SCRIPTS

# Topic 1 – Agile Methods and Artefacts

## Slide 1:

Hello, my name is Kanaga Manikandan. Today, I’m going to talk about a few agile methods that were used to support the software development exercise. click

## Slide 2:

Click. So, what are Agile Methods? Click Agile Methods are utilised by a team to work effectively and produce the best product in the end. click. We’ll be covering a few agile methods and how they were used in the software development exercise such as, click. 1), 2) and 3). click

## Slide 3:

click So, what are Scrum Meetings? click These meetings take place daily to review the current progress of the development. Click. The team members share their stories on what happened, what they plan to do and what problems they encountered. click The Scrum Master of the team conducts these meetings and usually last for about 15 minutes. click

## Slide 4:

click So, how were these meetings helpful in the exercise. click Each team member got a knowledge of which team member is going to do which task and within how much time. It’s less of a status update meeting and more of a meeting where team members commit to themselves. For example, a team member in the team said he would get the Feasibility Study done by the next Scrum Meeting. This statement allowed the other team members to get a knowledge of who’s doing what and can expect the results by the next meeting. Click. The team did not have a Scrum Master to conduct the meetings so, the team members do it among themselves every week. The members couldn’t meet up daily because of their personal commitments but the team made sure they get their work done by the next Scrum Meeting and discuss about it. Click. By addressing the completed tasks, pending tasks and problems, the team was able to proceed with the development effectively. A team member shouldn’t be clueless of what’s happening around him. Thus, it’s important for him/her to attend these meetings and get a grasp of the current situation and adapt accordingly. In this case, every team member managed to attend the weekly meeting and the update report can be seen in the picture. click

## Slide 5:

click Now, let us talk about the next method: The Product Backlog. click Product Backlog is nothing but the list of things that needs to be done within the project. It contains the user requested features that the user expects to see in the final product. They are derived from user-stories. The list is prioritised by importance and story points denote the weight and complexity of the task. click The Product Owner owns the Product Backlog and decides which items in the list make it into the final product. Click. The product backlog is a living document. New items/features may be added to the product backlog and the existing features may be modified or even removed and that is solely up to the Product Owner. click

## Slide 6:

click So, how was the Product Backlog helpful in the software development exercise. It provided a wealth of knowledge on the scope and weight of the project. Click. Items in the Product Backlog became tasks estimated in hours and were placed in various sprints. click This allowed the team to complete a chunk of work within 2-30 days. Just like how I said in the previous slide, new features were added to the product backlog and the team allotted time accordingly to get the new features done. click

## Slide 7:

click Now let us talk about the last method: The Burndown Chart. click Burndown Chart is a visual representation of the amount of work completed against time remaining. The picture to the right is the team’s Burndown Chart. You can see a red line in the picture which is drawn from the start point to the end point. Click. This line indicates the estimated rate of completion. The start point is the total number of tasks when the project commenced, and the end point is 0 indicating that there are no tasks left and the product should be ready for release. click If the blue line, which is representing the actual rate of completion, is above the red line, then that means that the team is behind schedule and have more work pending. This wouldn’t necessarily be the team members’ fault. This can also happen when more tasks have been added as said before. click

## Slide 8:

click So, how was the burndown chart helpful in the exercise. Click. The visual representation enabled the team members to have a clear idea on the current state of progress. The burndown chart was also reviewed in the weekly Scrum meetings and the team managed to keep a track of how much work is done, how much of it is left and how much time is left. click It also allowed the team members to quickly adapt to the current situation and manage their more efficiently. Click. The future of the development process was easily predicted with the help of the burndown. Click. click

# Topic 2: Expanded Use Case

## Slide 1:

Hello, my name is Kanaga Manikandan. Today, I am going to talk about expanded use cases and how they were used in the software development exercise. Click.

## Slide 2:

Click. So, what are expanded use cases? Expanded Use Case is a detailed description of the processes used to complete various system functions. Click. We’ll be covering on how expanded use cases were used to identify requirements and organise the management of the software development exercise. Click. We’ll also be covering on how it provided more knowledge than that of an existing use case diagram. Click.

## Slide 3:

Click. Now, let’s talk about how expanded use case helped the team identify the requirements. The team used the features from the user stories to derive the software’s functions and processes. Click. From these functions, the team was able to categorise each function into primary and secondary functions. Click. Primary functions are those which are very essential to the software product. Secondary features are those which will help the user to perform extra features. The software will be able to work even without these functions but it’s better to have them included. Now, that the team had derived the two types of functions, it was time to come up with a plan on how these functions are going to be carried out and what necessary steps will have to be taken by the user to complete a specific function. Click

## Slide 4:

Click. We will now see how the use case diagram became inferior to the expanded use case. Click. This use case diagram provided a clear visual flow of the steps taken by the user to complete the registration of a new member but as you can see, it was unable to describe some steps in more detail. This will only allow the team to understand the flow but not capture the complete picture. Click. If more features were to be added, the diagram becomes more cluttered and will only get bigger. Click. As you can see, the expanded use case displays a lot more information on a single function and the typical course of events displays both what the user does and what the system does in response. After this, the team was able to gain more knowledge on the same function and understand the concept in more depth. Click

## Slide 5:

Click. Now, let’s see how this was used to organise the management of the software development exercise. The detailed description from the expanded use case helped the team to sort out the weight of the tasks and assign those to each member equally. Click. The typical course of events from the expanded use case enabled the team to understand the flow of the processes in more detail and structure the query appropriately. Click. Click.

