Customer Relationship Manager Rewrite

Software Process and Project Management SWEN-256

le group une

Celeste Gambardella
Mallory Bridge
Jacquelyn Leung
Giovanni Melchionne

Professor

Samuel Malachowsky

Artifact 1

Overview, Organizational Structure, and Scope

Problem/Opportunity Statement

After failing to outperform competitors, Sell-IT, the producers of a Customer Relationship Management program, has hired a new CEO and development team. The team has been assigned to rewrite, Sell-IT, to meet the market's expectations and be easier to use. The CRM collects user data to maximize sales and customer retention. The team must have a modular design that can be expanded or updated over time to cater client's needs.

Goals & Project Benefits

Goals

- Outcompete similar CRM products
- Easier use of system
- Optimize workflows
- Modular design
- Provide in-depth analysis of system functions (for business and clients)
- Minimal viable product to be marketed and deployed within one year

Benefits

- Increase clients' satisfaction
- Attract more high-profile clients
- Easy to maintain
- Do not force onto users features that users will not use
- Cater to each client's needs

Stakeholders

Internal

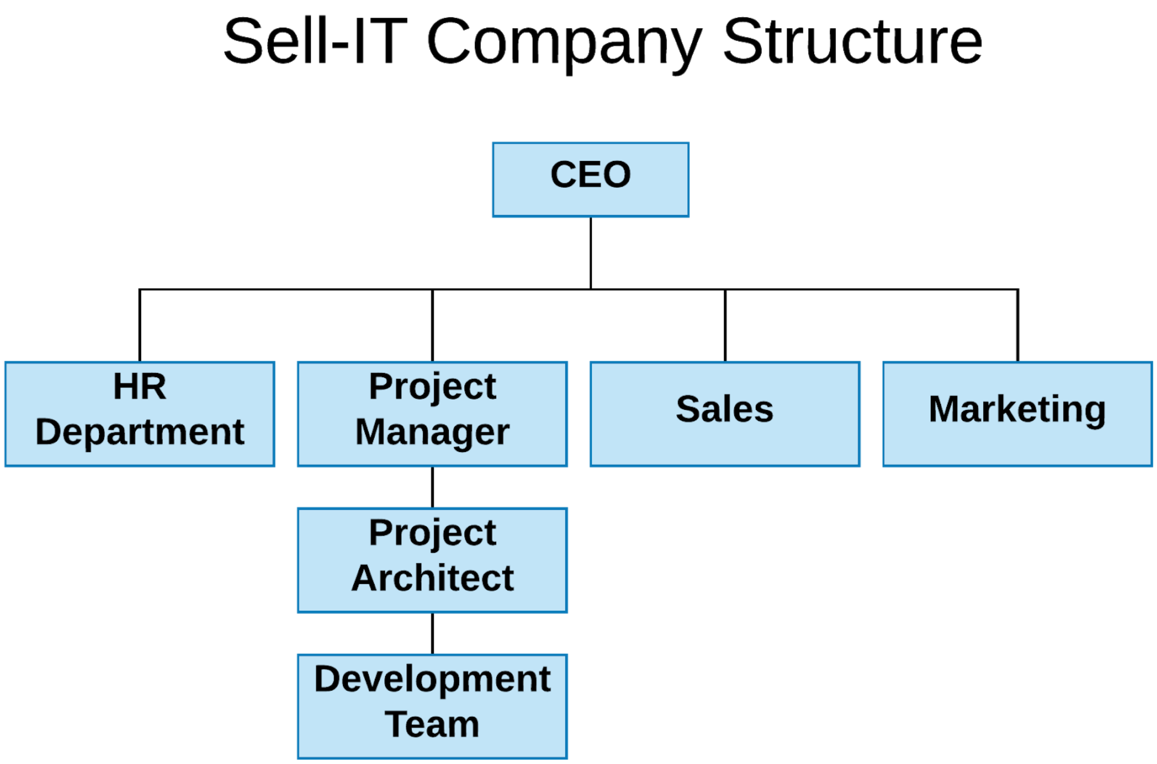
- CEO - Head of company, hired the team, sets goals for Sell-IT
- Development Team - Responsible for development of Sell-IT program
- Marketing - Responsible for branding of Sell-IT, ensuring it looks modern and highlighting features of program

- Sales -Looks for customers for Sell-IT who will hopefully test and later buy the product. They figure out what the customers really want to see

External

- Clients - Those who are going to buy and use the Sell-IT product
- Clients users - The people Sell-IT will be collecting data from
- CRM competitors - Other CRM companies in the market Sell-IT needs to compete with

Team & Organizational Structure



Description

- CEO - The new head of the company. Oversees all departments in the company and sets the overall goal for the Sell-IT product.
- HR Department - Handles the Human Resources of the company. Currently only two people.

