**Insertion Sort Application Development Process**

**Agile Scrum & XP**

**Advanced Software Process (CSPS-­544).**

**Prof. Chang­Hyun Jo.**

**California State University,Fullerton**

**(Team-4)**

**Chirag Padsala. (**[**cpadsala@csu.fullerton.edu**](mailto:cpadsala@csu.fullerton.edu)**)**

**Prathmesh Pardhiye. (prathmesh@csu.fullerton.edu)**

**Revision History:**

|  |  |  |
| --- | --- | --- |
| **Date** | **Assigned to** | **Task Completed** |
| Sept. 8, 2016 | Chirag Padsala | Created Title page. |
| Sept. 8, 2016 | Prathmesh Pardhiye | Defined XP, and Scrum process. |
| Sept. 9, 2016 | Prathmesh Pardhiye | Defined work product, roles and practises for Scrum and XP. |
| Sept. 9, 2016 | Chirag Padsala | Listed vision, goals and requirement.. |
| Sept. 10, 2016 | Chirag Padsala | Listed User Stories. |
| Sept. 10, 2016 | Prathmesh Pardhiye | Created Functional, non functional Requirement and AIF’s. |
| Sept 11, 2016 | Chirag Padsala | Created product backlog and estimation.  Assigned Roles and responsibility.  Updated table of content. |
| Sept 12, 2016 | Prathmesh Pardhiye | Defined Release Planning.  Explained Technological preparation. |
| Sept 13, 2016 | Prathmesh Pardhiye | Updated figures and table of contents. |
| Sept. 15, 2016 | Chirag Padsala | Listed and explained Iteration I. |
| Sept. 16, 2016 | Prathmesh Pardhiye | Added user stories. |
| Sept. 22, 2016 | Prathmesh Pardhiye | Listed and explained Iteration II. |
| Sept. 23, 2016 | Prathmesh Pardhiye | Explained other activities- project planning and monitoring. |
| Sept 25, 2016 | Chirag Padsala | Explained Iteration III and other activities-Risk Management.. |
| Sept 25, 2016 | Prathmesh Pardhiye | Explained Configuration management, Quality assurance. |
| Sept 26, 2016 | Prathmesh Pardhiye | Explained documentation, training, operating and feedback. |
| Sept. 26, 2016 | Chirag Padsala | What we learned as individual and as a group. |

**Table of Contents**

1.0 Process Definition..……………………………………………………………………..4

Extreme Programming (XP)………………………………………………………..4

Scrum…………………………………………………………………………………5

2.0 Work Products, Roles, and Practices…….…………………………………………..6

Scrum…………………………………………………………………………………6

XP…………………………………………………………………………………....11

3.0 Pre­Game: Planning and Staging……..……………………………………………..13

Vision……………………………………………………………………………….13

Requirements…………………………………...…………………………………16

User Stories……………………………………………………………………..…17

Product Backlog and Estimates…………………………………………….……18

Architectural Spike…...……………………………………………………………19

Release Planning…………………………………………………………...…….19

Technology Preparation: Android Development Environment…………..…..20

4.0 Development……..…………………………………………………………………….21

Iteration I………………………………………………………………………….. 21

Iteration II………………………………………………………………….……….24

Iteration III……………….…………………………………………………...…… 26

Acceptance Testing……………………………………………………………….28

5.0 Release………………………………..………………………………………………..28

Documentation…………………………………………………………………….28

Training………………………….…………………………..............…………….28

Operating. ……………………….………………………………….....................28

Feedback………………………….………………………...........….…..…….....29

6.0 Other Activities………………………...…………………………......................……30

Estimation………………………...………………………………......…..……….30

Project Planning…………………………………………………………...………31

Project Monitoring and Control……………………………..............................32

Configuration Management………………………………………………………33

Quality Assurance…………………………………………………………………34

Risk Management…………………………………………………………………35

7.0 Lesson Learned……………...……………….......….…….……….......................37

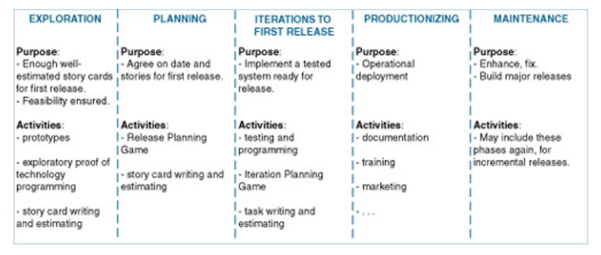
8.0 Reference………………………………………….…………………………………39

Team Charter………………………………………………………………………………40

Evaluation report…………………………………………………………….……….…….43

# **1.0 Process Definition**

Extreme Programming (XP)

Extreme Programming (XP) is a part of the Agile Process that spotlights on short advancement cycles to create quality programming and responsiveness to changing client’s necessity. 

There are five noteworthy stages in XP lifecycle:

1. Exploration

2. Planning

3. Iterations to first release

4. Productionizing

5. Maintenance.

Exploration:

This is the first phase in XP which encompasses the initial requirement processing and initial activities modeling aspect of agile process of development lifecycle.

Planning:

Following exploration phase is the planning phase in which user stories are studied and agreement dates of first release is planned.

Iteration to first release:

Iteration to release phase is primary effort of XP process, which consist of modeling, programming, testing and integration for iteration release.

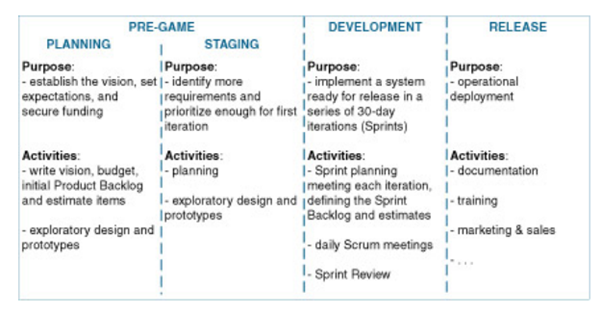
Productionizing:

This phase is close to deployment where documentation and marketing of product is performed.

Maintenance:

In this phase the XP lifecycle is performed to evolve product with fix and enhancement.

Scrum:

Scrum is an Agile software development method used for building applications. Scrum is a repetitive and incremental agile software development framework for overseeing as well as handling product improvement.

The scrum lifecycle consist of four phases:

1. Pr-game planning

a. Planning

b. Staging

2. Development

3. Release

Pre-game planning and staging:

In the planning phase, the group needs to make an arrangement that match their vision of what the application is required to do. The team needs to arrange their financial plan and evaluate the time required to do everything in the item backlog. In the arranging stage, the clients and the product owner ought to recognize prerequisites also, organize them in the product backlog.

Development:

In the development phase, the scrum group's motivation is to make a framework in incremental releases in every cycle. Besides, there ought to be daily scrum gatherings where the individuals need to overhaul (update) each other on what was finished after every sprints.

Release:

In the release phase of scrum, the practise is to release a completely operational product. And also, documentation should be accessible. Training can be given to show the client how to use and utilize the item.

