<Project Name>

Software Development Plan (Small Project)

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Introduction 3](#_Toc523117788)

[1.1 Purpose **Error! Bookmark not defined.**](#_Toc523117789)

[1.2 Scope **Error! Bookmark not defined.**](#_Toc523117790)

[1.3 Definitions, Acronyms, and Abbreviations **Error! Bookmark not defined.**](#_Toc523117791)

[1.4 References **Error! Bookmark not defined.**](#_Toc523117792)

[1.5 Overview **Error! Bookmark not defined.**](#_Toc523117793)

[2. Project Overview Error! Bookmark not defined.](#_Toc523117794)

[2.1 Project Purpose, Scope, and Objectives **Error! Bookmark not defined.**](#_Toc523117795)

[2.2 Assumptions and Constraints **Error! Bookmark not defined.**](#_Toc523117796)

[2.3 Project Deliverables **Error! Bookmark not defined.**](#_Toc523117797)

[2.4 Evolution of the Software Development Plan **Error! Bookmark not defined.**](#_Toc523117798)

[3. Project Organization Error! Bookmark not defined.](#_Toc523117799)

[3.1 Organizational Structure **Error! Bookmark not defined.**](#_Toc523117800)

[3.2 External Interfaces **Error! Bookmark not defined.**](#_Toc523117801)

[3.3 Roles and Responsibilities **Error! Bookmark not defined.**](#_Toc523117802)

[4. Management Process Error! Bookmark not defined.](#_Toc523117803)

[4.1 Project Estimates **Error! Bookmark not defined.**](#_Toc523117804)

[4.2 Project Plan **Error! Bookmark not defined.**](#_Toc523117805)

[4.2.1 Phase Plan **Error! Bookmark not defined.**](#_Toc523117806)

[4.2.2 Iteration Objectives **Error! Bookmark not defined.**](#_Toc523117807)

[4.2.3 Releases **Error! Bookmark not defined.**](#_Toc523117808)

[4.2.4 Project Schedule **Error! Bookmark not defined.**](#_Toc523117809)

[4.2.5 Project Resourcing **Error! Bookmark not defined.**](#_Toc523117810)

[4.3 Project Monitoring and Control 3](#_Toc523117811)

[5. Annexes 3](#_Toc523117821)

Software Development Plan (Small Project)

# 

# Introduction

The software development plan is mainly focused on collecting the necessary information to initiate, develop, and modify the AkiProPlus project, which was developed to overhaul the existing AkiPro management system from AkiBakery. This overhaul, which tackles issues related to handling invoices, data storage, inventory management, and customer feedback will be governed by the software development plan. Furthermore, this document is divided into several phases which are governed by the needs and deliverables from AkiBakery. To increase the comprehensiveness of the software development plan a section dedicated to definitions, acronyms, and abbreviations will be included. A references section will also be provided to ensure a list of documents referenced throughout the software development plan. To conclude, an overview of the rest of the points of interest in the software development plan is featured.

## Purpose

[Specify the purpose of this **Software Development Plan**. The text below is provided as an example**.** ]

This Software Development Plan aims to collect the necessary information to initiate, develop, and modify the AkiProPlus project. This will be executed by: having the development plan function as the top-level plan used by managers to direct the development cycle and by describing the approach to the development of the new management system. It will be used to direct the project's development from the project's inception (requirements planning) to the launch of the AkiProPlus Project.

Members using this plan will be:

* + The project manager
  + Project team members
  + Company Stakeholders

## Scope

[A brief description of the scope of this **Software Development Plan**; what Project(s) it is associated with and anything else that is affected or influenced by this document. The text below is provided as an example.]

*Justification*

The scope of the Software Development Plan is divided into several phases. These phases are governed by the needs and deliverables obtained from AkiBakery. The needs of the business require an in-depth automated system capable of strengthening core business processes that enable production and distribution.

*Objectives*

Primarily the intended system seeks to rectify issues pertaining to invoicing, data storage, inventory management, and customer feedback. Invoices should be automated and dynamic. To be specific, the system will allow remote use to facilitate purchase order changes in real time. The whole premise of the intended system is efficiency. The goal is optimize the core processes in order to better facilitate executive decision making.

