

***SOFTWARE PROJECT MANAGEMENT PLAN FOR Osríc's
Office Appliances and Decor (OOA&D)***

Team Ewoks:

Sudhindra Desai

Kelsey Joseph

Selvamani Masilamani

Erin Rhodes

Rich Stretanski

Contents

1.0 Introduction	4
1.1 Problem statement	4
1.2 Project scope	4
1.2.1 Inclusions	4
1.2.2 Exclusions	4
1.3 Major software functions	4
1.4 Performance/Behavior constraints	5
1.5 Management and technical constraints	5
1.5.1 Management constraints	5
1.5.2 Technical constraints	5
2.0 Project Estimates	5
2.1 Historical data used for estimates	5
2.2 Initial Estimate	5
2.2.1 SPMP Completion Estimate	5
2.2.2 Overall project estimate	6
2.3 Estimation techniques applied and results	8
2.3.1 Estimation technique 1 – lines of code	8
2.3.2 Estimate for technique 1 – lines of code	8
2.3.3 Estimation technique 2 – function points	8
2.3.4 Estimation technique 3 – process/task	8
2.3.5 Estimate for technique 3 – process/task	8
2.4 Reconciled Estimate	8
2.5 Project Resources	8
3.0 Risk Management	8
3.1 Project Risk Table	9
3.2 Overview of Risk Mitigation, Monitoring, Management	11
4.0 Project Schedule	12
4.1 Project task set	12
4.2 Task network	12
4.3 Timeline chart	12

5.0 Staff Organization	12
5.1 Team structure	12
5.2 Management reporting and communication	12
6.0 Tracking and Control Mechanisms	13
6.1 Quality assurance and control	13
6.2 Change management and control	13
6.3 Tools	13
7.0 Appendix	14

1.0 Introduction

1.1 Problem statement

Osric Ormondsey owns Osric's Office Appliances and Decor (OOA&D) unable to keep up with the customer request by providing the service based on priority due to high demand.

1.2 Project scope

1.2.1 Inclusions

Develop the web based service tracking tool to schedule and keep track of the services and priorities based on the company.

1.2.2 Exclusions

This plan will not include any travel needs (if needed).

Hardware procurement and Installation are not included (OS, DB or Webserver)

1.3 Major software functions

Plan the entire development into 6 Sprint of each 2 weeks.

1. Requirement gathering.
2. Break the individual requirements into userstory to build backlog.
3. Sprint planning
4. Retrospective
- Sprint 1:
 - Story 1: Create the DB tables.
 - Story 2: Define the users with role, "Admin", "Assistant", "Manager" & "Technician".
 - Story 3: Develop login mechanism/page
- Sprint 2: Assistant Page Development
 - Story 4: Develop Home page and enable/disable certain features based on user role
- Sprint 3:
 - Story 5: Develop functionality to create new customer.
 - Story 6: Define workflow for new customer (Manager Approval).
 - Story 7: Develop functionality to create new service request, assign priority & assign to technician.
- Sprint 4:
 - Story 8: Develop search functionality. User can search the company based on their ID to get details like # previous calls.
 - Story 9: Search functionality to get estimated service time.
- Sprint 5:
 - Story 10: Develop functionality to update the service call completion with number of day and night blocks.

- o Story 11: Manager home page show all the pending request for his/her approval for new customer.
- Sprint 6:
 - o Story 12: Build the logic of auto rest priority based on “2 full day blocks”.
 - o Story 13: Auto generate bill and send to customer by email.

1.4 Performance/Behavior constraints

- System is required to be available for use between 6am and 10pm, 7 days a week. Any necessary maintenance must occur between 10pm and 5am, as no changes will be made to the schedule during this time.
- Transactions should take less than 3 seconds.

1.5 Management and technical constraints

1.5.1 Management constraints

- Limited to a team of 5
- Must be completed before end of the 2017 calendar year

1.5.2 Technical constraints

All the Tools mentioned under section 6.3 have to be licensed versions, so that there won't be any hindrance throughout the lifecycle.

2.0 Project Estimates

2.1 Historical data used for estimates

Although each project is different we came up with this estimation based on past project with other customer that included most of the functionality that Osric is looking for.