**2.0 Work-Products, Roles and Practices**

**Scrum**

Work-products

There are five work-products in a Scrum development process:

· Product Backlog

· Sprint Backlog

· Sprint Burn-down Chart

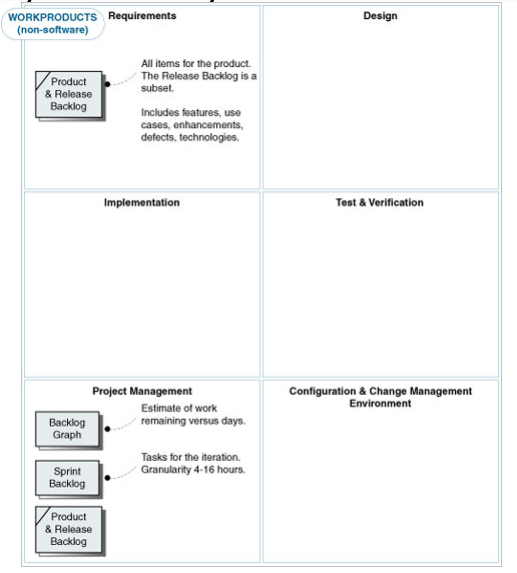
· Task Board

Product Backlog:

The Product Backlog is the expert list of all the desired elements in the product. When a project is started there is no comprehensive effort, which requires much time to write all the tasks or foreseeable requirements. The Product Backlog can be maintained in an Excel spreadsheet. Estimates are made by developers, but they are inaccurate and are only used for a rough allocation of tasks in several sprints.

Sprint Backlog:

The Sprint Backlog is a list of tasks that the Scrum Team is committed to complete the current Sprint. Sprint Backlog items are taken from the Product Backlog. The team, based on the defined priority and the perception of time, will be needed to complete the task. It is critical that the team identify the items and the size of Sprint Backlog. Because they are committed to completing the tasks, they are who should choose what they will commit. Sprint Backlog, very often, is maintained as an Excel spreadsheet, but you can also use your bug tracking system or various software products developed specifically for Scrum or Agile.

Figure:The work-products in a Scrum development process.

Sprint Burn-down Chart:

Sprint Burndown Chart is a graphical representation of the remaining work on Sprint. The remaining work estimated at Sprint is calculated daily and graphically, resulting in a Sprint Burndown Chart. The vertical axis shows the remaining stress at the Sprint hours. The horizontal axis represents the days of the sprint. The Burndown must be represented by a line out of the start of the Sprint with the initial hours, down to the final sprint, leaving no time left.

Figure: Burn-down Chart.

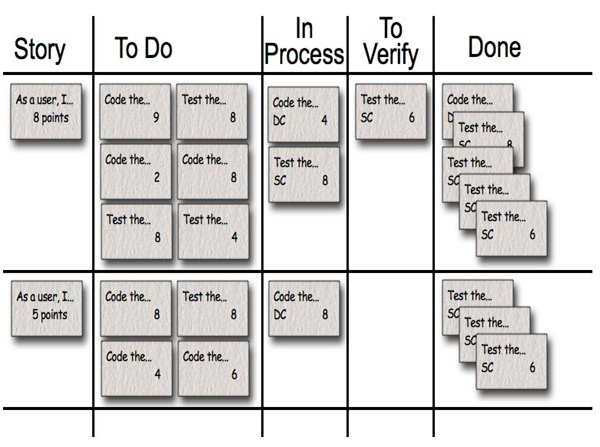
The team does its best to allocate the right amount of work in the Sprint, but

Sometimes during the Sprint Planning Meeting, it allocates activities more or more and, in these cases, the team needs to add or remove activities. In the Sprint Burndown Chart above you can see that the team has allocated activities more initially and still had approximately 600 hours to day 16/05/02. In this case the Product Owner was consulted and it was agreed to remove some activities of Sprint, which caused a large drop in the graph between the days 16/05/02 (619 hours) and 17/05/02. Since then, the team maintained a consistent progress and finished successfully Sprint.

Task Board:

Report is progress to team members and other stakeholders. The task board shows what all the team members are doing during a Sprint. It is continually updated during the Sprint - if someone thinks of some new task, they write in a new card and place it in the frame. Before or during a Daily Scrum, estimates are changed (up or down) and cards are moved around the board.

Each row in the Task Table is a Product Backlog item (in this example, stories). During the Sprint Planning Meeting the team defines the Product Backlog items that they can complete during the next sprint. Each row in the Task Table is a Product Backlog item (in this example, stories). During the Sprint Planning Meeting the team defines the Product Backlog items that they can complete during the next sprint. Each item in the Product Backlog is split into multiple items from Sprint Backlog. Each is represented by a task card that is placed on the task board in the "To Do".

Figure Task Board (Sprint Backlog)

The columns are:

· Story – Description of user requirement. Eg: (“As a user I want this application to sort numbers")

· To do – This column holds details of tasks to be completed.

· In Process – This column consist of details of the ongoing process and task manager.

· To Verify – The tasks which are completed are verified if they are as per requirements.

· Done – All the completed tasks are mentioned in this column.

Roles

Customers:

The customers play a fundamental part in the development forms. The clients are too called the users and product proprietor. They make the user stories for the item that the scrum team will construct. The customers alongside the group will choose which user stories ought to be created. The users are responsible for specific duties to the advancement group. An included client will keep any miscommunication and uncertain user prerequisites. The client should esteem the group's technical abilities and trust the group's supposition.

Moreover, the client ought not micromanage the group, yet permit it them to make the product without meddling in the process. The objectives set forward by the user should to be possible and practical as per the time period. There are two distinctive sorts of clients: inside and outside. Inside clients are individuals that work inside the organization and the item created is for the organization. The items produced for inward clients are not delivered out. Preference for creating an inside customer is availability. Besides, the development group could get a speedier feedback from an inner user. Thus, it is generally simpler to build up an item for inner users.

Development team:

A development group is critical to make a fruitful product. Each individual in the group plays a critical part inside the development procedure. Scrum groups contain fewer than 7 individuals. A small group permits individuals to facilitate together. Communication between colleagues is necessary. As indicated by scrum practices, colleagues don't have a title, however it is essential that the individuals can code and create tests for the program. The improvement group must have the capacity to remain focused to meet due dates and complete the delegated client stories before the end of every iteration. Individuals are relied upon to volunteer for assignments during each iteration and finish them before iteration ends. Thus, it is fundamental that the team collaborates well together and support each other. Also, the advancement group must coordinate with the product owner and scrum expert. It is the group's duty to develop a product as indicated by the product owner's vision. The improvement group must have the capacity to fulfill a large portion of the user’s needs.

Moreover, the development team should also consider the organization's business esteem.

Management

The management has the following responsibilities:

· Set product features;

· Decide the release date and content of the release;

· Reply by the profitability of the product (ROI);

· Prioritize features according to market value;

· Adjust features and priority every 30 days, as required; and

· Accept or reject work results.

The management is responsible for the first of three formalities

Practices:

The Agile procedure utilizes many practices to deliver software more beneficially. There is different practise process as listed below:

1. Pre-game planning and staging

2. Sprint planning

3. Self-directed and self-organizing teams

4. Scrum Meeting

5. Scrum master firewall

6. Daily build

7. Sprint review

**XP**

Work Products:

Workproduct in Xp consist of developing user stories, which are paper card with brief description of features request. It also consists of task list, which is the list of task to be accomplished. It also consists of index card, which gives a brief class responsibility and collaborative ideas.

Roles

There are different individual with different roles in XP programming.

­ Customers: Role of customers is to write user stories, which is the description of user’s requirement for the product. Using user stories acceptance tests are created.

­ Programmer

They define tasks from user stories and task estimation. They implement user stories and Unit tests.

­ Tester

They implements and perform Unit test and Functional test.