The Objectives can be classified as followed:

Specific

Measureable

Attainable

Realistic

Time Bound

*Description*

The end product of the system will satisfy all functional requirements of AkiBakery.These requirements include automated billing and procurement, CRM features, data storage, data recovery and some measure of interoperability.

This *Software Development Plan* describes the overall plan to be used by the <project name> project, including deployment of the product. The details of the individual iterations will be described in the Iteration Plans.  
The plans as outlined in this document are based upon the product requirements as defined in the *Vision Document*.

## Definitions, Acronyms, and Abbreviations

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the **Software Development Plan**. This information may be provided by reference to the project’s Glossary

* + **AkiPro**: The original management software system used by AkiBakery which will eventually be replaced by the new management system *AkiProPlus*.
  + **AkiBakery**: The name of the bakery using the AkiPro management system.
  + **Activity Diagrams**: Diagrams that tend to show the general flow of logic that has to interact with various activities of the system.
  + **RDBMS**: Relational Database Management System.
  + **Actors**: Usually external entities that are able to interact with the system
  + **Use-Cases:** Written use-cases generally shows how a system’s behavior reacts under various conditions based on the requests from clients.
  + **Entity-Relationship Diagram (ERD):** A detailed view or representation of an ER-Model.
  + **Data-Flow-Diagram (DFD):** A diagram depicting the movement of data between subsequent processes, data stores and entities which are generally external.
  + **Level-0 diagram:** A diagram that represents all of the major systems processes at a much higher level.
  + **Context Diagram:** Shows the systems boundaries, external entities and the major flows between the system and entities of an organization.
  + **Gantt chart:** A Gantt chart is generally a pictorial representation of a project that shows the different tasks that are to be completed within a specific time period.
  + **Work-Break-Down Structure:** This can be defined as incrementally chopping the project into smaller modules so that it can be used more efficiently.
  + **Trigger:** Triggers are basically events that initiates the beginning of a use case.
  + **Rational Unified Process (RUP):** A development methodology which consists of four phases: inception, elaboration, construction and transition.

See the Project Glossary.

## References

[This subsection provides a complete list of all documents referenced elsewhere in the **Software Development Plan**. Identify each document by title, report number if applicable, date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.

For the **Software Development Plan**, the list of referenced artifacts includes:

* RUP for Small Projects Website
* Iteration Plans
* Development Case
* Vision
* Glossary
* Any other supporting plans or documentation.

## Overview

[This subsection describes what the rest of the **Software Development Plan** contains and explains how the document is organized. The text below is provided as an example.]

This Software Development Plan contains the following information:

* + **Project Overview** — provides a description of the project's purpose, scope, and objectives.  It also defines the deliverables that the project is expected to deliver.
  + **Project Organization** — describes the organizational structure of the project team.
  + **Management Process** — explains the estimated cost and schedule, defines the major phases and milestones for the project, and describes how the project will be monitored.
  + **Applicable Plans and Guidelines** — provides an overview of the software development process, including methods, tools and techniques to be followed.

# Project Overview

## Project Purpose, Scope, and Objectives

## Project Purpose, Scope, and Objectives

The AkiProPlus project aims to primarily overhaul existing functions (business processes) of the current AkiPro management system and introduce new functions to improve the effectiveness of the improved management system over the previous one. The project ultimately must be able to fully replace the previous AkiPro management system for a new and improved one.

Below in this section is a bullet point list of the functional and non-functional requirements each entity of the new system must be able to fulfill in order for the project to be a success. Some of these functional requirements contain deliverables that are integral to its corresponding functional requirement. These entities include: customers, clerks, the AkiProPlus management system, van drivers, merchandisers, and suppliers.

***Functional Requirements***

**Customers:**

* Customers must be able to gain access to their resources through authentication via a username and password.
* Customers must be able to request for a change of their password if they happen to forget it.
* Customers must be able to request a standing order or order.
* Customers must be able to check their invoices and balances.
* Customers and must be able to cancel orders or standing orders via the system's interface.
* The customer must be able to make phone orders, orders via the use of AkiProPlus web interface and also order via cash payments.
* Customers must be able to request modifications to their standing orders or orders.

