**Design Doc**

|  |
| --- |
| American Video Game Company |
| CRM Proposal |
| Business Vision Document / Business Requirements |

|  |
| --- |
| Aime Thayer  4-20-2018  Version 1.0 |

Contents

[1. Introduction 3](#_Toc512843928)

[1.1. PUrpose Statement 3](#_Toc512843929)

[1.2. Overview of THE PROBLEM 3](#_Toc512843930)

[1.3. Goals and Objectives 3](#_Toc512843931)

[1.4. Prerequisites 3](#_Toc512843932)

[1.5. Scope 3](#_Toc512843933)

[1.1. Environment 3](#_Toc512843934)

[2. Requirements 5](#_Toc512843935)

[2.1. Business Requirements 5](#_Toc512843936)

[2.2. User Requirements 5](#_Toc512843937)

[2.3. Functional Requirements 14](#_Toc512843938)

[2.4. NonFunctional Requirements 16](#_Toc512843939)

[3. SOFTWARE DEVELOPMENT METHODOLOGY 18](#_Toc512843940)

[3.1. Advantages of the waterfall method 18](#_Toc512843941)

[3.2. disAdvantages of the waterfall method 18](#_Toc512843942)

[3.3. Best SUITED 19](#_Toc512843943)

[Design 20](#_Toc512843944)

[3.4. Flowchart 20](#_Toc512843945)

[3.5. GUI 21](#_Toc512843946)

[4. Testing 22](#_Toc512843947)

[4.1. Testing Details 22](#_Toc512843948)

[4.1.1. Create a contact 22](#_Toc512843949)

[4.1.2. update a contact 22](#_Toc512843950)

[4.1.3. delete a contact 23](#_Toc512843951)

[5. Sources 25](#_Toc512843952)

# Introduction

American Video Game Company needs a new customer relationship management (CRM) system in order to maintain current growth. In the following sections, we describe the sales force’s needs, outline the goals and objectives for this tool, propose an architecture, and describe a testing plan.

# PUrpose Statement

This document forms our recommendation for a software solution, a tool that fits American Video Game Company’s needs for a new CRM system for its sales force.

# Overview of THE PROBLEM

The sales force needs a new tool to keep up with growth. American Video Game Company’s sales have been up by 42% in the past two years, and finds itself outgrowing its existing CRM systems. The new solution should provide American Video Game Company a tool to manage reporting, do activity management, track sales, and manage client contacts.

# Goals and Objectives

We will build a new system, which will reflect existing business processes as much as possible, by allowing users to archive information, maintain versions of records, audit, roll back, manage workflows, and so on (as listed in the sections below). The system will be hosted in our server farm, as will the database that backs it. The internal and third-party users who need to use our system will be able to access it by visiting the webpage: crm.americanvideogames.com.

# Prerequisites

Any aspects that need to be in place prior to the design, development, and implementation of the project proposed in this document.

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Prerequisite | Description | Completion Date |
| 1 | n/a | Requirements for the workflow defined – without clear success criteria, the project cannot be successful! | 4/14/2018 |
| 2 | 1 | Requirements in this document approved by leadership | 4/20/2018 |
| 2 | 1, 2 | Budget for this project approved & obtained | 5/01/2018 |

# Scope

This document describes the business and user requirements, functional and non-functional. We describe the environment, software compatibility, and technical details along with a software solution and its architecture. Finally, we provide a testing plan for this software.

# Environment

The system must be compatible with the desktop browsers in use across the company today:

* Chrome and Chromium (latest version)
* Firefox (latest version)
* IE 9 or greater
* Safari 6.0

As well as the following mobile / tablet browsers:

* iOS7 Safari
* iOS7 Chrome and Firefox
* Android 4.0 Chrome

There is an existing active directory server which the system must integrate with.

We expect to use the existing servers in our server farm to host the webpage and the database, and each users’ individual workstation or mobile device to access the new tool.

# Requirements

American Video Game Company needs a new CRM tool to keep up with its growth. Sales have been up by 42% in the past two years, and the company thus is outgrowing the existing CRM systems. The new solution should provide American Video Game Company a tool to manage reporting and related activity, track sales, and manage client contacts, in accordance with the following detailed requirements. We will custom-build a bespoke CRM that meets each of American Video Game Company’s specific needs.

# Business Requirements

There are 2,000 users who will access the new system, of which 500 are expected to concurrently use the system during peak usage times. The system must scale to meet reasonable growth beyond this.