- Project Manager - Organizes and manages the development of the Sell-It Product. Communicates with other departments to keep transparency on the product high and relays messages from customers and other departments to the team
- Project Architect - Works closely with the development team on the actual creation of the product. Their job is to make the plan for how the project is structured and make sure the team follows it.
- Development Team - Make the actual Sell-IT program. Currently only five people are employed by the company (not including the project manager and project architect). More developers will be hired as the project expands and eventually there may be multiple development teams.
- Sales - These are the people going out and bringing in customers to test and hopefully buy the Sell-IT product when it is ready. Only three people have been hired so far.
- Marketing - Creates the brand of Sell-IT. They are currently focused on revamping to the Sell-IT brand to be very modern and fresh. Two people have been hired here so far.

Scope

The Project MUST Include (Minimal Viable Product)

- Must use built in integrations with existing enterprise systems
- Multiple workflows:
  - Data collection : data from ad views & clicks, news mentions, social media views, and purchasing / pricing page views
  - Data analysis : Perform and report analysis
  - Marketing tasks : Create, revise, approve/reject, and disseminate marketing plans
- Manage customer and potential customer
- Track purchases of client products & services
- Organize elements [customers, purchased products & services, potential customers] into groups with personalized features : metadata filtering, analytics, events & message notification organized by group & element type
- Integrations with Salesforce, SharePoint, & Oracle Enterprise Resource Planning (ERP) -- (HR) (Legacy)
- Single sign On
- Importing real-time data
- Must use built in integrations with existing enterprise systems

The client would LIKE the Project to include

- Collecting data from website traffic, social media, subsystem to analyse performance, anticipated revenue, un-contracted customers
- Display collected data on dashboard
- Integration of emails

- Data collection from emails
- Optimize workflow for marketing, customer sign on, data retrieval, data analysis, reproting
- Automate workflow for client & clients' customers
- Find best customers for client through analysis

The Project will NOT Include

- Hardset Features that does not allow a modular design
- AI Engine to analyze aggregated data
- Analysis of psychographic factors and demographics
- Proof of concept sign-off, planning, & tracking
- Free version of the software outside of testing/trial period
- Mobile version
- Different pay tiers outside of extending functionality (e.g. remove 10 user limit by paying more, you can have a reasonably large number of users from the get-go)
- Full built-in training seminar

Artifact 2

Methodology Analysis, Risks, and Decision

Methodology Options

Plan-Driven

Team Software Process (TSP)

- supports the development of industrial strength software through the use of team building, planning, and control.
- Strengths:
  - Take ownership of their process and plans to make realistic commitments to the project
  - Consistently using processes and methods they select in order to stay on track
  - Allows team and project managers to have clear visibility of the progress as the team progresses week to week
- Weaknesses:
  - Being artifact centric and high ceremony, or in other terms having many deliverables, meetings, etc.
  - Does not do a great just as scaling for small team and shorter projects
- Meshing:
  - Provides the organization needed for a project like this using artifacts
  - Help by having a team leader or coach that will oversee the project from a managers perspective, which in the end is a great benefit to our stakeholders in the project
- Risk Register

Name	Description	Mitigation
Overestimating time is takes to complete artifacts	Tech team overestimates their capabilities for finishing their requirements which results in the process being delayed	After a project requirement does not meet a set deadline switch the predetermined schedule to allow for more leeway next time.
The Team Leader or Coach leaves	The Team Leader or Coach has to leave for an unknown amount of time due to a family emergency.	The team comes together and discusses who they think is the most qualified to take over as a Team Leader/Coach based on experience and other team

		members' feedback.
Members of the team are not trained in the TSP process	There is a need for a lot of training for TSP to work, and the team's members have not been properly trained before starting the developmental process.	Set clear training guidelines that developers are required to go through. Making sure they clearly understand the type of process we are using. Asking each individual developer how comfortable they are with the process is being used.
Company income much less than expected, leading to project scale decrease	The CEO greatly overestimated the income of the company making it so the MVP was the full project. The ceremony of TSP becomes too much for the small project, and team members get frustrated and lose productivity.	The CEO hires a professional accountant to handle finances, so extra money is reserved to get through low points.