**Clerks:**

* Clerks must be able to approve or disapprove a customer's request to make a modification to a standing order.
* Clerks must be able to generate an order based on a customer request for an order.
* Clerks must be able to receive purchase orders from suppliers to either accept or decline a purchase order.
* Clerks must be able to financially clear purchase orders for the suppliers to ship their products to the bakery.
* Purchase orders from the suppliers must be stored in the central database of the AkiProPlus management system in order to keep check of the ingredients the business currently have in stock .

**The AkiProPlus management system:**

* The AkiProPlus management system must be able to provide a customer a new password if the customer happened to forget it.
* The AkiProPlus management system must make provisions for the clerks so they can adjust orders in case there are changes to be made, example return of items and discrepancies in orders.
* **The AkiProPlus management system must provide the bank and suppliers with documents including reports about ...**
* The AkiProPlus management system must provide options for the customer to check the status of their orders made and view their respective balances.
* The AkiProPlus management system must provide the customer with feedback containing information about if the order made was successful and provide options in which transactions can be made via a credit or debit card.
* The AkiProPlus management system must be able to differentiate between the different types of customers who are categorized as daily, weekly and monthly paid customers.
* Once a standing order is approved by a clerk, the AkiProPlus management system must generate a list of the various orders that must be carried out for that particular day.
* The system must be able to update inventory data regarding inventory replenishment based on received goods from suppliers via received purchase orders.
* The AkiProPlus management system must be able to store relevant data of recipes and inventory.
* The AkiProPlus management system must be able to create financial graphs on a timely basis (in real-time).
* The AkiProPlus management system must be able to create weekly reports outlining the status of the inventory.

**Van Drivers:**

* Van Drivers must be able to upload a localized copy of the database management system on their company tablets.
* Van Drivers must be able to synchronize the localized copy of the database management system on their company tablets to the main database management system.
* Van drivers must be able to print invoices.
* Van Drivers must be able to request a copy of a list with the respective orders that must be carried out for a particular route from the system which will be stored their tablets.

**Merchandisers:**

* Merchandisers must be able to upload a localized copy of the database management system on their company tablets.
* Merchandisers must be able to synchronize the localized copy of the database management system on their company tablets to the main database management system.
* Merchandisers must be able to make changes on standing orders using their company tablets.

Merchandisers must be able to indicate to the AkiProPlus management system if certain products are not selling as well as forecasted.

**Management (Department):**

* The Management department must be able to analyze the real-time created financial graphs created by the AkiProPlus management system to assist in decision-making business processes.
* The Management department must be able to access weekly reports outlining the status of the inventory to make the required purchase orders to suppliers for inventory replenishment.

**Suppliers:**

* The Supplier must come with a purchase order to AkiBakery to be accepted and payed by a clerk
* The Supplier must be able to replenish the inventory once the associated purchase order has been accepted and financially cleared.

***Non-Functional Requirements***

The system must be able to fulfill the following non-functional requirements:

* **Shutdown/Crash Recovery Mechanism**: The system must be able to recover from an unexpected shutdown or system crash.
* **Redundant power feeds during power failure**: The system must be able to continue operating during a power outage using redundant power feeds for a minimum of 2 days.
* **User Friendliness**: The system must have a user friendly UI.
* **Maintainability**: The system must be maintainable by database administrators and other qualifiable staff.
* **Adequate security measures**: The system must have adequate means of security (only authorized users have access to certain functions).
* **Cross-platform interoperability**: The system must allow for seamless use between all platforms and devices.

## Assumptions and Constraints

[A list of assumptions that this plan is based and any constraints, for example. budget, staff, equipment, schedule, that apply to the project.]

**Contraints**:

* + The project must be completed within a 12 week time period
  + The system must allow staff and customers to be able to have access to the system via any device eg. Tablets, Pc's, Mobile devies etc.
  + The tablets which will be given to van drivers should be at a low - cost in order to maintain a profitable state.

