Designing Agile Engineering Frameworks: A Product Management Approach to Developing Customer-Centric Software Platforms

Ridhima Arora[1], Manu Handa[2]

Abstract:
The Software Engineering Plan outlines a comprehensive development roadmap, utilizing agile and Scrumban methodologies to align with product management principles. It prioritizes customer-centric requirements using the Kano and MoSCoW models, ensuring maximum value delivery at each product stage. The plan details effort, cost, and staffing estimates to facilitate resource allocation, supporting a phased development approach with an initial pilot release for product-market fit validation, followed by feature and scalability enhancements. Integrated risk management strategies address schedule, scope, and security challenges, while a targeted MVP launch and profitability aim within two years demonstrate alignment with strategic business objectives. Emphasizing iterative testing, stakeholder feedback, marketing efforts, and strategic partnerships, the plan ensures the product meets evolving customer needs and drives adoption effectively.

Keywords: Agile Methodology, Scrumban, Product Management, Customer-Centric Approach, Requirement Prioritization, Kano Model, MoSCoW Method, Product Lifecycle, Pilot Release, MVP (Minimum Viable Product), Product-Market Fit, Iterative Development, Risk Management, Resource Allocation, Cost Estimation, Feature Scalability, Stakeholder Feedback, Continuous Testing, Marketing Strategy, Strategic Partnerships.

1. Introduction

The purpose of the document is to provide a software company with a Software Engineering Plan that builds on the methodology recommendation they have recently received. This engineering plan is intended for the VP of Engineering and other key stakeholders of the software product to provide a basis for their software development lifecycle. The following report details the engineering estimates, schedules, and additional information related to the roadmap of the software product. Furthermore, the plan is embedded with practices from the recommended software methodology of Scrumban that will help a software company team better reach their business objectives of creating a quality product and bringing the platform to market quickly. The structure of the document begins with a discussion of the project management decisions, which is then followed by the project planning, and then concludes with the project risks. As the software product project may have scope changes during the development, the software engineering plan must be adjusted for future evolutions. The software engineering plan listed in this document serves as a foundation for the software product project with the current project context and requirements in mind.

2. Project Management Decisions

2.1 Requirement Prioritization

The software product Requirements document was broken down and analyzed across various attributes. Additionally, non-functional requirements were added to benefit the quality and longevity of the platform. These characteristics combined with a software company business objectives helped decide which user stories would be prioritized for each release during the software development life cycle.

2.1.1 Attributes

2.1.1.1 Customer value

The attribute is important to differentiate requirements based on the customer needs using the Kano Model[1]. The requirement priority focuses on fulfilling the basic needs of the customer first in Release 1 (High Customer Value), then focusing on the improvements of the features in Release 2 (Medium Customer Value), and then focusing on the performance and scalability of the product in the Release 3 (Low Customer Value).

2.1.1.2 Difficulty/Complexity of implementation

The attribute segregates the requirements/user stories based on the complexity and degree of implementation. The Release 1 plan focuses on non-complex requirements which do not delay the time to market the product. Complex requirements were considered in Release 2 and Release 3 phases of the project.

2.1.1.3 Dependency on the completion of other stories

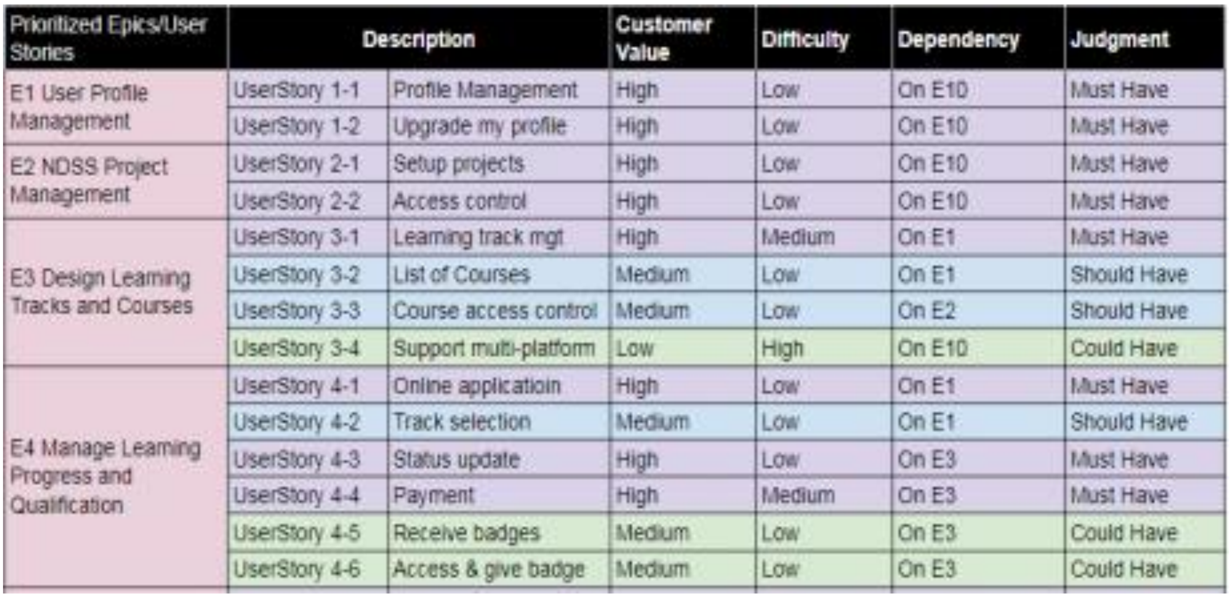
The attribute helps in identifying the requirements which are dependent on other requirements. It gives the sense that until the non-dependent requirement is finished, the dependent requirement cannot begin.

2.1.2 Release Planning (MoSCoW model)

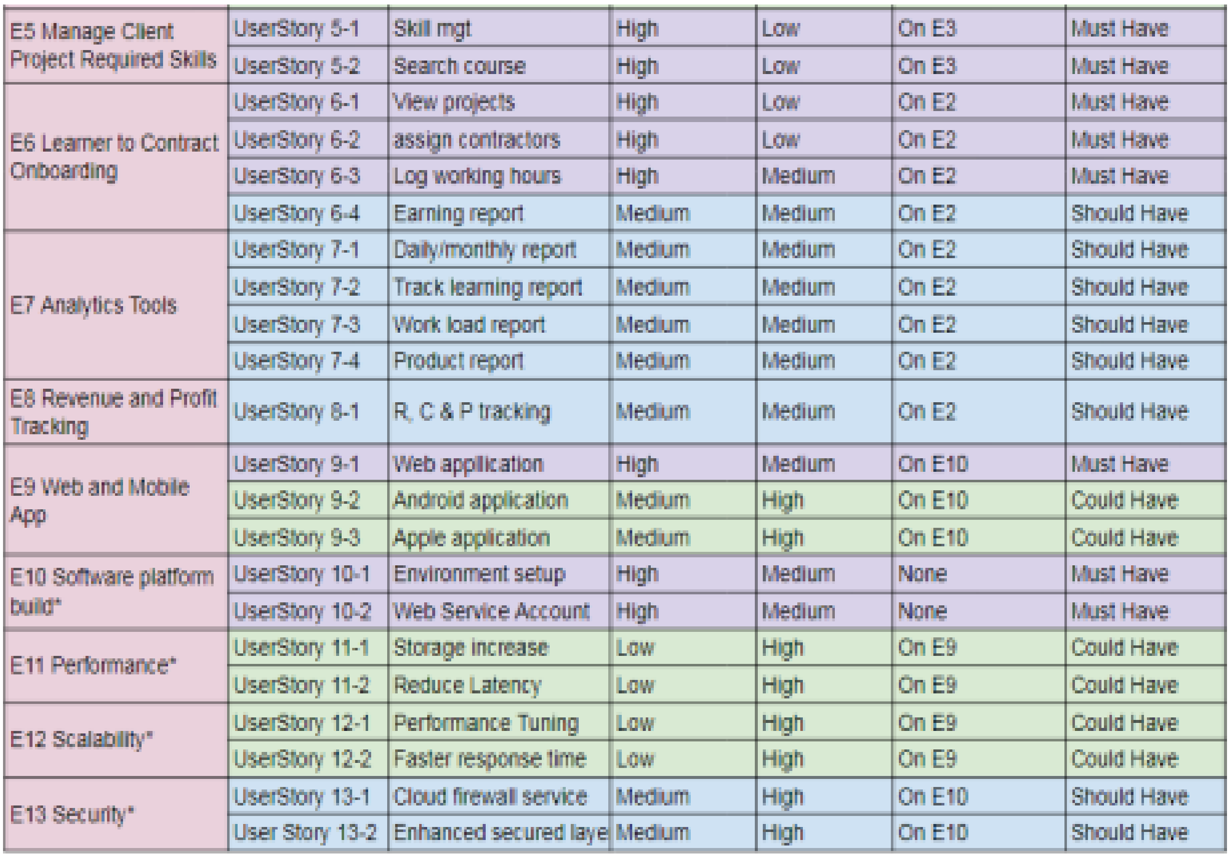
Once the requirements were categorized under the attributes bucket, they were further prioritized using the MoSCoW[2]requirement prioritization method (Judgment column of the below table.)

● "Must-Have" goes into Release 1. The attributes that derive the prioritizations are:
○ Customer Value - High (satisfying the basic customer needs)
○ Difficulty - Low
○ Dependency - Only to High Customer value user stories
● "Should-Have" goes into Release 2. The attributes that derive the prioritizations are:
○ Customer Value - Medium (Improvements on the feature)
○ Difficulty - Low to Medium
○ Dependency - To High/Medium Customer value user stories
● "Could-Have" goes into Release 3. The attributes that derive the prioritizations are:
○ Customer Value - Medium/Low (Performance of the product)
○ Difficulty - Medium to High
○ Dependency - Only to Medium/Low Customer value user stories

[1]"Kano model - Wikipedia." https://en.wikipedia.org/wiki/Kano_model. Accessed 23 Feb. 2019.
[2]"MoSCoW method - Wikipedia." https://en.wikipedia.org/wiki/MoSCoW_method. Accessed 23 Feb. 2019.