­ Coach

They monitor the process. They guide and help the process continuation.

Tracker

They keep track of process and help programmer with anything required. They communicate with programmer and coach to prevent the process going off track.

Practice

The XP process has 10 core practices:

1) Planning Game- Planning is a passionate minefield. It is a practice which Creates somewhat emotional separation from planning by regarding it as a game. The game has an objective, playing pieces, players, and standards for passable moves.

2) Pair programming – Pair programming is an Extreme Programming Practice in which two engineers take an interest in one development at one workstation. Every individual plays out the activity which the other is presently not doing: While one writes in Unit Tests alternate ponders the class that will fulfill the test.

3) Continuous Integration – The macro procedure of item arranged improvement is one of continuous integration. Every time your code enhances, you should send each refactor and addition that enhanced it to every one of your associates. These refactors should overlay their code, straightforwardly, as they write.

4) Test driven development – In test driven development when you code, exchange these exercises:

· include a test, motivate it to fail, and compose code to pass the test

· evacuate duplication

5) Design Improvement – The purpose of Refactor Mercilessly is to enhance the design of the code you are working with. Design improvement does not generally mean refactoring. At times it implies persuading individuals to delete features in the code that are not by any means required, or which will never be required for the present release of the item.

6) Small Release- Small Releases are a core routine of Xp. Every cycle is short and you just give the code to a little arrangement of functionality (e.g. you just add a couple User Stories every cycle).

7) Coding Standard – In this type of territories where individuals on a group frequently have diverse preferences and might need to concede to something. It's difficult to argue that any one framework has an immense favorable position over some other framework. But, consistency on some of these things has a major favorable position over irregularity: everybody on a group can read code effortlessly and make parts of the system refer to different parts of the framework effectively.

8) Simple design – The design strategy in XP is to make the simple design that meets your present necessities, as reflected in the present experiments. In numerous areas, "simple design" is uncertain; however it is an all-around characterized term in XP.

9) System metaphor – The way, Extreme Programming (XP) uses to bring together an engineering and give naming traditions. A basic shared story of how the framework functions is called metaphor. This story normally includes a bunch of classes and examples that shape the center pattern of the framework being manufactured.

10) Sustainable pace – The idea of sustainable pace is that efficiency does not increment with hours worked. Tired developers are less profitable than very much rested ones. Teams in venture banks, new companies, and "quick moving" situations should know without a doubt whether working 100 hour weeks is very beneficial than some other number.

11) Collective Code Ownership - Extreme Programming considers code to have a place with the project, not to an individual architect. As designers create required functionality, they may skim into and modify any class. They are in charge of keeping all the Unit Tests running and composing new ones for new functionality.

**3.0 Pre-Game: Planning and Staging**

Pre-game planning is part of scrum life cycle which consists of establishing vision, setting expectation, secure funding and prioritizing the requirements. There are different activities as explained below.

**Vision**

Description of application:

We as a group will be developing an android application for insertion sort. Application will take integer inputs (range 0-9) from user in text field. User has to enter minimum of 2 inputs and maximum of 8 inputs. After entering the inputs the user will click the sort button which will then lead to sorting of the array integer. The result will be sorted and each iteration of sorting will be displayed followed by the final result. If user doesn’t follow the threshold values required by the application an error will be displayed accordingly. Application also consists of exit button for user to exit the app.

Goals of project:

The goal is to successfully develop an android application as per requirements and constraints. The application should be as per user requirement, tested and without bugs. Application will be created using Scrum and XP methodology. The application should be flawless and completed in defined time frame. The project is tested so it functions as per requirement before deployment.

Business/Organization Goals:

To create application by following Scrum and XP process. We should define a process as per Scrum and XP methodology which will help us to develop the application as per requirement. We will develop application in iterations so that we can improve the app functionality in every iteration release.

System Features:

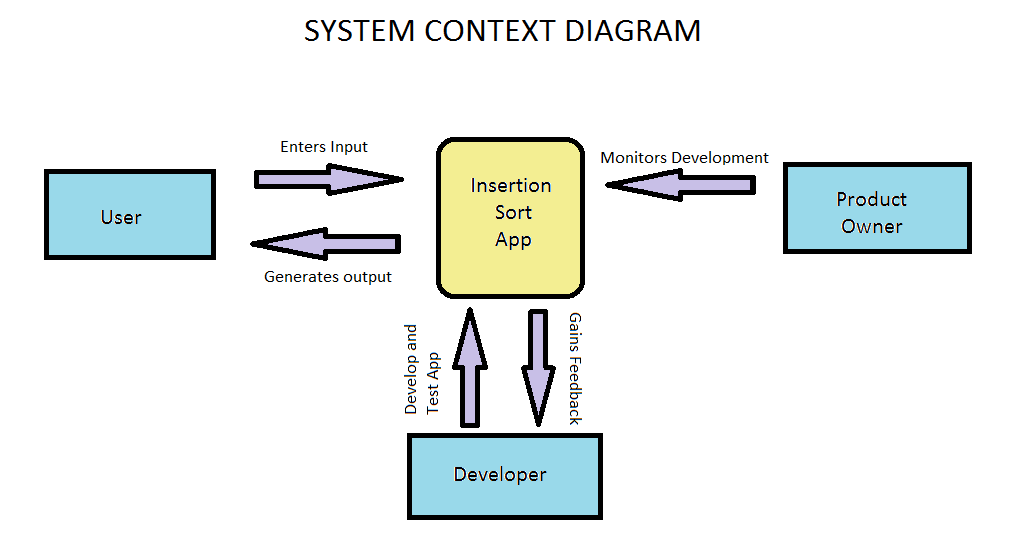
System feature of application should have the following requirements.

1. Application for insertion sort.

1. User can enter single digit numbers between 0-9.
2. User has to enter 3-8 integers.
3. User has to confirm input by clicking button.
4. Output will display all iteration of sorting process.
5. User can reset input fields to enter new integer for sorting.
6. User will be able to exit the application.

System Context Diagram

System context diagram is a diagram which defines the boundaries between system and environment and shows how the entities interact with each other. It shows a high level view of the system. In the below diagram the yellow part is the application which is built in android studio, which takes input from user and displays output on click of button. Product owner generally monitors the development of app and guide team to the track if misguided. Developer is the one who creates the app via coding and functionality of methods. User finally is the one who uses product; they interact with app as providing input and viewing the output. There are many factors who also plays important in the development process which does not comes under system context diagram because it is generally used to provide the overview of interacting entities among themselves.

Figure: System Context Diagram

Stakeholders

Stakeholders are the individuals who have interest and are the partial owner of the project. There are different types of stakeholders which are CEO, managers, sponsors, customers, scrum team and many more. Scrum team is the crucial stakeholders in a project. The team includes owner, Scrum master and the development team. Each of these three individuals assumes a key part in the advancement of the item. They choose the achievement or disappointment of an item. The CEO and managers inside the organization don't play a direct part inside the improvement procedure. In any case, they ought to have trust that the Scrum group can self-manage to make an item for the clients. The clients could be interior, for example, when the item being made will be made for the organization. Sponsors are a partner since they give the cash and assets to the group to make their product.

