Unified Documentation Strategy - DRAFT

Review Sign-Off

Name/Title	
Signature/Date	
	❑ Accepted in Current State ❑ Accepted with Noted Revisions ❑ Rejected ❑ Other (please indicate)
Comments	

Revision History

Version	Date	Name	Description of Changes
1.0	09/08/2000	Tom Foley	Issued for review.

Introducing the UDS

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.

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

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.

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).

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.

4. **Technical Design Specifications**

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

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.

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.

7. Product Test Plans

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

8. 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.

9. 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.

Dependencies

The dependencies are illustrated by the following:

1. Functional Requirements Specification (FRS)

1.1. Functional Design Specification (FDS)

1.1.1. Project Plan

1.1.2. Product Design

1.1.2.1. Product Manuals and Help

1.1.2.1.1. Product Training Materials

1.1.2.1.2. Product Implementation Guide

1.1.2.2. Product Test Plans

1.1.3. 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.

Scope of this Document

This document addresses structures and formats for the following documents in the UDS which are created as part of the development cycle:

1. Functional Requirements Specification (FRS)

2. Functional Design Specification (FDS)

3. Technical Design Specifications (TDS)

4. Product Manuals and Help

5. Product Test Plans

These structures provide the source materials for the templates that will be used to generate the project documents.

The structure and format of the Product Training Materials, Product Implementation Guide and Marketing and Sales Collateral should be addressed by their respective teams.

UDS Folder and File Structure

A successful UDS requires a rigid folder and file structure for maintaining the large number of files that support the UDS, as well as the individual project.

1. \\Odin\ftp\ftp\pmti\projects

All projects should be stored in this folder within an appropriate subfolder.

Project plans are stored in this directory and use the naming convention of **r&d.projectname.mpp**.

Avoid storing any other files in this folder.

1.1. r&d.projectname

Each project should have its own folder that is labeled to reflect the project code name using **r&d.projectname** format. Within that folder, subfolders should be created to contain all documents for the project, including working, published and archived versions of each. Avoid storing files in this folder.

1.1.1. design

This folder contains the Technical Design Specification for the project.

The naming convention for the Design Document should be **projectname.design.doc**. If more than one file exists, use projectname.design.filedesc.doc (e.g., nova.design.ui.doc or nova.design.chap01.doc).

1.1.2. doc&help

This folder contains the product documentation and help files required for the project. Subfolders for each product document should be created as needed.

1.1.3. other

This folder contains any files received from other sources, or files that don't fit into one of the other folder categories. This could include copies of standards, data tables, interface requirements, old documentation, etc. Include subfolders as needed.

1.1.4. qa

This folder contains the test plans and related documents for the project.

The naming convention for the test plan should be **projectname.test.doc**. If more than one file exists, use projectname.test.filedesc.doc (e.g., nova.test.subsystem01.doc).

1.1.5. specs

This folder contains the Functional Requirements Specification (FRS) and Functional Design Specification (FDS) for the project.

The naming convention for the FRS should be **projectname.frs.doc**. If more than one file exists, create an FRS subfolder and save all FRS files into that folder, using a naming convention based on projectname.frs.filedesc.doc (e.g., nova.frs.chap01.doc).

The naming convention for the FDS should be and **projectname.fds.doc**. If more than one file exists, create an FDS subfolder and save all FDS files into that folder, using a naming convention based on projectname.fds.filedesc.doc (e.g., nova.fds.chap01.doc).

UDS Archives and Working Versions

The versions of the documents stored in the above folders should be saved as read-only documents to prevent accidental overwriting or editing. Each of the above folders, excluding sources, should then contain the following subfolders:

1. **Archive**

   This folder contains previous published versions of the documents contained with the folder. Before the published versions are updated, the existing published version should be moved to the archive folder. The last modification date should be appended to the beginning of the filenames for the files just moved. For example, if you are working on the FRS for the "nova" project, the naming convention for the file would be sirius.frs.doc. If it was last modified on August 14, 2000, the filename should be changed to 20000814.nova.frs.doc.

   If the current version folder contains subfolders, then create a subfolder inside the archive folder using the date, and then copy/move all subfolders into that subfolder. For example, if the proddoc folder contains six subfolders and they were last modified on August 14, 2000, the archiving should be named 20000814, and the subfolders should be copied into that folder.

2. **Working**

   This folder contains the working version that is intended to be published next.

   NEVER work on the published version.

   Once you are ready to publish a new version, archive the current version and then copy the files and/or subfolders from the working folder into the root folder for the documents.