The new tool will:

1. Consolidate client and customer contact information, and all other relevant business info
2. Report and manage activities and interactions with contacts
3. Use role-based access control to allow internal and remote users feature access
4. Manage permissions
5. Enable 3rd party marketing contractors to access the system
6. Track sales and manage sales reporting
7. Integrate with other internal company systems to allow data sharing
8. Be secure, scalable, and easy to use
9. Be enhance-able at reasonable cost, with a known support and maintenance structure
10. Have a clear roadmap for future updates, upgrades, and feature development
11. Have a clear licensing model and defined ownership rights
12. Work with the company’s internal hosting infrastructure or have good reason to do otherwise
13. Be compliant with relevant law, best practices, and regulations
14. Store data in the US and not allow data to leave the US without prior approval

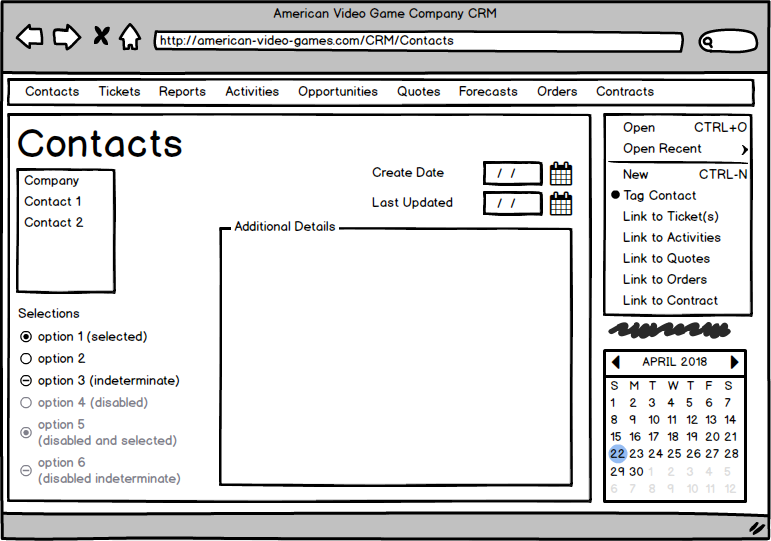
# User Requirements

Specific tasks that the CRM will need to perform include contact management, ticketing, report generation and management, activity management for sales’ visits and meetings, opportunity tracking and management, quote generation and archiving, forecasting, and order & contract management. Some of these requirements are duplicated in the following requirements sections.

Sales has generated the following requirements for contact management:

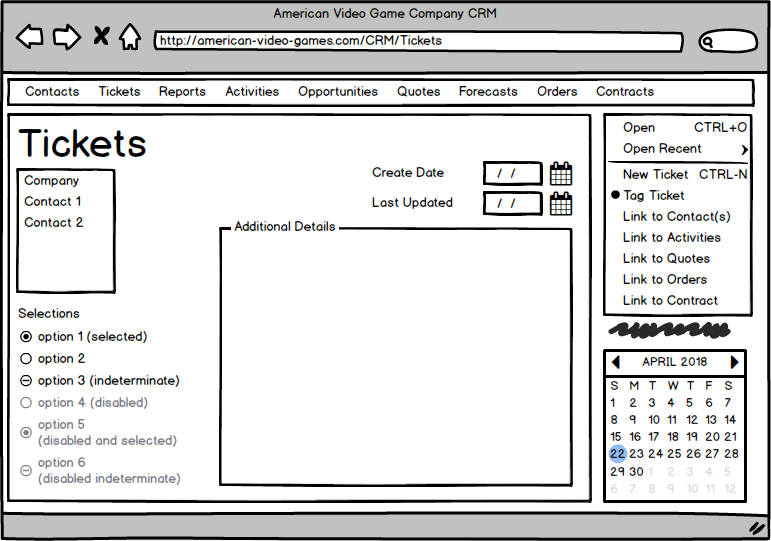
Users need to be categorized into business or end users, which will have a different T&C and preferences on how they like to be contacted, and how their details may be used (e.g. some businesses will want to enter into a strategic partnership and their logo can be used on American’s advertising). The CRM will have check boxes, text boxes, and areas to attach additional documentation to support these needs.

Users need to self-serve, for example to set their own contact settings, marketing preferences, and so on – whether it’s when they sign on with American or during a contact from a team member.



