

# **Software Project Management Plan**

## **Highland Basic Order Tracking System**

Prepared For: Highland Office Products  
Prepared By: John Steele  
Elucidata

Document ID: 100-01-01  
Version: 01.1

Copyright © 2001 Elucidata.  
All Rights Reserved.

---

## TABLE OF CONTENTS

---

<b>INTRODUCTION</b> .....	<b>5</b>
STANDARDS .....	6
<b>VISION &amp; SCOPE</b> .....	<b>7</b>
BACKGROUND .....	7
SOLUTION OVERVIEW .....	9
SCOPE & LIMITATIONS .....	11
<b>FEASIBILITY AND RISK ANALYSIS</b> .....	<b>13</b>
CRITICAL SUCCESS FACTORS.....	13
PROJECT SIZE.....	15
PROJECT RISK .....	16
FEASIBILITY AND RISK SUMMARY.....	16
<b>MANAGEMENT APPROACH</b> .....	<b>17</b>
PROJECT STRUCTURE .....	17
SOFTWARE QUALITY ASSURANCE (SQA) .....	18
SOFTWARE CONFIGURATION MANAGEMENT (SCM) .....	19
<b>TECHNICAL APPROACH</b> .....	<b>20</b>
ANTICIPATED IMPLEMENTATION ARCHITECTURE .....	20
DOCUMENTATION DELIVERY FORMATS .....	21
COMMUNICATIONS FORMAT .....	21
PROJECT HOME PAGE .....	21
<b>TRACEABILITY LISTING</b> .....	<b>22</b>

<b>GLOSSARY OF PROJECT-SPECIFIC TERMS</b>	
Glossary of Software Engineering Terms	A standard <a href="#">Glossary of Software Engineering Terms</a> is maintained separately. Terms specific to this project are maintained below.
Basic Order Tracking System	Software to be developed during this project.
BOTS	Acronym for Basic Order Tracking

---

## INTRODUCTION

---

This document serves as the project plan for the Basic Order Tracking System (BOTS) software development effort. The content of this document is divided into four major chapters: Vision & Scope, Feasibility & Risk Analysis, Management Approach, and Technical Approach.

The Vision & Scope chapter describes the conditions driving the development of BOTS. It provides an overview of the application as initially envisioned, and describes the scope and limitations of the design effort.

The Feasibility & Risk Analysis chapter addresses the issues of application complexity as well as the anticipated solution domain, database load and project size. Initial analysis indicates that this is a small, low-risk project with no abnormal database record volumes or transaction loads.

The Management Approach chapter describes the high-level structure and processes of the project as well as processes for software quality assurance and configuration management.

The Technical Approach chapter describes the anticipated development tools, implementation software, documentation delivery formats and formal communications methods for the project. Although a final determination of suitable implementation tools cannot be made until the design stage is concluded, a "best guess" estimate of tools based on high-level requirements helps to focus the processes of requirements elicitation and systems design.

Support items related to this document include BOTS [Use Cases](#), from which high-level requirements were derived, and [high-level project schedule](#).

## **STANDARDS**

---

The following standards were used as guides to develop this project plan. The standards were reviewed and this content tailored to the the specific needs of this project.

- ANSI/IEEE 1058: Standard for Software Project Plans
- ANSI/IEEE 730.1: Standard for Software Quality Assurance Plans
- ANSI/IEEE 828: Standard for Software Configuration Management Plans
- SEI/CMM: Software Project Planning Key Process Area

---

## VISION & SCOPE

---

The purpose of this chapter is to provide a high-level overview of the BOTS application. The three major sections of this chapter include the Background, the Solution Overview, and the Scope & Limitations of this project. This information is used to define and constrain the scope of the subsequent requirements and design stages of this software development effort.

### BACKGROUND

---

#### PROBLEM STATEMENT

Highland Office Products has been growing steadily over the past 10 years. During this time, Highland has incurred significant expense in upgrading their order entry software every few years as their client base and order volume grows. Each time they upgrade, the new vendor has difficulty transferring the old data to the new system, training costs are significant, and the transition requires a great deal of time and effort to minimize and recover from errors incurred during the process.