UDS Folder Structure for Single-file FRSs and FDSs

The resulting structure would be as follows:

1. UDS

- 1.1. r&d.project1name
  - 1.1.1. design
    - 1.1.1.1. Archive
    - 1.1.1.2. Working
  - 1.1.2. doc&help
    - 1.1.2.1. docproj1
      - 1.1.2.1.1. Archive
      - 1.1.2.1.2. Working
    - 1.1.2.2. docproj2
      - 1.1.2.2.1. Archive
      - 1.1.2.2.2. Working
    - 1.1.2.3. docproj3
      - 1.1.2.3.1. Archive
      - 1.1.2.3.2. Working
  - 1.1.3. other
  - 1.1.4. qa
    - 1.1.4.1. Archive
    - 1.1.4.2. Working
  - 1.1.5. specs
    - 1.1.5.1. Archive
    - 1.1.5.2. Working
- 1.2. r&d.project2name (as needed)

UDS Folder Structure for Multi-file FRSs and FDSs

If the FRS and FDS are multifile documents, the resulting structure would be as follows:

1. UDS

- 1.1. r&d.project1name
  - 1.1.1. design
    - 1.1.1.1. Archive
    - 1.1.1.2. Working
  - 1.1.2. doc&help
    - 1.1.2.1. docproj1
      - 1.1.2.1.1. Archive
      - 1.1.2.1.2. Working
    - 1.1.2.2. docproj2
      - 1.1.2.2.1. Archive
      - 1.1.2.2.2. Working
    - 1.1.2.3. docproj3
      - 1.1.2.3.1. Archive
      - 1.1.2.3.2. Working
  - 1.1.3. other
  - 1.1.4. qa
    - 1.1.4.1. Archive
    - 1.1.4.2. Working
  - 1.1.5. specs
    - 1.1.5.1. fds
      - 1.1.5.1.1. Archive
      - 1.1.5.1.2. Working
    - 1.1.5.2. frs
      - 1.1.5.2.1. Archive
      - 1.1.5.2.2. Working
- 1.2. project2name (as needed)

Project Plan

The project plan should identify all processes and milestones necessary for deliverying the proposed solution to the client and/or marketplace. In addition to product management and development processes, this plan should include the project plans of the teams identified in section 4 to make sure that the solution can be delivered in accordance with the expectations of the client and/or sales and marketing.

Microsoft Project Document

See //Odin/ftp/ftp/pmti/projects/r&d.working/r&d._generic.mpp for a sample project plan.

Function Requirements Specifications (FRS)

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:

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.

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. **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.

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.

Some Do's and Don'ts When Creating an FRS

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:

1. 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:

1.1. 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.

1.2. 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.

1.3. 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.

2. 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. 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.

FRS General Components

1. FRS Title Page

The title page should be of presentation quality (printed in color, if possible) and include the following elements:

1.1. Project Code Name – Functional Requirements Specification

This is the code name assigned to the project, followed by the name of the document.

Avoid using published product names and version numbers, as they could be subject to change. Such changes can cause future confusion with other projects within the same product suite or series.

1.2. Company Name

1.3. Project Leads and/or Primary Authors

List the full names of the project leads and/or primary authors for the FRS. You do not have to include contact information, as that is included elsewhere in the document.

1.4. Document Creation Date

This should be a field that is automatically updated when a project-specific FRS is created from the FRS template.

1.5. Document Modification Date

This should be a field that is automatically updated when the FRS is edited and saved.

1.6. Document Version Control Number

This should be a field that indicates the revision number of the FRS.

1.7. Document Location

This is the path and filename where the "record copy" of the FRS is stored. This copy should be in read-only format. You can use a field for this element.

Corporate logos and related artwork are nice additions to the title page, but avoid too much clutter.

2. FRS Confidentiality and Copyright Statements

This page includes any confidentialty or copyright statements that are required by the corporate legal counsel for all specification and design documents. If the project is for a specific client, be sure to include their appropriate confidentiality and copyright language as well.

Headers and footers within the document contents should contain some abbreviated form of these statements as directed by corporate legal counsel.

3. FRS Sign-off

Include a sign-off block for each signature member of the team.

Name/Title	
Signature/Date	
	❑ Accepted in Current State ❑ Accepted with Noted Revisions ❑ Rejected ❑ Other (please indicate)
Comments	

4. FRS Change History

Each time the FRS is published for review and/or consideration, make an entry in the following table.

Version	Date	Posted By	Description of Changes

5. FRS Project Team Members and Contact Information

The following table is used to record contact information for each member of the project team, as well as their role in the project. If there is a PMTI/LAW.com team and a third-party team, such as a client or customer, use different tables for each.

Name and Title	Role in Project	Phone and Fax	E-mail and Office

5.1. PMTI/LAW.com Team Members

Use this table for the PMTI/LAW.com Team Members if there are third-party teams involved in the project.

Name and Title	Role in Project	Phone and Fax	E-mail and Office

5.2. Third-Party Team Members (e.g., client)

Use this table for the third-party team members, if any, that are involved in the project. If there is more than one third-party team, insert a separate Level 2 line item and table for each.

Name and Title	Role in Project	Phone and Fax	E-mail and Office

6. FRS Table of Contents

The table of contents should include the first three levels of the FRS. To make the table of contents more usable, avoid using lengthy wording for FRS line items. Instead, use continuation paragraphs to expand the theme of each FRS line item.