Prioritized Epics/User Stories	Description		Customer Value	Difficulty	Dependency	Judgment
E1 User Profile Management	UserStory 1-1	Profile Management	High	Low	On E10	Must Have
	UserStory 1-2	Upgrade my profile	High	Low	On E10	Must Have
E2 NDSS Project Management	UserStory 2-1	Setup projects	High	Low	On E10	Must Have
	UserStory 2-2	Access control	High	Low	On E10	Must Have
E3 Design Learning Tracks and Courses	UserStory 3-1	Learning track mgt	High	Medium	On E1	Must Have
	UserStory 3-2	List of Courses	Medium	Low	On E1	Should Have
	UserStory 3-3	Course access control	Medium	Low	On E2	Should Have
	UserStory 3-4	Support multi-platform	Low	High	On E10	Could Have
E4 Manage Learning Progress and Qualification	UserStory 4-1	Online applicatioin	High	Low	On E1	Must Have
	UserStory 4-2	Track selection	Medium	Low	On E1	Should Have
	UserStory 4-3	Status update	High	Low	On E3	Must Have
	UserStory 4-4	Payment	High	Medium	On E3	Must Have
	UserStory 4-5	Receive badges	Medium	Low	On E3	Could Have
	UserStory 4-6	Access & give badge	Medium	Low	On E3	Could Have



E5 Manage Client Project Required Skills	UserStory 5-1	Skill mgt	High	Low	On E3	Must Have
	UserStory 5-2	Search course	High	Low	On E3	Must Have
E6 Learner to Contract Onboarding	UserStory 6-1	View projects	High	Low	On E2	Must Have
	UserStory 6-2	assign contractors	High	Low	On E2	Must Have
	UserStory 6-3	Log working hours	High	Medium	On E2	Must Have
	UserStory 6-4	Earning report	Medium	Medium	On E2	Should Have
E7 Analytics Tools	UserStory 7-1	Daily/monthly report	Medium	Medium	On E2	Should Have
	UserStory 7-2	Track learning report	Medium	Medium	On E2	Should Have
	UserStory 7-3	Work load report	Medium	Medium	On E2	Should Have
	UserStory 7-4	Product report	Medium	Medium	On E2	Should Have
E8 Revenue and Profit Tracking	UserStory 8-1	R, C & P tracking	Medium	Medium	On E2	Should Have
E9 Web and Mobile App	UserStory 9-1	Web application	High	Medium	On E10	Must Have
	UserStory 9-2	Android application	Medium	High	On E10	Could Have
	UserStory 9-3	Apple application	Medium	High	On E10	Could Have
E10 Software platform build*	UserStory 10-1	Environment setup	High	Medium	None	Must Have
	UserStory 10-2	Web Service Account	High	Medium	None	Must Have
E11 Performance*	UserStory 11-1	Storage increase	Low	High	On E9	Could Have
	UserStory 11-2	Reduce Latency	Low	High	On E9	Could Have
E12 Scalability*	UserStory 12-1	Performance Tuning	Low	High	On E9	Could Have
	UserStory 12-2	Faster response time	Low	High	On E9	Could Have
E13 Security*	UserStory 13-1	Cloud firewall service	Medium	High	On E10	Should Have
	User Story 13-2	Enhanced secured laye	Medium	High	On E10	Should Have

Figure 1: software product Requirement Prioritization

*New Epics/User Stories non-functional/technical stories created which are required as part of Product platform setup, product/feature improvement, and product scalability.

2.2 Project Estimates

2.2.1 Assumptions

Team

- The team will work 8-hour workdays in 40-hour work weeks.
- Management has experience working on large-scale projects.
- Developers have sufficient skill sets to develop the platform.
- The Product Manager and Project Manager will have the knowledge of leading as a Product Owner and Scrum Master respectively.
- The team will have training beforehand to familiarize with Scrumban practices.
- The team's time is split between many a software company projects, but software product is their number one priority in terms of tasking.
- The team will not grow in size during the initial development of software product.
- The development team will be able to leverage open-source libraries, tools, and resources to accomplish development tasks.
- Project
- During planning poker, each story point represents 2 person-days (16 hours).
- The current scope of the project can increase due to new and changing requirements.
- While team members could be working on multiple projects at a software company, their top priority is the software product project at a software company.
- After the first year, the whole team will be retained but will have project commitment reduced to 40%.
- When calculating staffing costs, salary raises will be disregarded.

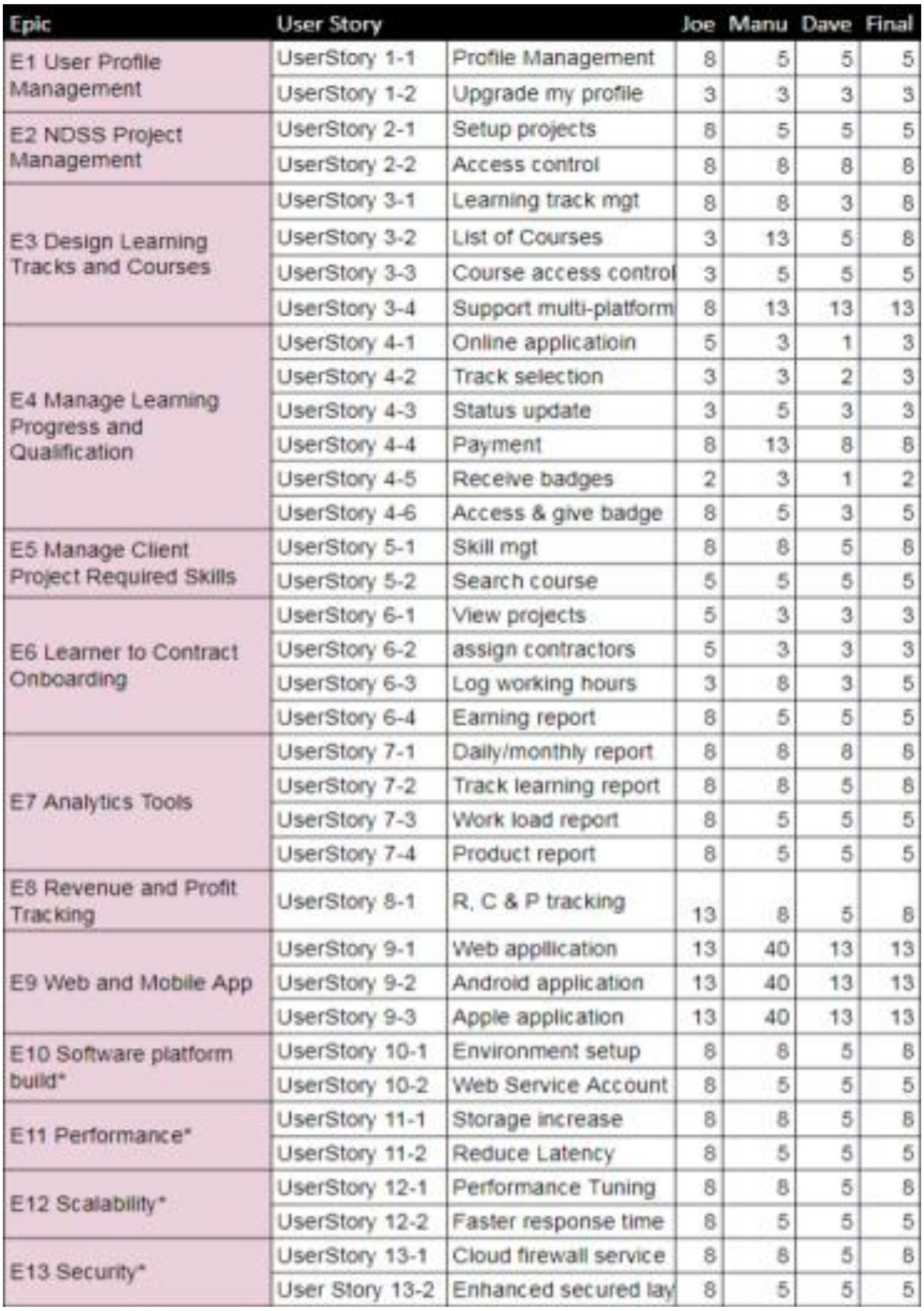
2.2.2 Effort Estimates

2.2.2.1 User Story Estimation

User Stories are estimated using a Wideband Delphi Estimation Technique[3]and Planning Poker Technique[4]to calculate their story points/effort size. Each story point is equivalent to 2 person-days effort.

[3] "Estimation Techniques Wideband Delphi - Tutorialspoint." https://www.tutorialspoint.com/estimation_techniques/estimation_techniques_wideband_delphi.htm. Accessed 23 Feb. 2019.

[4] "Planning Poker: An Agile Estimating and ...." https://www.mountaingoatsoftware.com/agile/planning-poker. Accessed 23 Feb. 2019.



Epic	User Story		Joe	Manu	Dave	Final
E1 User Profile Management	UserStory 1-1	Profile Management	8	5	5	5
	UserStory 1-2	Upgrade my profile	3	3	3	3
E2 NDSS Project Management	UserStory 2-1	Setup projects	8	5	5	5
	UserStory 2-2	Access control	8	8	8	8
E3 Design Learning Tracks and Courses	UserStory 3-1	Learning track mgt	8	8	3	8
	UserStory 3-2	List of Courses	3	13	5	8
	UserStory 3-3	Course access control	3	5	5	5
	UserStory 3-4	Support multi-platform	8	13	13	13
E4 Manage Learning Progress and Qualification	UserStory 4-1	Online applicatioin	5	3	1	3
	UserStory 4-2	Track selection	3	3	2	3
	UserStory 4-3	Status update	3	5	3	3
	UserStory 4-4	Payment	8	13	8	8
	UserStory 4-5	Receive badges	2	3	1	2
	UserStory 4-6	Access & give badge	8	5	3	5
E5 Manage Client Project Required Skills	UserStory 5-1	Skill mgt	8	8	5	8
	UserStory 5-2	Search course	5	5	5	5
E6 Learner to Contract Onboarding	UserStory 6-1	View projects	5	3	3	3
	UserStory 6-2	assign contractors	5	3	3	3
	UserStory 6-3	Log working hours	3	8	3	5
	UserStory 6-4	Earning report	8	5	5	5
E7 Analytics Tools	UserStory 7-1	Daily/monthly report	8	8	8	8
	UserStory 7-2	Track learning report	8	8	5	8
	UserStory 7-3	Work load report	8	5	5	5
	UserStory 7-4	Product report	8	5	5	5
E8 Revenue and Profit Tracking	UserStory 8-1	R, C & P tracking	13	8	5	8
E9 Web and Mobile App	UserStory 9-1	Web appllication	13	40	13	13
	UserStory 9-2	Android application	13	40	13	13
	UserStory 9-3	Apple application	13	40	13	13
E10 Software platform build*	UserStory 10-1	Environment setup	8	8	5	8
	UserStory 10-2	Web Service Account	8	5	5	5
E11 Performance*	UserStory 11-1	Storage increase	8	8	5	8
	UserStory 11-2	Reduce Latency	8	5	5	5
E12 Scalability*	UserStory 12-1	Performance Tuning	8	8	5	8
	UserStory 12-2	Faster response time	8	5	5	5
E13 Security*	UserStory 13-1	Cloud firewall service	8	8	5	8
	User Story 13-2	Enhanced secured lay	8	5	5	5

*New Epics/User Stories non-functional/technical stories created which are required as part of Product platform setup, product/feature improvement, and product scalability.