We are planning to deploy the team with experience of 50+ years

	Experience
Member 1	5
Member 2	10
Member 3	12
Scrum Master	5
Product Owner	20
Total Experience	52

Average price per hr = $\$81900/1800 = \45.5 per hour which is much less than you might have already paying.

2.2 Initial Estimate

2.2.1 SPMP Completion Estimate

120 Hrs (5 Members * 3 Days)

2.2.2 Overall project estimate

2.2.2.1 Line-of Code Estimate

Story	Task	File Type	# Lines
Story 1	Create DB	SQL	40
Story 2	UI	JSP	60
	Servlet	Java	140
	JS	JS	80
Story 3	UI	JSP	150
	Servlet	Java	100
	JS	JS	50
Story 4	UI	JSP	150
	Servlet	Java	180
	JS	JS	150
Story 5	UI	JSP	150
	Servlet	Java	80
	JS	JS	150
Story 6	UI	JSP	150
	Servlet	Java	80
	JS	JS	150
Story 7	UI	JSP	50
	Servlet	Java	40
	JS	JS	170
Story 8	UI	JSP	20
	Servlet	Java	20

	JS	JS	0
Story 9	UI	JSP	0
	Servlet	Java	150
	JS	JS	40
Story 10	UI	JSP	200
	Servlet	Java	50
	JS	JS	100
Story 11	UI	JSP	90
	Servlet	Java	30
	JS	JS	40
Story 12	UI	JSP	40
	Servlet	Java	50
	JS	JS	110
Story 13	UI	JSP	30
	Servlet	Java	50
	JS	JS	30
Story 14	UI	JSP	0
	Servlet	Java	110
	JS	JS	100
Story 15	UI	JSP	0
	Servlet	Java	50
	JS	JS	70
Total # Lines			3500

2.2.2.2 Function Estimate

Sprint	Week	Working Hrs Per Day	Working Days Per Week	Working Hr Per week	Member 1	Member 2	Member 3	Scrum Master	Project Manager	Total Hrs
Sprint 1	Week 1	8	5	40	40	40	40	10	20	150
	Week 2	8	5	40	40	40	40	10	20	150
Sprint 2	Week 1	8	5	40	40	40	40	10	20	150
	Week 2	8	5	40	40	40	40	10	20	150
Sprint 3	Week 1	8	5	40	40	40	40	10	20	150
	Week 2	8	5	40	40	40	40	10	20	150
Sprint 4	Week 1	8	5	40	40	40	40	10	20	150
	Week 2	8	5	40	40	40	40	10	20	150
Sprint 5	Week 1	8	5	40	40	40	40	10	20	150
	Week 2	8	5	40	40	40	40	10	20	150
Sprint 6	Week 1	8	5	40	40	40	40	10	20	150
	Week 2	8	5	40	40	40	40	10	20	150

2.2.2.3 Tasks Estimate

Sprint	Week	Member 1	Member 2	Member 3	Scrum Master	Project Manager	Total Hrs
Sprint 1	Story 1	30	30	30	10	20	120
	Story 2	20	20	20	5	10	75
	Story 3	30	30	30	5	10	105
	Time Spent	80	80	80	20	40	300
Sprint 2	Story 4	30	30	30	10	20	120
	Story 5	20	20	20	5	10	75
	Story 6	30	30	30	5	10	105
	Time Spent	80	80	80	20	40	300
Sprint 3	Story 7	30	30	30	10	20	120
	Story 8	20	20	20	5	10	75
	Story 9	30	30	30	5	10	105
	Time Spent	80	80	80	20	40	300
Sprint 4	Story 10	40	40	40	10	20	150
	Story 11	40	40	40	10	20	150
	Time Spent	80	80	80	20	40	300
Sprint 5	Story 12	40	40	40	10	20	150
	Story 13	40	40	40	10	20	150
	Time Spent	80	80	80	20	40	300
Sprint 6	Story 14	40	40	40	10	20	150
	Story 15	40	40	40	10	20	150
	Time Spent	80	80	80	20	40	300

2.2.2.4 Total overall project time estimate in hours of effort

Total estimated hrs. : 1800 Hrs

2.3 Estimation techniques applied and results

2.3.1 Estimation technique 1 – lines of code

SLOCCount utility helps to find the line of code

2.3.2 Estimate for technique 1 – lines of code

lines per Hr = # code lines / # hours spent

2.3.3 Estimation technique 2 – function points

End of each sprint product is shipped to customer to make sure all the committed functionalities are compiled as customer needs. # stories deviled provides the count of functionality.