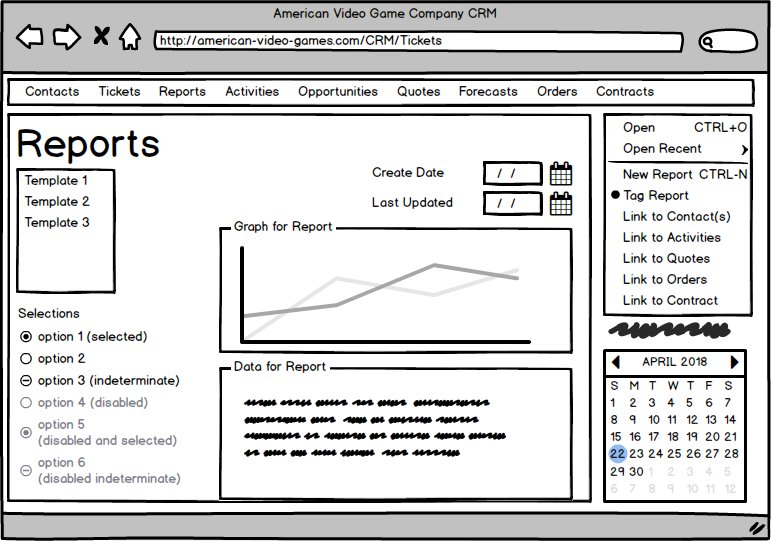
Sales has generated the following requirements for ticketing:

The ticketing system will keep our contact information, by allowing the sales force to enter and track every communication and inquiry, on a per-contact basis. The ticket will include form fields for who called, reason for calling, time, date, follow-up, and relevant details. Email replies are picked up by the system and also tracked on a per-contact basis.



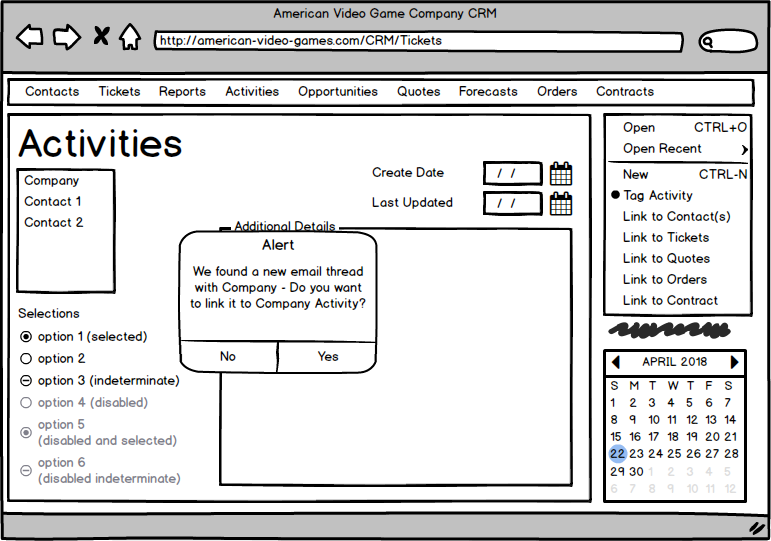
Sales has generated the following requirements for reporting:

The CRM system delivers predefined and user-defined reports on any and all data it holds. The user interface will be easy to use, and allow filtering, formatting, querying, and exporting data. The reporting will allow for detailed and not-so-granular reports. It will allow for the creation of dashboards, creation of historical data reports, and templates for executive-level summaries. Users may click through the data in the UI, save it, or save and reuse any filters they’ve applied for future use.



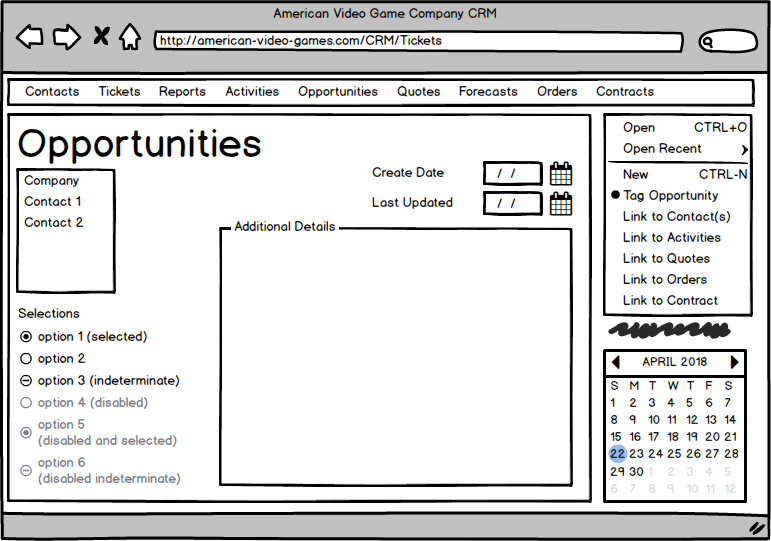
Sales has generated the following requirements for activity management for visits & meetings:

The system acts as a centralized repository for records of our meetings and visits with stakeholders. It tracks emails, and acts as a ticketing system. Meetings and visits may be tagged to specific employees.



