law.com

Practice Manager Applications

# Software Development Life Cycle

Created on February 15, 2001
by Tom Foley

Revision No. **3**

Last Saved on **February 15, 2001**
By Tom Foley

The electronic form of this focument can be found in
c:\tom.stuff\pmti\uds\ldc.sdlc.doc

# Practice Manager Applications • Software Development Life Cycle

Created on February 15, 2001
By Tom Foley.

Revision No. **3**, Last Saved on **February 15, 2001**
By Tom Foley

For additional information, contact:

**Law.com**

1781 Mars Hill Road
Watkinsville, GA 30677

Tel: 706.310.1515
Fax: 706.310.1323

# Sign-off

| | |
|---|---|
| Name/Title | **Dave Schroeder**<br>**Chief Technology Officer** |
| Signature/Date | |
| | ❑ Accepted in Current State<br>❑ Accepted with Noted Revisions<br>❑ Rejected<br>❑ Other (please indicate) |
| Comments | |

| | |
|---|---|
| Name/Title | **Fredrick W. Huszagh, II**<br>**Director, Practice Manager Application Development** |
| Signature/Date | |
| | ❑ Accepted in Current State<br>❑ Accepted with Noted Revisions<br>❑ Rejected<br>❑ Other (please indicate) |
| Comments | |

# Change History

| Version No. | Date | Name | Description of Changes |
|---|---|---|---|
| | | | |
| | | | |
| | | | |

# Contents

**1. Release Definitions ........ 1**

1.1. Feature Release (FR) ........ 1

1.2. Enhancement Release (ER) ........ 1

1.3. Bug Fix Release (BFR) ........ 1

1.4. Emergency Bug Fix (EBF) ........ 1

**2. Release Planning ........ 1**

**3. Support Document Definitions – Knowledge Transfer ........ 2**

3.1. Document Set ........ 2

3.1.1. Project Plan ........ 2

3.1.2. Functional Requirements Specification (FRS) ........ 3

3.1.3. Functional Design Specification (FDS) ........ 5

3.1.4. Technical Design Specifications ........ 5

3.1.5. Marketing and Sales Collateral ........ 5

3.1.6. Product Manuals and Help ........ 6

3.1.7. Turnover Documents ........ 6

3.1.8. Product Test Plans ........ 6

3.1.9. Product Training Materials ........ 6

3.1.10. Product Implementation Guide ........ 6

3.2. Dependencies ........ 7

3.2.1. Functional Requirements Specification (FRS) ........ 7

**4. The Software Development Life Cycle -- Composite ........ 7**

**5. Software Development Flow ........ 9**

**6. SDLC Responsible Entities ........ 10**

**7. SDLC Storyboard – Feature Release ........ 11**

**8. SDLC Storyboard – Enhancement Release ........ 12**

# Practice Manager Applications • Software Development Life Cycle

## 1. Release Definitions

The scope of the Software Development Life Cycle (SDLC) is determined by the type of release. The following describes those releases:

### 1.1. Feature Release (FR)

A feature release is a new version of the software that includes new features, enhancements to existing features, and bug fixes. This type of release is traditionally the longest in every aspect, including specification development which requires functional requirements and design, extensive coding in both the user interface and data model, full regression testing of all modules and components, and staged implementation, typically with training.

### 1.2. Enhancement Release (ER)

An enhancement release is a new version of the software that includes enhancements to existing features and bug fixes. No new features are added in an enhancement release. The intent is to minimize the impact on the database, and avoiding problems with data conversions and the implementation of new data structure, both of which traditionally expose an application to potential problems. This type of release is somewhat shorter that a feature release in that specification development is typically limited to just a functional design, coding is limited the user interface with minimal data structure changes, testing does not need to be fully regressive across the entire product and implementation consists of a few scripts and a new executable.

### 1.3. Bug Fix Release (BFR)

An bug fix release only fixes bugs that have been found in either a feature release or enhancement release. Traditionally, these are reported to product management by application support and, because they are not critical fixes, are scheduled for a planned release instead of an emergency release. There is no specification required, as the behavior that is being corrected has already been specificied in the functional design specification associated with the last feature or enhancement release.