Figure 2: Story Point sizing using Planning Poker

2.2.2.2 Project Phase Estimation

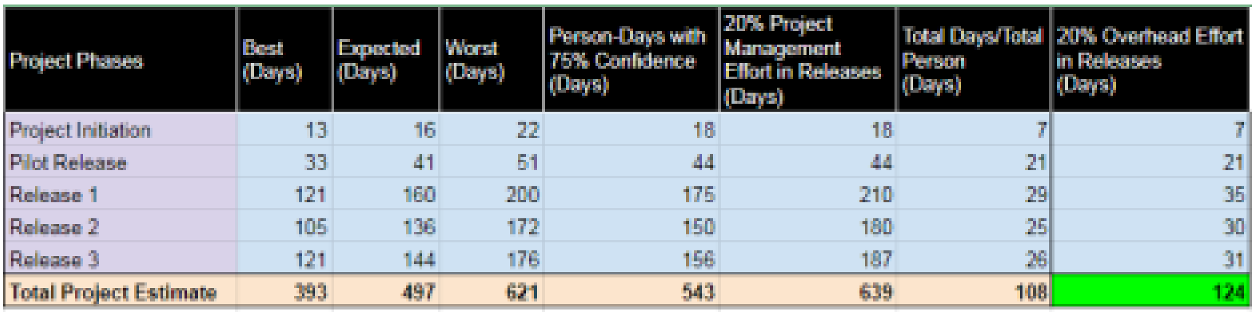
Using the PERT 3-Point estimate technique[5], as shown in Appendix 1, effort estimate is calculated for each phase of the project based on the 75% confidence level resulting in the calculation of total effort of the project which is 124 person-days, as shown in the below table.

Project Effort accounted for Project Initiation time which is for Project Planning, Kick Off meeting, and Project training required for Jr. Developers to make them understand the project landscape and platform creation.

Further, the effort accounts for sizing of Functional requirement specified in the Product requirement document in addition to a technical/non-functional requirement which is not specified in the requirement document.

Additionally, in the calculation of project/phase estimates, 20% of the Project Management effort is accounted in addition to the 20% of the overhead effort for the leave time, meeting, and non-productive time in the three Releases of the project.

Project Initiation and Pilot release phases of the project are time-bound phases and do not require a full team to work in these phases. Hence overhead effort is not accounted in these phases.



Project Phases	Best (Days)	Expected (Days)	Worst (Days)	Person-Days with 75% Confidence (Days)	20% Project Management Effort in Releases (Days)	Total Days/Total Person (Days)	20% Overhead Effort in Releases (Days)
Project Initiation	13	16	22	18	18	7	7
Pilot Release	33	41	51	44	44	21	21
Release 1	121	160	200	175	210	29	35
Release 2	105	136	172	150	180	25	30
Release 3	121	144	176	156	187	26	31
Total Project Estimate	393	497	621	543	639	108	124

Figure 3: Project Effort in person-days

In summary, the Project Effort table displays that in 4 weeks (21 days) the Pilot Release will be launched internally to validate the feasibility of the project (details are covered in the Schedule estimate section of the document.) The product will be launched in the market by Week 11 with basic features of the product followed by the second release by Week 17 with improved features. The final release is planned to release by Week 24 which is 124 days from the beginning of the project.

2.2.3 Staffing Requirements and Cost Estimates

2.2.3.1 Staffing Requirements

To fulfill the software product project development and management, a total of 11 staff members are required. On the software development side, two senior software developers and four junior software developer are needed for full stack software implementation. Senior software developers would help guide the team with practices selected: Emergent Architecture, Interaction Design, Emergent Design, Coding Standards, Test-first programming, Integration Testing and Continuous System Testing. Additionally, junior developers are needed to work with senior developers to implement all required functions.

On the management side, one Product Manager is needed throughout the three releases of the software product project, who is the owner of the product. One Project Manager and one Engineering Manager are needed 50% of the time for project and engineering management.

[5] "Three-point estimation - Wikipedia." https://en.wikipedia.org/wiki/Three-point_estimation. Accessed 23 Feb. 2019.

On the UX & supporting side, a UX Designer is needed full time for the pilot release, and on demand for 1st, second and third release for new UX designs when necessary. Support specialists are not required until the

2nd and third release when the project will have a customer base.

The staffing requirements are summarized in the tabled below, with annual salary reflecting current market values of those positions in Silicon Valley. An additional 30% cost of their annual salary is taken into account as an overhead cost, to account for offices and benefits.

Table 1: Staffing Requirements

[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	$160,000	1	30%
[redacted]	$140,000	1	30%
[redacted]	$170,000	1	30%
[redacted]	$120,000	1	30%
[redacted]	$150,000	2	30%
[redacted]	$90,000	4	30%
[redacted]	$80,000	1	30%

2.2.3.2 Working Days Estimate

Working days estimate are based on "days of work" results for each release from the PERT 3-point estimate. An additional 20% overhead of working days are taken into consideration to account for hours not generating actual work which is seen through meetings and sick leaves. For example, the PERT estimates suggest 25 working days are needed for release 2, and the final working days estimate comes out to be 30 days with 20% overhead accounted for.

For Project and Engineering managers, their working days are only 50% allocated to the project. This is because, in the industry, those positions often manage more than one project in parallel.

UX Designers spend less time post-first release. This is largely because there will be less required design tasks after the pilot.

Support Specialist is not needed until the product is open up to meaningful number of external customers, which is release 2 and after.

A summary of working days estimate is shown in the table below.

Table 2: Staffing Working Days

[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	5	35	30	31
[redacted]	5	18	15	15
[redacted]	5	18	15	15
[redacted]	14	12	12	12

[redacted]	5	35	30	31
[redacted]	5	35	30	31
[redacted]	0	0	12	12

2.2.3.3 Cost Estimate

The final cost estimate is a product of the staff requirement, and their salary estimates with a 30% overhead of salary. This is then used with their specific working days estimate in each release that already accounts for 20% overhead of working hours.

A total of 250 working days per year, which considers about 11-12 paid vacation days, is used to translate the annual salary into per day cost.

With this information, the final cost estimate is shown in the following table.

Table 3: Cost Estimate

[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	$4,160	$29,120	$24,960	$25,792	$84,000
[redacted]	$3,640	$13,104	$10,920	$10,920	$39,000
[redacted]	$4,420	$15,912	$13,260	$13,260	$47,00
[redacted]	$8,736	$7,488	$7,488	$7,488	$31,000
[redacted]	$7,800	$54,600	$46,800	$48,360	$158,000
[redacted]	$9,360	$65,520	$56,160	$58,032	$189,000
[redacted]	$0	$0	$4,992	$4,992	$10,000
[redacted]	$38,000	$186,000	$165,000	$167,000	**$558,000**

2.2.4 Revenue and Profit Estimates

Since our efforts and scheduling estimates portray the software product project will be accomplished within a year, there are certain assumptions that are made in terms of staffing costs. After Release 3, the whole team will be retained but will have project commitment reduced to 40%. This reduced project dedication will allow for maintenance, user experience improvements, and potential development upgrades for the next generation version of the software product platform. By utilizing the average costs of the full releases with high team capacity (Release 1 - 3), a calculation can be made to determine the average costs per week and at a reduced time dedication. The totals costs of release 0 to 3 were added to the estimated cost of the remaining work weeks which helped us better approximate the first year of staffing costs for the software product project. Years 2, 3, and 4 were calculated using the average costs per week at 40% capacity under the assumptions that salary raises are disregarded, and there are 50 work weeks in the year to follow industry averages for vacation time.

Table 4: software product Revenue and Cost for the first four years

[redacted]	[redacted]	[redacted]	[redacted]
1	$960,000	$0	-$960,000
2	$650,000	$900,000	$250,000
3	$650,000	$1,500,000	$850,000
4	$650,000	$2,250,000	$1,600,000

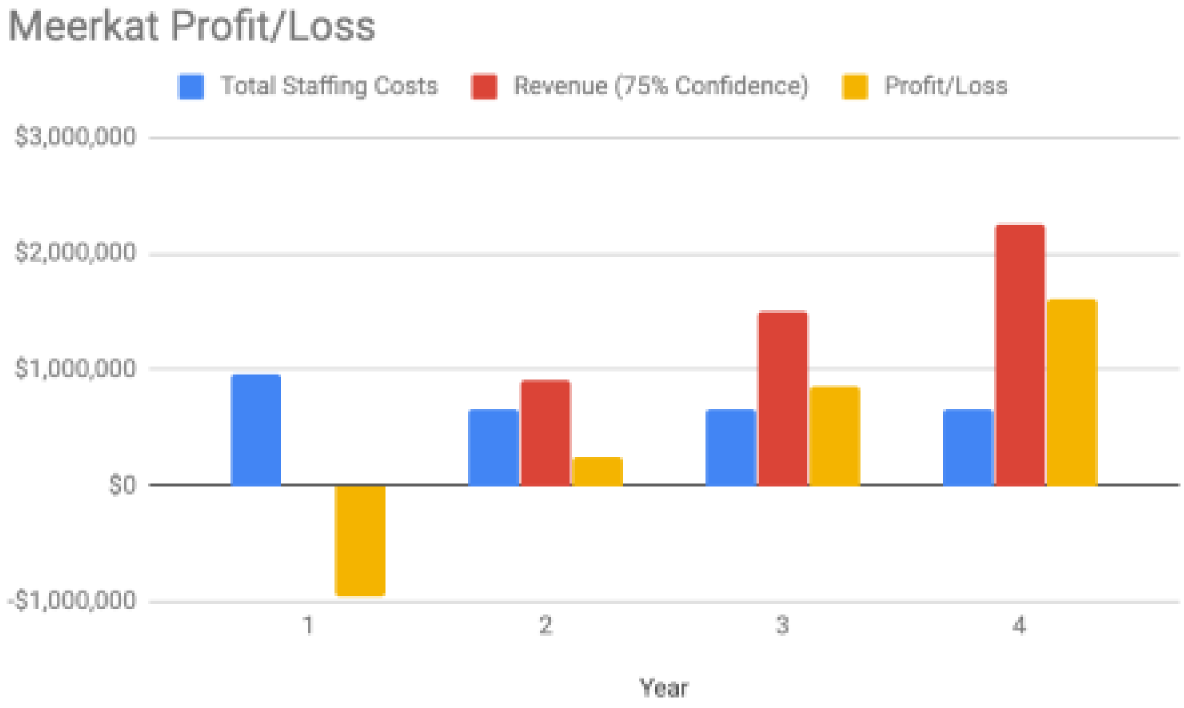


Figure 4: software product Profit/Loss Chart for the first four years

2.2.5 Scope Estimates

2.2.5.1 Pilot Release

Pilot Release takes 14 working days and costs $38,000, as discussed in Effort Estimate, Staffing Requirement, and Cost Estimate section. A demo of the product is built using the Scrumban method in this release as a proof of concept for internal users.