Highland management has decided that it is time to begin development of their own software, using scalable relational database technology. This will allow Highland to benefit from owning their own code and stabilizing the user interface and processes under which they will operate in the future.

#### BUSINESS RISKS

Failure to complete the development of BOTS will result in continued excess expenditures by management to support upgrades of systems that are not scalable and which do not provide adequate interfaces with other applications. Lack of adequate interfaces will result in continued expenditures of time and effort by staff to duplicate or replicate data outside of the existing system.

Risks incurred by implementing BOTS include missing, incomplete, or erroneous features, inadequate performance, and the potential for introducing errors into existing data during the data migration process. These risks will be mitigated by

formal processes for review and acceptance of requirements, design elements, and implemented code modules. In addition, a full data migration to a pre-production test system will be installed and verified against the existing production data prior to acceptance of the final system.

## **PRINCIPAL PROJECT PARTICIPANTS**

Software projects by nature include a wide variety of personnel with varying degrees of involvement as the project progresses. Rather than attempt to document all possible project participants, this project plan identifies those personnel tasked with direct responsibilities for the success of the project.

### **PROJECT EXECUTIVE SPONSOR**

Ted Adams is the Highland executive in charge of approving and funding this project.

### **PRIMARY END-USER REPRESENTATIVE (PER)**

Sara Conner will act as the primary point of contact and interface to BOTS end-users. The PER will work directly with the Primary Developer Representative to identify subject matter experts, coordinate meetings, conduct reviews, and track identified issues, consolidating and coordinating the responses of all end-users into a cohesive set of communications.

### **PRIMARY DEVELOPER REPRESENTATIVE (PDR)**

Joe Smith will act as the primary point of contact and interface to the development team. The PDR will work directly with the Primary End-user Representative to coordinate meetings, conduct reviews, and track identified issues. In addition, the PDR will be responsible for project tracking, customer communications, and insuring that defined standards for software quality assurance and configuration management are met.

### **QUALITY ASSURANCE (QA) REVIEWER**

Jim Jones will conduct periodic reviews of the work products associated with this project as defined in the Software Quality Assurance (SQA) section of the Management chapter of this project plan. The QA reviewer will work independently from the development team to ensure objective audits and reviews of the work products and processes of this software development project.

## **SOLUTION OVERVIEW**

---

### **VISION STATEMENT**

Highland management envisions a database application that will support basic order entry and shipping operations, and transmit fulfilled order data to the Highland billing and accounting systems. Although the initial development effort will be only to support core features that are critically required for successful order entry and fulfillment, the application will be developed using open, standards-based technology that facilitates integration with other systems, additional development, and significant scalability.

### **HIGH-LEVEL REQUIREMENTS**

High-level requirements are the "goals" that the system is to attain. All requirements, their associated design elements, and the subsequent test cases are traceable to one or more of these goals. The high-level use cases from which these goals were derived are available on the [project home page](#). The high-level requirements for the BOTS application are as follows:

#### **G1-BOTS**

##### **Client/server RDBMS**

Provide a graphical user interface into a Relational DataBase Management System (RDBMS) for BOTS users. The database engine will provide the standard features expected of a RDBMS, including data transfer, storage, bulk modification of data, relationship management and access control. End users will connect to the RDBMS via a graphical user interface client, query and reporting tool, or web browser, as appropriate to the task.

#### **G2-BOTS**

##### **Manage operational data**

Provide the ability to enter, update, and maintain operational data, such as customers, products, and customer orders.

#### **G3-BOTS**

##### **Manage reference data**

Collect and maintain reference data, such as product categories and customer demographic categories.

#### **G4-BOTS**

##### **Access control**

Restrict access to BOTS data to specified sets of authorized users.

## **G5-BOTS**

### **Standard reports**

Provide the ability to periodically query the database and generate standardized reports, such as order volume by period, customers by demographic group, and product demand.

## **G6-BOTS**

### **Ad hoc queries and reports**

Provide the ability for end-users to develop ad hoc queries and columnar reports as necessary to support day-to-day operations.

## **G7-BOTS**

### **Data transfer**

Transfer data regarding fulfilled orders to the Highland billing and accounting system. Transfer data to Microsoft Word and Excel for other uses and analyses.