### 1.4. Emergency Bug Fix (EBF)

An emergency bug fix is one that resolves an issue reported to application support by a client. The difference between a BFR and an EBF is that the reported bug causes the user to no longer be able to use the application, or their system, in the manner defined by the functional design, such as a system lock-up or data error. These release are not scheduled, and are delivered to the client in the most timely manner possible.

## 2. Release Planning

The success of the the Software Development Life Cycle is rooted in a successful release plan model. If releases are not scheduled throughout the year, then scope within a release has a tendency to “creep” instead of being truly planned. The following describes an optimal release schedule (intervals between releases are approximate)

- Feature Release

- Bug Fix Release = Feature Release Date + 7 weeks
- Enhancement Release = Feature Release Date + 14 weeks
- Bug Fix Release = Enhancement Release Date + 7 weeks
- Feature elease = Enhancement Release Date + 14 weeks

For example:

- 04/03 could be the date of the first Feature Release the year
- 05/21 would be the next Bug Fix Release, and would be based on the code and data structure found in the Feature Release
- 07/29 would be the first Enhancement Release of the year
- 08/27 would be the next Bug Fix Release, and would be based on the code and data structure found in the Enhancement Release
- 10/15 would be the date of the first Feature Release the year
- 12/03 would be the next Bug Fix Release, and would be based on the code and data structure found in the Feature Release

## 3. Support Document Definitions – Knowledge Transfer

The two most important assets a company can have are its people and the knowledge they possess. But without a proper transfer of that knowledge, both assets can be wasted or under-utilized. There are few instances where the impact is greater than in high-tech information service companies. Without the proper transfer of knowledge, the wants and needs of customers cannot be translated into the applications, tools and services – the “solution” – they need to properly support their business and operations. And solutions that are developed for customers cannot be properly documented, tested, delivered and trained unless the transfer of knowledge continues throughout the development and delivery process. And without the proper transfer of knowledge, the marketing and sales teams cannot appropriately sell the solution (i.e., they sell the wrong solution to customers, or don’t sell one at all even though one exists).

In a setting without a proper strategy for the transfer of knowledge, information becomes lost as development and delivery of a solution progresses from conceptualization to realization. The most effective way to insure that information is properly and accurately transferred is to implement a Unified Documentation Strategy (UDS). A UDS employees a methodology whereby information gathered at each step of the development and delivery process is captured and recorded in a structured format, typically a document or set of documents. These documents are then distributed to those persons whose processes are subordinate and superordinate to the current step.

### 3.1. Document Set

In the development and delivery of a solution, the specific set of documents used to support the UDS is as follows:

#### 3.1.1. Project Plan

Using the FRS and FDS, as well as input from the various members of the project team, the project manager develops a project plan. The project plan begins with sign-off of the FDS by the product manager, and concludes with placing the products and services developed within the project into general availability production (GA). This plan should take into account development and testing cycles, as well as alpha, beta and pilot releases. If other projects or teams are impacted by this project, their project or team plans should be incorporated into this project plan in order to identify dependencies.

### 3.1.2. Functional Requirements Specification (FRS)

The capture of information begins with the product manager. Working with various entities, the product manager identifies the customer's business function and defines a solution for meeting their wants and needs. This information is recorded in the Functional Requirements Specification (FRS), and is used as the source materials for the Functional Design Specification (FDS).

The Function Requirements Specifications (FRS) is the first link in the Knowledge Transfer process, and the foundation of the UDS. The goal of the FRS is as follows:

#### 3.1.2.1. Capture and identify the business functions as they relate to the project.

The project may exist to fill the needs of clients and/or an industry in general, to maintain market position with respect to competition, and/or to create a new, previously undefined market.

The sources of this functional description are subject matter experts (SMEs), business analysts (BAs), sales representatives, training and support personnel, and end users.