**Requirements:**

It is divided into 3 categories:

1. Functional Requirement

2. Non-Functional Requirement

3. Architecturally Influential Factors

Functional Requirements:

Functional requirement are the requirements which makes the application work flawlessly. It consists of the essential requirement to be performed by the application. Functional requirements are decided using user stories. Following are the functional requirements we used for developing the application:

1. User should enter input for sorting.

2. User should provide acknowledgement to sort integer by clicking button.

3. User should be able view all iteration of sorting.

4. User can retype integers for sorting without restarting the application.

5. User should be able to exit the app.

6. User should enter minimum of 2 and maximum of 8 integers.

To describe in detail 1, 2, 6.

The FR1, FR2 and FR6 are the most crucial requirement of this system. The insertion sort application will not perform sorting without these requirements. Thus, these requirements must be added in the first sprint.

Non-functional requirement:

Non-functional requirements are requirements which describe the application and the minimum requirements and constraints that it must follow. Its consist of different attributes and constraints ranging from performance, availability, efficiency.

1. User should see error message if input integer is not between 0-9.

2. User sees error message if inputs in less than 2 or exceeds 8.

3. Application should perform smoothly.

4. Application should be available every time.

5. Response time should be minimum.

6. Restore time should be minimum.

7. Application should run on Android platform.

In this NR1, NR2 and NR4 are the important priority in nonfunctional requirements as these are the minimum requirements of the project for the application to run without any problem.

If NR1 and NR2 are necessary because if this condition is not satisfied it then it will affect the system’s design.

Architecturally influential factors:

Architectural Influential factors include risks, availability security, and modification, functional and nonfunctional requirements which might damage the system’s architecture. The Architectural Influential factors are explained below:

AIF 1: Cost is the important factor in the AIFs as without efficient money it is difficult to develop a project. If the funds are low then it difficult to develop a project with

Good system architecture. The customer’s won’t get the desired application.

AIF 2: Modification is the another AIFs which might affect the project. Modification is necessary for the project development. But, if some major changes are done in the project then it might affect the system architecture. Thus modification should not be done without considering system architecture

AIF 3: The third AIF which might affect the system is the security. It does not include protection for the application but also for the customers who are accessing this application. The application without security is not safe to deploy. Customers are first concerned about the security of the application.

**User stories**

Planning:

User stories are hand written or written on index cards. User stories specify a brief about the requirements to be considered for developing the project. User stories represent components of the application that the users want. Scrum team should be able to finish their user stories. As indicated by XP standards, the user stories ought not to be point by point of the fact that a client representative should be accessible for illumination. It is the duty of the product proprietor to pick which user stories to organize and select the stories that the team ought to take a shot at for every emphasis. Also, client stories can be changed or disposed of as the procedure requires.

Our team created six user stories to develop the insertion sorting application. These stories were written on index cards. These user stories were then separated into litter errands for simpler time estimation. Then the development team decided which user stories must be selected for the first iteration. When we are done with this iteration then feedback from the customers would be taken to decide which user stories are necessary and which are not.

List of User Stories

User stories are written by user to about what the system should do. User stories are reviewed by product owner and provided to the team members. User stories are listed and sorted as per priority and time estimation to achieve. We created multiple user stories which were the listed and release plan was created, using scrum we implemented 3 iterations, and as per iterations user stories were divided. User stories used for our project are listed below.

US 1: As a user, I want to create android application.

US 2: As a user, I want Application to sort integers.

US 3: As a user, I want this app to run on all phones and tablet.

US 4: As a user, I want button to display the output.

US 5: As a user, I want app to take input from user.

US 6: As a user, I want placeholders for text fields.

US 7: As a user i want reset button to clear field.

US 8: As a user i want this application to display every step of sorting.

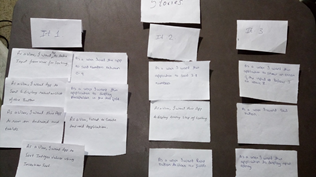
US 9: As a user i want this application to sort 3-8 numbers.

US 10: As a user I want this application to show an error if the input is below 3 or

Above 8

US 11: As a user I want exit button

US 12: As a user I want this application to display input array

Figure: User Stories.

Product Backlog for Estimation:

The Product Backlog is the expert list of all the desired elements in the product. Product backlog needs to be created during a meeting. As discussed all the elements should be in application. Product backlog should be visible to everyone for better planning.

It changes various ancient requirement artifacts. The product owner arranges task in the product backlog. The owner decided various tasks and prioritizes them in order to complete the first iteration.

Product Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Type** | **Priority** | **Estimate(hr.)** |
| Create UI | Feature | Critical | 2 |
| Create a Insertion sort method | Feature | Critical | 3 |
| Create input and outputs fields | Feature | Critical | 30 min |
| Create Sort button | Feature | Critical | 30 min |
| Convert input to string | Feature | Critical | 2 |
| Unit test for insertion method | Test | Critical | 3 |
| Create reset button | Feature | Critical | 1 |
| Create exit button | Feature | Critical | 1 |
| Display all iteration of sort | Feature | Critical | 2 |
| UI unit testing | Test | Critical | 1 |
| Create an error toast for inputs | Feature | Critical | 1 |
| Enhance UI | Enhancement | Critical | 30 min |
| Create an icon for app | Enhancement | Minor | 1 |

**Architectural Spikes:**

The purpose of architectural spike is to recognize the area with risk. We must find the solution of risks before the development stage .Once the risks are recognized, and then we found a better solution of how the task will be completed. For architectural spike we must define system metaphor.

System metaphor

The purpose of system metaphor is to understand a particular part in a better way. We must understand how the system works and we have to build it. In this project we will understand how we build the insertion sort app on android studio and allow the user to perform various functions in our application.

Before the first iteration, we discussed various risks in our application. This helped us to understand the risks in our project and prepared us to handle various problems which might occur in iteration.

Release Planning:

Release planning is the discussion of which iteration to be released on which date and what are the are changes in the product. Before the first iteration, we discussed estimated start of the a particular iteration and the end date. We also discussed the estimated type of changes and requirements in each iteration.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration Stage | Functionality | Start Date | End Date | Plan |
| Iteration I | Basic Functions | Sept 8 | Sept 12 | Create application that lets user sort integer onclick of button. |
| Iteration II | App Enhancement | Sept 13 | Sept 19 | Create application which shows all steps of sorting and reset button. |
| Iteration III | App Enhancement | Sept 20 | Sept 24 | Added app icon and exit button. |

**Technological Preparation:**

Working with android studio:

Insertion sort application was our first project in android development. We downloaded the latest build of android studio. During installation we added supporting packages which were needed for the project. We were not familiar with the tags and labels which were used in android studio. With the help from youtube tutorial video’s, we understood the concept of how to use and develop application in android studio. One of the major issue which came up was we were not able to run any program on android studio. The issue was we both had AMD processors and android emulator didn't have support for AMD processor. They recently extended their support for AMD so after surfing the internet and finding for solution we came to know the way to run emulator on AMD. For that we had to download supporting package for the processor which is ARM image, but it was very slow compared to the intel processor. After developing the project on AMD for some weeks, we decided to find a friends laptop/pc with intel processor to run android studio. We finally borrowed a friends laptop and continued the development of the project. We then setup AVD (Android Virtual Device) again on new laptop to use an actual android device, it was much faster to load the AVD and app now. Intel is 10x faster for using android studio compared to AMD. By this experience we also learnt that many programming software runs slow on AMD.

After setting up android studio we installed all the supporting packages, HAXM, android api’s as we were familiar with the installation. We also explored many tiny things in android studio such as changing theme, adding line number to code and many small things. Overall it was a good experience developing our first application.