Agile

Feature Driven Design (FDD)

- Blends a number of industry-recognized best practices into a cohesive whole; to deliver tangible, working software repeatedly in a timely manner.
- Strengths
  - Useful for scaling a small agile development to a long-term project with a large team
  - Allows for large projects to be broken down into more manageable chunks easily
  - Provides a process flow designed to deliver working software in a timely manner
  - Client-valued (feature) perspective
  - Requires fewer meetings than Agile methodologies
- Weaknesses
  - Doesn't work well for smaller projects
  - Places high dependence on a chief programmer (acts as a coordinator, lead designer, and mentor)
  - Produces no client-facing documentation
- Meshing
  - FDD defines the following main roles:
    - Project manager
    - Chief architect
    - Development manager
    - Chief programmer

        - Class owner
        - Domain expert
    - These roles would work fairly well with the company and team organizational structures laid out in Artifact 1.
- Risk Register

Risk Name	**Description of Risk**	**Mitigation**
Chief Programmer leaves	The Chief Programmer leaves the company during the development of a product.	Have more than one person familiar with the work done by the Chief Programmer, acting as a backup.
Management switches existing project to FDD	FDD may not work well with existing systems because there is no model to define the system.	Spend time to examine the existing system and create a general model, if possible. Otherwise the project may need to be started over.
Developer goes on vacation	A developer working on a class (part of a feature being currently worked on) goes on vacation.	Have all members of the feature teams be familiar with the classes comprising their feature at a basic level.
Large feature is introduced	A feature is added to the project scope that cannot be completed in its entirety within 2 weeks.	Break down the feature into multiple smaller features.

Scrum

- Scrum is an agile project management methodology or framework used primarily for software development projects with the goal of delivering new software capability every 2-4 weeks.
- Strengths
  - Allows for teams to develop deliverables quickly and efficiently
  - Large projects divided into smaller sprints
  - Works well for fast moving development projects
  - Teams have clear visibility from scrum meetings
  - Adopts feedback from project manager and stakeholders
  - Allows for changes from feedback to be made easier
- Weaknesses
  - Lack of definite end date leads to scope creep
  - Adopting scum to large teams can be difficult
  - Needs experienced team members
  - Quality hard to implement unless goes through aggressive testing process
  - If team member leaves in middle of project, can cause a huge negative impact
- Meshing
  - This methodology allows for the stakeholders to be more involved in the development cycle and allows for high communication between the developers and stakeholders. There will be multiple smaller teams working on different aspects of the CRM to ensure an efficient pace on deliverables.
- Risk Register

Risk Name	**Description of Risk**	**Single Mitigation**
Experienced team member quits to work at a FAANG company in the middle of a sprint.	During the middle of development a senior architect quits the company to join Apple. There is no other person on the team with this developers skill set, which is essential to completing this sprint on time. The developer failed to leave any comments in his most recent changes.	The team promotes a developer to become the senior architect and hires a new developer to the team. While making it mandatory for everyone to document their code before committing it.
Management onboards 10 new development team members	Management onboards 10 new team members that are all going to be managed by the same manager and work together on the CRM project.	The project manager splits up the new team members into smaller teams to work on specific features of the CRM.

	There’s too much ceremony in scrum meetings to be efficient, the team gets frustrated attending 30 minute long daily stand up meetings.	
Management onboards 3 fresh hires with no experience in Agile development	Management hires 3 fresh graduates with no previous experience of scrum or agile development and has only worked on waterfall teams.	The new hires are taught by current developers about scrum and work together to ensure work is being completed in two week sprints.
Team member refuses to attend daily/weekly scrum meetings	A developer refuses to attend daily stand ups and weekly scrum meetings to give his updates and progress on his development.	The project manager speaks with the developer about their performance and lack of attendance to get the developer attending daily standups and scrum meetings.