**Assumptions:**

## Project Deliverables

[A list of the artifacts to be created during the project, including target delivery dates. The text below is provided as an example.]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Deliverable Number | Deliverable Title | | Deliverable Date | |
| 1 | | Software Development Project Plan | | 11/2/2014 | |
| 2 | | Requirements Specification | | 15/01/2014---11/2/2014 | |
|  | | Analysis [DFD’s, ERD’s,User interface] | |  | |
| 3 | | Architecture Specification | | 15/2/2014--- 21/2/2014 | |
|  | | Component/Object Specification | | 15/2/2014--- 21/2/2014 | |
| 4 | | Test Plan | | N/A | |
| 5 | | Final Product w/ Demo | | N/A | |

Deliverables for each project phase are identified in the Development Case. Deliverables are delivered towards the end of the iteration, as specified in section *4.2.4 Project Schedule*.

## Evolution of the Software Development Plan

[A table of proposed versions of the **Software Development Plan**, and the criteria for the unscheduled revision and reissue of this plan. The text below is provided as an example.]

The *Software Development Plan* will be revised prior to the start of each Iteration phase.

# Project Organization

## Organizational Structure

[Describe the organizational structure of the project team, including management and other review authorities.]

A layout of the team’s organizational is depicted below. This diagram will show where each member is positioned along with their primary responsibilities. Responsibilities can be changed periodically by the team leader depending on constraints that must be met. For an in depth detailed description of the member’s roles and responsibilities refer to section 3.3.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## External Interfaces

[Describe how the project interfaces with external groups. For each external group, identify the internal and external contact names. This should include responsibilities related to deployment and acceptance of the product.]

**Merchandisers** :

* + Will be able to monitor stock movements provided that all products have been distributed.
  + Will be allowed to make changes to standing orders once a product is not selling as it should.
  + Are allowed to monitor reports, data trends and also figure analysis using an Electronic Point of Sale System (EPoS). Using the EPoS system merchandisers will have the ability to detect or spot areas where the demand on certain products lie.
  + Using the provided technology Merchandisers can calculate which selling product has the most profit and even monitor customers purchasing habits which they will then input all this data into the system so AkiPro Plus can make changes which will improve the overall reliability of the business.

**Van drivers:**

* Are allowed to utilize the route information from the tablet to deliver goods to the relevant customers.
* Will be allowed to do on demand sales, this type responsibility belongs to the van driver exclusively.

http://www.retailhumanresources.com/job-seekers/retail-job-descriptions/merchandiser-job-description

http://epos.uk.com/what-is-epos.html

## Roles and Responsibilities

[Identify the project organizational units that will be responsible for each of the disciplines,

workflow details, and supporting processes. The text below is provided as an example.]