## **G8-BOTS**

### **User community**

Personnel involved with operating and maintaining BOTS are divided into five major classes:

#### **DATABASE ADMINISTRATION**

Database administration personnel are responsible for maintenance of the authorized user list, managing the reference data, and managing certain operational data such as product information. Database administrators are also responsible for performance management, operating system and database server upgrades, and periodic backup of the database.

#### **SALES PERSONNEL**

Sales personnel are primarily responsible for entry and maintenance of customer data and orders. Sales personnel may also periodically transfer data to Microsoft Word for the generation of mailings.

#### **SHIPPING PERSONNEL**

Shipping personnel are responsible for identifying which order items have been shipped and which have been placed on back order by updating the appropriate order line items.

#### **ACCOUNTING PERSONNEL**

Accounting personnel are responsible for periodically initiating the transfer of fulfilled order data from BOTS to the Highland billing and accounting system.

### **MANAGEMENT**

Management personnel are responsible for generating standardized and *ad hoc* queries and reports as necessary.

## **SCOPE & LIMITATIONS**

This section describes the scope and limitations of the project, not of the software. The scope and limitations of the software under design have been fully described in the solution overview section above.

### **SCOPE OF CURRENT EFFORT**

This project will gather requirements, perform analyses, develop design elements and produce software artifacts for a complete implementation BOTS. The goal of this project is to produce a fully functional and acceptable software product that satisfies the high-level requirements described above. The activities associated with this project are described in the associated [Software Development Life Cycle](#) document for this project.

The specific activities anticipated for each stage are identified in the high-level project schedule as each previous stage is completed. For example, when the planning stage is completed and this project plan is finalized, the high-level project schedule will contain a detailed listing of the anticipated activities for the requirements stage, but the design stage will only be identified as a single consolidated activity with an estimated duration. The software development team has found that it is very difficult to provide accurate, detailed plans for anything beyond the next stage at any point in the software development life cycle.

### **SCOPE OF FUTURE EFFORTS**

Future efforts may include the addition of new capabilities and enhancements of existing features, as required by Highland personnel.

## **CONSTRAINTS & LIMITATIONS**

### **CONSTRAINT: SPECIALIZED SOFTWARE**

The software under development is intended to be special-purpose software, focused tightly on customer needs. As such, the software is not intended to serve a broad market or multiple customers, and the implementation of the software is not required to be compatible with multiple platforms or interface standards.

**LIMITATION: SOFTWARE PROTOTYPES**

From time to time, the software development team, to clarify requirements and/or design elements, may generate mockups and prototypes of screens, reports, and processes. Although some of the prototypes may appear to be very substantial, they're generally similar to a movie set: everything looks good from the front but there's nothing in the back.

When a prototype is generated, the developer produces the minimum amount of code necessary to clarify the requirements or design elements under consideration. No effort is made to comply with coding standards, provide robust error management, or integrate with other database tables or modules. As a result, it is generally more expensive to retrofit a prototype with the necessary elements to produce a production module than it is to develop the module from scratch using the final system design document.

For these reasons, prototypes are never intended for business use, and are generally crippled in one way or another to prevent them from being mistakenly used as production modules by end-users.

---

## FEASIBILITY AND RISK ANALYSIS

---

This chapter evaluates the size (level of effort), feasibility and risks associated with this software development effort, and includes a listing of critical success factors, a project sizing estimate, and a project risk assessment.

### **CRITICAL SUCCESS FACTORS**

---

Critical success factors consist of assumptions and dependencies that are generally outside the control of the software development team. If an assumption proves to be false, or a dependency proves to be unreliable, the success of the project is placed at significant risk.

#### **ASSUMPTION: IMPLEMENTATION PLATFORM STABILITY**

It is assumed that, once specified, the implementation platforms for client and server partitions of BOTS will not be subject to change in the foreseeable future. This is especially important in terms of the client platform; adding a requirement to support a substantially different client operating system at a later date will threaten the viability of the project.

#### **ASSUMPTION: INFRASTRUCTURE STABILITY**

It is assumed that, once specified, the network operating system and the respective operating systems of the file servers, database servers and Web servers upon which BOTS will be dependent will not be subject to significant change in the foreseeable future. Portions of BOTS will be dependent upon networking protocols, operating system calling parameters and native language support provided by the currently implemented operating systems for the various servers and network. Changes in any of these software infrastructure elements will require significant modification of BOTS and will threaten the viability of the project.