2.2.5.2 Release 1

Release 1 takes 35 working days and costs $186,000, as discussed in the Effort Estimates, Staffing Requirement, and Cost Estimates section. The “must have” functions of the product are built using the Scrumban methodology in this release. With the conclusion of this release, the product is ready for general public users.

2.2.5.3 Release 2

Release 2 takes 30 working days and cost $165,000, as discussed in Effort Estimate, Staffing Requirement, and Cost Estimate section. The “should have” functions of the product are built using Scrumban method in this release. With this release, the product is more feature rich.

2.2.5.4 Release 3

Release 3 takes 31 working days and cost $168,844, as discussed in Effort Estimate, Staffing Requirement, and Cost Estimate section. The “could have” functions of the product are built using Scrumban method in this release. With this release, the product is more secure with multi-platform support.

2.2.6 Schedule Estimates

software product project timeline is divided into five phases of the project, and each phase of the project has its milestone completion after achieving the “Definition of Done.” All the phases of the project are plotted on the Gantt Chart against the time in weeks where W1 corresponds to Week 1, W2 corresponds to Week 2, and likewise.

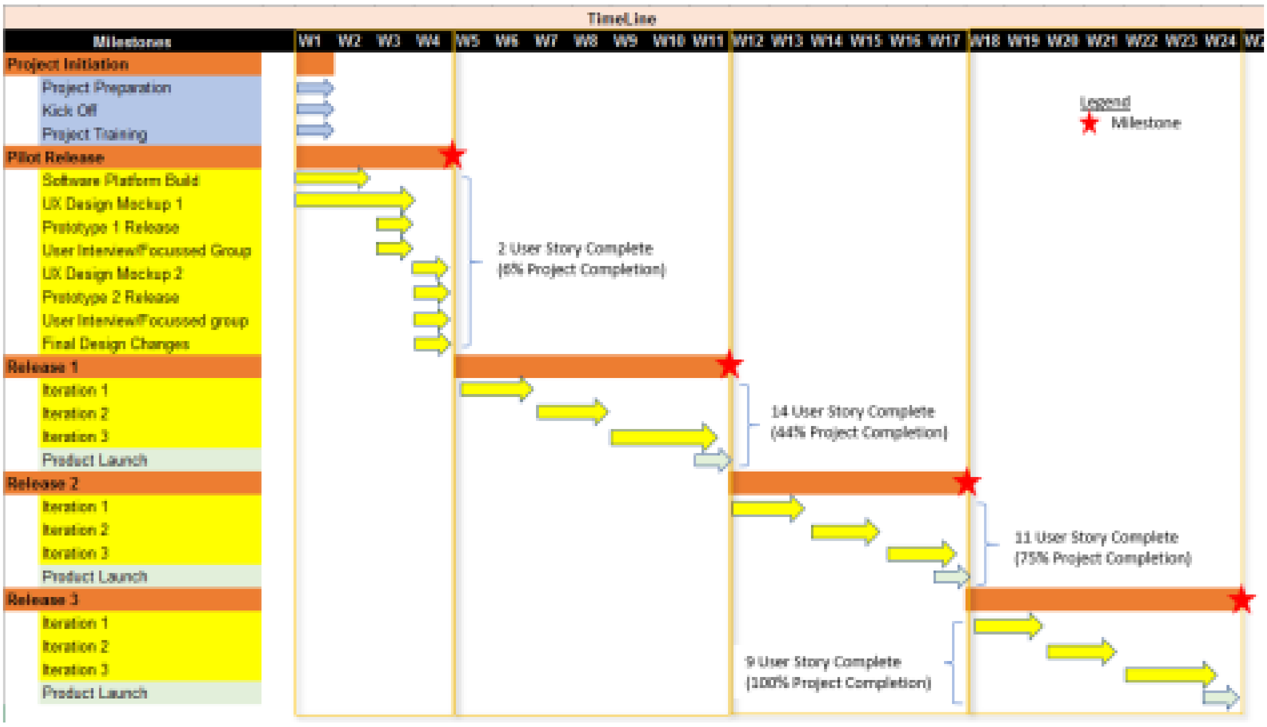


Figure 5: software product Project Timeline

Five phases of the Project:

2.2.6.1 Project Initiation

Project Initiation is primarily a Project Manager activity conducted during the first week of the project which involves project preparation, setting up the Kanban board, creating tasks for the team, allocating staff in the phases of the project, planning the sprints, and updating the team members on the project outline by conducting a Kick-Off meeting. Also, this week, Product Manager grooms the product backlog and defines the roadmap of the product.

Furthermore, the phase includes project training which is an Engineering activity conducted by the two Senior Developers to train four Junior developers.

2.2.6.2 Pilot Release

software product project is a new project, and the requirements were written without discussion with the customers. Hence the POC (proof of concept) to understand the viability and feasibility of the project is necessary. The project incorporates a four-week timeline (Week 1 to Week 4) starting in parallel with the project initiation phase.

The Pilot release phase is primarily a Product Manager and the UX Designer activity to build a Prototype (Interaction Design) of a basic product that involves screen to screen navigation. The product is released internally to discuss with customers in a Focus Group session and Customer interview and seek feedback on the usability of the product.

Before the Interaction Design begins, the phase of the project includes two weeks of Engineering activity to build the software platform, seek relevant web services licenses and validate the stability of the software environment.

2.2.6.3 Release 1

Release 1 is the first external release to the market and is a seven-week (Week 5 to Week 11) long phase of the project focusing on the basic needs of the customers. It covers 14 user stories with three Iteration cycles (two cycles are two weeks long, and the third cycle is two and a half weeklong). At the completion of the Iterations, the phase will constitute 44% of the entire project completion.

For the first release, the Product launch is very important, hence a Launch week is kept for the release which will include marketing of the product, training to support, internal and external communication to stakeholders and customers.

2.2.6.4 Release 2

Release 2 is a six week (Week 12 to Week 17) long phase of the project and focuses on the feature improvement user stories. It covers 11 user stories and at the completion of three two-week Iteration cycles, will constitute 75% of the entire project completion. Feature release will be communicated externally and internally through social media, and newsletter.

2.2.6.5 Release 3

Release 3 is a final release of the project which is a six week (Week 18 to Week 24) long phase of the project and focuses on the performance improvement features plus delighters. It covers 9 user stories and at the completion of three two-week Iteration cycles, will constitute 100% completion of the project.

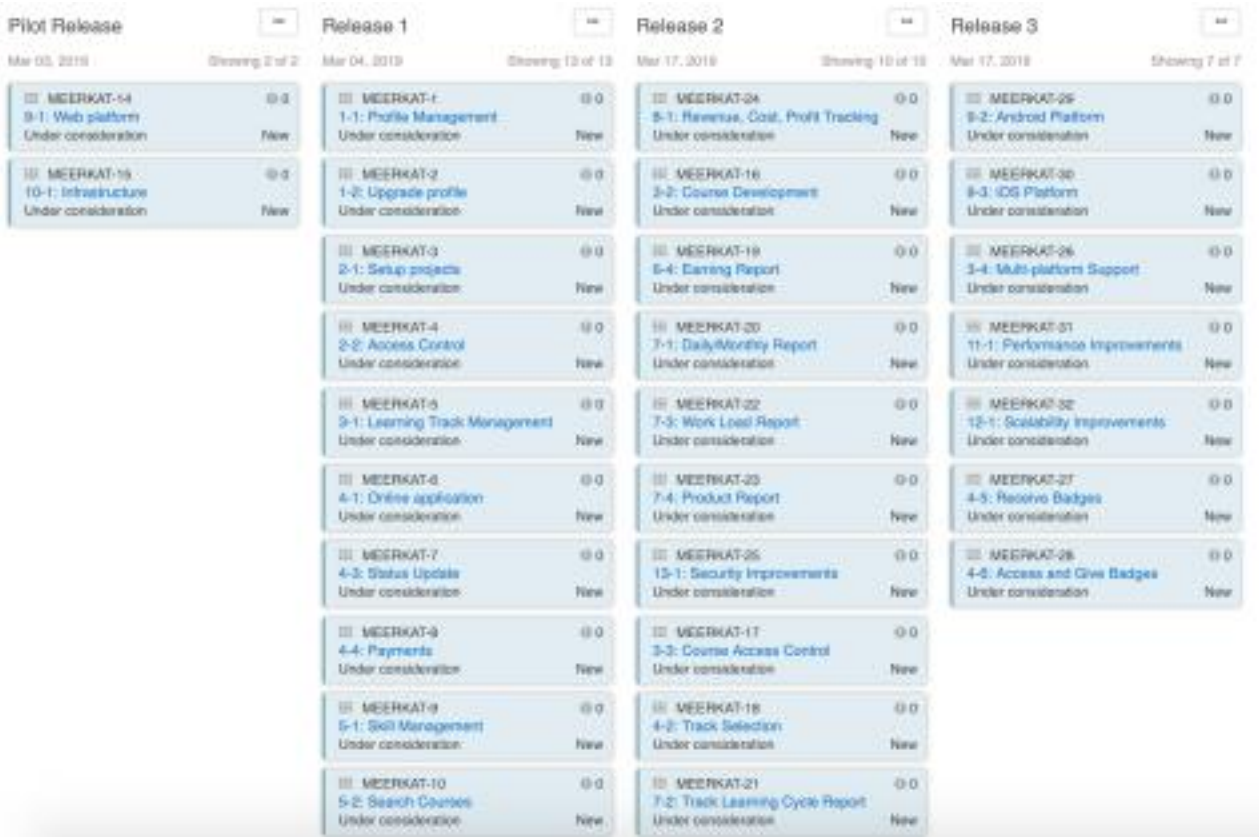
2.3. Project Planning

2.3.1 Product Roadmap

Every user requirement was examined to develop an overall release strategy that would align with a software company business objectives: Deliver a high-quality software product Platform and Get the product to market quickly. Adopting the Scrumban methodology, the software engineering plan has been broken down into different developmental releases and can be further broken down into 2-week iterations.