#### 3.1.2.2. Propose a solution for the project.

Once the project functionality has been fully explored, product management works with senior development personnel to define a proposed solution. This proposed solution explores the workflow and business rules associated with the solution, as well as defines what is in and out of scope for the project.

An analysis of the solution is conducted with respect to other subsystems to determine if there are external impacts and dependencies that could prevent bringing the project to successful conclusion.

Business, financial, sales, marketing, legal and geo-political issues associated with the project are also explored and defined. Senior personnel representing these areas are asked to review any issues they may have with respect to the project, and identify whether a resolution is available that will permit bringing the project to successful conclusion.

#### 3.1.2.3. Establish a development process for the project.

Once a solution has been proposed, a process must be established for developing and delivering that solution. Product management works with senior development personnel to define the development process.

Values for the project are established with respect to time, cost and quality.

Potential areas of sacrifice are identified with respect to functionality and usability which may be taken advantage of in order to maintain the project values.

3.1.2.4. Identify dependencies and requirements outside of product management or development to allow for successful completion of the project.

Projects typically involve more than just the development team. Other teams involved in the overall delivery and fulfillment process need to be aware of what projects will be coming their way so they can put together their own project plans. This includes operations, implementation, support and training.

The ultimate goal of the FRS is to establish the scope and expectations for the project. This means you want to quickly capture the requirements, propose a solution, and achieve sign-off. There are several "gotchas" that can cause this process to grind to a halt:

3.1.2.5. Don't Combined the FRS and the FDS

There is often a temptation to combine the FRS and FDS into one document, as the target audience is the same and both require a sign-off from the same people. There are three reasons why combining these two documents is not a good idea:

3.1.2.5.1. Because the FDS includes the data/system models, UI design, and use cases, it takes a much longer time to generate than the FRS. This delays the first step in the process – achieving sign-off on scope.

3.1.2.5.2. If you create the FDS without first establishing an approved FRS, you may actually end up designing models, creating UI mockups, and writing use cases for items that are out-of-scope or deferred to another project. This is a waste of valuable time and resources.

3.1.2.5.3. Because the FDS contains so much detail in the models, mockups and use cases, the persons conducting a scope review on the requirements portion may get lost in or distracted by the detail. The FDS also tends to be a much larger document than the FRS, and this can intimidate people or, at the least, make them "dread" reviewing the document.

3.1.2.6. Don't Create a Multi-phase Development Project

If the solution is to be developed in multiple phases, create a separate project for each phase. In the scope section, put the projected phase number in the "Out" column, and make a note that the particular scope item is deferred to another phase.

By not combining phases, you simplify the process of final acceptance of the project. Final acceptance is the process of comparing the delivered solution against the FRS. If the FRS includes elements that will be in future phases, then bringing closure to the project can be difficult.

You also do not want to be creating the subordinate documents that originate from the FRS if you do not need to. This will help shorten the development cycles at most stages.

If there are strong dependencies between phases, you may need to develop the FRS and FDS for each phase in parallel, or at least with a slight stagger.

Keep in mind that multi-phase development and multi-phase implementation are two different things.  If a single solution is to be implemented in multiple phases or stages, then keep development as one project and break out the multiple implementation phases in the project plan.

##### 3.1.2.7. Don't Include Changes to Other Components or Subsystems in this FRS

If the solution impacts other subsystems or components, it is best to create a separate set of specifications and design documents for each subsystem or component, rather than combining the changes into the specifications and design documents for the primary project.  This is because the impacted subsystem or component may also impact other subsystems or components that are not directly related to the proposed solution, and these are only identified in their specifications.

If you do have to create specifications and design documents for multiple components or subsystems, make sure you include the following:

- In the primary project specifications and design documents, reference the specifications and design documents of the impacted components or subsystems.
- In the specifications and design documents of the impacted components or subsystems, reference the specifications and design documents of the primary project.

Keep in mind that it is perfectly acceptable for a single project to be defined by multiple sets of specifications and design documents, though one set should be identified as the superordinate set, with all others are subordinate to that one.