7. FRS Back Cover

If the document ends on an even page, include a back cover page after the last section of the FRS. If the document ends on an odd page, no back cover as needed as the even side will be blank. This let's the reader know there are no more pages to review.

8. FRS Binding

Use some sort of formal binding for your FRS. A three-ring binder with cover and spine inserts makes a good presentation. If you need to reduce the "bulkiness" of the document, use a velobind system with a clear front cover and an opaque back cover.

FRS Contents

1. Scope of Functional Requirements Specification

The objective of this section is to describe the overall scope of this document, as well as the goals and objectives for this project. This section should also describe, at a high-level, the processes that will take place upon sign-off of this document, including development of the FDS (you might want to include a brief description of the contents of the FDS). As the FRS is developed, keep in mind that it will be marked as public, made available to clients, included in statement of work documents, provide support for marketing efforts, and used as the foundation for all other documents delivered as part of the solution.

[start text here]

2. Functional Description

The objective of this section is to convince the reader in several pages (or less) that we (1) have a solid understanding of their business (2) have correctly identified and understands a problem impacting their business and (3) are prepared to deliver an effective and efficient solution to that problem. This description should be divided into the following seven areas.

2.1. Business Function

Describe, at a high level, in a narrative non-technical form, and in terminology familiar to the client, the business problem for which we are providing a solution.

[start text here]

2.2. Business Environment

Describe the environment in which the business function exists. Is the business function part of an internal operation? Does the business function include the general public? Is there pre-existing technology that must be considered?

[start text here]

2.3. Business Function Workflow

Describe the workflow in the current business model. This section should accomplish two things: (1) inform developers who may not be familiar with the business workflow they are to fit within, and (2) provide a context where one may identify issues/problems with the current workflow which require a programmed/computerized solution.

[start text here]

2.4. Business-related Issues/Problems

Call out and describe specific known and anticipated issues or problems which might be experienced in the workflow. Not all issues and problems identified herein may be susceptible to a programmed/computerized solution, but they should still be identified. And even if they are, we may not be planning to address them all in this project, but deferring them to a future or parallel project. The issues and problems should be described in terms of the issues or problems, not in terms of the solution. Each issue and/or problem identified herein should be included in the Solution Scope area of the Proposed Solution section. It is there that we indicate whether the issue or problem is in scope or out of scope for this project. And if the issue or problem is out of scope, is it deferred to another project.

2.4.1. [Issue/Problem1]

[start text here]

2.4.2. [Issue/Problem2 (as needed)]

[start text here]

2.5. Additional Business-related Requests

Call out and describe specific requests for features that the client wants to have included in the project ("needs" should be included in 2.4). This is sometimes referred to as the "Wish List" or "Nice-to-Haves." It should be understood that the items contained herein should not be of the highest priority in the event they must be sacrificed or deferred in order to meet other project expectations (cost, time, required functionality, etc.). The issues/problems identified in 2.4 will be addressed first. Also keep in mind that not all requests identified herein may be susceptible to a programmed/computerized solution, but they should still be identified. And even if they are, we may not be planning to address them all in this project, but deferring them to a future or parallel project. The requests should be described in terms of the request, not in terms of the solution. Each request identified herein should be included in the Solution Scope area of the Proposed Solution section. It is there that we indicate whether the request is in scope or out of scope for this project. And if the request is out of scope, is it deferred to another project.

2.5.1. [Request1]

[start text here]

2.5.2. [Request2 (as needed)]

[start text here]

2.6. Business Market / Competition / Clients

List and/or describe why this project must be carried out, and what the consequences are if it is not.

[start text here]

2.7. Business Stakeholders

List and/or describe who cares about this project.

[start text here]

2.8. Business Terminology

List the business terminology with supporting definitions that can be understood by those not directly associated with this business. This section also shows the client that we understand what they are saying when they use their own terminology. The objective of this section is to get all members of the project team to speak a common "language" with respect to the solution.

2.8.1. [Term1]

[definition]

2.8.2. [Term2 (as needed)]

[definition]

3. Proposed Solution

The objective of this section is to (1) focus on issues and problems identified in the Functional Description, (2) describe the solution PMTI/LAW.com is proposing, (3) establish the scope of the solution by indicating which issues are being included, which are being excluded, and which are being deferred, (4) describe who will be using the solution and how, and (5) begin identifying the processes, dependencies and issues that could directly or indirectly impact the successful delivery of the solution. This section should be divided into the following 11 areas.

3.1. Solution Description

Describe, at a high level, in a narrative non-technical form, and in terminology familiar to the client, the solution we are proposing.

[start text here]

3.1.1. Solution Workflow

Describe the solution workflow at a high-level, in either narrative or graphical (the latter is preferred). Keep in mind that these are NOT the storyboards or detailed workflows that will be found in the Functional Design, though they can serve as the basis for them. If there is more than one workflow to describe, add subheadings for each.