The roadmap below illustrates. Release 0 is primarily focused on delivering a software product pilot to validate assumptions, receive customer feedback, and understand technical and market feasibility. Additionally, the pilot that results from the initial release helps align organization vision and proposes an initial design for the development team. The next release, Release 1, covers essential features that cover the varying user personas and business functionality to generate revenue. Release 2 builds on top of software product foundation with more features and analytics adding to the customer value proposition. Lastly, Release 3 focuses on multi-platform support, and product improvement through the completion of non-functional requirements.

The product roadmap and backlog will be primarily adjusted by the product owner of the team, and the scrum master will be in charge of process facilitation. The use of online dashboards, features boards, and Kanban boards will make it easier for team members to visualize the product direction and implement a design that best aligns with business goals.



Pilot Release
Mar 03, 2019 Showing 2 of 2

MEERKAT-14 0
9-1: Web platform
Under consideration New

MEERKAT-15 0
10-1: Infrastructure
Under consideration New

Release 1
Mar 04, 2019 Showing 13 of 13

MEERKAT-1 0
1-1: Profile Management
Under consideration New

MEERKAT-2 0
1-2: Upgrade profile
Under consideration New

MEERKAT-3 0
2-1: Setup projects
Under consideration New

MEERKAT-4 0
2-2: Access Control
Under consideration New

MEERKAT-5 0
3-1: Learning Track Management
Under consideration New

MEERKAT-6 0
4-1: Online application
Under consideration New

MEERKAT-7 0
4-3: Status Update
Under consideration New

MEERKAT-8 0
4-4: Payments
Under consideration New

MEERKAT-9 0
5-1: Skill Management
Under consideration New

MEERKAT-10 0
5-2: Search Courses
Under consideration New

Release 2
Mar 17, 2019 Showing 10 of 10

MEERKAT-24 0
8-1: Revenue, Cost, Profit Tracking
Under consideration New

MEERKAT-16 0
3-2: Course Development
Under consideration New

MEERKAT-19 0
6-4: Earning Report
Under consideration New

MEERKAT-20 0
7-1: Daily/Monthly Report
Under consideration New

MEERKAT-22 0
7-3: Work Load Report
Under consideration New

MEERKAT-23 0
7-4: Product Report
Under consideration New

MEERKAT-25 0
13-1: Security Improvements
Under consideration New

MEERKAT-17 0
3-3: Course Access Control
Under consideration New

MEERKAT-18 0
4-2: Track Selection
Under consideration New

MEERKAT-21 0
7-2: Track Learning Cycle Report
Under consideration New

Release 3
Mar 17, 2019 Showing 7 of 7

MEERKAT-29 0
9-2: Android Platform
Under consideration New

MEERKAT-30 0
9-3: iOS Platform
Under consideration New

MEERKAT-26 0
3-4: Multi-platform Support
Under consideration New

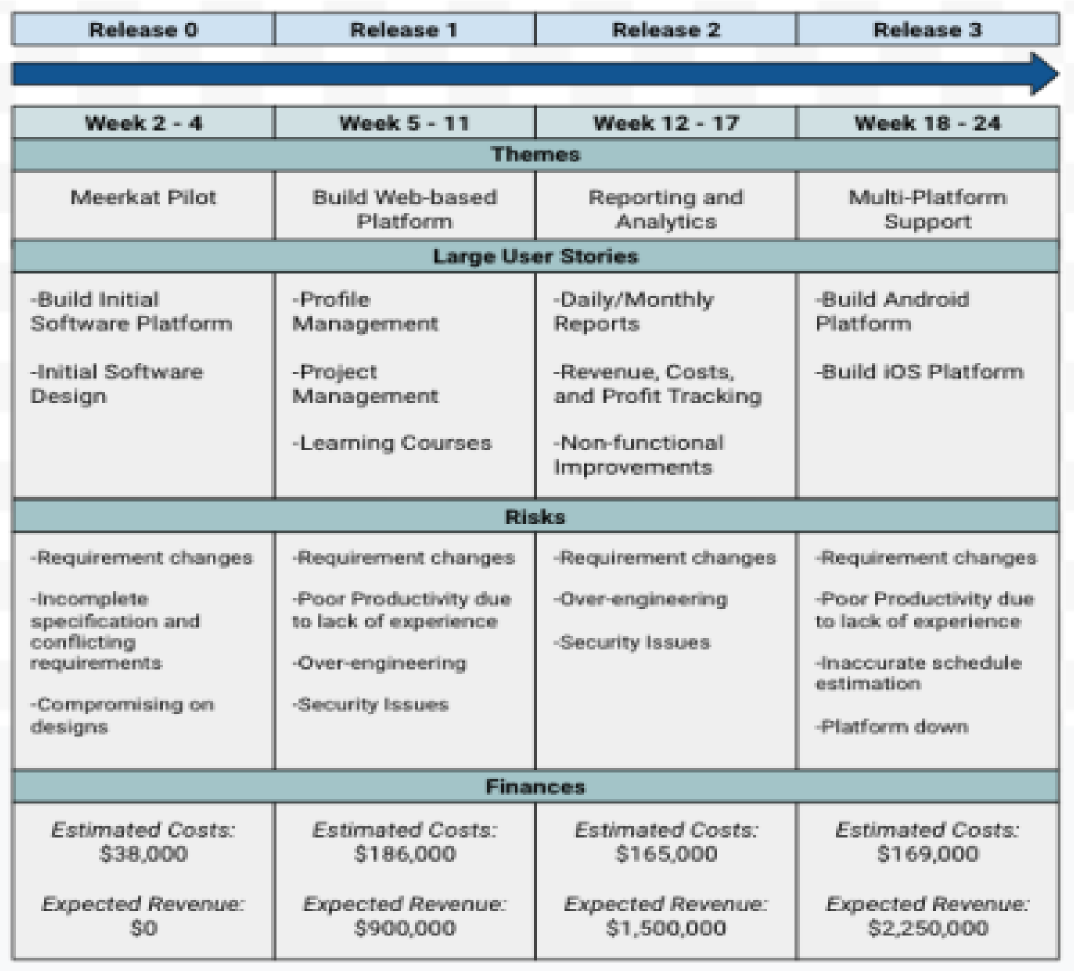
MEERKAT-31 0
11-1: Performance Improvements
Under consideration New

MEERKAT-32 0
12-1: Scalability Improvements
Under consideration New

MEERKAT-27 0
4-5: Receive Badges
Under consideration New

MEERKAT-28 0
4-6: Access and Give Badges
Under consideration New

Figure 6: software product Features Board



Release 0	Release 1	Release 2	Release 3
Week 2 - 4	Week 5 - 11	Week 12 - 17	Week 18 - 24
Themes			
Meerkat Pilot	Build Web-based Platform	Reporting and Analytics	Multi-Platform Support
Large User Stories			
-Build Initial Software Platform -Initial Software Design	-Profile Management -Project Management -Learning Courses	-Daily/Monthly Reports -Revenue, Costs, and Profit Tracking -Non-functional Improvements	-Build Android Platform -Build iOS Platform
Risks			
-Requirement changes -Incomplete specification and conflicting requirements -Compromising on designs	-Requirement changes -Poor Productivity due to lack of experience -Over-engineering -Security Issues	-Requirement changes -Over-engineering -Security Issues	-Requirement changes -Poor Productivity due to lack of experience -Inaccurate schedule estimation -Platform down
Finances			
Estimated Costs: $38,000 *Expected Revenue:* $0	*Estimated Costs:* $186,000 *Expected Revenue:* $900,000	*Estimated Costs:* $165,000 *Expected Revenue:* $1,500,000	*Estimated Costs:* $169,000 *Expected Revenue:* $2,250,000

Figure 7: software product Roadmap

2.3.2 Release Planning

2.3.2.1 First Release Plan

The first release is critical for the project software product as it sets the foundation of the project to attract learners and position them to the projects in a software company pipeline. The release is divided into two parts, first to validate proof of concept which is the Pilot release which will be released internally. Second is the first external release which covers the basic features that satisfied the needs of the users.

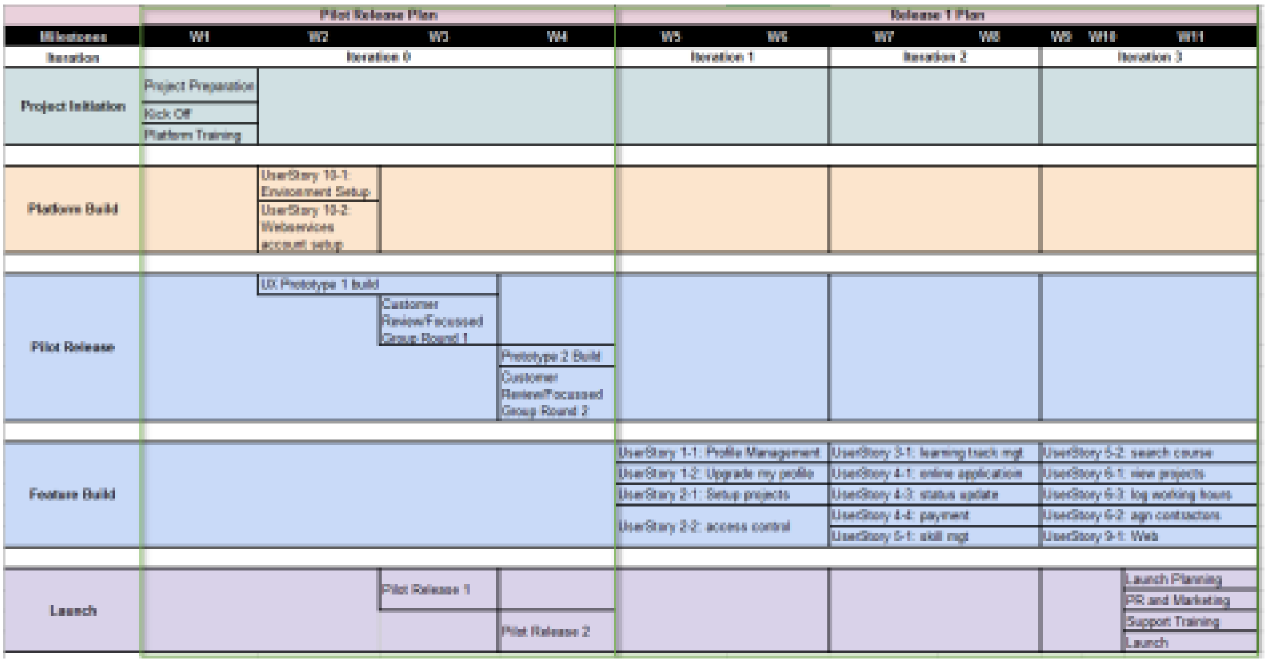
Pilot Release

The release is scheduled on a four-week plan which heavily focuses on the UX designs and setting up of the environment. The purpose of the release is to validate the feasibility of the software platform and the requirements.