Developing and Communication:

As of developing the project and communication, we use to communicate via whatsapp, emails and google docs for writing our report simultaneously. We were not able to host meeting everyday as other work and projects. We mainly used to communicate via whatsapp. Both of us used to check our message daily for the task assigned.

As a part of generating documents and reports we use to use google docs. We created our report on google drive. So if one member makes any changes in the report the other member can view the changes instantly. We can even see what the other member is working on right now in the report which helped in developing the report efficiently. Everyone read the document if anything new was added and continued with their work. We were also able to add comments in google docs unlike desktop word application, which helps each other to know about the work pending or any changes to be updated. There are no special skills required to learn google drive. Their functionality is same as of Microsoft Word. We were also able to add comment in google docs at any particular part to write down review or any pending work.

**4.0 Development**:

**Iteration 1:**

Our team consists of two member. So each roles were divided among the two of us. Chirag was the scrum master. Prathmesh was the product owner.In the first iteration Development team was done in paired programming as there were only two members

User stories

US 1: As a user, I want to create android application.

US 2: As a user, I want Application to sort integers.

US 3: As a user, I want this app to run on all phones and tablet.

US 4: As a user, I want button to display the output.

US 5: As a user, I want app to take input from user.

US 6: As a user, I want placeholders for text fields.

Before the first iteration we came up with various user stories as listed above. These stories were then listed and then the owner selected necessary user stories which were required for the first iteration. The product owner then prioritized the user stories and assigned the task to the team to develop the insertion sort application.

Iteration Planning

Planning of the software product is done to ensure that we are working on things that we need to do. Everyone in the team needs to coordinate with other people. If unexpected event arises we need to understand the consequences. The initial part of planning is to create user stories. They are written by customers as well as product owner to explain what the application needs to do. Release plan is created to assign task for each individual. Scheduling is done for each user stories that are created. Release date and time when each user story must be completed is done in release planning. The software application is divided in different iteration and the product is released at each iteration.

The different user stories which were created are first approved by the development team. The product owner then prioritized the tasks. Then we planned our meeting dates for each iteration. Then the estimated time needed to complete the first iteration was then calculated.

Sprint Backlog

In sprint backlog we created different columns like what is left to do, in process, verified and tasks which are done. The product owner chirag then selected the important task and then all of these tasks were placed in the to do list. The task which started were then added to the in process column and when processing was done then they were added in the verification task. Once the if all the process were completed then each task are placed in the done list.

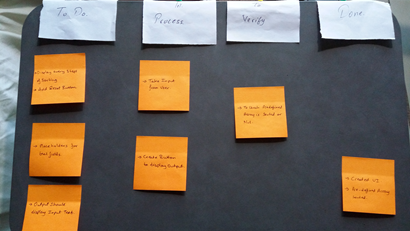


Figure: Sprint backlog after Iteration 1.

Managing

One of the important factor that is required is open work space. There should be enough space in the room to hold standup meeting and whiteboard as well as conference table is necessary for group discussion. The management should make sure that the application is working at a sustainable pace. Stand up meetings should be done everyday and each member of the team must contribute in each meeting. The pace of the project must be calculated. Each task must be completed according to the given period of time. Cross training is required in each project development. We did pair programming for different iteration to reduce load. Fixing the particular process in XP is necessary.

Our team had only two members so different work and task were divided equally. We used to work in a closed room. We were not familiar with android studio. So we spent our initial time understanding android studio to develop our project. Then as we were clear with the concepts we then started working at steady speed. We decided our working hours and in those working hours we took breaks. So that we don't feel tired or bored to work on this projects.

Designing

Simplicity is the main type for a successful design. We should start with simple design if we need a better software application. If there is complex design in the application then we break it down in simple part. There should be architectural structure. The another form of simple design is create easy creating standard classes and methods. CRC cards must be included to move from procedural mock object based technology. Spike solution must be created to solve tough design. Refractory code must be used to develop a successful application.

In the first iteration we had to create a functional user input insertion sort application. The application included a field where the user would type his input and then click on the sort button and the desired sorted output would be displayed. The system metaphor helped us to create a simple description design. We developed the application in android studio with all the necessary user requirements. We also created risk solutions and ensured that the system is working properly

Code

Customer must be available not only during coding phase but should be available at the whole development stage. Coding standards must be maintained to keep the code consistent and simple for each member in the team must understand. Paired programming must done so that more ideas and simple methods class are created. Unit test must be performed at each and every iteration. A single computer must be used for sequential release when development group is co-located. Collective ownership should be implemented so that each and every member ideas would be shared in the project.

We started coding at a sustainable pace and considering the required task that are necessary for the application. We made sure that each task was completed according to the given period of time. In this iteration we were able to finish the required application functionality. Pair programming was conducted so that we both could express our ideas and views which helped in enhancing the project.

Testing

Every Time a code is written unit test must be performed at every iteration. Unit test framework must be created for automated unit tests. Each code must pass unit test before release. Test must be done if there a bug in the application and should be cleared before moving to the next task. Acceptance testing must be carried out at every iteration before the release. Acceptance of the project from the customer’s as well as the whole team must be done.

We made sure that the user interface had a simple design and not complex form which would be difficult for the user to understand. The position of each button and other inputs were in the right position. We did not insert any input and clicked on the sort button. Execution time taken for the application to run was then tested. The response time taken to sort the numbers was checked. We performed unit testing to check functionality within the class. We also verified that the insertion sort class is operating as required.

Small Releases

After the testing phase, we released our product so that we can get feedback from the users. The feedback from users will help us to find bugs or any problem that is occurring in the application. This feedback will help us to make necessary changes in the application and enhance is to make the application better and efficient.

Daily Scrum meetings and Sprint Review

We decided to organize our scrum meeting in the afternoon at around 2pm everyday as both of us were available at that time. We discussed various topics that are required for the development of the application. The topics included related to the design, coding and many more. We would discuss things like what we did in our last meeting? What are the things we are going to discuss in the next meeting? What are problems arising while development?

These were topics that were discussed in the scrum meeting.

In sprint review our team considered which all part of the process was good for the system or which parts were bad for the system. We compare sprint backlog and product backlog. We would discuss what to do in the next iteration and all changes need to be made.

**Iteration 2**

This iteration is about enhancing the given application. The product owner and scrum master for this iteration is chirag and prathmesh will handle development. In this iteration we have improve the application by customer's feedback and product owner.

User Stories:

US 7: As a user i want reset button to clear field.

US 8: As a user i want this application to display every step of sorting.

US 9: As a user i want this application to sort 3-8 numbers.

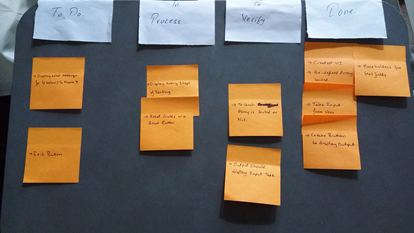
The seventh user story was suggested by the customers. A lot of customers were having problem to use sorting application repeatedly as they had to exit the application, if wanted to use it again. Everyone agreed to this issue to provide efficiency for the customers.

The US8 was also suggested by the product owner so that it would easy for the customers to understand each and every part of insertion sort. The customers can even understand how each integer is compared and how insertion sort is performed. Both the member in the team agreed to his user story and decided to implement every step of sorting.

The US 9 was also suggested by the product owner because during the testing of first iteration if there were more than 8 integer it displayed garbage value. Hence, this limitation had to be implemented in the application.