[start text here]

3.1.1.1. Workflow1 (as needed)

[start text here]

3.1.2. Business Rules

Describe the business rules at a high-level and in narrative form. Keep in mind that these are NOT the detailed business rules that will be found in the Functional Design, though though they can serve as the basis for them. If there is more than one business rule "grouping" to describe, add subheadings for each.

[start text here]

3.2. Solution Scope

Using the Business-related Issues/Problems identified in area 2.4 beginning on page 19 and the Additional Business-related Requests identified in area 2.5 beginning on page 20, create a list items and indicated whether they are included and excluded from the scope of this project. If the item is included in the project, place a checkmark (Objects – Checkmark) in the "In" column, and briefly describe the scope of the solution. If the item is excluded from the project and is not the responsibility of PMTI/LAW.com, place a checkmark (Objects – Checkmark) in the "Out" column. If the item is excluded from this project but will be included in another project, type "DEF" in the "Out" column (for "Deferred"), and enter "Deferred to ProjName" in the "Scope of Solution" column, where ProjName identifies the project where the scope item will be addressed. It is a good idea to go ahead and create the FRS document for that project so you can begin capturing the solution requirements. If there are scope items to be included in the project that are not related

Scope Item	In	Out	Scope of Solution
2.4.1 - [Issue/Problem1]			

3.3. Solution Impact on Other Systems or Components

If delivery of the solution defined by the project project will cause other components of a larger system to change, you should identify the changes in this area of the FRS, and create a separate FRS for the impacted component. In the "Scope of Functional Requirements Specification" for that component, you should reference this project document as the source of the change being defined.

3.3.1. [System/Component1]

[start text here]

3.3.2. [System/Component2 (as needed)]

[start text here]

3.4. Actors

Lists the actors that will use the solution.

3.4.1. [Actor1]

Business Role	Describe the role of this actor within the business.
High-Level Goal	Describe what the actor will accomplish with this solution.
Skill Level	Describe the skill level of a typical actor. Include a range from "novice" to "advanced."
Environment Factors	Describe environmental conditions that will impact the actor's use of the system.
Performance & Productivity	Describe the actor's sensitivities with respect to performance and productivity.
Security Consideration	Describe security considerations that should be taken into account for the actor with respect to system and data access and manipulation. This discussion should be at a high-level and in general terms, as it will be more rigidly described and defined in the FDS.

3.4.2. [Actor2 (as needed)]

[insert table here]

3.5. Usage Descriptions

For each actor described in the previous section, list the specific usages of the system. First identify the goal-level usage description (e.g., Maintain User Accounts). Then within that goal, list the tasks-level use cases that will be needed to accomplish the goal (e.g., Create User Accounts, Find Existing User Accounts, View User Account Details, Modify User Account Details, Print User Account Details, Activate New User Accounts, Expire User Accounts, Reactivate User Accounts, Archive Expired/Inactive User Accounts and Delete Expired/Inactive User Accounts). Step text for each task-level use case is defined in the Function Design Specification (FDS). The purpose of this section is to merely identify those use cases.

It is possible for two actors to have goal-level usage descriptions and/or task-level use cases in common. In this section, you should repeat the goal-level usage description and/or task-level use case as needed, but in the Use Cases section of the FDS, you should only list them once by creating a composite for all actors.

3.5.1. [Actor1]

3.5.1.1. [Goal-level Usage Description 1]

3.5.1.1.1. [Task-level Use Case 1A]

3.5.1.1.2. [Task-level Use Case 1B (as needed)]

3.5.1.2. [Goal-level Usage Description 2 (as needed)]

3.5.1.2.1. [Task-level Use Case 2A]

3.5.1.2.2. [Task-level Use Case 2B (as needed)]

3.5.2. [Actor2 (repeat as needed)]

3.5.2.1. [Goal-level Usage Description 3]

3.5.2.1.1. [Task-level Use Case 3A]

3.5.2.1.2. [Task-level Use Case 3B (as needed)]

3.5.2.2. [Goal-level Usage Description 4 (as needed)]

3.5.2.2.1. [Task-level Use Case 4A]

3.5.2.2.2. [Task-level Use Case 4B (as needed)]

3.5.3. [SystemComponent1 (if and as needed)]

3.5.3.1. [Goal-level Usage Description 5]

3.5.3.1.1. [Task-level Use Case 5A]

3.5.3.1.2. [Task-level Use Case 5B (as needed)]

3.5.3.2. [Goal-level Usage Description 6 (as needed)]

3.5.3.2.1. [Task-level Use Case 6A]

3.5.3.2.2. [Task-level Use Case 6B (as needed)]

3.6. Solution-related Issues

Use this section to identify issues resulting out of delivering the proposed solution. These issues can include the ones identified in area 2.4 (Business-related Issues/Problems) beginning on page 19 and in area 2.5 (Additional Business-related Requests) beginning on page 20. Below are some categories where issues may exist once the solution is delivered. Add categories as needed. Representatives from each of these areas should address the issues listed in this document or in separate documents to be delivered later.

3.6.1. Financial and Business

List and/or describe financial and business issues related to the project.