XP

- Dynamically changing software requirements
- Risks caused by fixed time projects using new technology
- Small, co-located extended development team
- The technology you are using allows for automated unit and functional tests
- Strengths
    - Receive customer feedback at end of each release
    - Able to mold the project exactly to what the customers want based on feedback
    - Useful in small teams
    - Acceptance tests are a necessary part of each story, guarantees functionality
    - Daily meetings keep team members reliable
    - Frequent code reviews ensure a complete understanding of code base throughout project
    - Easily get a functioning product quickly
- Weaknesses -
    - Does not scale well with larger team
    - Harder to integrate with legacy/ outside code
    - Need constant customer communication
- Meshing
    - Does give a start up feel with small, close team
    - Product needs customer feedback so it can fit their needs
    - Can get a functioning product out the door fast

Risk Name	Description of Risk	Mitigation
Two team members are unagreeable with each other and refuse to work together.	Two of the members on the team are always disagreeing with each other during the daily meetings and refuse to code in pairs. This greatly reduces the productivity of them in a pair and the team as a whole because the entire vibe is uncomfortable.	Hold an extensive behavioral interview before a person is hired to make sure their personality agrees with the team.
Highly infectious virus breaks out in the city, causing mandatory quarantine.	Pair programming can no longer happen in the same room. Programming is no longer as efficient, leading to loss progress and development delays.	Create a thorough work-from-home plan that developers are comfortable with so they can remain productive.
During flu season half of the development team becomes too sick to come into work.	Half of the developers are no longer able to come into work due to being sick. The typical pair programming groups can no longer work together, causing a delay in productivity.	Provide hand sanitizer, vitamin C, and a comprehensive health plan to workers to avoid becoming sick or at least recover sooner.
Lack of role definition in XP causes some developers to feel over-empowered.	With no clear role definition, some members of the team decide to make themselves the boss of others, making other team members too mad to do productive work. This delays the progress of the product.	The Product Manager classifies a Team Lead that all the developers can go to with any questions they have. The Team Lead is responsible for updating the teams Product Manager which any concerns developers may be faced with and help provide some team structure.

Selected Methodology: Feature Driven Development (FDD)

We chose FDD because it most closely aligns with Sell-IT’s ideal modular design. Each iteration of the process would allow the team to meet with customers to receive feedback, molding the product to exactly what the market wants. At the end of each iteration working software is delivered, so as soon as the minimal viable product is working, it is available to the customers. FDD also allows for the company to emulate startup culture while being an older company having to integrate with legacy software. Developers do not have to be meeting everyday and looking over each other's shoulder, but still can change their design based on changing requirements rather easily. FDD also makes expansion of the team easy; as the project grows and more customers are acquired the team will need to expand too.

Potential Risks

Risk Name	Description of Risk	Probability	Impact	Risk Exposure	First Indicator	Mitigation 1	Mitigation 2
Chief Programmer leaves	The Chief Programmer leaves the company during the development of a product.	20%	30 weeks	6 weeks	Showing signs of uninterest in work being done, not caring as much as before as compared to when they were first hired.	Have more than one person familiar with the work done by the Chief Programmer, acting as a backup.	The team comes together and discusses who they think is the most qualified to take over as a new Chief Programmer based on experience and other team members' feedback.
Management switches existing project to FDD	FDD may not work well with existing systems because there is no model to define the system.	5%	15 weeks	0.75 weeks	Management starts to complain about the current process taking place in terms of development. Mentioning they want the large projects to	Spend time to examine the existing system and create a general model, if possible. Otherwise the project may need to be started over.	Making sure the system’s code has clear documentation and well structured to allow for developmental adaptability.

					be broken down into more manageable chunks.		
Developer goes on vacation right before critical feature release	A developer working on a class (part of a feature being currently worked on) goes on vacation for two weeks	40%	4 weeks	1.6 weeks	Taking their time completing things as they know they are leaving soon.	Have all members of the feature teams be familiar with the classes comprising their feature at a basic level.	Have all employees give at least a two week notice for vacations longer than three days, and get it approved by upper management.
Large feature is introduced	A feature is added to the project scope that cannot be completed in its entirety within 2 weeks.	70%	3 weeks	2.1 weeks	Stakeholders start to mention that they are looking for a new update to the CRM, that has to be revolutionary.	Break down the feature into multiple smaller features.	Make sure management is communicating with the developers of the stakeholders potential requests so developers can start thinking ahead if they decide to move forward with the new feature update.
Missing /Unclear documentation	Poor Documentation leading to long transitions for new team members.	80%	2 weeks	1.6 weeks	New hires or new team members unsure of current state of project, members taking a long time to read over new code.	Updating Design documentation after each sprint of the project. Conduct code reviews for other team member's code.	Have code reviews take place before anything goes into production in which all code has to have correct documentation.