|  |  |
| --- | --- |
| **Person** | **Rational Unified Process Role** |
| *Manasseh Castello, Project Manager* | Responsible as a:  [System Analyst](file:///C:\process\workers\wk_sysan.htm)  [Test Analyst](file:///C:\process\workers\wk_tstanl.htm)  Conflict manager (for settling conflicts)  [Requirements Specifier](file:///C:\process\workers\wk_ucaut.htm)  [Design Reviewer](file:///C:\process\workers\wk_desrv.htm)  [Test Manager](file:///C:\process\workers\wk_tstmng.htm)  The project manager works closely with the project leader as both these roles are very similar in leadership, communication, and problem solving disciplines.  The project manager has a closer level of communication with the other organizational units for e.g. conflict resolution. The project manager |
| *Lloyd Philbert, Project Leader* | Responsible as a:  [System Analyst](file:///C:\process\workers\wk_sysan.htm)  [Test Analyst](file:///C:\process\workers\wk_tstanl.htm)  Customer Relations Manager  [Requirements Reviewer](file:///C:\process\workers\wk_reqrv.htm)  [Change Control Manager](file:///C:\process\workers\wk_ccmgr.htm) [Requirements Specifier](file:///C:\process\workers\wk_ucaut.htm) [Design Reviewer](file:///C:\process\workers\wk_desrv.htm)  [Test Manager](file:///C:\process\workers\wk_tstmng.htm)  The project leader works closely with the project manager as both these roles are very similar in leadership, communication, and problem solving disciplines. The main difference is that the project leader has the authority to review requirements while the project manager does not. |
| *Johann Blaides, Software Engineer* | Responsible as a: [Designer](file:///C:\process\workers\wk_dsgnr.htm) [Implementer](file:///C:\process\workers\wk_implm.htm) (Coder)  [User Interface Designer](file:///C:\process\workers\wk_uides.htm) [Software Architect](file:///C:\process\workers\wk_archt.htm)  Software Administrator [Code Reviewer](file:///C:\process\workers\wk_codrv.htm) [Integrator](file:///C:\process\workers\wk_syint.htm) [Test Designer](file:///C:\process\workers\wk_tstds.htm)  To a lesser extent also does the following: [Tester](file:///C:\process\workers\wk_tstr.htm) [Technical Writer](file:///C:\process\workers\wk_tchwr.htm)  As the individual responsible for implementing the coding and eventual testing of the main software running the new AkiProPlus management system this individual will have to work closely with the database engineer. This is to make sure the software compenents work seamlessly with the database components of the new management system to avoid inconsistencies in interoperability between the software and database |
| *Jevon Phillips, Database Engineer* | Responsible as a:  [Designer](file:///C:\process\workers\wk_dsgnr.htm) [Implementer](file:///C:\process\workers\wk_implm.htm) (Coder) [Code Reviewer](file:///C:\process\workers\wk_codrv.htm) [Integrator](file:///C:\process\workers\wk_syint.htm)  Database Administrator  [Test Designer](file:///C:\process\workers\wk_tstds.htm)  To a lesser extent also does the following roles:  [Tester](file:///C:\process\workers\wk_tstr.htm) [Technical Writer](file:///C:\process\workers\wk_tchwr.htm)  As the individual responsible for implementing the coding and eventual testing of the database component of the new AkiProPlus management system this individual will have to work closely with the database engineer. |

Anyone on the project can perform [Any Role](file:///C:\process\workers\wk_any.htm) activities.

# 

# Management Process

## Project Estimates

[Provide the estimated cost and schedule for the project, as well as the basis for those estimates, and the points and circumstances in the project when re-estimation will occur.]

## Project Plan

[This section contains the schedule and resources for the project.]

### Phase Plan

[Include the following:

 Work Breakdown Structure (WBS) — optional for small projects

 a timeline or Gantt chart showing the allocation of time to the project phases or iterations

 identify major milestones with their achievement criteria

Define any important release points and demos.]

### Iteration Objectives

[List the objectives to be accomplished for each of the iterations.]

### Releases

[A brief description of each software release and whether it’s demo, beta, and so on.]

### Project Schedule

[Diagrams or tables showing target dates for completion of iterations and phases, release points, demos, and other milestones.]

### Project Resourcing

 [Identify the numbers and type of staff required here, including any special skills or experience, scheduled by project phase or iteration.

 Describe how you will approach finding and acquiring the staff needed for the project.

 List any special training project team members will require, with target dates for when this training should be completed.

 Allocation of costs against the WBS and the Phase Plan.]

## Project Monitoring and Control

 [The following is a checklist of items to consider:

