Modern methods in Software Engineering ID 2207 KTH

SEP Swedish Event Planning And Swedish United Solutions USS

Project report

Course Coordinator: Mihhail Matskin
Teacher: Shatha Jaradat
Teacher: Cosar ghandeharioon

Students
Haseeb Aslam Butt
Adam shafai

Table of contents

Introduction

An internal system management is implemented using the (Agile) XP-programming approach.
Instruction of how the program should be implemented was given by the course instructor which focused on the following business.

Swedish event planning SEP is a company which organise business events and they need an internal system management to be developed in order to automate their system.
The organization have different departments such as customer service CS which register the event requests into the system.
The idea is that every employee should login to the system with some authentication and the employees have different accessibility and responsibility depending on their position in the organization.
Since the time and scope of this project was limited we were instructed to focus on some workflows as follows.

We will focus on the following functionalities in SEP business:
1. Workflow of event requests.
2. Workflow of tasks distribution to services/production departments.
3. Staff recruitment management (through HR)
4. Financial requests management.

Client management, reports management, the employees' records, scheduling issues, salaries and other parts of the problem are not required for the project.

The main elements of XP in this project will be:
1. Developing user stories.
2. Release planning.
3. Iteration planning.
4. Selecting a system metaphor that could be suitable for the problem solving.
5. Developing a system in a test-driven fashion for selected user stories.
6. Refactoring the programs.
7. Pair programming.
8. Daily Stand-up meeting.

User stories

Some GUI parts are excluded from user stories but exits in code and in the iteration plan
Table 1

User Stories	**Descriptions**	**Time**	**Value**	**Risk**
login	Employee I want to login with my unique username and password.The login should provide the functionality to verify employees credentials and based on he or she's predefined position in the SEP organization.	20	H	L
new Event Request	CS should be able to initiate new event request also should be able to fill the event name , start date , finish date, budgets, event type and details about the event from customer point of view.	30	H	L
new Client	SCS may have ability to register new clients, view clients list, She may fill the form within client's name , phone number, and email address. She may first check client's records if the customer is already in the database	20	H	L
Client records	AM , SCS and FM may have ability to Check/view/update the client's info in the database/list if the client is registered in the database	40	H	M
Employee/user	Every employees in SEP should have their unique username name and password,There should be possibility to add new user	30	M	L
Employee Position /Rank	The employees in SEP should be categorised by their rank and position every employee based in their responsibility should be assigned accessibility to view , check or modify events and user's e.t.	20	L	L
FinancialRequest	In case of any financial issue the department managers should be able	30	M	L

	to request finance and send to financial manager the should be able to view and close the requests.			
StaffRequest	The department managers should have ability to send recruitment request to HR in case need more crew.	30	L	H
Status	The employee based in their rank should view the status of event requests and should be able to submit their own decision in the status bar(accepted or rejected)	40	H	H
Task	The department manager should have the ability to assign tasks to sub team employee	30	M	L
Team	The department manager should have ability to build sub team in order to assign them certain task to complete	30	L	L
UserRecords	save all user information in database/ lists	40	H	M
EventRecords	all events should be categorised and saves in a database/lists	20	H	M
ListClientView	The CS, SCS,AM and FM can search and view for clients records	50	L	L
ListUpdaterview	should update and view the status of a request	30	H	H
ListStaffRequest	The staff request may come from PM or SM which should be viewed as a list	30	L	L
Edit Task	The department manager have ability to edit the task, The employee may also have access to modify the task	10	H	L
MainView	The system should have a user friendly interface GUI		L	L
New account	The system may have functionality which HR can create new user	20	H	L

	account for the new employee			
LoginView	login interface should have user friendly interface GUI	10	L	L
ListFinancialRequest	The financial request may can be send from PM or SM or Sub team which should be view as list	30	M	L
createEventRequest	The system may have functionality to accept as many event as needed and the event should be viewed as a list	30	L	L
Controller	Handle the interaction between model classes and view classes	30	H	M

Release plan

The following user stories were been chosen among those in table 1.1. The choice is based within the scope of the project timing,some user stories which involve data base and front end development are excluded in order to deliver the project in time.

