusses the processes that take place at each phase of TA's software development processes. These phases are:

1. Client Needs Analysis
2. Planning
3. Coding
4. Testing
5. Implementation

The processes involved with each phase will be explained in detail in this document. Figure 1 below summarizes the process.



Client Needs Analysis

This phase takes place after a contract has been finalized with a client and they have allowed the software development activities to commence. The first step of the client needs analysis is to schedule a "go-live call" with the client site. TA ensures that there are members from both the clinical and information technology side for the client. During this call, TA will determine the needs for the specific client project. There are two scenarios that may take place at this stage:

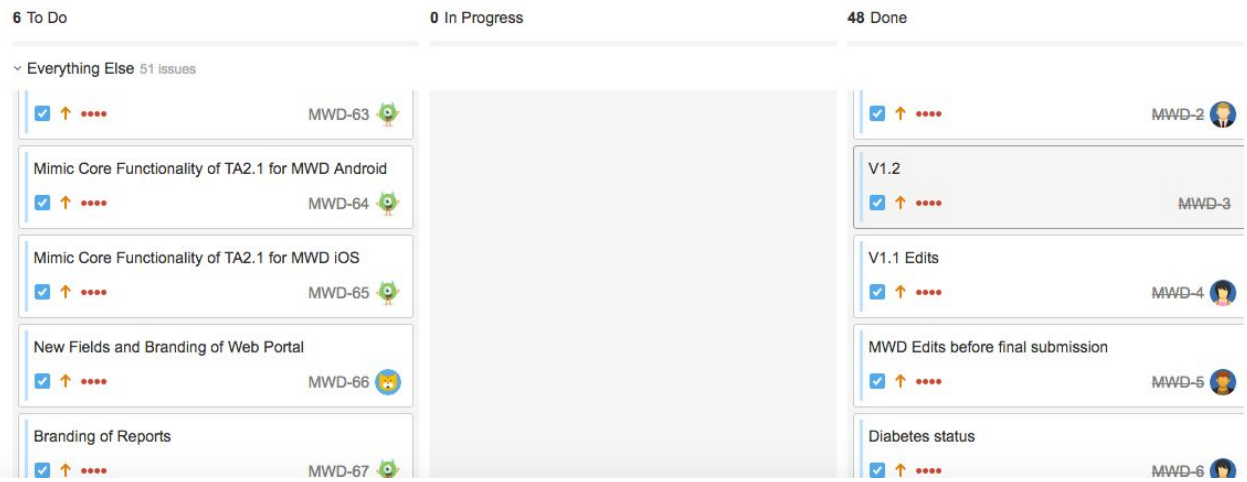
Scenario 1: Tissue Analytics' current product satisfies all of the customer's existing needs. In this case, TA's software development team can skip straight to the implementation phase.

Scenario 2: Additional customization is required for the client to go live. This requires implementation of the full SDLC process.

Assuming that scenario two is applicable, TA will use the client go-live call to put together an initial list of requirements and specifications in order to meet the client's needs. TA will schedule additional calls or meetings, if necessary, to extract all of the requirements necessary for a smooth implementation.

Planning

After requirements are extracted from the client, TA will schedule one or more internal meetings to establish a detailed software architecture intended to serve as an implementation guide for the software development team. This software architecture must be approved by TA management. Finally, TA project management resources will formalize the planned architecture into a series of granular tasks using Jira, which is a state of the art project management software. Each task has a deadline associated it. This is in keeping with TA's use of the **agile development methodology.** See figure 2 below for a screenshot of TA's use of Jira for project management.



Coding

TA uses best practices in software development to develop products and/or new features. TA software developers adhere to the tasks assigned to them via Jira as specified in the previous section. TA uses the following languages/frameworks in order to develop its software:

- Java 8: Android and back-end service development
- Swift 3.0: iOS development

- Python: Imaging algorithm implementation
- Knockout.js: web development
- postgreSQL 10: database development

Use of any core technologies apart from the above must be approved by TA management prior to use.

For information on TA's core system architecture, please see **TA System Architecture Diagram** document. For information on the secure coding techniques TA uses to ensure data is encrypted at all times, please see the **Data Encryption Policy.**

After all coding is complete, TA performs a static code scan using state of the art code scanning technology to ensure that no vulnerabilities exist as a result of the new feature development.

Testing

TA has a Quality Assurance (QA) team that is responsible for performing testing tasks after coding has been completed. The QA team performs unit testing of all aspects of the newly developed software and logs a report, using the template specified in the **Patch Management and Testing Document**, on any issues found during the testing phase. Any bugs or issue are logged in Jira and addressed iteratively until management has approved the current status of the new software development.

In addition to this, software components are constantly tested even if they are not in the midst of core development. For more information on testing protocols, please see the **Patch Management Policy.** For more information on the periodic software component audits TA performs, please see the **QA Team Reporting Document.**

Implementation

The final stage of the process is implementing the software so that the client can use it on live patients. This involves installing the software on client devices and workstations to facilitate its use. This also involves implementing the necessary system integrations, which will have already been developed and tested in the previous phases. TA has a series of integrations, including integrations with Active Directory (AD) software and Electronic Medical Record (EMR) vendor software.

Once the integrations have been completed, the TA Onboarding Team (OT) is engaged to train the client to use the software optimally. The OT will either go onsite or perform remote training based on the client's needs. Once training is complete, the OT works with the software developers to monitor the client's progress and track any issues that arise. At this stage, any new issues that arise or features requested reinitiate the SDLC process.

Revision History

Date of Change	Version Number	Responsible	Summary of Change