[start text here]

3.6.2. Marketing and Sales

List and/or describe marketing and sales issues related to the project.

[start text here]

3.6.3. Legal

List and/or describe legal issues related to the project.

[start text here]

3.6.4. Geo-political

List and/or describe geo-political issues related to the project.

[start text here]

3.7. Solution Terminology

List solution-specific terminology with supporting definitions that can be understood by those not directly associated with the development and delivery processes. The objective of this section is to get all members of the project team to speak a common "language" with respect to the solution.

3.7.1. [SolutionTerm1]

[definition]

3.7.2. [Solution Term2 (as needed)]

[definition]

3.8. Feedback and Visibility

List prototypes and demos required. If usability studies are required, they should be identified in this section as well.

[start text here]

3.9. Project Values

List and/or define values for the project. These values "set the bar" for the project team when planning on how to deliver the solution.

3.9.1. Time

Describe the time constraints for this project, including external factors which may impact the delivery date (e.g., "Needs to be completed in this fiscal year.")

[start text here]

3.9.2. Cost

Without necessarily going into numerical detail, describe the cost considerations for this project.

[start text here]

3.9.3. Quality

While we always strive to produce the highest quality products, our expectations my exceed those of our clients or business partners. Describe the quality level that must be achieved as part of the solution.

[start text here]

3.9.4. Other

List and/or define other project values here.

[start text here]

3.10. Available Project Sacrifices

There is an old saying in the manufacturing industry - "We can make it fast, inexpensive and top-quality, pick two." Review the project values above, and identify where sacrifices could be made. For example, if the customer needs it fast and of top-quality, we may have to hire additional staff and pay some overtime. If time is not a consideration, but cost and quality are, then we may need to look at lengthening the project time to avoid overtime and build in additional test cycles. If time and cost are of paramont concern, then we may need sacrifice some "quality" in non-critical areas in order to meet the client's expectations. In addition to time, cost and quality, functionality is another area where sacrifices can be made (e.g., Can we use something we already have that meets/exceeds the requirements, and then modify it later?). Usability is another area where sacrifices can be made, especially in early stages of development, such as pre-release versions (e.g., a solution may be developed for the general public, but the first few phases are internal only, where tolerance is higher and "risk" is much lower).

3.10.1. Functionality

Identify that functionality that may not be critical and could be sacrificed in order to meet the time, cost and quality expectations for the project. For example, if there are ways of using a "command line" interface for carrying out certain administrative functions, can we defer development of a UI to a future release if needed? The items on this list are typically those items from Additional Business-related Requests identified in area 2.5 beginning on page 20 that have been identified in the Solution Scope (see page 21) as being included in the project.

[start text here]

3.10.2. Usability

Identify what areas of the solution will be used the least, and therefore do not need to be as "user-friendly" as areas of the solution that are used on a regular basis. For example, if there is an administrative function that is only used once a month, can usability analysis and implementation for that function be deferred to a future release if needed?

[start text here]

3.10.3. Other Poject Sacrifices

List and/or define other project sacrifices here.

[start text here]

3.11. System requirements (hardware, software and connectivity)

3.11.1. Servers

List and/or define the server configurations and related parameters that will be needed for and/or supported by the solution. Actual specifications should be prepared by the system engineers as part of their project deliverables.

[start text here]

3.11.2. Clients

List and/or define the client configurations and related parameters that will be needed for and/or supported by the solution. Actual specifications should be prepared by the system engineers as part of their project deliverables.

[start text here]

3.11.3. Other

List and/or define other system requirements that will be needed for and/or supported by the solution. Actual specifications should be prepared by the system engineers as part of their project deliverables.

[start text here]

3.12. Interface requirements

3.12.1. Standards

List and/or define interface standards that will be needed for and/or supported by the solution.

[start text here]

3.12.2. Inputs

If this solution is accepting inputs from some external system, or even an existing internal system, specify that here. Unless it is a very short standard, don't include the whole standard definition here; put it in an appendix, and refer to that appendix here. Alternatively, if there is an external document, refer to it. Specify which edition of the document you are referring to. If there are multiple interfaces, provide this information on each interface. If there are no external standards from which the inputs of the system will come, state that here.

[start text here]

3.12.2.1. First input requirement listed here if more than one.

[start text here]

3.12.2.2. Second input requirement listed here if more than one.

[start text here]

3.12.3. Outputs

If there are external interfaces to which the solution outputs must conform, discuss it here. Unless it is a very short standard, don't include the whole standard definition here. Instead, put it in an appendix, and refer to that appendix here. Alternatively, if there is an external document, refer to it. Specify which edition of the document you are referring to. If there are multiple interfaces, provide this information on each interface. If there are no external standards to which the outputs of the system must conform, state that here.

[start text here or…]

3.12.3.1. [first ouput requirement listed here if more than one]

[start text here]

3.12.3.2. [second ouput requirement listed here if more than one]

[start text here]

3.12.4. Connectivity

List and/or define connectivity requirements that will be needed for and/or supported by the solution. Additionally, describe any issues related to connectivity that will affect the solution, such as, protocol, leased lines used, etc.