Sprint backlog

As in this iteration we changed the role and product owner was played by prathmesh, where he decided some task which was then executed by developer chirag. We used the same format as shown in above iteration.

Figure: Sprint backlog after iteration 2.

Managing

In this iteration the programming part was handled by Prathmesh. While chirag used to review the code that was programmed. The reviewer must be vigilant for any change in function or class that was not on purpose. Chirag also handle different task and made sure everything is working efficiently.

Design

In this the development team had to add another button in the design. Thus by considering the system metaphor the reset button was created. The thing was needed to be added in this iteration was detailed steps of insertion sort. Chirag made sure that the iteration of insertion sort was displayed in a proper way.

Coding

In this iteration prathmesh had to apply append function to display each steps of sorting. Then button needed to be added to reset the text field and error message to be displayed if the user did not type numbers between 3-8.

Testing

In this phase the testing of the reset button was checked whether it is functioning according to the way it was desired and even all the steps of sorting were displayed properly. Unit testing was performed for the number limit which we had to apply and made sure that it displayed proper error message if the input was not as desired.

Small Release

In the second iteration, we released the insertion sorting application with the new button and some detailed sorting steps.Some limitation was also added for the application to work efficiently.

Daily Scrum meetings and Sprint Review.

In the scrum meeting we discussed the contents we could add in the application and to improve the performance. When all the tasks were completed the acceptance testing was done.

**Iteration 3**

In the the iteration we tried to improve the efficiency of the application. User can see error message if they wrote something wrong in the input screen so it helps users understand what is necessary. Prathmesh was appointed as scrum master and product owner while chirag handled the development part for this iteration.

User stories

US 10: As a user I want this application to show an error if the input is below 3 or above 8

US 11: As a user I want exit button

US 12: As a user I want this application to display input array

US 13: As a user I want application icon

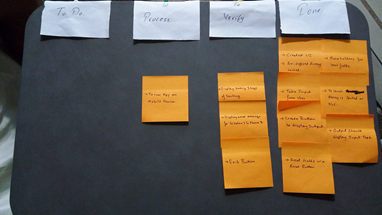
The product owner decided to show error message which will help the users to understand what he needs to type in the application and the limit of the application input. Prathmesh also suggested adding details in the error message so that the user can understand their boundaries.

US 11 was to improve efficiency of the application. User were not able to close the application and if the clicked the home button the application would just minimize. The task of closing the application used to consume time. So it was necessary to add the exit button. Each member in the team agreed to this point.

After the second iteration user gave a feedback to add input array in the output screen. This story was not that necessary as they can see the input in the input screen. But it was the need of the customers and would be convenient for the customers to view exactly which input was sorted.

Sprint Backlog

We used the same format as shown in above iterations and after this iteration our to do list was empty as shown in figure below, which means the product was as per user’s requirements.

Figure: Sprint backlog for final iteration.

Managing

In this iteration the management part was not much of a burden as we were as the final iteration. The team had to add some minor of necessary changes in the application to improve the efficiency, consistency and convenience of the application. Prathmesh just had to make sure that every change was in its right place and didn’t affect the software framework.

Design

We had to make changes in the user interface. We had to add exit button in the application. We made sure that it didn’t affect our system architecture. System metaphor was also considered to keep the design of the application simple.

Code

The developers had to add a user input array in the output field. For this text field was added to display the output. Users input was converted into an integer array and then displayed.

Testing

In this phase the testing was basically manual testing. The positioning of all the elements was checked. The testing for error message was done with unit testing. The error message was checked if it was showing the message at the right time.

Small release

The third iteration was about improving the efficiency of the product. Some new features like exit button, error message, displaying input array and icon. This iteration improved the design and made the application looked enhanced. The regularity caused the application to be convenient to the user.

Daily Scrum meetings and Sprint Review

In daily scum our discussion was about improving the efficiency, design and convenience for the customers. We discussed various issues, pros and cons of implementation of various attributes.

We then reviewed our final product, checked its compatibility and every team member agreed to the final products structure and functionality. Each stories that were implemented were reviewed.

**Acceptance Testing**

Acceptance testing was carried out after each iteration. They used to make sure that the product satisfies the needs of the user. Acceptance test was done by our team pretending as customers. We even had some volunteers who did the acceptance testing for our insertion sort application. Everyone checked for any bugs while running application is running. The user interface of the application was working properly. The input and other popup messages were functioning as intended. All the functions were running properly and the desired output is displayed were tested for a full functional application.

**5.0 Release**

**Documentation**

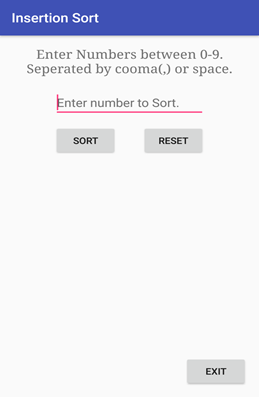
While developing the project we used lots of references for learning the agile process and android studio. We developed the app on android studio; we took help of YouTube tutorials as this was our first android application. For developing the report we used google docs which help more than one people to work on same file. Users can update and edit at the same time, also users can comment on a particular part which can be referred by others.

**Training**

Our training aim was to teach the user how to use the insertion sort application. The customer must understand the different functions of the application. What are necessary measures that were required to run the application and how to use is conveniently? Such forms of training were provided. We arranged a training session at specific intervals which would help the customer understand the application if he did not understand certain functionality. Customers would come with various queries in different session. For the training session we had different goals so that we know what should be conveyed to the user.

**Operating**

For the Insertion sort to operate user needs to install the app on android system and open it. User will see the interface as shown in the pic below.

Figure: User Interface of Application.

First is the field saying users to enter the numbers. After entering the number user can click sort to sort the numbers. User can also reset the input and output field via reset button. And finally the user can exit the app via exit button.

Sort Button:

This will sort the integers entered by user and show every steps of sorting as output.

Reset Button:

This will clear the input field for the user to enter new values to sort.

Exit Button:

By clicking this button user can quit the app easily.

**Feedback**

Our project received favorable response from users and volunteers. They did not find any bugs in the application. The application design was very simple so it convenient for the users to use this application efficiently. The functionality and the methods were not complex too. The users are satisfied with the insertion sort application.

**6.0 Other Activities**

**Estimation**

Estimation occurs when product backlog is created. Whenever we added new content in the product backlog, each product needs to be estimated. It is the most important part of the software development cycle. Without estimation it not possible to control and plan effectively.

If there is no estimation then we have to suffer more than estimated. The four basic steps in project estimation which are as follows

* Size of the product
* Effort
* Schedule
* Product cost

Calculating the size of the product is the first step for convenient software estimation.

We did the estimation size by considering various factors which are customer requirement, system requirement and software requirement. The backlog included various product like the exit button, reset button and many other contents that were customers need to develop the product. We had issues with the system requirement as the software application which is the Android studio was not efficient with AMD processor. The application was very slow and would take about 10 minutes for the emulator to open. A individual volunteered his INTEL processor laptop for our project to finish the project in time.

Another form of estimation is the effort taken to complete the project. In this the coding part is not the only part which is considered for the effort. It also includes efforts taken in design, development and testing the software. Content writing, reviewing data, designing framework and testing the written code were done by our team to achieve our desired goal. We estimated necessary efforts to be taken in a specific field. We were not familiar with the android studio coding. So efforts taken to understand the concepts and methods took lot of effort in the project development. Reviewing was another effort which took time for development. Reviewing each iteration and trying to find bugs in the application so that user won’t have any errors while operating the application. Efforts taken per person in terms of hourly basis were also calculated so that the application is ready according to the desired date.