2.3.4 Estimation technique 3 – process/task

lines of code task (story) = (# lines of code per sprint / # Story delivered in that sprint)

2.3.5 Estimate for technique 3 – process/task

task(story) per hr = # Stories developed / # hours Spent

2.4 Reconciled Estimate

Cost (In \$): \$81900 (Team Member – \$60 per hr , Scrum Master – \$50 per hr, Project Manager – \$70 per hr)

Hours: 1800

Team Size: 05

Time to complete: 12 weeks

2.5 Project Resources

- People : 3 Team members, 1 Scrum Master, 1 Project Manager
- Hardware : Win 10 Enterprise with 32 GB RAM and 500 GB (HDD)
- Software & Tools :
 - Source Control: Subversion 1.9.5
 - Development Environment: Eclipse Java EE IDE for Web Developers
 - Database: MySQL 5.7
 - Webserver: Apache Tomcat 9.0.0.M21
 - Runtime Environment: JRE 1.8 Update 121

3.0 Risk Management

3.1 Project Risk Table

ID	Name of Risk	Description	Probability	Impact	Mitigation	Contingency
1	Estimates are inaccurate	Inaccurate estimates is a common project risk.	MEDIUM	HIGH	Estimates were peer reviewed by the Senior Management, Senior Technical Management to avoid being In-accurate	New methodology will be used to re-estimate
2	Inaccurate customer priorities	When customers are not Prioritized accordingly	MEDIUM	HIGH	Every customer is assigned with a Priority Value at the time of Service request creation.	Customer will be attended immediately without considering the pre-assigned Priority
3	Customer Cancels Service request	Customer cancels the Service request placed already	LOW	LOW	Assistant calls the customer to re-confirm the problem existence	Customer who cancels more than three times will be least prioritized
4	Technician misunderstand problem	When problems are misinterpreted by the Technician, it will take time for them to fix the issue	LOW	HIGH	Technicians are well trained to understand the problem at hand	New Technician is assigned with proper knowledge to handle the Task
5	Technician shortfalls	Inability to secure sufficient Technicians for the project.	LOW	HIGH	Highly skilled Technicians are identified, trained and deployed on work	New Technicians has to be recruited by checking for a right knowledge
6	System Outages	Application crashes/ Database crashes	LOW	HIGH	Regular System maintenance is carried on by calling in the System engineers	Regular backup is done to overcome system outages
7	Failure to update the Service request	Assistant fail to update the service request in the system due to	LOW	HIGH	Assistant generates the Service request number for every request received so that no request is	Assistant uses the ledger to track down the Service request in case of

		system outage			missed	system outage
8	Delay in Customer Payback	Customer delays in paying back for the service done	LOW	MEDIUM	Customer pays back via online transaction based on the generated Invoice	Customer will be notified about the payment due every week once through E-mail until payment is done
9	Team Member Leaves	A team member leaves the team because of extenuating circumstances, opportunity, or other personal reasons.	LOW	HIGH	Create a work environment that team members don't want to leave.	Predetermine how the work would be divided up, should a team member leave.
10	Software Familiarity & Experience	Team Member unfamiliarity with software environment causes longer development time and/or more defects.	MEDIUM	LOW	Make team members aware of the dev environment ahead of time.	Make a list of skills that each team member has, including familiar software. Have the inexperienced team members go to the experienced team member for assistance.
11	Project Requirements Change	Customers threaten to leave for another competitor because of wait times. Pressure is added to complete the project ahead of schedule.	LOW	HIGH	Strive to complete sprints ahead of time, just in case.	Have an accelerated schedule made .
12	Team Member Personality Clash	Team members are not working well together.	LOW	MEDIUM	Schedule team bonding exercises/events.	Change leadership or find new members.
13	Bad Team Morale	Low team morale causes low productivity.	MEDIUM	MEDIUM	Keep the work environment feeling fresh.	Have a couple days of slack planned so that the team can take time off to

						refresh if needed.
--	--	--	--	--	--	--------------------

3.2 Overview of Risk Mitigation, Monitoring, Management

Customer Risks:

Customer Cancels Service Request:

- **Mitigation:** The Assistant call the customer to reconfirm the existence of the problem. This will reduce the amount of times a technician is sent to a customer who no longer has a problem, freeing them up to be used for other service requests.
- **Monitoring:** The Assistant will monitor and keep track of any cancellations from customers.
- **Management:** A single cancellation does not have a drastic impact on the business, however if it is a common occurrence it will have a significant impact on the wait times of other customers. Therefore, any customer who cancels a service requests more than three times will have their customer priority number reduced.