[start text here or…]

3.12.4.1. [first connectivity requirement listed here if more than one]

[start text here]

3.12.4.2. [second connectivity requirement listed here if more than one]

[start text here]

4. Development Process

The objective of this section is to define the development process, and begin translating the proposed solution into a delivered one. This includes identifying key elements of product engineering, documentation, and testing, such as deliverables, tools and participants. This is a good opportunity to identify any dependencies that may directly or indirectly impact the development team, such as availability of resources.

4.1. Product Engineering

4.1.1. Engineering Deliverables

List and/or define the engineering deliverables for the solution.

[start text here]

4.1.1.1. Products

List and/or define the products that will be delivered as part of the solution.

[start text here]

4.1.1.2. Tools

List and/or define the tools that will be delivered as part of the solution.

[start text here]

4.1.1.3. Data

List and/or define any data that will be delivered as part of the solution. This includes global data for supporting the production environments, as well as data needed for demos and training. If the solution uses templates, list the templates that will be delivered as part of the solution, indicating whether they are for demo, training or production purposes.

[start text here]

4.1.2. Engineering Participants

Who will be working on this project? You do not need to list by name, but more by organization. For example, is this an "engine-only" project that does not require a UI?

[start text here]

4.1.3. Buy or Build?

Describe pluses and minuses of buying a solution or building a solution.

[start text here]

4.1.4. Engineering Tools

Are there tools that will be needed by engineering to deliver the solution? If so, do we build or buy the tools. This is only a consideration if the solution is utlilizing technology that is new to us, including legacy systems with which we have not previously interfaced.

[start text here]

4.1.5. Engineering Environments

4.1.5.1. Coding Environment

List and/or identify the hardware/software environment to be used by developers while coding the solution.

[start text here]

4.1.5.2. Testing Environment

List and/or identify the hardware/software environment to be used by developers while testing the solution prior to turnover to QA.

[start text here]

4.1.5.3. EBF Environment

List and/or identify the hardware/software environment to be used by developers for emergency bug fixes (EBFs) after the solution has been put into product. If an EBF is reported in a production environment, this environment would be used by development to correct and test the code causing the EBF prior to submitting to QA for certification.

[start text here]

4.2. Product Documentation

4.2.1. Documentation Deliverables

Identify the documentation deliverables for this project. List by name and/or description. For each, identify the audience, whether it will be available internally and/or externally, and in what formats it will be delivered.

Component Description	Audience	Int.	Ext.	Print	HTM	PDF	Doc
Functional Requirements Specification (this document)	Primary = Development Secondary = All Others	Yes	Clnt Only	Yes	No	Yes	Yes
Functional Design Specification	Primary = Development Secondary = All Others	Yes	Clnt Only	Yes	No	Yes	Yes
Design	Engineering	Yes	No	Yes	No	Yes	Yes
(proddoc1)							
(proddoc2)							

[start text here]

4.2.2. Documentation Participants

Identify who will be working on this project.

[start text here]

4.2.3. Documentation Tools

Identify what tools will be used on this project.

[start text here]

4.3. Product Testing

4.3.1. Certification

4.3.1.1. Certification Participants

List and/or identify who will be participating in the certification effort on this project.

[start text here]

4.3.1.2. Certification Scope

List and/or identify the scope of certification (e.g., full regression or new/impacted functions only).

[start text here]

4.3.1.3. Certification Tools

List and/or identify any specific tools needed for certication.

[start text here]

4.3.1.4. Certification Environment

List and/or identify the hardware/software environment to be used by QA while certifying the solution. This should be at a high-level and in narrative form, as the system engineers will be providing detailed specifications as part of their deliverables for this project.

[start text here]

4.3.1.5. EBF Environment

List and/or identify the hardware/software environment to be used by QA for emergency bug fixes (EBFs) after the solution has been put into product. If an EBF is reported in a production environment, this environment would be used by QA to replicate the EBF before submitting to development for correction. Once the EBF code has been corrected, this environment is used by QA to certify the fix. This should be at a high-level and in narrative form, as the system engineers will be providing detailed specifications as part of their deliverables for this project.

[start text here]

4.3.2. Benchmarking

4.3.2.1. Benchmarking Participants

List and/or identify who will be participating in the benchmarking effort on this project.

[start text here]

4.3.2.2. Benchmarking Scope

List and/or identify the scope of benchmarking.

[start text here]

4.3.2.3. Benchmarking Tools

List and/or identify any specific tools needed for benchmarking.

[start text here]

4.3.2.4. Benchmarking Environment

List and/or identify the hardware/software environment to be used by QA while benchmarking the solution. This should be at a high-level and in narrative form, as the system engineers will be providing detailed specifications as part of their deliverables for this project.

[start text here]

4.4. Engineering Dependencies

List and/or define other dependencies on this project.