Company has layoffs and team is cut in half	There is a worldwide pandemic and the company is unable to pay for all of the developers on the team and needs to lay off a lot of developers. The CRM team is cut down to only have four developers versus the eight at the beginning of the project.	1%	52 weeks	0.52 weeks	Travel ban/flight prices decrease, Schools extending spring break, Online classes taking place, Non-Essential businesses forced to shut down	Have a risk assessment plan in place just in case a major crisis occurs and a flexible budget to minimize the number of employees needing to be laid off.	Lots of hand sanitizer and make sure everyone has at least one way to work remotely
There is a security breach within the system	Mismanagement of security keys leads to a large employee data breach.	75%	4 weeks	3 weeks	Team notices loss of data, or other oddity	Make sure to hire an employee with some experience in security	Outsource security to another company before the company is developed enough to have its own.
New CEO mismanages the company	The new CEO is unable to manage the company efficiently. The CEO has extremely high expectations of the new CRM system and team.	5%	20 weeks	1 weeks	CEO doesn’t listen to the developers, struggling to handle everything and manage the company, Can’t communicate with other managers properly.	Discuss with the CEO about concerns with his/her management	Discuss with Stakeholders and investors about the CEO’s mismanagement of the company to find a new CEO.

Artifact 3

Estimation and Scheduling

Work-Breakdown Structure (WBS)

- 1. Sell-IT Customer Relations Management Program
- 2. Create Concept
  - 2.1. Gather Requirements
    - 2.1.1. Define functional requirements
    - 2.1.2. Define non-functional requirements
    - 2.1.3. Meet with current CRM customers of other products
      - 2.1.3.1. Document most and least wanted features
    - 2.1.4. Meet with previous Sell-IT users for feedback on previous Sell-IT
      - 2.1.4.1. Document most used features
      - 2.1.4.2. Gather general feedback on performance and wants
  - 2.2. Evaluate Previous Version of Sell-IT
    - 2.2.1. Gather information on performance of last system
    - 2.2.2. Document reusable functionality
  - 2.3. Create Risk Management Plan
    - 2.3.1. Generate risks register
    - 2.3.2. Develop plan to mitigate / avoid certain risks
  - 2.4. Create and Approve Project Charter
- 3. Design System
  - 3.1. Create Product Timeline
    - 3.1.1. Define Roadmap
      - 3.1.1.1. Create high-level view of the requirements
      - 3.1.1.2. Meet with developers to discuss their strengths and weaknesses
      - 3.1.1.3. Readjust timeline to fit needs of the developers if possible
  - 3.2. Model the System Architecture
    - 3.2.1. Create a Domain Model
      - 3.2.1.1. Gather the essential language used by CRM Producer
      - 3.2.1.2. Classify main entities to include
      - 3.2.1.3. Meet with stakeholders for feedback on new system model
      - 3.2.1.4. Update Domain Model to meet any critiques by stakeholders
    - 3.2.2. Create a Product Backlog
      - 3.2.2.1. Order the stories by priority
      - 3.2.2.2. Set up planning poker with Developers
        - 3.2.2.2.1. Rank each story
    - 3.2.3. Create a Release Plan

- 3.2.3.1. Define when stories should be completed by
- 3.2.3.2. Readjust timeline if needed
- 3.2.3.3. Update Product Manager
- 3.2.4. Establish Coding Standards
- 4. Develop Product
  - 4.1. Develop Architecture
    - 4.1.1. Set up work environment for development
  - 4.2. Set up Database
    - 4.2.1. Install and set up database software and storage
    - 4.2.2. Create database architecture
  - 4.3. Develop Data Collection Workflow
    - 4.3.1. Develop Ad views & clicks data collection
    - 4.3.2. Develop News mentions data collection
    - 4.3.3. Develop social media views data collection
    - 4.3.4. Develop purchasing / pricing page views collection
    - 4.3.5. Develop purchase of product or service data collection / tracking
  - 4.4. Develop Analysis Workflow
    - 4.4.1. Develop analysis creation from data
    - 4.4.2. Develop report generation from analysis
  - 4.5. Develop Marketing Workflow
    - 4.5.1. Develop marketing plans creation
    - 4.5.2. Develop marketing plans approval / rejection system
    - 4.5.3. Develop marketing plans dissemination process
  - 4.6. Develop Sign-In
  - 4.7. Develop Customer & Sales Management Feature
    - 4.7.1. Develop personal customer grouping
  - 4.8. Develop Integrations
    - 4.8.1. Develop Integration with Salesforce
    - 4.8.2. Develop Integration with SharePoint
    - 4.8.3. Develop Integration with Oracle Enterprise Resource Planning
    - 4.8.4. Develop Email Integration with Outlook
    - 4.8.5. Develop Email Integration with Gmail
- 5. Test Product
  - 5.1. Define Objectives
    - 5.1.1. Gather requirements needing testing
    - 5.1.2. Define the depth of testing
    - 5.1.3. Classify the impact of the requirement being tested
  - 5.2. Develop Tests
    - 5.2.1. Define type of testing to be completed
      - 5.2.1.1. Classify Black Box or White Box Testing
      - 5.2.1.2. Set up the test environment
    - 5.2.2. Create Testing Suite
      - 5.2.2.1. Define testing structure