The first week starts with project planning and training Junior Developers. The second week starts with setting up the software platform and getting web service licenses. In parallel, the UX designs are built and continue until Week 3. Once the prototypes are ready, the internal release will be released named as Pilot Release 1 in which customer review and focus groups will be conducted to validate the requirements and seek feedback from the learners/users. Week 4 will focus on the design revisions based on the feedback, and the second round of customer interview will be conducted as part of internal Pilot Release 2 to finalize the design.

Release 1

The release is scheduled on a seven-week plan out which primarily focuses on the development of minimum viable features from each Epic that satisfies the basic needs of the customers. The release is divided into three Iteration cycles with the first two cycles running for two weeks each, and the third cycle will run through two and a half weeks covering 14 user stories altogether. Additionally, the last week (seventh week) - Week 11 of the project focuses on preparing for the launch of the product to the market. The product release will be communicated to the stakeholders of the company through the newsletter, and to the external users through social media. Further, the part of the week will be dedicated to training the support specialist.



	Pilot Release Plan				Release 1 Plan						
Milestones	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11
Iteration	Iteration 0				Iteration 1		Iteration 2		Iteration 3		
Project Initiation	Project Preparation Kick Off Platform Training										
Platform Build		UserStory 10-1: Environment Setup UserStory 10-2: Webservices account setup									
Pilot Release		UX Prototype 1 build	Customer Review/Focussed Group Round 1	Prototype 2 Build Customer Review/Focussed Group Round 2							
Feature Build					UserStory 1-1: Profile Management UserStory 1-2: Upgrade my profile UserStory 2-1: Setup projects UserStory 2-2: access control		UserStory 3-1: learning track mgt UserStory 4-1: online application UserStory 4-3: status update UserStory 4-4: payment UserStory 5-1: skill mgt		UserStory 5-2: search course UserStory 6-1: view projects UserStory 6-3: log working hours UserStory 6-2: agn contractors UserStory 9-1: Web		
Launch			Pilot Release 1	Pilot Release 2						Launch Planning PR and Marketing Support Training Launch	

Figure 8: software product First Release Plan

2.3.2.2 First Release Effort

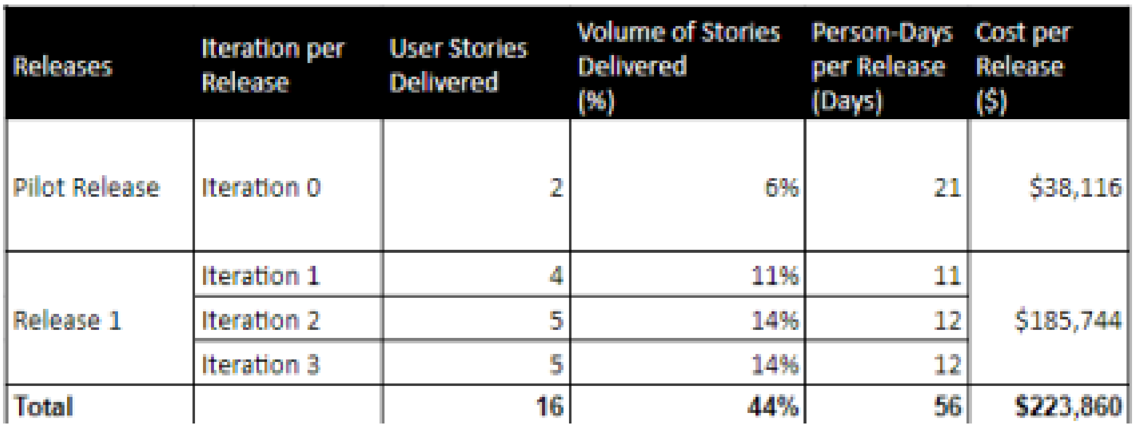
As stated above, the first release is divided into Pilot release, and Release 1, the effort for each release is broken down into four components.

1. **User stories delivered** - Number of user stories completed in both phases of the project. In the Pilot release, since the focus is on creating the prototypes, only two stories will be covered which are related to

setting up the software platform and acquire web service licenses. Release 1 is a major release with MVP features. Hence the total user stories completed will be 14, which makes the total of 16 user stories for both the releases.

2. **The volume of stories completed** - It is the percentage of user stories completed which is equivalent to the percentage of the project completion. It illustrates the progress of the project which is 44% in total combining both the releases.
3. **Person-days release effor**t - The effort spent in both the releases is 56 person-days.
4. **Cost per Release** - The cost of the Pilot release to validate the proof of concept is $38k which is a significant amount but will be counted as money well spent in the future as it will help in uncovering the design gaps and validate the viability of the project. Further, the Release 1 will cost around $186k making the total cost of $224k to deliver the basic viable product.

Table 5: Release Effort

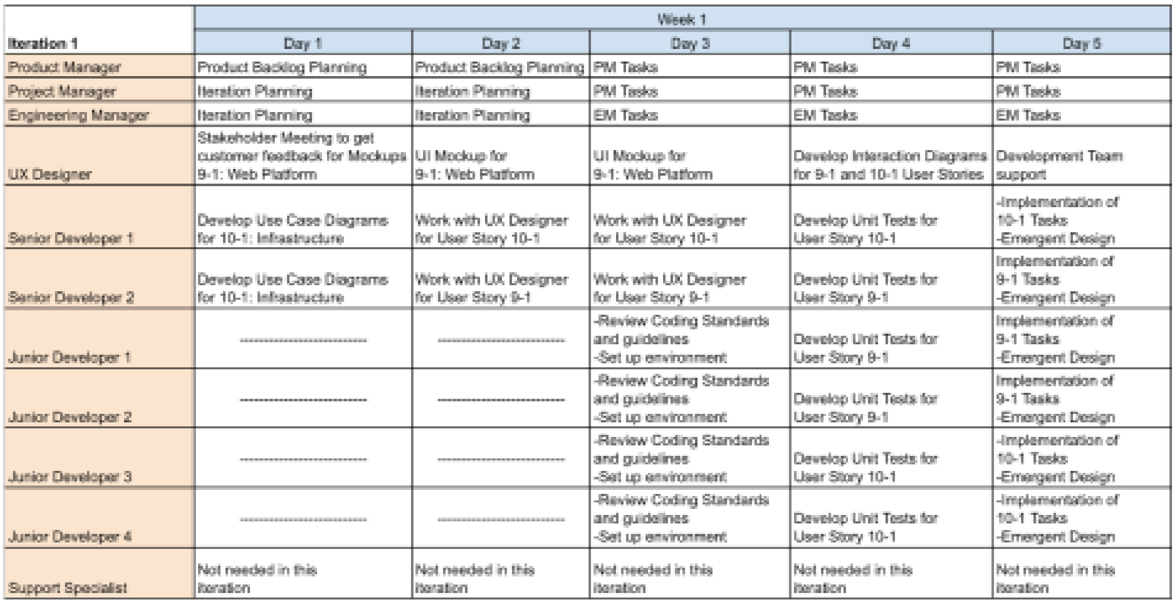


Releases	Iteration per Release	User Stories Delivered	Volume of Stories Delivered (%)	Person-Days per Release (Days)	Cost per Release ($)
Pilot Release	Iteration 0	2	6%	21	$38,116
Release 1	Iteration 1	4	11%	11	$185,744
	Iteration 2	5	14%	12	
	Iteration 3	5	14%	12	
Total		16	44%	56	$223,860

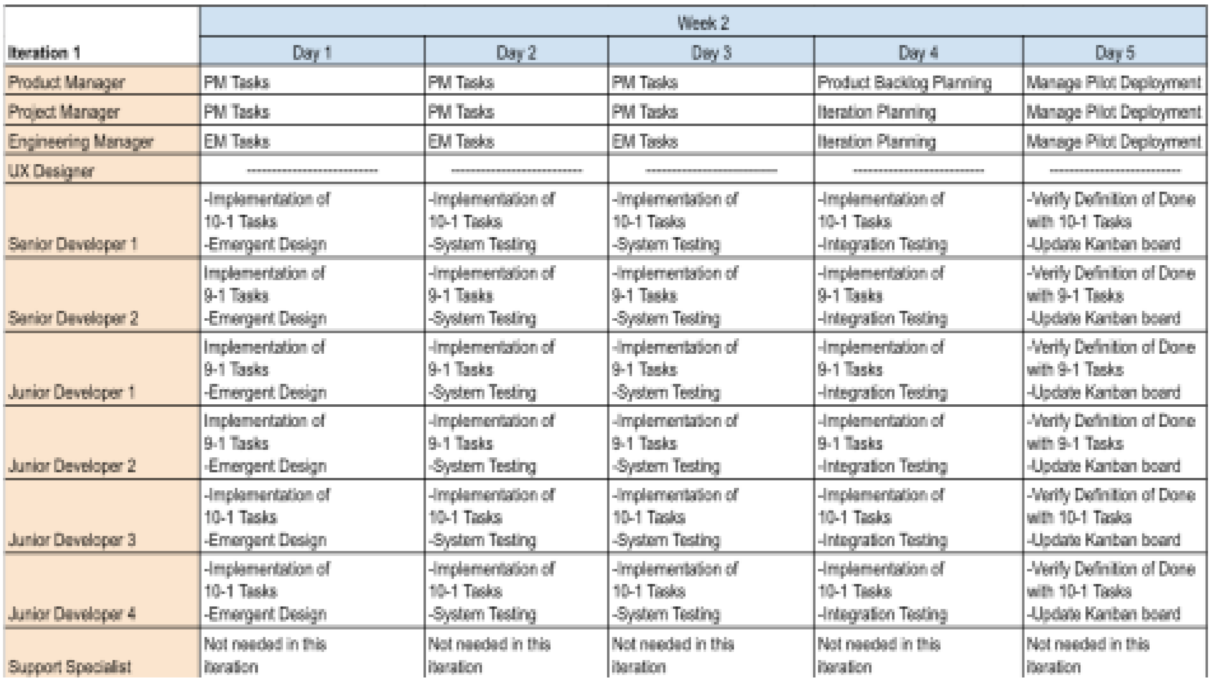
2.3.3 Iteration Planning

2.3.3.1 Iteration Plan

This iteration plan below details the first iteration of the pilot release. Combined with their tasks that are specific for the current iteration, each position also has recurring tasks that are seen throughout the development lifecycle. In the iteration plan below, there are table cells that indicate these recurring tasks as seen with “PM Tasks.” These tasks include practices associated with the recommended methodology that will help a software company better reach their business objectives. For further details, these practices are listed in a table below the Iteration Plan.