### **ASSUMPTION: PROJECT PRIORITY**

It is assumed that Highland management will encourage those personnel assigned to the project to place a high priority on tasks associated with the project. Inadequate customer personnel participation is one of the leading causes of software project failures.

### **ASSUMPTION: SOFTWARE QUALITY ASSURANCE (SQA) PROCESS**

The SQA process essentially controls the review and approval cycle for project documents and software artifacts. Since Highland has no formal SQA process in place for this project, SQA activities will be initiated by the development team in accordance with the process defined in the Management Approach: SQA portion of this document.

### **ASSUMPTION: SOFTWARE CONFIGURATION MANAGEMENT (SCM) PROCESS**

Configuration management is often referred to as change management. The SCM process essentially controls document and software artifact classification, storage, revision and retrieval. Since Highland has no formal SCM process in place for this project, SCM activities will be initiated by the development team in accordance with the process defined in the Management Approach: SCM portion of this document.

### **DEPENDENCY: SUBJECT MATTER EXPERTS**

The primary end-user representative will identify a Subject Matter Expert (SME) for each high-level requirements defined in the Solution Overview section of this document. SMEs are considered to be experts in the processes supporting the high-level requirements. SMEs will act as primary sources of requirements and will be members of the core review team charged with reviewing documents, providing feedback, and coordinating test activities for software modules in their domain of expertise.

### **DEPENDENCY: STABILITY OF END-USER REPRESENTATION**

A software development project is highly dependent upon the stability of the population of customer personnel assigned to the project. A constantly shifting set of customer personnel leads to a constantly shifting set of requirements and design elements; this is another leading cause of software project failures.

### **DEPENDENCY: RAPID REVIEW CYCLE**

People work best when information is fresh in their minds. Therefore, it is essential that documentation generated by the software development team be

reviewed as quickly as possible, while the discussions that resulted in the documentation are fresh in the minds of customer personnel. Review comments need to be posted quickly to allow the documentation to be updated in a timely manner. A lengthy review cycle typically leads to a project becoming stagnant, and is one of the leading causes of project abandonment.

**DEPENDENCY: EXTERNAL SYSTEMS INTERFACE SPECIFICATION**

The BOTS development team will need to have access to detailed specifications for the exchange of information between BOTS and other external systems. A "trial and error" approach here will place significant risk on the success of the project.

**PROJECT SIZE**

Project size is estimated by averaging the amount calculated for each of six project size factors. Full definitions for each of these project size factors are contained in the [Glossary of Software Engineering Terms](#). In the table below, the calculated ranges are highlighted in red for each sizing factor.

**Project Size Factors**

	Small	Medium	Large
Number of Operational Data Areas	<b>1-8</b>	9-15	16+
Number of Support Data Areas	<b>0-10</b>	16-30	31+
Number of External Interfaces	<b>0-2</b>	3-5	6+
Number of Simultaneous Users	<b>1-10</b>	11-40	41+
Initial Data Load (kR = Thousand Records)	<b>0-10kR</b>	10-50kR	>50kR
Operational Transaction Load (kTpd = Thousand Transactions Per Day)	<0.5kTpd	<b>0.5-3kTpd</b>	>3kTpd

Based on the values calculated for each project size factor, this project is estimated to be a small development effort.

## PROJECT RISK

Project risk is estimated by selecting the most appropriate attributes for each of six project risk factors. Full definitions for each of these project risk factors are contained in a standard [Glossary of Software Engineering Terms](#). In the table below, the selected attributes are highlighted for each risk factor.

### Project Risk Factors

	Low Risk	Medium	High
Implementation Technology	<b>Well known</b>	Some new	Cutting edge
External Interface Complexity	<b>All simple</b>	Minority complex	Majority complex
Developer Skills / Resources	<b>Readily available</b>	Some contention	Need to be acquired
Customer Resources (UC = Use Case)	>1 SME per UC available	<b>1 SME per UC available</b>	Outside SME(s) used
Business Process Maturity	<b>Standard, stable</b>	Non-standard, stable	Automating new/unusual process(es)
Business Process Complexity	All simple	<b>Minority complex</b>	Majority complex

Based on the attributes selected for each project risk factor, this project is estimated to be a low risk development effort.