Login
EventRequest
Client
Position
ResourceRequest
StaffRequest
Status
Task
Team
User
CreateClientView
CreateEventView
ListClientView
ListStaffRequest
ListResourceRequest
ListUpdaterView
LoingView
MainView
Controller

Risk/Value	**High**	**Medium**	**Low**
High	2	0	1
Medium	4	0	0
Low	5	4	7

Iteration plan

The architectural style MVC is used to group the user stories in three category. The implementation follows the test driven approach TDD. The user stories as planned were categorised and implemented in three iteration.
Refactoring have been made randomly but mostly in the end of every task to remove redundancy and eliminate unused functionality.
The implementation is based in Gradle and the environment used is intellij IDEA with Junit for testing purpose.

Iteration 1

LoginTest Login LoginView UserTest User ClientTest Client Position MainView EventRequestTest EventRequest
Implementation took 2 days

Iteration 2

ResourceRequest StaffRequestTest StaffRequest Status TaskTest Task TeamTest Team Controller
Implementation took 3 days

Iteration 3

CreateClientView CreateEventView CreateResourceRequest CreatStaffRequest CreateTaskView ListClientView ListStaffRequest ListResourceRequest ListTaskView ListUpdaterView
Implementation took 3 days

Metaphor

A **Postbox:** think of a letter sending by post.The letter must be put(Submit by you) in a postbox and it later view and picked by the postman and he or she submit it later to the post terminal which finally based on the address you wrote it will be delivered. and the other hand you may or may not get a response from the person or organization you wrote to.
The same here the customer officer fills the event form(write letter) and submit(postbox) to the system (postman).

An internal system management which focus on communication and interaction between employees in a company, the communication is time independent.Employees have the ability to **Create/View/update** and **Submit/send** the **messages/letters /Status**.

Test

The unit test class is made before the actual class and some trivial getter and setter have been omitted testing and more focused was on the important classes.
The Junit test framework have been used to tests.
The test begins with making some static class and make some objects meanwhile one make the object the test directly fails since there is no class exists for the object created and later use the junit Assertion to put argument to the constructor which also immediately fails the next step is to implement the constructor and run the test again.

Acceptance Test

Two acceptance test have been implemented one for the **Login** since the criteria was that this test should be done for the class which has input and output.
The acceptance test for **Login** is an static class make object of the **LoginView** and use the junit assertion to take input after that it view the LoginView interface which disappear as soon as one click on the button.
The **ListView** acceptance test use the same terminology as the **loginView.**

Project Design Methodology

We used Agile Methodology for developing this project. Pair Program Technique was used to implement the project. Based on this technique, we both developers worked simultaneously on one computer. One person was coding at a time and the other person was observing and giving live feedback. Moreover, the role was also switched multiple times during the working day. While working using this technique we realized that it helped in enhancing our productivity as the errors were spotted very early and corrected instantly which saved a lot of time and effort.

Comparison

The object oriented and analysis design approach is more “formal” approach. In this approach clients requirements are gathered in the start and they are considered fix and change in requirements at the later stage become very difficult. Moreover, a lot of formal documentation is produced during different phases of the project.
The Agile approach is more flexible approach and welcomes changes. Also Agile approach emphasises more on communication between clients and programmers. Moreover, in Agile approach, focus is on developing the working release of the software with some critical business functionality implemented earlier in the timeline.

Meetings

Project Day 1
Participants: Haseeb & Adam
Agenda: To logically divide the intended system in different components. Discuss an outlook on the schedule and the possible buildup for timely completion of all the required functionalities. Moreover, possible mechanisms for Authentication using Username & passwords were also discussed.

Project Day 2
Participants: Haseeb & Adam
Agenda: Discuss the progress made in the implementation of Authentication mechanism in previous day. Contemplate on the User Interface that can fulfil our non-functional requirement of user-friendly UI. Design details for the Implementation of the functionality of entering new event request by CS representative was discussed. Moreover, the method to test the correct functionality of this use case was also discussed.

References

1. Lectures slides of the course MMSE 2017
Minhail Mitskin, KTH
2. SEP Business Case Description
3. http://www.extremeprogramming.org/