[start text here]

4.4.1. LAW.com Engineering and Development

List and/or define project dependencies related to LAW.com engineering and development.

[start text here]

4.4.2. Client

List and/or define project dependencies related to the client.

[start text here]

5. Implementation and Support Processes

The objective of this section is to define the implementation and support process. This includes a discussion of upgrading existing systems, as well as new installations. This is a good opportunity to identify any dependencies that may directly or indirectly impact the implementation and support teams, such as availability of resources. All information provided herein should be at a high-level and in narrative form, as Professional Services will be providing a detailed guide as part of their deliverables for this project.

5.1. Distribution

List and/or define how the solution defined by the project will be distributed.

[start text here]

5.2. Integration

List and/or define how the solution defined by the project will be integrated into legacy systems.

[start text here]

5.3. Production Environment(s)

5.3.1. Staging

List and/or identify the hardware/software environment to be used by the system engineers for staging the next release of product software. This is where operations personnel will "test" upgrading the GA production environment to make sure the processes are known.

[start text here]

5.3.2. Pilot

List and/or identify the hardware/software environment to be used by the system engineers for piloting the next release of product software. This is where external users "test" the solution before it goes into GA. This can also be a pre-GA environment for new users.

[start text here]

5.3.3. General Availability

List and/or identify the hardware/software environment to be used by the system engineers for the current production release of product software.

[start text here]

5.4. Operation and Maintenance

Describe, at a high level and in a narrative non-technical form, the operation and maintenance policies and procedures that must be carried out by the system engineers in support of the solution. Detailed operation and maintenance policies and procedures should be fully documented and record by the system engineering team. Those documents should be referenced in this FRS and made available to the client as application.

[start text here]

5.5. Support

5.5.1. Internal Support

List and/or define project dependencies related to internally suporting the solution defined by the project.

[start text here]

5.5.2. External Support

List and/or define project dependencies related to externally suporting the solution defined by the project.

[start text here]

5.6. Training

5.6.1. Internal Training

Describe, at a high level and in a narrative non-technical form, the internal training that will be provided by Professional Services in support of the solution for each of the following groups (add groups as needed). Materials will be provided by the Professional Services training team as part of their deliverables for this project.

5.6.1.1. Marketing and Sales

[start text here]

5.6.1.2. Systems

[start text here]

5.6.1.3. Implementation

[start text here]

5.6.1.4. Support

[start text here]

5.6.2. External Training

Describe, at a high level and in a narrative non-technical form, the external training that will be provided by Professional Services in support of the solution for each of the following groups (add groups as needed). Materials will be provided by the Professional Services training team as part of their deliverables for this project.

5.6.2.1. End users

[start text here]

5.6.2.2. Marketing and Sales Partners

[start text here]

5.6.2.3. Integration Partners

[start text here]

6. [Appendix (as needed)]

Functional Design Specification (FDS)

This document is used to capture the high-level engineering specifications, including the UI design (storyboards and mockups), data models, security considerations, use cases and step text. This document provides the foundation for subordinate project documents, including but not limited to product documentation and test plans. Work should not begin on this document until the FRS has been approved by all appropriate members of the project team.

Some Do's and Don'ts When Creating an FDS

[to be written]

FDS General

1. FDS Front Cover

The front cover should be of presentation quality and include the following elements (the text should be the same as that found on the FRS):

1.1. Project Code Name – Functional Design Specification

1.2. Project Leads and/or Primary Authors

1.3. Document Creation Date

1.4. Document Modification Date

1.5. Document Version Control Number

1.6. Document Location

The Functional Design Specification (FDS) should be stored in the FuncReqs folder for the project. Refer to the Functional Requirements Specification (FRS) section for more information on archiving and working with UDS files.

2. FDS Confidentiality and Copyright Statements

This page includes any confidentialty or copyright statements that are required by the corporate legal department for all specifications. If the project is for a specific client, be sure to include their appropriate confidentiality and copyright language as well.

Headers and footers within the document contents should also contain some abbreviated form of these statements.

3. FDS Sign-off

Include a sign-off block for each "approving" member of the team.

Name/Title	
Signature/Date	
	❑ Accepted in Current State ❑ Accepted with Noted Revisions ❑ Rejected ❑ Other (please indicate)
Comments	

4. FDS Change History

Each time the FDS is published for review and or consideration, make an entry in the following table.

Version	Date	Name	Description of Changes

5. FDS Table of Contents

The table of contents should include the first three levels of the FDS. To make the table of contents more usable, avoid using lengthy wording for FDS line items. Instead, use continuation paragraphs to expand the theme of each FDS line item.

6. FDS Back Cover

Include a back cover page after the last section of the FDS. This let's the reader know there are no more pages to review.

7. FDS Binding

Use some sort of formal binding for your FDS. A three-ring binder with cover and spine inserts makes a good presentation. If you need to reduce the "bulkiness" of the document, use a velobind system with a front and back cover.

FDS Content

1. Scope of Function Design Specification

What is the overall scope of this document.

[start text here]

2. UI Design

Insert images of UI pages here.