## FEASIBILITY AND RISK SUMMARY

The previous analyses indicate this is a small, low risk software development effort. The development team has not identified any significant issues pertaining to level of effort or probability of success, outside of the previously described critical success factors.

---

## MANAGEMENT APPROACH

---

The management of a software project includes three primary focus areas. The first area focuses on defining the structure, or stages and sequence of activities, of the project. The second area focuses on insuring the development and delivery of high-quality work products through the application of a Software Quality Assurance (SQA) process. The third area focuses on the coordination of changes to current work products through the application of a Software Configuration Management (SCM) process. This chapter addresses these three focus areas.

## PROJECT STRUCTURE

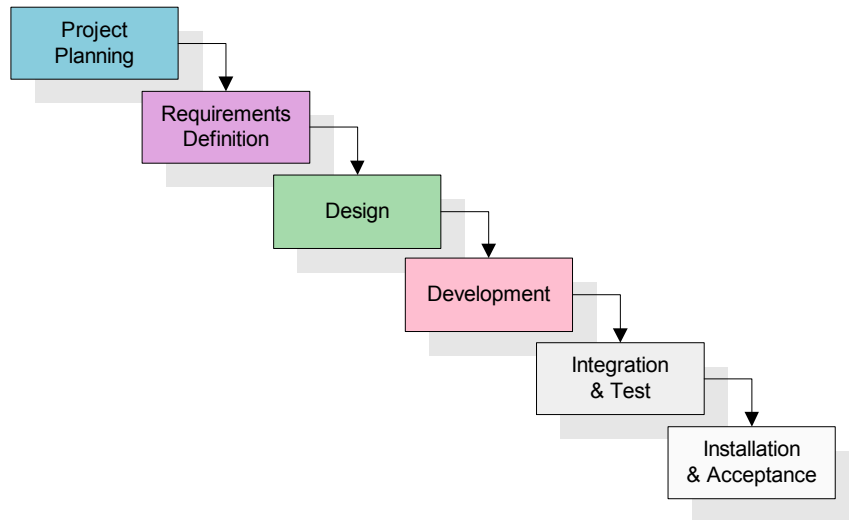
---

This section includes an overview of and links to documents describing the software development life cycle, the software quality assurance process, and the software configuration management process for this project.

The [project schedule](#) presents a comprehensive listing of the tasks and activities associated with the planning and requirements gathering stages for this project. Subsequent stages are described only at a very high level. In software development, it is very difficult to identify detailed tasks and activities associated with a life cycle stage that is more than one stage beyond the current stage of development.

## **SOFTWARE DEVELOPMENT LIFECYCLE**

Software projects of this size are generally broken down into six stages:



The Software Development Life-Cycle (SDLC) shown above utilizes the waterfall model with modifications for rapid prototyping. The complete software development lifecycle for this project is described in a separate document, available at:

<http://www.elucidata.com/refs/sdlc.pdf>

Please refer to this document for a description of the structure, inputs to and outputs from each of the stages shown above, as well as a comparison to other SDLC models.

## **SOFTWARE QUALITY ASSURANCE (SQA)**

The software quality assurance process implements periodic reviews of current draft deliverables and selected work products in each stage of development. The primary objective of the SQA process is to ensure the production of high-quality work products according to stated requirements and established standards. SQA incorporates a formal process for evaluating and documenting the quality of the work products produced during each stage of the software development lifecycle.

The complete SQA process for this project is described in a separate [Software Quality Assurance Plan](#). Please refer to this document for a description of the structure and timing of SQA activities for this project.

## **SOFTWARE CONFIGURATION MANAGEMENT (SCM)**

---

The software configuration management process controls the identification and retention of all the various artifacts (documents, source code, executables, e-mail, etc.) generated during the software development lifecycle. The primary objective of the SCM process is to coordinate the use of software artifacts among the project participants, making sure everyone is working with the same versions of various artifacts (change control) and making sure that nothing gets lost (retention control).

The complete SCM process for this project is described in a separate [Software Configuration Management Plan](#). Please refer to this document for a description of the structure and timing of SCM activities for this project.