Delay in Customer Payment:

- **Mitigation:** We can mitigate delayed payments from customers by offering online payment transaction based on the generated invoice.
- **Monitoring:** The Manager should monitor all customer payment statuses
- **Management:** Should the customer not pay in a timely fashion, the customer will receive an automatic email notification of the invoice.

Customer Threatens to Leave:

- **Mitigation:** Customers are frustrated with wait times and could potentially pressure us to speed up development by threatening to leave. We should mitigate this outcome by keeping this in mind and focusing on completing each sprint on-time or ahead of time, so that should this outcome occur, it will not be catastrophic to the project.
- **Monitoring:** The project manager will be responsible for monitoring the status of each sprint as well as comparisons to the estimates. It will also be the manager's responsibility to create an alternate accelerated schedule.
- **Management:** In the event that a high priority customer threatens to leave, the project manager will create an alternative accelerated schedule which will be implemented.

Technical Risks:

Technician Misunderstood Problem:

- **Mitigation:** Keep High standards for hiring practices, so that Technicians will be well trained to understand the problem at hand.
- **Monitoring:** It is upon the technicians themselves to recognize when they are unable to solve the problem, and then notify an Assistant.
- **Management:** In the event that a technician is unable to solve the problem for the customer, a new technician will be deployed to the customer.

Technician Shortfalls:

- **Mitigation:** Keep High standards for hiring practices and encourage technicians to share the solutions to new problems that they encounter with other technicians so that the amount of shortcomings that a technician might have is mitigated.

- **Monitoring:** The Manager writes performance reviews based on customer feedback and timely resolution of service requests.
- **Management:** Should a current technician have drastic shortcomings, a decision must be made to either invest extra training into the technician, or look to hire a new technician.

System Outages:

- **Mitigation:** In order to mitigate the effect of system outages, there will be scheduled system maintenance by the system engineers.
- **Monitoring:** The system engineers will monitor the status of the system when they come in for their regularly scheduled system maintenance.
- **Management:** In the event that system outages become a common occurrence, a regular backup of the system will be implemented.

Failure to Update the Service Request:

- **Mitigation:** To mitigate this, the assistant will generate a service request number for every request so that no request is missed.
- **Monitoring:** The assistant is responsible for keeping track of service request numbers.
- **Management:** In the event of an outage, the assistant use the ledger to track down the service request number.

Team Risks:

Team Member Leaves:

- **Mitigation:** A team member leaving could be a huge setback. Keeping an interesting and challenging work environment would keep some members from leaving for other opportunities.
- **Monitoring:** The team manager is responsible for keeping track of each team member's responsibilities in case a member decides to leave.
- **Management:** In the event that a team member leaves, the workload of that individual will be divided up based on a preexisting arrangement for how to divide up each member's responsibilities.

Software Familiarity and Experience:

- **Mitigation:** In order to mitigate problems with familiarity of the dev environment, the full details of that environment will be made available to the team as early as possible prior to the project start.
- **Monitoring:** The Team Manager will make a list of skills that each team member has, including familiar software, and they will familiarize themselves with each member's strengths.
- **Management:** In the event where a team member is struggling with an aspect of the dev environment, the team manager will have the inexperienced team members go to the experienced team member for assistance.

Team Members Not working Well Together:

- **Mitigation:** To decrease the chances of team members not working well together, regularly scheduled team bonding exercises/events will be held.
- **Monitoring:** The project manager will be responsible for scheduling team building events, as well as monitoring the team's overall effectiveness at working together.
- **Management:** Should a team member be unwilling to work with others to a degree that is drastically affecting how the team is performing, then either a team member change

will be made or a team manager change will be made.