#### 3.1.3. Functional Design Specification (FDS)

Using the FRS, the product manager solicits input from the various members of the project team, including the data specialists, developers, graphic designer, integration specialists, project manager, QA engineers, system engineers and technical writers.  Their input is used to develop workflow and data models, create the UI standards and mockups, and identify the deliverables necessary to bring the project to a successful conclusion.  The impact on other subsystems and teams is also identified.  This information is recorded in the Functional Design Specification (FDS) and appended to the FRS.  The FRS and FDS are used as the source materials for the project plan, product design, and marketing and sales collaterals.

#### 3.1.4. Technical Design Specifications

Using the FRS and FDS, the development team translates the business requirements and solution into a technical design.

#### 3.1.5. Marketing and Sales Collateral

If the product is intended for a general release, the FRS and FDS can be used to create the collaterals that will be used by the marketing and sales teams. Development of these collaterals should not be included in the project plan, but training of the marketing and sales personnel should be a consideration.

### 3.1.6. Product Manuals and Help

Using the FRS, FDS and Product Design, the technical communication team develops the manuals and help system for the product. Prior to the final turnover of the product to QA, the manuals and help are incorporated into the product so they can be included in the final certification.

### 3.1.7. Turnover Documents

These are the forms that accompany delivery of product within the software development life cycle:

#### 3.1.7.1. Engineering to Quality Assurance

This document, which accompanies every delivery from engineering to quality assurance, identifies the content of the turnover, including scripts which need to be run and executables that need to ne installed.

Special instructions related to setup and configuration of the supporting hardware and software are also included.

This document also identifies any new features or enhancements that have been included, as well as any bugs that have been fixed.

#### 3.1.7.2. Quality Assurance to Client Services

This is not a unique document, per se, but is, instead, a chronological accumulation of the engineering turnover documents, which then accompany every delivery from quality assurance to client services.

If needed, QA may elect to provide a summary of the turnovers into a single document.

### 3.1.8. Product Test Plans

Using the FRS, FDS and Product Design, the quality assurance team develops test plans for certifying the product, manuals and help.

### 3.1.9. Product Training Materials

Using the FRS, FDS, Product Manuals and Product Help, the training department develops the materials required for each of the courses they will be conducting in support of the product.

### 3.1.10. Product Implementation Guide

Using the FRS, FDS, Product Manuals and Product Help, the Client Services team develops the Product Implementation Guide. This document includes any information necessary for installing or upgrading the product, as well as any processes for data import and/or conversion.

### 3.2. Dependencies

The dependencies are illustrated by the following:

3.2.1. Functional Requirements Specification (FRS)

3.2.1.1. Functional Design Specification (FDS)

3.2.1.1.1. Project Plan

3.2.1.1.2. Product Design

3.2.1.1.2.1. Product Manuals and Help

3.2.1.1.2.2. Product Training Materials

3.2.1.1.2.3. Product Implementation Guide

3.2.1.1.2.4. Product Test Plans

3.2.1.2. Marketing and Sales Collateral

As you can see from the above list, the success or failure of the UDS begins with the FRS. If this document is not generated, or the generated version is not thorough and accurate, the remaining documents will not be properly and accurately generated.