---

## TECHNICAL APPROACH

---

This chapter describes the anticipated development tools, implementation software, documentation delivery formats and formal communications methods for the project. Although a final determination of suitable implementation tools cannot be made until the design stage is concluded, a "best guess" estimate of tools based on high-level requirements helps to focus the processes of requirements elicitation and systems design.

### ANTICIPATED IMPLEMENTATION ARCHITECTURE

---

The anticipated BOTS system will use a three-tier application approach, consisting of a database server, an application server, and a browser-based web client. The end user will access the application via a web home page, which provides top-level access to the major features of the application. Individual functions will be provided by normal html pages, active server pages, and visual basic applets, as appropriate. These application components will be stored on the central application server, and delivered to the browser as required.

### ANTICIPATED DEVELOPMENT TOOLS

The anticipated development toolset consists of Microsoft SQL server for back-end database storage and retrieval, Microsoft Internet Information Server (IIS) for web and application serving, Seagate Crystal Reports for *ad hoc* and standardized report generation, and Microsoft Internet Explorer for client tasks.

Configuration management of source items will be handled by Microsoft Visual Source Safe. Code development will be done via Microsoft Visual Basic. Microsoft FrontPage will be used for web site development. Online help will be developed and delivered in HTML Help format.

The final application will be delivered as a pre-configured SQL Server database and an IIS compatible set of web pages, active server pages, and visual basic applets.

## **DOCUMENTATION DELIVERY FORMATS**

---

All documentation will be delivered as Adobe Acrobat Portable Document Format (pdf) files, with the exception of the online help system. This will allow all participants to review and annotate documents during the review cycle, as well as fill out online forms and apply digital signatures. All project participants who will be reviewing deliverables will need to have Adobe Acrobat Business Tools installed on their workstations.

## **COMMUNICATIONS FORMAT**

---

Informal communications between project participants can take place in person, via telephone, or via electronic mail. Minutes from face-to-face and web teleconference meetings will be distributed via e-mail for review and clarification if necessary. For one-on-one meetings or telephone calls between client and development team personnel, the member of the development team will produce an e-mail summary of the conversation and send it to the other party for clarification if necessary. All e-mail communications will be forwarded to a central project archive account for later reference.

Formal communications consist of stage deliverable review forms and memoranda of acceptance. These are digitally signed and stored in the project archive.

## **PROJECT HOME PAGE**

---

The development team maintains a Home page for this project at [http://www.elucidata.com/projects/elcd\\_100/100home.html](http://www.elucidata.com/projects/elcd_100/100home.html). The purpose of the project home page is to act as a central information repository for the project, allowing all participants direct access to the most recent versions of project artifacts. In addition, the home page contains links to current prototypes, documents under development, and other project support items.

Note that this project site is open to all, but all documents are restricted to current project participants via a confidential project password. Unauthorized personnel do not have access to password, and project participants do not have access to other project passwords managed by the development team. Authorized personnel are identified by the PER and PDR.

---

## TRACEABILITY LISTING

---

The following listing is automatically maintained by the word processor for this document. It is of use only to systems analysts when verifying requirements traceability across documents.

**G1-BOTS**  
**G2-BOTS**  
**G3-BOTS**  
**G4-BOTS**  
**G5-BOTS**  
**G6-BOTS**  
**G7-BOTS**  
**G8-BOTS**

Filename: 100-01-01  
Directory: C:\Elucidata\100-Highland BOTS\Stages\1 -  
01-Plan\Source  
Template: C:\Documents and Settings\John Zoltai\Application  
Data\Microsoft\Templates\Shell Templates\Standard Shell Document.dot  
Title: Software Project Management Plan  
Subject: Highland Basic Order Tracking System  
Author: John Steele  
Keywords:  
Comments: Copyright © 2001 Elucidata  
Creation Date: 9/1/2000 6:31 PM  
Change Number: 35  
Last Saved On: 11/28/2001 4:26 PM  
Last Saved By: John Steele  
Total Editing Time: 695 Minutes  
Last Printed On: 11/28/2001 4:27 PM  
As of Last Complete Printing  
Number of Pages: 22  
Number of Words: 4,736 (approx.)  
Number of Characters: 26,997 (approx.)