[start text here]

2.1. Storyboards

Insert introductory text here for this "Storyboards" section. This text should explain why storyboards are being provided, and what the reviewer is expected to do with them.

[start text here]

2.1.1. First Storyboard

Insert the first storyboard here. Where practical, try and cross-reference storyboard elements with usage descriptions and/or use case step text.

[start text here]

2.1.2. Second Storyboard (as needed)

[start text here]

2.2. Mockups

Insert introductory text here for this "Mockups" section. This text should explain why mockups are being provided, and what the reviewer is expected to do with them.

[start text here]

2.2.1. First Mockup

Insert the first mockup here, followed by a properties table to identify the various components.

[start text here]

GENERAL PAGE ELEMENTS

Element	Description / Function
Help ID	
Title Bar Text	

BUTTONS

Label	Function	Use Case

LINKS

Label	Function	Use Case

FIELDS

Label	Function	Use Case

COLUMNS

Label	Function	Use Case

2.2.2. Second Mockup (as needed)

[start text here]

3. Data Models

Insert data models here.

[start text here]

4. Security

Define the security requirements here. This includes login processes, user validation, and access to the various pages and data within the solution. Add subheadings as needed.

[start text here]

5. Use Cases

Use cases describe the way a user uses the system. For our purposes, a "user" is a person, the "way" is a device (e.g., PC, PDA, terminal, etc.) running some sort of interface software, and the "system" is where information is stored. Obviously, there is a lot more to it than that, but this provides a fundamental definition from which to work. The question also surfaces regarding the system – Is the system a user with its own use cases? The answer is "yes" with respect to how the system handles external interactions with the user (i.e., process a request), but "no" with respect to internal processes, such as parsing and storing individual data elements. The latter should be addressed in the Design document.

Insert introductory text here for this "Use Cases and Step Text" section. This text should explain why use cases and step text are being provided, and what the reviewer is expected to do with them.

The goal-level usage descriptions should be a composite of those listed in the Usage Descriptions section of the FRS, regardless of actor. The task-level use cases should be a composite of those listed for each usage description in the Usage Descriptions section of the FRS, regardless of actor.

[start text here]

5.1. Goal-level Usage Decription 1

Identify the goal-level task for this collection of use cases (e.g., "Maintain User Accounts"), and provide a brief explanation of its usage within the system.

[start text here]

5.1.1. Task-level Use Case 1A Name

The name is the goal as a memorable short active verb phrase (e.g., Create User Accounts, Find Existing User Accounts, View User Account Details, Modify User Account Details, Print User Account Details, Activate New User Accounts, Expire User Accounts, Reactivate User Accounts, Archive Expired/Inactive User Accounts and Delete Expired/Inactive User Accounts).

Goal in Context	A longer statement of the goal in context, if needed.
Scope & Level	What system is being considered black box under design – one of : Summary, Primary Task, Subfunction.
Preconditions	What we expect is already the state of the world.
Success End Condition	The state of the world upon successful completion.
Failed End Condition	The state of the world if goal abandoned.
Primary Actors	Role names or descriptions for the primary actors.
Secondary Actors	Roles names or descriptions for any secondary actors or systems.
Trigger	The action upon the system that starts the use case.

STEPS

Step	Action
1.	Put here the steps of the scenario from trigger to goal delivery,and any cleanup after.
2.	

EXTENSIONS

Rel. Step	Branching Action
1.	a) Condition causing branching in step 1 – action or name of sub-use case, if any. b)
2.	a)

SUBVARIATIONS

Rel. Step	Branching Action
1.	a) List of variations to step 1. b)
2.	a)

RELATED INFORMATION

Priority	How critical to your system / organization.
Performance	The amount of time this use case should take.
Frequency	How often it is expected to happen.
Channels to Actors	For example, interactive, static files, database, timeouts.
Open Issues	List of issues awaiting decision affecting this use case.
Due Date	Date or release needed.
Other Management Information.	As needed.
Superordinates	Optional, name of use case(s) that includes this one.
Subordinates	Optional, depending on tools, links to sub.use cases.

5.1.2. Task-level Use Case 1B Name (as needed)

Insert the Use Case Table at the beginning of the line below.

5.2. Goal-level Usage Decription 2 (as needed)

5.2.1. Task-level Use Case 2A Name

Insert the Use Case Table at the beginning of the line below.

5.2.2. Task-level Use Case 2B Name (as needed)

Insert the Use Case Table at the beginning of the line below.

6. [Appendix (as needed)]

Working Design

Using the FRS and FDS, the development team translates the business requirements and proposed solution into a working design, which is then captured in this document.

1. Outline needs to be developed.

Documentation and Help

1. Outline Needs to Be Developed

Product Test Plans

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

1. Outline needs to be developed.

Item #	Item Description	Expected Results
1		
2		
3		

Pass/Fail: _____
Client Signature/Date (if applicable): _____
PMTI/LAW.com QA Signature/Date: _____

This page intentionally left blank.