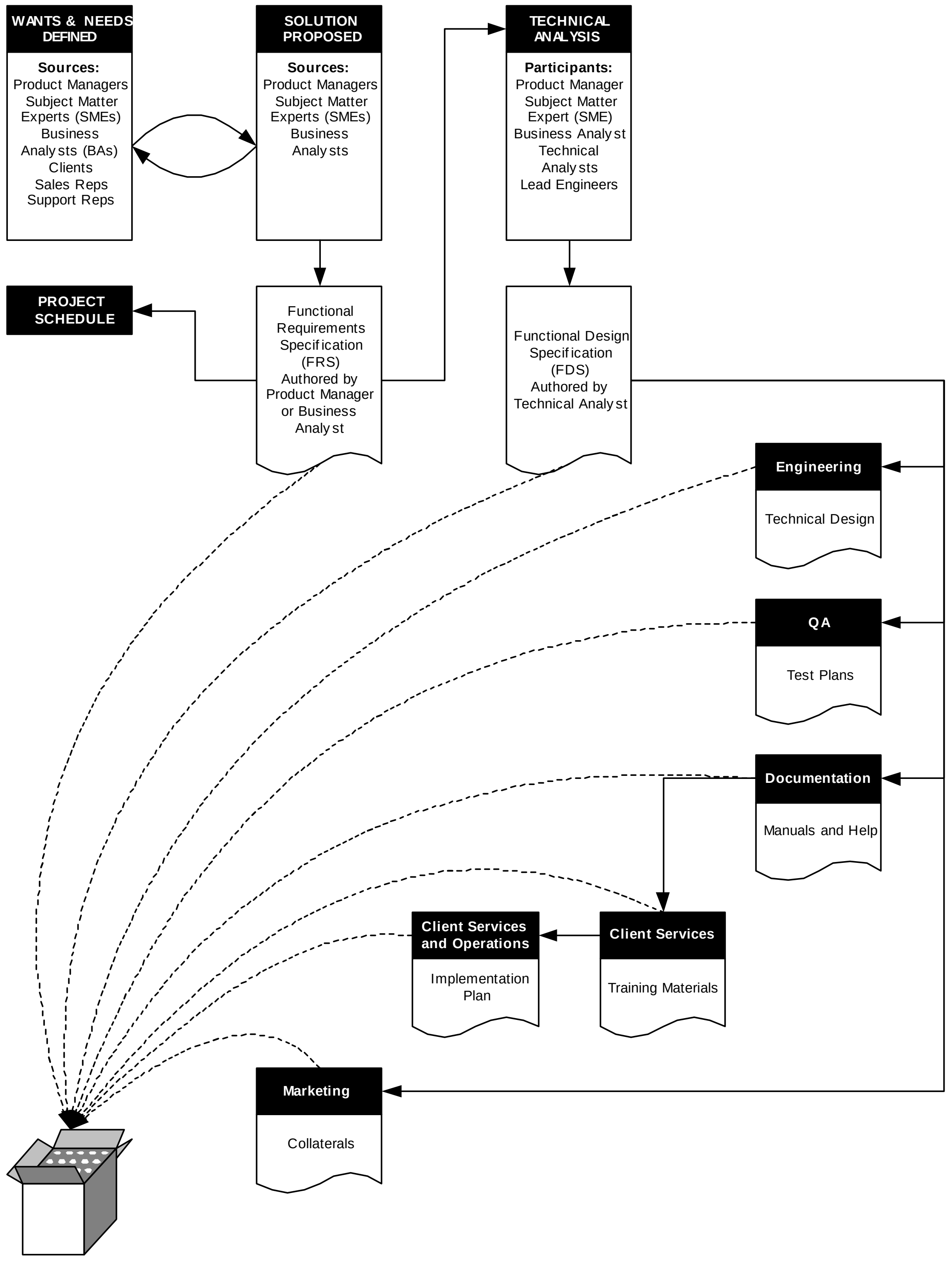
## 4. The Software Development Life Cycle -- Composite

The scope of the Software Development Life Cycle depends on the type of release for which the release is planned.

| SDLC Component | Feature Release | Enhancement Release | Bug Fix Release | Emergency Bug Fix |
|---|---|---|---|---|
| Requirements Gathering for Functional Requirements Specification | 2 weeks<br>Gathered from:<br>App Support, Clients, Client Services, Prod Mgmt, and Sales | N/A<br>Requirements are inherited from the Feature Release | N/A<br>Requirements are inherited from the Feature Release | N/A<br>Requirements are inherited from the Feature Release |
| Authoring, reviewing and approving Functional Requirements Specification | 1 to 2 weeks | N/A | N/A | N/A |
| Authoring, reviewing and approving Functional Design Specification | 3 to 4 weeks<br>Based on approved Functional Requirements Specifications | 1 to 2 weeks<br>Based on approved Functional Requirements Specifications from the last Feature Release | N/A<br>Design is inherited from the last Feature or Enhancement Release | N/A<br>Design is inherited from the last Feature or Enhancement Release |
| Typical Coding Period | 6 weeks | 4 weeks | 2 weeks | Less than 2 days |
| Test Plan Development | 4 weeks<br>Based on Functional | 2 weeks<br>Based on Functional | N/A<br>Testing is inherited from | N/A<br>Testing is inherited from |