* Requirements Management : Specify the information and control mechanisms which will be collected and used for measuring, reporting, and controlling changes to the product requirements.
* Schedule and Budget Control:Describe the approach to be taken to monitor spending against the project budget and progress against the planned schedule. Describe how to take corrective action when required.
* Quality Control:Describe the timing and methods to be used to control the quality of the project deliverables and how to take corrective action when required. Include techniques, metrics, criteria, and procedures used for evaluation— this will include walkthroughs, inspections, and reviews. Note that this is in addition to the Test Plan, which is not enclosed in the Software Development Plan.
* Reporting and Measurement: Describe internal and external reports to be generated, and the frequency and distribution of publication. Specify which metrics should be collected and why.
* Risk Management: Describe the approach that will be used to identify, analyze, prioritize, monitor and mitigate risks. Include a list of risks and their current status.
* Project Close-out: Describe the activities for the orderly completion of the project, including staff reassignment, archiving of project materials, post-mortem debriefings and reports, and so forth.
* Configuration Management: Describe the process by which problems and changes are submitted, reviewed, and dispositioned. Describe how project or product artifacts are to be named, marked, and numbered, including hardware, system software, Commercial-Off-The-Shelf (COTS), plans, models, components, test software, results and data, executables, and so on. Describe retention policies, and the back-up, disaster, and recovery plans. Also describe how the media is to be retained—online, offline, media type, and format.
* Problem Resolution: Describe the approach to be taken to resolve disagreements with the customer, including how to handle schedule slips, scope, and contractual disagreements.
* Subcontractor Management: Describe how subcontractors will be managed.
* Process Improvement Plan: Describe how the effectiveness of the process will be assessed and improved.

The text that follows is provided as an example.]

**Requirements Management**

The requirements for this system are captured in the Vision document. Requested changes to requirements are captured in Change Requests, and are approved as part of the Configuration Management process.

**Schedule and Budget Control**

Expenses are monitored by the project manager, and reported and assessed monthly. (See Reporting and Measurement below).

The project manager maintains a schedule showing the expected date of each milestone. The line items in the schedule include work packages assigned to individuals. Each individual who is assigned a work package provides %completion information to the project manager on a weekly basis. Changes in the schedule will be escalated to the project sponsors, who will then decide whether to alter scope in order to preserve target completion dates.

**Quality Control**

Defects will be recorded and tracked as Change Requests, and defect metrics will be gathered (see Reporting and Measurement below).

All deliverables are required to go through the appropriate review process, as described in the Development Case. The review is required to ensure that each deliverable is of acceptable quality, using guidelines described in the RUP for Small Projects review guidelines and checklists.

Any defects found during review which are not corrected prior to releasing for integration must be captured as Change Requests so that they are not forgotten.

**Reporting and Measurement**

Updated cost and schedule estimates, and metrics summary reports, will be generated at the end of each iteration.

The Minimal Set of Metrics, as described in the RUP [Guidelines: Metrics](file:///C:\process\modguide\md_metri.htm), will be gathered on a weekly basis. These include:

Earned value for completed tasks. This is used to re-estimate the schedule and budget for the remainder of the project, and/or to identify need for scope changes.

Total defects open and closed – shown as a trend graph. This is used to help estimate the effort remaining to correct defects.

Acceptance test cases passing – shown as a trend graph. This is used to demonstrate progress to stakeholders.

In addition, overall costs will be monitored against the project budget.

**Risk Management**

Risks will be identified in Inception Phase using the steps identified in the RUP for Small Projects activity “Identify and Assess Risks”. Project risk is evaluated at least once per iteration and documented in this table. The risks of the greatest magnitude are listed first in the table.

|  |  |  |
| --- | --- | --- |
| **Risk Ranking (High, Medium, Low)** | **Risk Description and Impact** | **Mitigation Strategy and/or Contingency Plan** |
|  |  |  |

**Configuration Management**

Appropriate tools will be selected which provide a database of Change Requests and a controlled versioned repository of project artifacts.

All source code, test scripts, and data files are included in baselines. Documentation related to the source code is also included in the baseline, such as design documentation. All customer deliverable artifacts are included in the final baseline of the iteration, including executables.

The Change Requests are reviewed and approved by one member of the project, the Change Control Manager role.

Full backups are performed monthly and incrementals are performed nightly.

# Annexes

[Additional material of use to the reader of the **Software Development Plan**. Reference or include any project technical standards and plans which apply to this project. This typically includes the Development Case, plans for infrastructure, and product acceptance. It also typically includes Programming Guidelines, Design Guidelines, and other process guidelines. The text that follows is provided as an example.]

The project will follow the RUP for Small Projects process, as tailored by the project Development Case.

Other applicable process plans are listed in the references section, including Programming Guidelines.