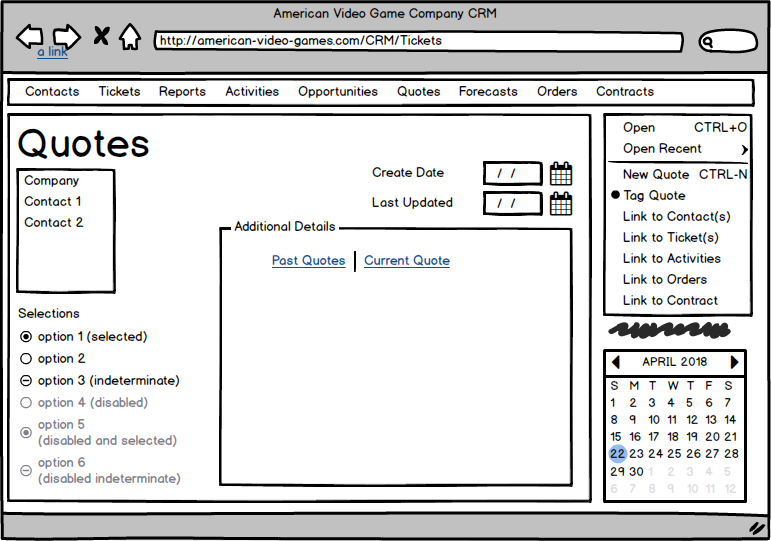
Sales has generated the following requirements for opportunity management:

The CRM gives sales the ability to track processes, manage pipelines, and analyze the competition. Thus, it will implement workflow management for all sales-related processes. It will create and track pipelines. And it will allow users to perform various competitive analyses, such as win/loss, competitive, competitive product, and discount approval analyses.



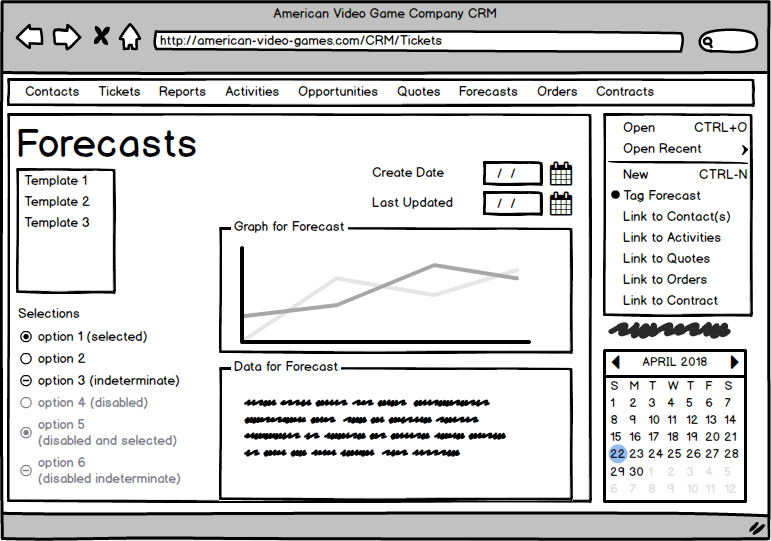
Sales has generated the following requirements for quoting:

The system will generate and assemble quotes; manage discounting, tax, and freight costs; track currency exchanged; ensure each document that needs one, has an electronic signature; configure price quotes and prices; manage substitutions; manage inventory; forecast shipping; manage contract pricing; manage the price catalog; and deliver quotes.



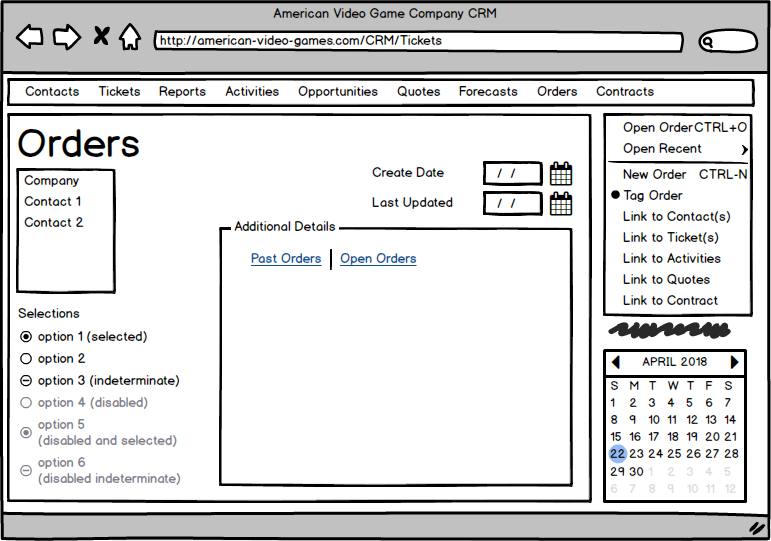
Sales has generated the following requirements for forecasting:

The CRM system will help them by including the following actions: manage foreign and domestic currency adjustments; allow baselining; create and update manager adjustments; upsides; machine forecasting; show sales distribution; create and display forecast periods; and forecast products.