| SDLC Component | Feature Release | Enhancement Release | Bug Fix Release | Emergency Bug Fix |
|---|---|---|---|---|
| | Design Specifications | Design Specifications | test plans of last feature or enhancement release, or based on behavior reported in bug | test plans of last feature or enhancement release, or based on behavior reported in bug |
| Alpha Release Testing<br>Internal Distribution Only<br>Full Regression | 3-4 weeks | N/A | N/A | N/A |
| Beta Release Testing<br>Internal Distribution Only<br>Full Regression | 2-3 weeks | 1-2 weeks | 1-2 weeks | N/A |
| Pilot or Pre-release Testing<br>Internal Management System Upgraded<br>Internal Demo System Upgraded<br>Internal Training Systems Upgraded<br>External release limited to Clients partipating in the pilot progam. | 2-3 weeks | 1-2 weeks | Less than 1 week | 1 day, maximumum |
| QA Final Certification for General Availabilty Release to Clients<br>Final upgrade of internal systems. | 1 week | 1 week | Less than 1 week | |

## 5. Software Development Flow



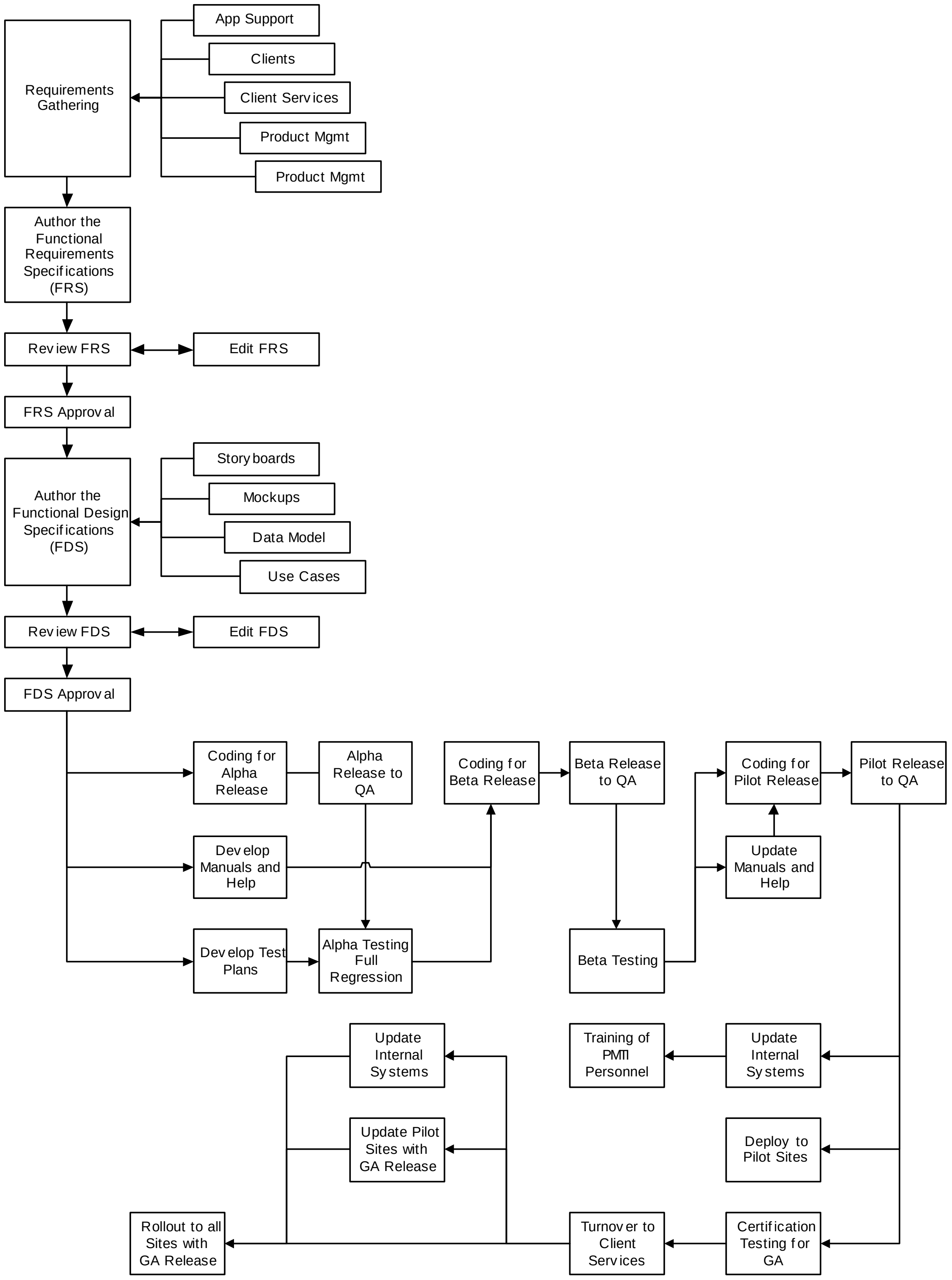
## 6. SDLC Responsible Entities

**UDS Responsibility Matrix**

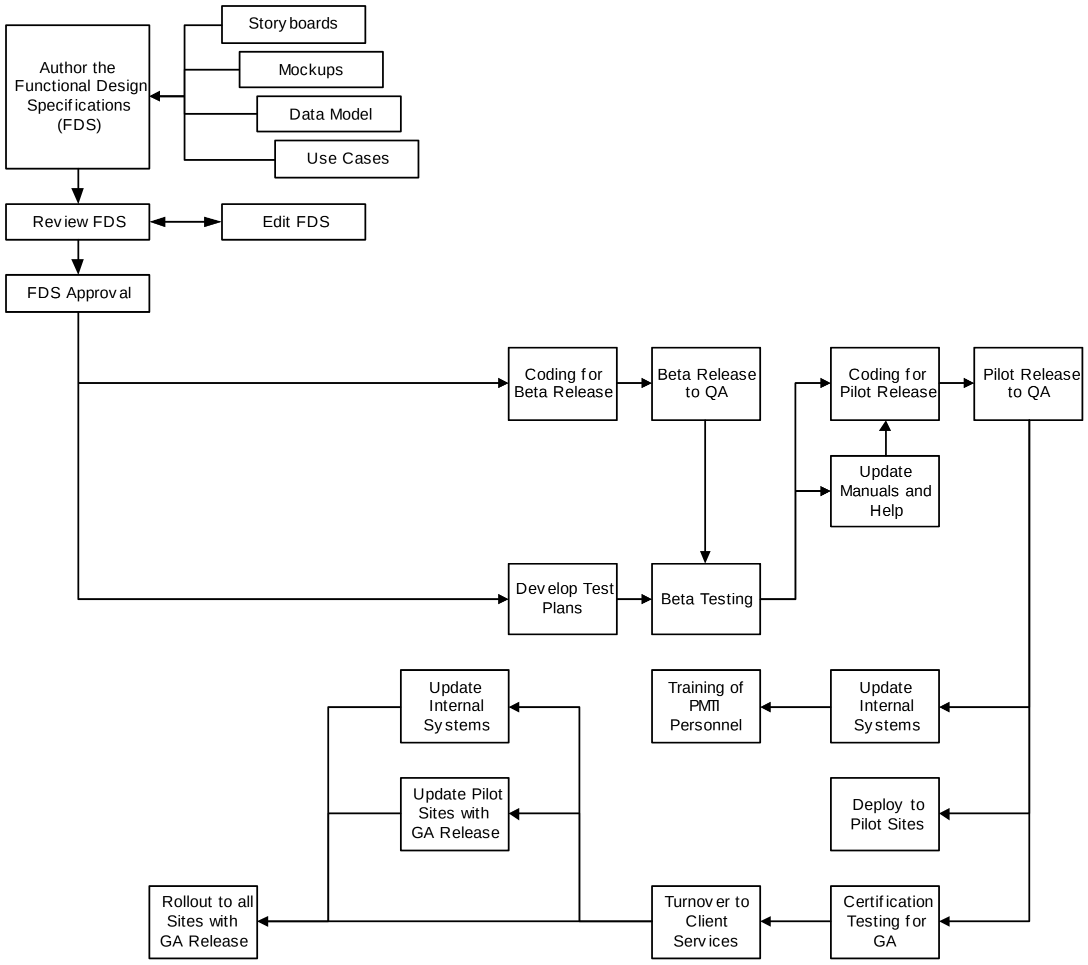
| | | Business Development & Management | | | | | | | Tecnical Development | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Document | Role | Product Manager | Business Analyst | Subject Matter Expert | Client | Sales & Marketing | Client Services | App Support | Technical Analyst | Engineering | QA | Documentation | Operations |