# Topic 4 – CRC Cards

## Slide 1:

Hello, my name is Kanaga Manikandan. Today, I’m going to talk about how CRC cards helped in the software development exercise. Click

## Slide 2:

So, what are CRC cards but mainly, what does it stand for? Click. CRC stands for Class, Responsibility, Collaboration. It is abbreviated in such a way because these cards are comprised of these components. Now, that we know what it stands for, what does it do? Click. They are used to gather a simple idea about what each class does within the project environment. Classes are essential when it comes to creating a complex software. Thus, the starting grounds or the foundation into creating them must be simple and solid. This was where the CRC cards made our lives easier. Click. We’ll be covering on how these were used to assign responsibilities to the classes and how they aided the design and development of the software development exercise. Click

## Slide 3:

Click. Now, let us talk about how the creation process began. Click. The team studied the requirements of this project from the user stories. Click. This enabled the team to form a basic outline of the classes and its functions. Click. Now, that the team had an idea, all that was left was to materialise it with the help of CRC Cards. Click.

## Slide 4:

Click. Now, how did it help assign responsibilities to each class. Click. A couple of classes were formed based on the basic outline the team had from before. Click. Responsibilities were assigned based on the functions they were planned to do which was derived from the user stories. Click. One of the primary classes is the Form class responsible for processing user commands, switch between pages and pop up successful/error messages. Click. The next primary class is the database class responsible for managing member details and make game bookings for customers or members but make event booking for only members. The other 3 classes hold respective information that will be needed when the user decides to view them. Click.

## Slide 5:

Click. Now, let’s talk about these CRC cards aided the design and development of the software development exercise. Click. With the help of the CRC cards, the team was able to create the class diagram. Click. The assigned responsibilities helped the team to understand each class better and it eased the workload. Click. The collaborations also helped the team to understand how each class is going to interact with another. Click. Thank you. Click.

# Topic 9 – Documentation Management and Version Control

## Slide 1:

Hello, my name is Kanaga Manikandan. Today, we’re going to be talking about documentation management and version control. Click.

## Slide 2:

Click. So, what are they exactly? Click. Documentation Management is a method used by a software team to preserve the files in an external storage where it’s secure and accessible by other team members. There are many tools and software products that allow you to do this, but I’ll show you what the team decided to use in just a moment. Click. Version Control is a popular documentation management method and allows each team member to keep track of all the modifications and contributions performed by the team and will allow any member of the team to go back to a certain point in development and start afresh. This is very useful because if any problem were to rise during the development of the project, the team can always go back to previous version and redo the tasks correctly. Click. We’ll be covering how documentation management and version control were used to support the management of the software development exercise. Click.

## Slide 3:

Now, let’s talk how the documents were managed. Click. Google Drive was the tool used by the team to securely store the files and it allowed the team to access them anytime. The picture to the left is the team’s work stored in a folder in Google Drive. Click. It allowed the team to make any changes to the files freely or work together to make spontaneous changes. The picture to the left is the team’s activity log. The visual feed helped the team to gather an idea on who contributed to which file and when did they do it. Click. The notifications about changes made in the documents allowed the team to be up to date with the progress. Click.

## Slide 4:

Now, let us talk about version control. Click. I used GitHub to store my files securely in an online storage. The picture to the left is my repository of this project and it will contain all the files associated with the software development exercise. You may ask as to why the team decided to have more than 1 storage. One of the reasons is that it’s always a good practice to have more than 1 storages to keep your files secure. The odds are less for both the storages to fail thus even if one fails, the other storage will still be active. The other reason is, Click, version control allows you to keep track of all the updates performed through the course of the development and just like I said in one of the previous slides, it was very essential to have a safe backup point. The picture to the left displays my commitments to my repository and my team can also have a look at this and know how I’m managing my files. Click. It proved better than Google Drive in terms of uploading and accessibility. Why uploading because Google Drive will not know what changes you have made unless you upload them or access the files directly to make changes whereas with GitHub, you can install the GitHub Desktop application and any changes you make in your local folder will be noted by the app and all you have to do is commit those changes and push it to the online storage. Click

## Slide 5:

Now, how did these tools pay off in the end? Click. The tools supported the team very well as they made communication easier and were very easy to use. The user-interface and the features it had made our lives a lot easier. Click. They also proved to very reliable and cost-efficient. Google is a very prestigious company and we were sure that our files would be secure and have no chance of leakage. Amidst of many software products which are priced, these two happened to be free and provide all the necessary features we wanted for documentation management. Click. The tools also supported our Agile development methods in a better way as they allowed us to work together on a single document and discuss about it at the same time and environment. Click. Thank you.