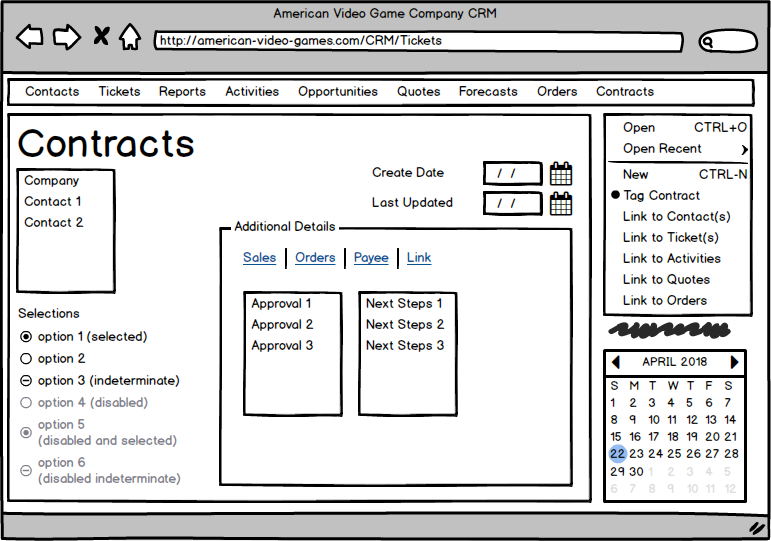
Sales has generated the following requirements for order management:

The CRM must create and track orders, allow sales to take orders, convert quotes to orders, and re-order.



Sales has generated the following requirements for contract management:

The CRM must track the process or creating, signing, and approving contracts. It will track terms, allow for approvals / authorization of contracts, and manage contract termination.



# Functional Requirements

User accounts must comply with role-based access control. The system must accurately control data access, work flows, and editorial control based on these roles, or permissions granted by the administrator. We can use the existing active directory server for authentication and authorization.

Administrators must have the ability to manage permissions by creating roles or granting individual users access to particular actions. Again, we can use the existing active directory server here, to save time and resource cost for this requirement, in relation to the rest of the system that we are building.

In addition to role & permissions management, the system must:

* Write-once, record activity by user identifier, and implement versioning i.e. archive information without deleting older records, in order to maintain history and allow audits, workflow management, and graceful roll-backs. We can implement this in the database, by turning on audit logs, and appropriately designing the database schema
* Implement “soft deletes” for most actions, and “hard deletes” (restricted to specific roles/permissions). We can do this in the orchestrator, by following the appropriate permissions in order to update relevant database records
* Be testable and scalable, with a clear support and maintenance structure. We can ensure this via the correct architecture for scale, and using our standard operating procedures for software launches
* Have a clear roadmap for future updates, upgrades, and feature development. This design doc is a first step in that direction.
* Have a clear licensing model and defined ownership rights. This will be easy, since we are building our own software.
* Store data in the US, and process, analyze, and share that data within the US only (unless otherwise approved). Since our servers are located in our offices, and our offices are in the US, we meet this requirement, but should pay attention to it if we ever expand internationally.
* Permissions and approval workflow must be well-defined. The requirements in this design doc are a first step in that direction.

Specific requirements for each workflow include:

***Contact Management***

Forms must dedupe existing company names and contacts, recommend merging similar-sounding names with existing entries, and can be tagged or categorized by type. This will keep the data ‘clean’ and in a logical hierarchy, making it easier for sales reps to find contacts as well as allowing users to self-serve. Businesses can have one or more contacts, offices, subcompanies, contracts, etc. Each contact (whether assigned to a business or not) will have one or more roles and permissions assigned to them, which may vary due to their job title, company, office location, etc. and can move or change at any time.

A contact does not need to remain assigned to one company and can be assigned to multiple companies, offices, roles, etc. as needed. Each form field has a recommended type and format, and is verified against the recommendation when creating or editing a business or contact. Partial entries are allowed, but will get flagged for review so they will be completed as appropriate.