Bad Team Morale:

- **Mitigation:** To decrease the likelihood of the team developing bad morale, the project manager will keep the workplace fresh, exciting, and challenging.
- **Monitoring:** The project manager will be responsible for monitoring the morale of the team.
- **Management:** Should the team morale status become bad, the manager is able to give paid time off assuming the schedule allows.

Operational Risks:

Estimates are Inaccurate:

- **Mitigation:** All estimates are to be peer reviewed by the Senior Management, Senior Technical Management to avoid being Inaccurate
- **Monitoring:** The Project Manager must monitor how closely the project estimates compare to the actual metrics throughout the project.
- **Management:** If the Project Manager decides after monitoring that the estimates will be way off, then they can decide to make re-estimates using a different methodology.

4.0 Project Schedule

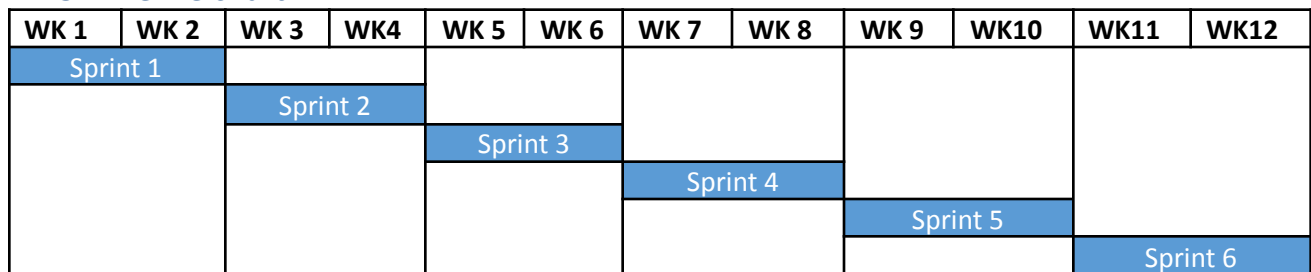
4.1 Project task set

- Sprint 1
 - Create database framework, login features and user categories
- Sprint 2
 - Create user profiles and user functionalities
- Sprint 3
 - Develop basic functionalities : create new customer, new service request, etc.
- Sprint 4
 - Expand program functionality to include search feature and scheduling
- Sprint 5
 - Refine functionality to adapt to scheduling changes
- Sprint 6
 - Continue refining functionality and workflow completion
- UAT testing