5.2.2.2. Define Mandatory Coding Standards

5.2.2.3. Classify framework to be used

5.2.3. Gather Testing Metrics

5.2.3.1. Document the code complexity

5.2.3.2. Document the code coverage

5.2.3.3. Document the lines of code

5.2.4. Prioritize requirements needing testing

5.2.4.1. Assign Developer(s) to a testing requirement

5.2.4.2. Create a Testing Timeline

5.2.4.3. Meet with Developers for feedback on given timeline

5.2.4.4. Readjust timeline if necessary

5.3. Document Findings

5.3.1. Create a log of testing completed

5.3.1.1. Define if test failed or passed

5.3.1.2. Define frequency of testing

5.3.1.3. Define the severity of problems found

5.3.1.4. Define who was responsible for testing certain requirement

5.4. Define a Code Review for Developers

5.4.1. Assign Developers to review other teams code base

5.4.2. Define Pass or Fail testing

5.4.3. Gather feedback from Developers

6. Roll Out

6.1. Hold Retrospective Meetings

6.2. Deliver Product to customers’ systems

6.3. Meet with customers to confirm product performance

6.4. Gather information on faults and failures

7. Support / Control

7.1. Generate usage documentation for data collection and analysis

7.2. Update risk management plan

7.3. Validate user requirements

Product Backlog

#	**Points**	**Story**
1	8	As a *User,* I want to *collect data from various websites in real time* so I can *determine potentials for clients*
~~**2**~~	~~5~~	~~As a *User*, I want to *have data collection* so I can *get data from ad views and clicks, new mentions, social media views, and purchasing/pricing page views*~~
3	~~8~~	As a *User*, I want to *have marketing tasks* so I can *create~~, revise, approve, reject, and disseminate~~ marketing plans*
4	13	As a *Client*, I want to *manage my current and potential customers*, so that I can *organize them into personalized groups*.
5	8	As a *Client*, I want to *be able to use email integration* so I can *easily communicate with a large range of customers*
6	5	As a *User*, I want to *easily sign-on customers* so I can *easily manipulate data relating to them*
7	8	As a *User*, I want to *be able to get data analysis*, so that I can *perform and report ~~and~~ different statistics*
8	5	As a *User*, I want to *retrieve data from multiple sources* so I can *easily track related information*
9	13	As a *User*, I want to *integrate with Salesforce* so that I can *have multiple platforms' features*
10	13	As a *User*, I want to *integrate with SharePoint* so that I can *have multiple platforms' features*
11	13	As a *User*, I want to *integrate with Oracle Enterprise Resource planning* so that I can *have multiple platforms' features*
12	13	As a *User*, I want to *integrate with other existing enterprise systems*, so that I can *have multiple platforms' features*
13	20	As a *Manager*, I want to *have parts of the sales process workflow automated* so that we can *be more competitive*
14	20	As a *Manager*, I want to be able to *expand the capabilities of the software* so that we can *increase productivity and successful campaigns*
15	3	As a *User*, I want to *have data collection* so I can *get data from ad views and clicks.*

16	3	As a *User*, I want to *have data collection* so I can *get data from news mentions*
17	3	As a *User*, I want to *have data collection* so I can *get data from purchasing/pricing page views*
18	3	As a *User*, I want to *have data collection* so I can *get data from social media views*
19	3	As a *User*, I want to *have data collection* so I can *get data from* purchase of product or service data collection / tracking
20	5	As a *User*, I want to *be able to create an analysis from data so I can analyze the data*
21	5	As a *User*, I want to *have marketing tasks* so I can *revise, marketing plans*
22	3	As a *User*, I want to *have marketing tasks* so I can *approve or reject marketing plans*
23	3	As a *User*, I want to *have marketing tasks* so I can *disseminate marketing plans*