During sales, followup, etc calls, users may tag inquiries to business and contact entries, or create a new business or contact and tag it to the existing call activity.

***Ticketing***

The ticketing system will keep our contact information, by allowing the sales force to enter and track every communication and inquiry, on a per-contact basis. The ticket will include form fields for who called, reason for calling, time, date, follow-up, and relevant details. Each ticket is unique. Email replies are picked up by the system and also tracked on a per-contact basis. The database supports all information required to make this workflow happen. The system must maintain an audit trail of all workflows and all database changes.

***Reporting***

The CRM system delivers predefined and user-defined reports on any and all data it holds. The user interface will be easy to use, and allow filtering, formatting, querying, and exporting data. The reporting will allow for detailed and not-so-granular reports. It will allow for the creation of dashboards, creation of historical data reports, and templates for executive-level summaries. Users may click through the data in the UI, save it, or save and reuse any filters they’ve applied for future use.

***Activity Management for Visits & Meetings***

The system acts as a centralized repository for records of our meetings and visits with stakeholders. The system allows one- or two-way communication between itself and MS Exchange/Outlook. The system includes ticketing functionality as described in the ticketing section. It shall export and re-import data for transfer to other internal systems in an efficient manner. It will include appropriate validation and minimize risk. Meetings and visits may be tagged to specific employees.

***Opportunity Management***

The CRM gives sales the ability to track processes, manage pipelines, and analyze the competition. Thus, it will implement workflow management for all sales-related processes. It will create and track pipelines. And it will allow users to perform various competitive analyses, such as win/loss, competitive, competitive product, and discount approval analyses.

***Quoting***

The system will generate and assemble quotes; manage discounting, tax, and freight costs; track currency exchanged; ensure each document that needs one, has an electronic signature; configure price quotes and prices; manage substitutions; manage inventory; forecast shipping; manage contract pricing; manage the price catalog; and deliver quotes.

***Forecasting***

Sales managers look to forecasts to help them understand revenue expectations and predict profitability. The CRM system will help them by including the following actions: manage foreign and domestic currency adjustments; allow baselining; create and update manager adjustments; upsides; machine forecasting; show sales distribution; create and display forecast periods; and forecast products.

***Order Management***

The CRM must track orders, create orders, allow sales to take orders, convert quotes to orders, re-order, all in a self-serve portal.

***Contract Management***

Since every deal needs a contract, contract management is critical to the sales force. The CRM must track the process or creating, signing, and approving contracts. It will track terms, allow for approvals / authorization of contracts, and manage contract termination.

# NonFunctional Requirements

The tool must maintain the same or better performance as the prior tool. It should have an intuitive user interface, and be generally easy-to-use / user-friendly. It must be easily extended, modified, or integrated with other tools and services in the future, without needing to rely on a single company or team to do so. It should be secure, but that needs to be balanced with usability. It needs to have a clear licensing model, with defined ownership rights. And it must be compliant with relevant law, best practices, and regulations. It also needs to integrate with the existing active directory server.

# SOFTWARE DEVELOPMENT METHODOLOGY

The company has selected the waterfall software development methodology for this project. The waterfall methodology defines a clear path through the following steps: requirements, design, implementation, verification, deployment, and maintenance. Waterfall is a good methodology for well-defined, predictable projects and teams with experience working on similar applications. Unlike Agile process, Waterfall has well-known milestones. The business appreciates the clear deadlines that Waterfall outlines, and can digest the different stages of development with very little additional training on the development process itself. Using something like Agile might be disruptive to the business, if only because it’s so outside “the norm”.

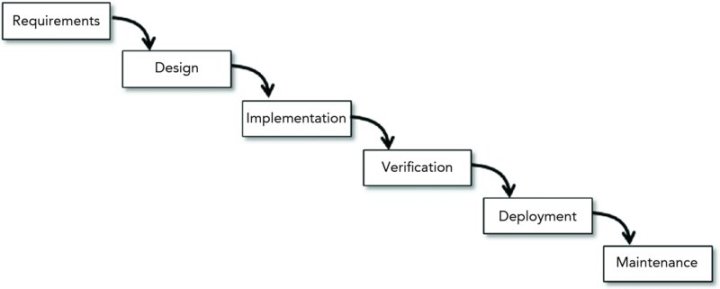


Fig. 12.1, UCertify Beginning Software Engineering, see sources section for details.

# Advantages of the waterfall method

The waterfall methodology provides a clear structure for delivery and allows us to work toward a predictable launch milestone. With architecture and features defined in advance and a clear path through development, we can be successful using Waterfall.

Compare this to Agile, where the freedom to define new processes, flexibility of not following a plan, and expectations of customer collaboration would be too radical for American Video Game Company.