4.2 Task network

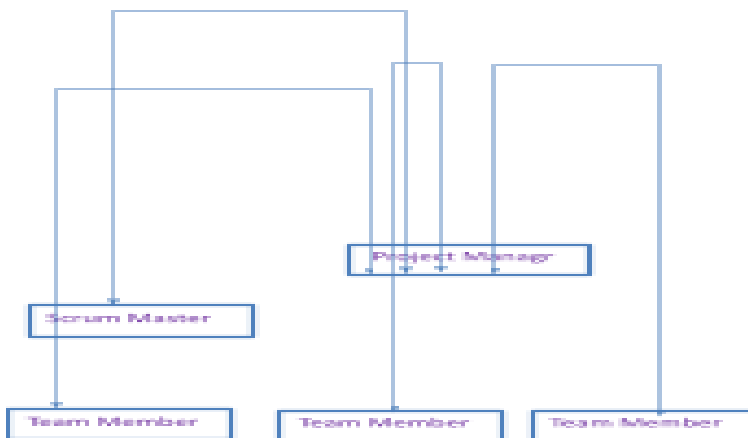
Task Mode	Task Name	Duration	Start	Finish	Predecessors
	Sprint 1	2 wks	Mon 9/11/17	Fri 9/22/17	
	Create db tables	2 wks	Mon 9/11/17	Fri 9/22/17	
	Create user role profiles	1 wk	Mon 9/18/17	Fri 9/22/17	
	Create login screen/mechanism	2 wks	Mon 9/11/17	Fri 9/22/17	
	Sprint 2	2 wks	Mon 9/25/17	Fri 10/6/17	1
	Develop Admin home page	2 wks	Mon 9/25/17	Fri 10/6/17	2,3
	Develop Assistant home page	2 wks	Mon 9/25/17	Fri 10/6/17	2,3
	Develop Manager home page	2 wks	Mon 9/25/17	Fri 10/6/17	2,3
	Develop Technician home page	2 wks	Mon 9/25/17	Fri 10/6/17	2,3
	Sprint 3	2 wks	Mon 10/9/17	Fri 10/20/17	5
	Develop the functionality to create a new customer	1 wk	Mon 10/9/17	Fri 10/13/17	2,3
	Develop a workflow for new customers requiring manager approval	1 wk	Mon 10/9/17	Fri 10/13/17	2,3
	Develop the functionality to create a new service request	1 wk	Mon 10/16/17	Fri 10/20/17	2,3,11,12
	Develop the functionality to prioritize new service requests and assign to a technician	1 wk	Mon 10/16/17	Fri 10/20/17	2,3,11,12
	Sprint 4	2 wks	Mon 10/23/17	Fri 11/3/17	10
	Develop the functionality to search for customers	2 wks	Mon 10/23/17	Fri 11/3/17	2
	Develop the functionality to search for scheduled appointments	2 wks	Mon 10/23/17	Fri 11/3/17	2
	Develop the functionality to determine estimated service time	2 wks	Mon 10/23/17	Fri 11/3/17	2
	Sprint 5	2 wks	Mon 11/6/17	Fri 11/17/17	15
	Develop the functionality to update the service call completion with numer of day and night blocks	2 wks	Mon 11/6/17	Fri 11/17/17	13,14
	Add functionality to Manager homepage to show all pending request for his/her approval for new customer	2 wks	Mon 11/6/17	Fri 11/17/17	13,14,8
	Sprint 6	2 wks	Mon 11/20/17	Fri 12/1/17	19
	Build the logic of auto reset priority based on 2 full day blocks	2 wks	Mon 11/20/17	Fri 12/1/17	14
	Develop an auto generated bill and send to each customer by email	2 wks	Mon 11/20/17	Fri 12/1/17	3
	UAT Testing/Quality Assurance	3 wks	Mon 12/4/17	Fri 12/22/17	22

4.3 Timeline chart



5.0 Staff Organization

5.1 Team structure



5.2 Management reporting and communication

- Master Plan that identifies key activities and the projected start and end dates for each sprint as well as percentage complete for the developers to share with the Team Lead on projected timing.
- KPI/KAI report that tracks a status (Red/Yellow/Green) to key identified metrics for the Team Lead to communicate with the management team.

Note: KPI – Key Performance Indicators; KAI – Key Activity Indicators

6.0 Tracking and Control Mechanisms

6.1 Quality assurance and control

- End of each Sprint product is delivered to customer for review, feedback and acceptance.
- Defect found in the previous sprint will be addressed in upcoming print and shipped to customer again for signoff.
- Test driver development help to minimize the issues.
- Application of Technical Methods (Employing proper methods and tools for developing software).
- Periodical Technical Self and Peer review is carried on to ensure the quality and functionality expectation is achieved.
- Testing of Software.
- Control of Change (Assess the need for change, document the change).
- Different Metrics are captured to measure the quality of the Software
- Artifacts are updated regularly, so that it can be archived to refer back in the future when required.
- Regular PCI Audit is carried on to ensure all the standards and estimated schedules are met.

6.2 Change management and control

A change management file will be created documenting all changes that are requested after development. Included in the file will be the following:

- Description of the requested change
- Departments/Entities affected by the change
- Priority of the change
- Severity of the change (i.e. will it affect other operations, or is it an easy singular fix)
- Approval of the change
 - Approval from all affected departments
 - Approval from Manager
- Projected Implementation Date of the change – estimated date change will be available in production site.
- Status
- Actual Implementation Date of the change – actual date change is available in production site

Changes can be requested through a help link and form, which will trigger a workflow to notify all the involved parties. Accepted changes will be sent to the development team to implement, while rejected changes will be returned to the requestor who can then edit and resubmit.

Once a change is approved and implemented, an implementation date will be determined and a change notice, (including a brief one point lesson, if needed) will be distributed to all registered users.

6.3 Tools

- o Source Control: Subversion 1.9.5
- o Development Environment: Eclipse Java EE IDE for Web Developers
- o Database: MySQL 5.7
- o Webserver: Apache Tomcat 9.0.0.M21
- o Runtime Environment: JRE 1.8 Update 121

7.0 Appendix