| Functional Requirements Specifications | Owner/Author | | X | | | | | | | | | | |
| | Contributor | X | | X | X | X | X | X | | | | | |
| | Reviewer | X | | X | X* | X | X | X | X | | | | |
| | Approver | X | | X | X* | X | X | X | X | | | | |
| Functional Design Specifications | Owner/Author | | | | | | | | X | | | | |
| | Contributor | X | X | X | X | X | X | X | | X | X | X | X |
| | Reviewer | X | X | X | X* | X | X | X | | X | X | X | X |
| | Approver | X | X | X | X* | X | X | X | | X | X | X | X |
| Technical Design Specifications | Owner/Author | | | | | | | | | X | | | |
| | Contributor | | | | | | | | | | | | |
| | Reviewer | | | | | | | | X | | X | | |
| | Approver | | | | | | | | X | | X | | |
| Marketing and Sales Collateral | Owner/Author | | | | | X | | | | | | | |
| | Contributor | X | X | X | | | | | X | | | | |
| | Reviewer | X | X | X | | | | | X | | | | |
| | Approver | X | X | X | | | | | X | | | | |
| Product Manuals and Help | Owner/Author | | | | | | | | | | | X | |
| | Contributor | X | X | X | | | X | X | X | X | X | | |
| | Reviewer | X | X | X | | | X | X | X | | X | | |
| | Approver | X | X | X | | | X | X | X | | X | | |
| Test Plans | Owner/Author | | | | | | | | | | X | | |