	Week 1				
Iteration 1	Day 1	Day 2	Day 3	Day 4	Day 5
Product Manager	Product Backlog Planning	Product Backlog Planning	PM Tasks	PM Tasks	PM Tasks
Project Manager	Iteration Planning	Iteration Planning	PM Tasks	PM Tasks	PM Tasks
Engineering Manager	Iteration Planning	Iteration Planning	EM Tasks	EM Tasks	EM Tasks
UX Designer	Stakeholder Meeting to get customer feedback for Mockups 9-1: Web Platform	UI Mockup for 9-1: Web Platform	UI Mockup for 9-1: Web Platform	Develop Interaction Diagrams for 9-1 and 10-1 User Stories	Development Team support
Senior Developer 1	Develop Use Case Diagrams for 10-1: Infrastructure	Work with UX Designer for User Story 10-1	Work with UX Designer for User Story 10-1	Develop Unit Tests for User Story 10-1	-Implementation of 10-1 Tasks -Emergent Design
Senior Developer 2	Develop Use Case Diagrams for 10-1: Infrastructure	Work with UX Designer for User Story 9-1	Work with UX Designer for User Story 9-1	Develop Unit Tests for User Story 9-1	Implementation of 9-1 Tasks -Emergent Design
Junior Developer 1	------------------------	------------------------	-Review Coding Standards and guidelines -Set up environment	Develop Unit Tests for User Story 9-1	Implementation of 9-1 Tasks -Emergent Design
Junior Developer 2	------------------------	------------------------	-Review Coding Standards and guidelines -Set up environment	Develop Unit Tests for User Story 9-1	Implementation of 9-1 Tasks -Emergent Design
Junior Developer 3	------------------------	------------------------	-Review Coding Standards and guidelines -Set up environment	Develop Unit Tests for User Story 10-1	-Implementation of 10-1 Tasks -Emergent Design
Junior Developer 4	------------------------	------------------------	-Review Coding Standards and guidelines -Set up environment	Develop Unit Tests for User Story 10-1	-Implementation of 10-1 Tasks -Emergent Design
Support Specialist	Not needed in this iteration	Not needed in this iteration	Not needed in this iteration	Not needed in this iteration	Not needed in this iteration



	Week 2				
Iteration 1	Day 1	Day 2	Day 3	Day 4	Day 5
Product Manager	PM Tasks	PM Tasks	PM Tasks	Product Backlog Planning	Manage Pilot Deployment
Project Manager	PM Tasks	PM Tasks	PM Tasks	Iteration Planning	Manage Pilot Deployment
Engineering Manager	EM Tasks	EM Tasks	EM Tasks	Iteration Planning	Manage Pilot Deployment
UX Designer	------------------------	------------------------	------------------------	------------------------	------------------------
Senior Developer 1	-Implementation of 10-1 Tasks -Emergent Design	-Implementation of 10-1 Tasks -System Testing	-Implementation of 10-1 Tasks -System Testing	-Implementation of 10-1 Tasks -Integration Testing	-Verify Definition of Done with 10-1 Tasks -Update Kanban board
Senior Developer 2	Implementation of 9-1 Tasks -Emergent Design	-Implementation of 9-1 Tasks -System Testing	-Implementation of 9-1 Tasks -System Testing	-Implementation of 9-1 Tasks -Integration Testing	-Verify Definition of Done with 9-1 Tasks -Update Kanban board
Junior Developer 1	Implementation of 9-1 Tasks -Emergent Design	-Implementation of 9-1 Tasks -System Testing	-Implementation of 9-1 Tasks -System Testing	-Implementation of 9-1 Tasks -Integration Testing	-Verify Definition of Done with 9-1 Tasks -Update Kanban board
Junior Developer 2	Implementation of 9-1 Tasks -Emergent Design	-Implementation of 9-1 Tasks -System Testing	-Implementation of 9-1 Tasks -System Testing	-Implementation of 9-1 Tasks -Integration Testing	-Verify Definition of Done with 9-1 Tasks -Update Kanban board
Junior Developer 3	-Implementation of 10-1 Tasks -Emergent Design	-Implementation of 10-1 Tasks -System Testing	-Implementation of 10-1 Tasks -System Testing	-Implementation of 10-1 Tasks -Integration Testing	-Verify Definition of Done with 10-1 Tasks -Update Kanban board
Junior Developer 4	-Implementation of 10-1 Tasks -Emergent Design	-Implementation of 10-1 Tasks -System Testing	-Implementation of 10-1 Tasks -System Testing	-Implementation of 10-1 Tasks -Integration Testing	-Verify Definition of Done with 10-1 Tasks -Update Kanban board
Support Specialist	Not needed in this iteration	Not needed in this iteration	Not needed in this iteration	Not needed in this iteration	Not needed in this iteration

Figure 9: Pilot Release Iteration 1 Plan

Table 6: Individual roles in the Iteration cycle[6]

Recurring Tasks		
Role/Tasks	**Detailed Description**	**Definition of Done**
Project Manager		
Iteration Planning	-Scheduling -Scope Analysis -Resource Allocation -Process Facilitation	-Schedule reflects customer's vision as well as engineering efforts and feasibility -Product backlog is refined -Defects fixed or postponed
Project Monitoring	-Evaluate Scope -Risk Monitoring -Quality Monitoring	-Risk-driven testing coverage -Software Quality Metrics updated daily -Risks updated daily and prioritized
Product Manager		
Product Roadmap	-User Story Definition -Groom Backlog -Update Kanban Board	-All requirements are clarified in the same format to make the team share the same understanding
	-Stakeholder Feedback	-Kanban Board reflects current state
Velocity	-Quality Monitoring	-DoD of User Stories are completed

[6] "Definition of Done - CA Agile Central - CA Technologies Documentation." https://docs.ca.com/en-us/ca-agile central/saas/definition-done. Accessed 24 Feb. 2019.

Tracking	-Burn-up charts	-Product Backlog updated
Engineering Manager		
Analyzing Technological Stack	-Analyze Technical Debt -Analyze Architecture	-Any configuration or build changes documented -Documentation updated
Deployment Management	-Risk Monitoring -Quality Monitoring	-Project builds without errors -Unit tests written and passing -OK from the team: UX designer, developers, Project Manager, Product Owner
Development Team		
Test-first Development	-Create Unit Tests -Verify Tests -Improve Tests	-Unit tests are created, improved, passed and cover the code written -More modularized, flexible, and extensible code and less bugs
Implementation	-Story completion -Follow coding standards - Continuous System Testing	-All code is checked in -All Developer tests passed -Feature/Task OK'd by Product Owner -Proper naming and usage of coding guidelines in the code
Integration	-Integrate subsystems -Integrate system -Integration Testing	-All unit and functional tests are green -Individual units are grouped together and tested
Support Specialist		
Supporting Team	-Development Tasks -Q&A Testing	-QA performed and issues resolved -Refactoring completed

2.4 Project Risks

Like any other software projects, risks are common, and they exist throughout the life cycles of software development. Based on studies in the field and the unique characteristics of this project, below are the top 10 project risks. Based on the discussions in the Cost Estimate section, the direct cost of 1 week of the development team is $27,000. The following chart and the section summarize the top 10 risks in the Risk Assessment Metrics and steps to mitigate the risk to bring the High Impact and High Probability risks (red zone) to the less risky zone (Amber or Green).

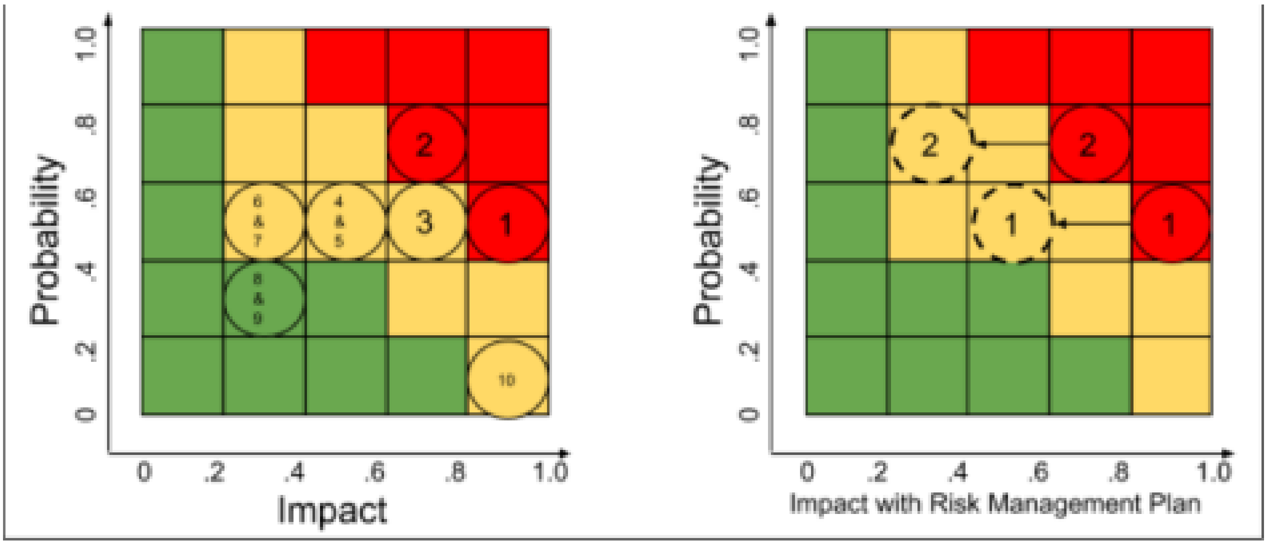


Figure 10: Risk Assessment Matrix (Probability and Impact)

2.4.1 Risk 1: Inaccurate schedule estimation

Even though the team might come up with the best estimate for the project, schedule flaw could happen due to unexpected increased in difficulties of variant tasks.