Sprint 1 Backlog

#	**Points**	**Story／Acceptance Criteria**
1	5	As a *User*, I want to *have data collection* so I can *get data from ad views and clicks, new mentions, social media views, and purchasing/pricing page views*
		1. Product is able to collect data from various website traffics, social media, and the system itself 2. Product is able to analyze marketing performance 3. Product is able to analyze anticipated revenu 4. Product is able to analyze uncontacted potential customers 5. Product is able to forecast data 6. Product must display data in an easy-to-use dashboard
2	13	As a *Client*, I want to *manage my current and potential customers*, so that I can *organize them into personalized groups*.
		1. User is able to organize customers, purchased products/services, and potential customers into groups 2. User is able to personalize groups 3. User is able to add metadata tags to groups

		4. User is able to use metadata for filtering, analytics, and message/event notifications 5. User is able to organize by group and element type
3	13	As a *User*, I want to *integrate with Salesforce* so that I can *have multiple platform's features*
		1. Product is able to integrate with legacy installation of Salesforce 2. Product seamlessly imports data in real-time 3. Product seamlessly exports data in real-time 4. Product is able to access Salesforce features
4	8	As a *User*, I want to *be able to get data analysis*, so that I can *perform and report and different statistics*
		1. Product is able to inspect given data 2. Product is able to evaluate data 3. Product is able to ensure data integrity 4. Product is able to report data findings
5	8	As a *User,* I want to *collect data from various websites in real time* so I can *determine potentials for clients*
		1. Product is able to collect data from websites in real-time 2. User is able to easily access collected data 3. User is able to determine marketing potentials for clients 4. Product dashboard is easy-to-use
Total	47	

Technical Process

Timing

- Sprint 0 would be 2-3 weeks long.
- Sprints would be 3-4 weeks long.
- Total time for MVP development is 16-24 weeks long.
- Total time for Feature development is 16-20 weeks long.
- Testing would be 3-5 weeks long.
- MVP Product will be rolled out after 4-6 months, with update features being introduced every 2-3 months.

Meetings

- Mandatory team meetings will be conducted Mondays, Wednesdays, and Fridays. These meetings are intended to ensure that all members of the team are aware of their responsibilities and how their individual responsibilities are related to other members of the team.
    - A team member will be picked via lottery to supply snacks for the team at each Monday meeting; the expense will be reimbursed through the Expensify application.
- Non-mandatory team meetings may be conducted at the discretion of either the Project Manager, Chief Architect, or Chief Programmer. These meetings are intended to be used in the circumstance that there is an issue, and should only require the involvement of the parties involved.
- Mandatory meetings with upper management will be conducted on the Tuesday of the first and third weeks of the month. These are intended to keep management up-to-date on the project process, as well as to keep them aware of any issues or panacea that have arisen regarding the project process. At least one chart or spreadsheet is required to be presented by the development manager at each of these meetings.

Sprint 0

#	Points	Story
1	5	As a Developer, I want to get together with my team and discuss a Domain-driven Design so that I can develop the product.
2	13	As a Developer, I want to learn the existing CRM product and technologies used so that I can develop the product.

3	5	As a Chief Architect, I want to create a design model with my team so that I can know everyone realizes the expectations I have.
4	13	As a Project manager, I want to create a product backlog so that my team can have stories to start development.
5	13	As a Chief Architect, I want to create the database so that we can collect data.
6	8	As a Chief Architect, I want to set up a development environment so that the developers can start working on the product.
Total	57	

Spikes

#	Points	Story
1	5	As a Developer, I want to get together with my team and discuss a Domain-driven Design so that I can develop the product.
2	5	As a Developer, I want to learn the existing CRM product and technologies used so that I can develop the product.
3	5	As a Chief Architect, I want to create a design model with my team so that I can know everyone realizes the expectations I have.
Total	15	

Epics

#	Points	Story
1	52	As a *User*, I want to *integrate with Salesforce, SharePoint and Oracle Enterprise Resource planning and existing enterprise systems*, so that I can *have multiple platform’s features*
2	21	As a User, I want to *have multiple workflows(data collection, data analysis, marketing tasks)*, so that I can do *all my work in one application*
Total	73	