There are many factors to consider when estimating the total cost of a project which include labor, hardware and software purchases or rentals, travel for meeting or testing purposes, telecommunications, training courses, office space, and so on. The product cost estimation was not calculated by our team as it was a very basic application which does not require enough resources.

Scheduling is the important form of estimation for a successful software development. It is calculated from the effort estimation process. We estimated who work on what part of the project and what is estimated time taken to finish that part. Each individual were divided in different part for each iteration. There were some parts were it would take time to complete the task. Then we made sure such task should be completed in the desired time. After allocation of task we created a calendar schedule which gave an idea when our application would be complete.

**Project Planning**

Humphrey, 1990 defined project plan as the work and how it will be done. Planning helps in organizing various parts of the system which is the estimation, resources and framework. Before developing the application, we planned the whole software process for the application run effectively. Various factors were taken into consideration while developing the software which is estimation, necessary resources, review, monitoring as well as control, configuration and risk management.

We first estimated the amount of resources, time and effort required to develop the application. There were a lot of resources which included various the requirement of the customer, software specification and system requirement. We had some customer requirement which were applied in the application for the user to access this application. The various forms of request from the user were removed for different reasons. The scheduling of the project development was another important part of estimation. We schedule our project according to the form of per person on hourly basis. After the scheduling is done, then the resources are allocated and tasks were assigned to each individual for development,

The monitoring and control of the application is another important part of the project planning. While developing a software application it is necessary to monitor and control the application to make sure the particular application is working according to the desired plan. Thus, monitoring was done at end of each iteration and control at start of each application.

The development team needs to be controlled and guided with what they are doing because there are times when a particular coder will do the part which he knows best and leave the part for the preceding stage. This, forms would create problems for the application and will also affect the architectural design. Efficient training needs to be done to tackle such situation.

After each iteration the project should be monitored and should be scheduled when the project monitoring needs to be done. Planning the monitoring process will help in giving the desired result.

Risk is another factor which might affect the system framework and code of the application in an adverse way. Thus planning the expected risks would help in avoiding various forms of problems. Before starting application development we listed the amount of risks or problems which might occur while developing the application. There were various risk which would have occurred while coding, executing application and many more. We made sure these risks were noted before starting that part of the project.

One of the most important part of project planning knows the goals and objective of the project. Every member in the team must know what he is doing and when a particular task needs to be done. They must know each and every phase and every part of the iteration which would be beneficial for the software application development.

**Project monitoring and control:**

Project Monitoring is collecting, recording information about project performance that project manager wish to know. Project controlling is using available reports and documents to monitor activities to ensure that project is flowing as planned.

The purpose of project monitoring and control is to track the activities of project and to ensure if the desired output is generated. By asking certain questions we can we can keep track of our project, if it’s going as expected or not, as shown in the figure below. The benefits of this process are to mitigate or avoid the risks and react appropriately. Generally the project manager helps to keep the project moving and allows necessary adjustment. Project manager should be kept up to date with the team progress and performance. The project goals, timeline and budget helps to ensure if the project is moving as planned. What do we monitor? We generally monitor human resources, materials, money, space, time and tasks.

Project monitoring can be done through a group meeting with the parties involved in project. In meetings using the backlog we can decide if the project is going as planned or not and can take actions as needed. Also the various factors are checked including time, money and estimate if the pending tasks can be completed in budget and timeline assigned. Project monitoring and controlling is a constant part of the monitoring and control.

Figure: Questioning techniques for project.

Keeping the vision in mind, while developing the sorting app we created and certified that all the activities are up to mark. User stories were reviewed at every steps to ensure that the project is as per user requirement.

Scrum meeting were held to check what was done and what was pending with the help of sprint backlog and tracker and if all the task completed are up to user requirement or not, all the resources were checked. With the help of sprint backlog we were able to monitor the progress of work. With the help of sprint backlog the task which were required to be done were decided giving every task its priority. As, we were group of two it was easy to communicate but was difficult at the same time to implement multiple tasks simultaneously.

After each iteration the reports and resources were provided to project manager, which then decided the process to be followed for next iteration. Also question were asked at each stages to keep track of the available time and resources. Accordingly, the schedules were decided for next iteration. Also the risk factors were taken into consideration at every stage, for identifying the risk and mitigating it within the decided timeframe.

After development of each iteration, Testing was done to know the bugs and to improve the project efficiency and make it bug free. We did testing on every components buttons and fields and were faced with many small and major bugs. We were developing the application for android for the first time, and so we also faced issues and bugs which were easy to solve and some of them took time more than expected. There were many bugs identified during each iterations, and all bug were successfully eliminated.

**Configuration Management**

Software configuration management is the process of tracking and controlling the software application. Humphrey stated that changes made to the requirement drive the design, and design changes affect code. Thus management must make sure the changes made in the application are not any major changes which affect the system architecture. The changes that are made should be discussed at the start of each iteration. Even if there is a small system there should be a changes management system which monitors the changes.

Humphrey defined some keys task for SCM which were implemented in our project. They were as follows

* Configuration Control
* Change management
* Revisions
* Versions
* Conditional Code

In configuration control different code must be created for each iteration. This might lead to consumption of space but helps to keep the code controlled. In any iteration we code we made sure we kept a separate copy of code. So when two members work differently in the same code then different solution arrives to the same problem.

Humphrey stated that development is thus a repetitive cycle of revision, integration and test. Anytime any process is performed it necessary to do a revision as well as check for errors and integrate that process. After every iteration we reviewed our application and made sure that the desired output is same as we decided and checked for different errors or faults in the software application. There times when we used to not get the desired result and thus revision helped in identifying the problem.

While developing the application we made sure that any changes that need to be done were performed before the iteration. So that it will not affect the system architecture. If there are changes in the application the change management helped us to solve the problem by working separately on the same issue and then discussion to solve makes that particular change.

There are various issues with iteration changes. There are many different changes that are made in the same module. Thus it is necessary to select the right module for each iteration. Only that particular module should thus be selected. For example we had different forms of input field to apply in the code. But it is necessary to select the right one even if the other has the same functionality. This would help to solve risk or errors in the future stages of iteration.

Conditional code is another form to handle variations between modules. This helps in solving various form of versions and handling change management.

**Quality Assurance**

Software quality assurance consists of a means of monitoring the [software engineering](https://en.wikipedia.org/wiki/Software_engineering) processes and methods used to ensure quality and the types by which this is accomplished are many and varied, and may include ensuring conformance to one or more standards, such as [ISO 9000](https://en.wikipedia.org/wiki/ISO_9000) or a model such as [CMMI](https://en.wikipedia.org/wiki/CMMI) (Wikipedia, 2016)**.** It also includes deciding the quality of the application at the start of the project. The purpose software quality assurance is improve the quality, to ensure the software product has established certain standards and any inadequate resources that are missing can be fixed

For quality assurance in our project was check by various ways. We used handle others workload so that we can monitor other’s work. This helps in testing the application. The individual can even understand the other person’s work and this will help in the better understanding of the project.