Table 7: Risk 1 Exposure

Probability	High at 50%
Impact	Up to 4 weeks of delay
Exposure:	50% * 4 weeks = 2 weeks of delay, which translate into $54,000 direct cost.

2.4.1.1 Risk Management Plan

Get the team more involved in planning and estimating. Get early feedback and address slips directly with stakeholders.[7] For instances, Product Backlog Planning for Project Planning, Workflow Visualization using Kanban Board would help better at estimation and visualization of what is going on. With those practices in place, expected delays are hoping to be reduced to 2 weeks.

[7] "The Top Five Software Project Risks - Project Smart." https://www.projectsmart.co.uk/top-five-software-project risks.php. Accessed 16 Feb. 2019.

2.4.2 *Risk 2: Requirement Changes*

As the HCaaS industry evolves, the project might have to face changes in requirements, from as small as user interface design pattern, to as big as business model changes.

Table 8: Risk 2 Exposure

Probability	High at 60%
Impact	Up to 3 weeks of delay
Exposure:	60% * 3 weeks = 1.8 weeks of delay, which translate into $48,600 direct cost.

2.4.2.1 Risk Management Plan

Constant involvement of customers and developers.[6]Iteration based practices like Emergent Architecture, Interaction Design, and Emergent Design works better for changing requirements, hence can help mitigate the risk. With those practices in place, expected delays are hoping to be reduced to 1.5 weeks.

2.4.3 Risk 3: Key team members leaving

The team has only 1 Project Manager and 1 Product Manager that have a full view of the big picture of the projects as a whole. From the engineering side, two senior engineers take on big share of the engineering design and responsibility. Any of those key members leaving would result in the delay of the project.

Table 9: Risk 3 Exposure

Probability	High at 50%
Impact	Up to 3 weeks of delay
Exposure:	50% * 3 weeks = 1.5 weeks of delay, which translate into $40,500 direct cost.

2.4.3.1 Risk Management Plan

Set aside a budget to counteroffer if the team member plans to leave the organization and set up budgets for team events to boost team morale. Increased collaboration and information sharing in the team to minimize the knowledge loss due to key team member departure.

2.4.4 Risk 4: Poor Productivity due to lack of experience

The team has limited experience in the Scrumban practices and Human Capital as a Services (HCaaS) domain. It is not reasonable to expect the team could come up to speed with all practices and process. Hence productivity might be poor from time to time whenever the team needs to learn new knowledge.

Table 10: Risk 4 Exposure

Probability	High at 50%
Impact	Up to 2 weeks of delay
Exposure:	50% * 2 weeks = 1 weeks of delay, which translate into $27,000 direct cost.

2.4.4.1 Risk Management Plan

Having short iterations and the right set of people in the team would make it easier for the team to catch up with speed. Also, coaching and team development are important to improve productivity. Additionally, practices like Coding Standards and Test-first programming could help boost the productivity of the development team.

2.4.5 Risk 5: Incomplete specification and conflicting requirements

It is observed quite frequently in the software development cycles that when the coding and integration begins, the requirement or/and technical specifications are either not completed or conflicting with each other.

Table 11: Risk 5 Exposure

Probability	High at 50%
Impact	Up to 2 weeks of delay

Exposure:	50% * 2 weeks = 1 weeks of delay, which translate into $27,000 direct cost.

2.4.5.1 Risk Management Plan

A dedicated Product manager to make trade-off decisions and communicate to the development team is critical here. Using Kanban board, User Stories and Use Case Diagrams to manage and visualize the requirements would help here as well.

2.4.6 Risk 6: Compromising on designs[8]

In order to get stuck into the next ‘real’ tasks, developers tend to rush the design process. This is a waste of programming hours, as designing is the most critical part of the software development.

Table 12: Risk 6 Exposure

Probability	Medium at 40%
Impact	Up to 2 weeks of delay
Exposure:	40% * 2 weeks = 0.8 weeks of delay, which translate into $21,600 direct cost.

2.4.6.1 Risk Management Plan

Embracing Interaction Design and Emergent Design during the design phase could help mitigate this risk. Those are iteration-based design practices that take into account changing requirements, which also works great for cases like “not complete” design.

2.4.7 Risk 7: Over-engineering (a.k.a gold plating)

Developers sometimes like to show off their skills by adding unnecessary features. For instance, a developer might add Flash to a basic login module to make it look ‘stylish’ which is a waste of programming hours.

Table 13: Risk 7 Exposure

Probability	Medium at 40%
Impact	Up to 2 weeks of delay
Exposure:	40% * 2 weeks = 0.8 weeks of delay, which translate into $21,600 direct cost.

2.4.7.1 Risk Management Plan

There are primarily two high risk gold plating: Requirement gold-plating and Developer gold-plating. For requirement gold-plating, using Kanban board to visualize requirements could help identify requirements that aren’t necessary. For developer gold-plating, Test-first Programming practices might help here, as developers focus would shift to pass the test with the necessary effort. Overall, risk of gold plating is mitigated by short iterations in Scrumban methods.

2.4.8 Risk 8: Lack of sufficient projects for learners

Due to strong competition or economic downturn, chances are that we don’t have sufficient projects for workers in the platform that’s ready to take on.

[8] "Top 10 Software Development Risks | ITProPortal." 14 Jun. 2010, https://www.itproportal.com/2010/06/14/top ten-software-development-risks/. Accessed 16 Feb. 2019.

Table 14: Risk 8 Exposure

Probability	Medium at 20%
Impact	Up to 10% revenue
Exposure:	20% * 10% revenue = 2% revenue, which translate into $18,000 direct cost for the 2nd year.

2.4.8.1 Risk management plan

Promotion plans for the projects that use the platform should be put in place by support specialist when the project is ready to open up for general public. In fact, sales pushes should begin earlier on, and customers should be in the pipeline waiting for participations.

2.4.9 Risk 9: Security issue leads to data leakage

Due to unexpected security issues in software, such as heartbleed[9], user data get exposed to the public. This will result in losing the customer or/and their trust.

Table 15: Risk 9 Exposure

Probability	Medium at 20%
Impact	Up to 10% revenue
Exposure:	20% * 10% revenue = 2% revenue, which translate into $18,000 direct cost.

2.4.9.1 Risk management plan

Embrace open-source software frameworks that are proved to be more secure than proprietary software. Additionally, embrace cloud platform to minimize the security burden, as most of the cloud platform will have built-in support for security.

2.4.10 Risk 10: Platform down

Due to unexpected, rare conditions, such as DSS attack or even fires in the data center, the whole platform could be down for an extended period.

Table 16: Risk 10 Exposure

Probability	Low at 1%
Impact	Up to 100% revenue
Exposure:	1% * 100% revenue = 1% revenue, which translate into $9,000 direct cost.

2.4.10.1 Risk management plan

The best risk management plan here would be to have “failover” and “business continues” plan ready. Today, most of the cloud platform is still suffering from the unexpected downtimes[10]. Hence it is not unreasonable to have the services ready in more than one cloud platforms.

[9] "Heartbleed Bug." http://heartbleed.com/. Accessed 24 Feb. 2019.

[10] "AWS Service Health Dashboard - Feb 24, 2019 PST." https://status.aws.amazon.com/. Accessed 24 Feb. 2019.

3. Conclusion

software product Software Engineering plan successfully achieves the three a software company objective. Firstly, to release the product quickly so that the company can start generating the revenue from the current projects in the pipeline. With the current software plan, the product is expected to be launched in 11 weeks from the beginning of the project. Secondly, the company expects to be profitable from year 2. Project cost analysis shows that the company will generate $250K from Year 2. Thirdly, to deliver the product with a reasonable set of features with the focus on quality. The project release plan considers the objectives and releases the product with basic, satisfying needs of the customer and considered the quality as a prime focus by analyzing the product viability in the Pilot release and testing the product continuously in each iteration cycle.

Further, the engineering plan eliminates a software company’s three business model issues (Revenue stream not predictable, Billable hours challenge, and limited growth by development resources) by sizing the complete project in detail, calculating the revenue and profitability of the product, and allocating time to train the developers.

From the Project planning perspective, the Project can achieve the required figures based on the degree of confidence level, but strategically placing the product and marketing the product is of paramount importance. The software product project team expects a strong marketing and sales team from a software company during the first six months of the project to market the product and spread awareness in order to have a strong customer base at the time of the release. Further, the team requires the sales team to engage an existing education provider in a partnership to design learning tracks and course list concurrently with the building of the product.

4. References

1. "Kano model - Wikipedia." https://en.wikipedia.org/wiki/Kano_model. Accessed 23 Feb. 2019. 2. "MoSCoW method - Wikipedia." https://en.wikipedia.org/wiki/MoSCoW_method. Accessed 23 Feb. 2019.
2. "Estimation Techniques Wideband Delphi - Tutorialspoint." https://www.tutorialspoint.com/estimation_techniques/estimation_techniques_wideband_delphi.ht m. Accessed 23 Feb. 2019.
3. "Planning Poker: An Agile Estimating and ...." https://www.mountaingoatsoftware.com/agile/planning-poker. Accessed 23 Feb. 2019. 5. "Three-point estimation - Wikipedia." https://en.wikipedia.org/wiki/Three-point_estimation. Accessed 23 Feb. 2019.
5. "Definition of Done - CA Agile Central - CA Technologies Documentation." https://docs.ca.com/en-us/ca-agile-central/saas/definition-done. Accessed 24 Feb. 2019. 7. "The Top Five Software Project Risks - Project Smart." https://www.projectsmart.co.uk/top-five software-project-risks.php. Accessed 16 Feb. 2019.
6. "Top 10 Software Development Risks | ITProPortal." 14 Jun. 2010, https://www.itproportal.com/2010/06/14/top-ten-software-development-risks/. Accessed 16 Feb. 2019.
7. "Heartbleed Bug." http://heartbleed.com/. Accessed 24 Feb. 2019.
8. "AWS Service Health Dashboard - Feb 24, 2019 PST." https://status.aws.amazon.com/. Accessed 24 Feb. 2019.