Artifact 4

Stakeholder Analysis, Communications Plan, and Metrics

Stakeholder Register

Name	**Classification**	**Level of Power**	**How they Impact**	**Level of Interest**	**Artifacts**	**Basic Stakeholder Management Strategy**
CEO	Head of the company	5	Manages the company and hires everyone	5	Schedule, Budget, Forecasts, Risk Register, Customer Satisfaction Metric Report	High-Power and High Interest. The CEO has the most amount of power and interest in the project to manage it with the most amount of care.
Development Team	Develops the product	4	They are making the product	5	Schedule, Risk Register, Forecasts, Progress Report, Lessons Learned, All Metric Reports	Medium-Power and High-Interest. The development team has high interest in the project and does not have as much power as the CEO but has medium power within the project. They should be informed and managed with care.
Marketing Department	Branding the software	2	They are promoting and in charge of the design	4	Schedule, Customer Satisfaction Metric	Low-Power and High-Interest. The marketing team has high interest in the project but low power with the development or say in project requirements. They should be kept informed of all changes within the project.
Sales Department	Looks for customers	2	They are selling and	2	Schedule, Budget,	Low-Power and Low-Interest. The

			speaking directly to customers		Customer Satisfaction Report	sales department doesn’t have much involvement in the project, nor do project details matter to them. They need a basic understanding of the product so they can sell it.
Clients	Customers buying/using Sell-IT	3	They give suggestions and feedback on the product	4	General Progress Report, Customer Satisfaction Report	Medium-Power and High Interest. The client is who the product is for and has some say in the product and what they want to see. They should be kept informed of the progress and treated with care.
Client Users	People Sell-IT will be collecting data from	1	Their information is being collected and they can give feedback	1	Basic,Positive Customer Satisfaction Report	Low-Power and Low interest. The client user uses the product and only should be monitored. They do not need much effort.
CRM	Competitors	3	Need to match them or do better	4	None	Low-Power and High-Interest. There are other CRM companies in the market that Sell-IT competes with. These Competitors do not have a say in what Sell-IT does but does guide the Sell-It company in knowing what customers are expecting from them. The Competitors do have a high interest in what Sell-IT is developing.

Communications Plan

What Needs to be Communicated	Why	Between Whom	Best Communication Method	Responsible for Send	When / How Often
Status of Project	Management & the client need to know how the project is progressing	Chief Programmer, CEO, Client	In Person	Chief Programmer	Both Tuesday meetings every month
New Risk	Management needs to know if a new risk has appeared	Project manager, CEO	In Person	Project manager	On occurrence
Risk Averted	Management needs to know if a risk has been averted	Project manager, CEO	E-mail	Project manager	On occurrence
Addition to Scope	Everyone needs to know if something has been added to the project scope	Client, Chief Architect, Chief Programmer, CEO, Development Manager, Project manager	E-mail	Client	On occurrence
Removal from Scope	Everyone needs to know if something has been removed from the project scope	Client, Chief Architect, Chief Programmer, CEO, Development Manager, Project manager	E-mail	Client	On occurrence
Change to Team	Team members need to know if someone had been added/removed from the team	Project manager, Domain Expert, CEO,	In Person	CEO	On occurrence

		Development team			
Results of Likert Test	Tells the company what parts of the product the customers like / don’t like.	Development team, Marketing Team, Sales Team, Project Manager	In person	Sales Team	After each release (every 2 weeks)
Number of defects	Developers need to know the defects as well as the CEO, Project Manager and Chief architect. This is to ensure that there are little defects on the product and the development team is not developing code that is high in defects.	CEO, Project Manager, Chief Architect, Development Team	In Person	Development Team	First Tuesday meeting of the month

Metrics

Metric Name	**Rationale**	**Measurements**	**Usage**
Number of defects found after a release	With feature driven design, before releasing new updates with new features we want to make sure that we are fixing the defects as well.	After each release, we will allow customers to report defects.	Shows the ability of the team to create quality code and the effectiveness of our testing.
Test coverage of specification	Within our module design we would	IDE automatically calculates the	Make sure that the team is making a

	want to make sure we minimize defects by having a high test coverage.	number of lines of code each test runs through divided by the total number of lines or code in each module.	quality test that covers each base. Modules can be compared to see if higher testing coverage results in fewer defects.
User or customer satisfaction	We are not making this product pro bono, the customer is the main source income for the product and we want to ensure that they are satisfied with the product and will continue to support it.	After each release, we will ask customers and users to fill out a Likert scale test on their satisfaction with the product based on categories: usability, user experience, etc.	We will take the results and use it to drive the design for the upcoming features and bug fixes.
Module Complexity	One of the main requirements wanted by our stakeholders. Will help us make sure we are meeting these needs and improve our customer feedback.	We will incrementally test each module being developed making sure they work smoothly with one another throughout the development process.	Allow our stakeholders to see we are looking to meet their requirements of the software being developed being modular throughout development.