| | Contributor | X | | | | | | | X | X | | X | |
| | Reviewer | X | | | | | | | X | X | | X | |
| | Approver | X | | | | | | | X | X | | X | |
| Training Materials | Owner/Author | | | | | | X | | | | | | |
| | Contributor | X | X | X | | | | X | X | | X | X | |
| | Reviewer | X | X | X | | | | X | X | | X | X | |
| | Approver | X | X | X | | | | X | X | | X | X | |
| Product Implementation Guide | Owner/Author | | | | | | X | | | | | | |
| | Contributor | X | | | | | | X | X | X | | | X |
| | Reviewer | X | | | | | | X | X | X | | | X |
| | Approver | X | | | | | | X | X | X | | | X |

* Client Reps only directly involved if they are one of the stakeholders in the product.
For example, if the project is custom, the client would be directly involved in the approval process.

## 7. SDLC Storyboard – Feature Release



## 8. SDLC Storyboard – Enhancement Release


Story boards
Mockups
Data Model
Use Cases
Author the Functional Design Specifications (FDS)
Review FDS
Edit FDS
FDS Approval
Coding for Beta Release
Beta Release to QA
Coding for Pilot Release
Pilot Release to QA
Update Manuals and Help
Develop Test Plans
Beta Testing
Update Internal Systems
Training of PMTI Personnel
Update Internal Systems
Update Pilot Sites with GA Release
Deploy to Pilot Sites
Rollout to all Sites with GA Release
Turnover to Client Services
Certification Testing for GA