# disAdvantages of the waterfall method

Disadvantages of using waterfall generally include the inability to react to risks and unknowns found later in the project. In specific, we are most concerned about the sales team deciding late in implementation or early in verification that, instead of following established business practices & work flows, that they want to try new features or workflows in the new tool. Waterfall development is rigid and time-based, and doesn’t offer the ability to change requirements late in the game.

Agile would be better suited if we ended up in that situation, because the emphasis on customer collaboration and people over process, even welcoming when requirements change, means that surprises can be dealt with appropriately, with a methodology individually crafted to each risk or surprise as it comes up.

# Best SUITED

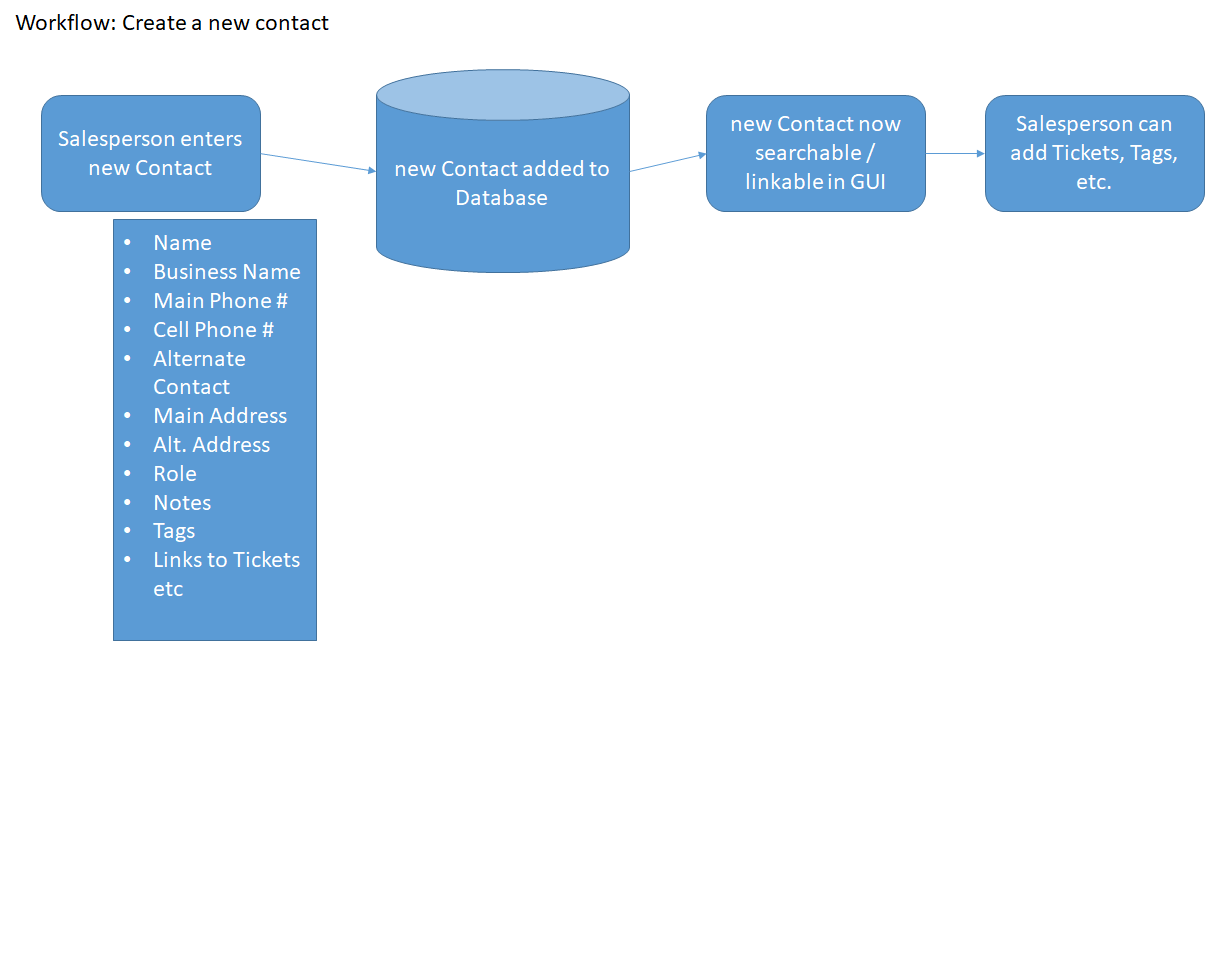
Because the sales team has expressed that they want to follow current CRM workflows with a faster tool, we choose Waterfall methodology over Agile development. We have an experienced team who have worked with similar applications in the past, we do not expect the requirements to change much, and we have plenty of time to do everything sequentially. If we parallelize some tasks, we might even bring the deadline in. This means we have a high chance of success following the clear milestones and structure within the Waterfall methodology.