The software quality must be checked by certain standards which include functionality, usability, reliability, performance and reliability. Checking the function of the application means checking out how it does it do? Instead of what does it do? Usability of the software is tested by considering accessibility, consistency with the user interface which would help the user to access the application conveniently. For assuring the quality we must check if the application is reliable. We must test the accuracy by checking the output is displaying as desired, check the necessary resources are available while running the application and if some data is lost then does the application has the ability to recover the lost data. Another important function which is necessary is whether the application supports the necessary standards.

For our application we carried out some these necessary functions to check the quality of the software application. We checked if all the necessary methods, functions and tags are available to the user and the user can understand these necessary functions. The user typed insertion sort is displaying the right output was checked. The another function is the recoverability of the application. This function was not possible for our application as there was no stored data in our application. The products adaptability in different platform was checked as well as the compatibility. Another important factor in the software quality was testing the products response time to give the desired result and time required to start the application was also checked.

The stages are categorized into areas for requirements capture, system design and coding and testing and finally release.

* Verification
* Validation
* Qualification

In the verification process it evaluates a system or component to check if the product of the given phase satisfies the condition with the start of the iteration. At every iteration we checked for the given product backlog and the sprint backlog to assure we have entered as we required. The process of evaluating the application is known was validation. At the development stage we used to make sure that all the component satisfies certain requirements. The last phase is the qualification phase. In this phase we checked the particular system or component is suitable any kind of operational use.

The third step in estimating a software development project is to determine the project schedule from the effort estimate. This generally involves estimating the number of people who will work on the project, what they will work on (the Work Breakdown Structure), when they will start working on the project and when they will finish (this is the “staffing profile”). Once you have this information, you need to lay it out into a calendar schedule. Again, historical data from your organization’s past projects or industry data models can be used to predict the number of people you will need for a project of a given size and how work can be broken down into a schedule.

Risk Management

Risk management is necessary in software process as it helps to keep finding the risk and help to mitigate it. Risk in software can be of many types and needs to be solved for an efficient project. In software process risk management goes as follows:

1. Risk Identification and analysis,

2. Risk Comparison,

3. Prioritize,

4. Mitigate,

5. Monitor.

Figure: Risk Management Factors.

In risk identification step risk in the software process or software is identified and listed down to take further action. Followed by that the listed risk are compared among each other which then helps to find the priority of the risk to be solved first. After the risk is prioritized the risk is mitigated and monitored if it’s solved or not.

Risk management actually helps improving the project. The risk can be classified into two types, one which can be avoided and the other which needs attention. The risks which can be avoided are generally the risk which does not affect the project and project timeline. Those risk falls out of the scope of project. And the risk which needs attention cannot be overseen because it can affect the whole project and project planning. If any critical risk is avoided it can lead to huge damage and the software process may need to be restarted.

According with project monitoring and controlling the risk management was considered at every iteration of our process. All the risk were first identified and listed down. Risk planning was done and after that all the risk were compared among each other thus generating the priority of risks as threat to the process development, the risk which needed attention first were kept on first priority to be solved. After that risk was analyzed, using proper techniques risk was eliminated and as per the risk next iterations were decided. During our scrum meeting, we use to look over and identify the risk and with proper monitoring and control revisiting the risk we were able to mitigate the risks. After the mitigation of risk we have to sure that the risk won’t occur again and so proper monitoring is necessary.

**7.0 Lesson Learned:**

Prathmesh Pardhiye:

Before starting this project, i did not have any experience in software process. I didn’t even know there were different types of process. I’m glad that i took this subject in this course. The first process which I learnt was scrum and Xp. It helps to build a software application in an fun and easy way. It was difficult for me to learn Scrum and Xp as I did not have any experience. There was no specific structure while developing the application. There were some rules for developing it. I made sure that I didn’t violate any principles while developing Scrum and Xp. The daily scrum meeting which we held was knowledgeable and useful. Sometimes we did have hosting our daily meeting because of other work. But we managed to attend as many meeting as we could. It helped me learn why daily scrum meetings are necessary. It helps us share work we did and it helped each other in solving various issues while developing. At the initial stage of the project we had to define process. It helped us understand the how the process actually works. We got the understanding of how release a product in forms of iteration rather than a final product. User stories were also the important part of Scrum. Not all user stories must be selected while developing scrum. Selected user stories were divided into different task and to add them into each iteration. I didn't know programming could be done in pairs. It helped me learning new libraries and to organize code. I had never worked in android studio. I had to learn from scratch. At the initial stage i had difficulty learning android studio. But later it was all smooth. Software development process taught me the importance of time management, how to work as a group, how to manage things and organization.

Chirag Padsala:

As i was new to the software development process and agile methodology, it was very challenging and quite difficult but after completing this project i got better insights about Scrum and XP development process. It is a good and enjoyable process and also compared to other development process it let us complete the project quite fast. I also learned using both the agile methodology together.

Using Scrum I learned that there are three roles to be played. Product Owner, Scrum Master and Development Team. As at each iteration we switched our roles it was a good experience of playing all three roles. As a product owner i learned how to come up with user stories to the team. As a Scrum master i learned how to work with product owner and maintain product backlog in shape for next sprint. Also it helped learning how to make team perform at highest level possible and conducting meetings. As being a part of development team I learned that it is important to focus on product backlog and user stories. Also i learned how it is to work in a group. We had to release app at every iteration as a part of development process and was done successfully.

Using Extreme Programming I Learned how to manage risk, spikes and mitigate it. I also learned pair programming, which helps a lot to learn from the other team member. It was very hard for us to work on android studio as it was our first project on this IDE. We first learned android studio with help of youtube tutorials. After which we started developing the application. Using XP we have to maintain code standards which are actually good as it makes the code simple and easy to understand.

Overall, I think these process are very effective and I am happy that these were the first Agile process which I learned. In future I also wish to see how the agile process is applied in real world.

As a Group:

The most important thing that we learnt was working as a team. The advantage of working in a small groups was it was easier to communicate. Both us were able to express our views. If both us had a different view, then that didn’t lead to an argument as it was small group. Setting up meetings was also easy. As communication is a main part of these agile process, we were able to complete the requirements assigned by product owner. Also at the same time it was quite hard to play multiple roles at the same time. Also we learned that no matter how we planned the project initially, there will always be changes in requirement as we go through the development process. But we made sure that changes did not affect the user’s requirement.

We conducted scrum meetings which helped us to gain a lot new knowledge from each other. Also using scrum we learned that Scrum meeting are very important to attend as we get to know the track of the product development. We also learned how important it is to complete the work in given deadline so that it does not affect the next iteration. It also helped us to solve various issues while developing the application. User stories were necessary for each iteration. We learnt that not all user stories are necessary. They were broken down into task for each iteration. We also learned that meeting each other is the key and as we used to live close to each other it was easy for us to collaborate and work together.

We spent our initial stage in learning android programming, As we were both were new to android studio. By surfing web we learnt about how the android programming works, developing the interface and coding under the same roof. We learnt how UI and code interacts with each other, also how buttons works in android. We were glad we got to work on android studio; we are also planning to develop another application in our spare time. Paired programming helped us alot. As we both were new to android it helped us to understand different methods, class and libraries.

Software process development helps us to understand how process works and how it is implemented for developing software. It taught us the importance of communication with team members and to work in a group. We came to know our strengths and weakness. Initially, we had difficulty developing the project as we were not familiar with the terms. But this project helped us in understanding the Scrum and XP process.

**8.0 Reference**

CMMI Wikipedia, 2016

<https://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration>

Humphrey, Watts S. Managing the Software Process . Addison Wesley, 1989.