# Design

The sales force needs a new tool to keep up with growth. American Video Game Company’s sales have been up by 42% in the past two years, and finds itself outgrowing its existing CRM systems. The new solution should provide American Video Game Company a tool to manage reporting, do activity management, track sales, and manage client contacts.

# Flowchart

This workflow shows what happens when a Salesperson creates a new Contact, and outlines potential next steps. We will create additional workflow diagrams as needed.



# GUI

This section intentionally left blank - Please see the customer requirements section for mockups of the GUI.

# Testing

We propose testing the Contact related workflows in the section below.

# Testing Details

We will test the following workflows: create a Contact; query the sales GUI from mobile; hard delete a Contract

# Create a contact

|  |
| --- |
| **Requirement to be tested**  Creating a new Contact |
| **Preconditions: Conditions that must be present before test case can successfully run**  We must have a database set up with the correct schema for the entire set of workflows to test this and future Contact related tests (listed below): Contacts, Tags, Tickets, etc.  We must have the APIs for the GUI for create a Contact. The QA tester does not need to interface with the GUI, but must at least use a command line client to use the API for the Contact test. |
| **Steps: The steps the tester must execute to test the feature.**   1. Gather data for fake Contact and make a separate record of this data 2. Call the new Contact API with the gathered data 3. Confirm successful creation by seeing that the correct fields were filled out in the database (can query db directly) 4. Confirm that the correct data is in each of the expected fields (can query db directly) |
| **Expected results: Expected results and any side effects such as updating a database, writing to a file, etc.**  The fake Contact should have the same number of fields, and the same data in each field, as in the gathered data for that Contact |
| **Pass/Fail: Mark whether the test case passed or failed. The results can be compiled and used to determine if the application is ready for delivery/release.**  Pass |

# Query the sales gui from mobile

|  |
| --- |
| **Requirement to be tested**  Query the sales GUI from mobile |
| **Preconditions: Conditions that must be present before test case can successfully run**  We must have a database set up with the correct schema for the entire set of workflows to test this and future Sales related tests: Contracts, Tags, Tickets, etc.  We must have the GUI available to test. The QA tester needs to interface with the GUI in order to make a query. |
| **Steps: The steps the tester must execute to test the feature.**   1. Create a simple fake Sale – can use the APIs via a CLI to create the test case 2. Now, on the GUI, search for that Sale. Use the search bar at the top and mark if the Sales query was successful 3. This time, make the same search but from the Search menu item, and mark if the Sales query was successful 4. Now, make the search from one of the hotlinks at the bottom of the other tabs, such as the Contract tab (may wish to test multiple tabs). Mark if the Sales query was successful |
| **Expected results: Expected results and any side effects such as updating a database, writing to a file, etc.**  Expect to be able to navigate to a fake Sale using the mobile interface and several navigation paths. |
| **Pass/Fail: Mark whether the test case passed or failed. The results can be compiled and used to determine if the application is ready for delivery/release.**  Pass |

# hard delete a contract

|  |
| --- |
| **Requirement to be tested**  Hard deleting a contract |
| **Preconditions: Conditions that must be present before test case can successfully run**  We must have a database set up with the correct schema for the entire set of workflows to test this and future Contract related tests (listed below): Contracts, Tags, Tickets, Contacts, etc.  We must have the APIs for the GUI for create a Contract. The QA tester does not need to interface with the GUI, but must at least use a command line client to use the API for the Contract test.  We must ensure that detailed audit logs are turned on in the database. |
| **Steps: The steps the tester must execute to test the feature.**   1. Call the delete Contract API with a fake Contract 2. Confirm successful deletion by seeing that the correct fields were removed in the database (can query db directly) 3. Examine the audit logs to see that the deletion was reflected correctly in the db audit logs |
| **Expected results: Expected results and any side effects such as updating a database, writing to a file, etc.**  Expect to be unable to find the test Contract in the db query  Expect to see a record of the deletion and who deleted the Contract in the audit logs |
| **Pass/Fail: Mark whether the test case passed or failed. The results can be compiled and used to determine if the application is ready for delivery/release.**  Pass |

# Sources

Fig 12.1: Waterfall software development image from the course materials from Beginning Software Engineering on UCertify.com, retrieved 4/17/2018, <https://wgu.ucertify.com/?func=ebook&chapter_no=14#02TfW>