**indh Madressatul Islam University**

**Department of Software Engineering**

**2nd Semester Fall 2022**

**Software Engineering (SEN102)**

**Faculty Name: Syeda Wajiha Naim**

# **Assignment # 3 & 4**

**(Total Marks: 20)**

### **Deadline**

Your assignment must be submitted before or on **5th January 2023**.

### **Objective**

This assignment is designed to let you learn the concepts of software engineering.

### **Assignment**

1. As a project manager your job is to manage a team of developers that is going to build an application similar to the ones they have built earlier which means they have experience building such applications. The application that they are going to build now is more complex and larger one, and the requirements have been thoroughly explained and documented by the customer. Justify your choice of process model for this project. Discuss the valid reason(s) in detail.
2. Discuss T-shirt sizing technique of story points estimation in Agile.
3. Give example of Risk mitigation, monitoring and management (RMMM) plan. [Build up a table.]

**Instructions**

* Submit within due date.
* Write in your own words.
* On title page mention the following:

**Insert Logo of the University**

Underneath the logo write:

**Name:**

**Semester & Section:**

**Student ID:**

**Course ID:**

* Do not use rephrasing tool.

As a project manager your job is to manage a team of developers that is going to build an application similar to the ones they have built earlier which means they have experience building such applications. The application that they are going to build now is more complex and larger one, and the requirements have been thoroughly explained and documented by the customer. Justify your choice of process model for this project. Discuss the valid reason(s) in detail.

## **Waterfall Model Process**

**Communication**

Manager has organized a development team and asked them to create an application that is long and unique in that the customer has given the complete details . After that will go to planning.

**Planning**

After the communication is complete, now he will see what planning should be done first, like if someone wants to build a house, then he will first make a map etc. After that will go to Modeling.

**Modeling**

Now we are going to the design, now we will do the design, like when we are building a house, what color should we use, how should we do the floor, how should we install marble or tiles? After that will go toConstruction.

**Construction**

Now that we have done everything, we have done the communication, we have done the planning, we have done the modeling, now we are going to build the application, just like we need to mix everything first to build a house.After that will go to deployment.

**Deployment**

Now when everything is done, now user will see and he will check that what the user communicated to the developer is correct then he will do okay then developer will deploy.

**Reason**

The waterfall model is used because the communication that took place from the user to the communication, the requirements in the communication were fixed, nothing we had to implement what the user said would be fixed if the user He said that now you make any model, later we have to change it, then we would have used the V model because in the V model, we did communication first and then planning.

Then after that we remembered that we had to edit these things in the requirements, then we went to the communication again and in the communication we edited or again in the direct we went to the designing. If we remember that we have to edit this thing in the communication after doing the communication, then we can't go backwards because we used the waterfall model because it had the requirement fix.

## All About T-Shirt Sizing Agile Estimation Technique

Companies using Agile find its one aspect really good and that it is not rigid. If the organizations using Agile get the essence of it, they can easily customize it to suit their needs to ensure maximum output. One of the best examples of this is Agile Estimation. The entire product is set through these estimation techniques and they make it easy for the team to get through the product.

Almost all the Agile teams do estimation and use different estimation techniques by breaking complicated products into smaller parts. In addition, when estimations are recorded, it is helpful in measuring the team's velocity and output and hence creating release plans in a better way. There are many Agile estimation techniques including affinity mapping, dot voting, story points, a bucket system, and t-shirt sizing. Each has its own benefits. In this article, let’s discuss t-shirt size estimation in Agile.

##### What Is T-Shirt Sizing In Agile

T-Shirt size estimation in Agile is a technique that uses relative estimation. This estimation technique is helpful in planning effectively for a longer time. The T-Shirt sizing technique is a tool that helps in both product estimation and capacity planning. It helps you in knowing how much time or effort a particular initiative will take. Usually, Agile begins with either a high-level estimation of the product or a macro view of the product. This means the teams are able to arrive at a long-term plan for the product. And one of the most popular techniques used for this purpose is t-shirt sizing which estimates story points using relative estimation. It estimates what time, budget, and effort would be required for a particular product. The T-Shirt sizing Agile estimation technique is particularly beneficial for those teams that have just started using Agile and want a relative estimation for a product. On the basis of the way you use this technique, a t-shirt size can depict the scope of the product, the effort needed for it, the work hours required for it, the complexity of the product, time estimates, or all of these.

The t-shirt sizing technique is based on the concept of basketing meaning items with similar sizes are grouped together. Teams mutually collaborate, discuss, and then arrive at a decision on size. The size of a User Story is set by the team relative to others. The t-shirt size estimation in Agile starts at a t-shirt size in order to plan the future releases. This planning is then split into story points for the purpose of Sprint planning. And the story points can be further broken into hours for executing Sprint. Irrespective of this, the important point here is that as the work gets closer and closer to a developer's keyboard, more accurate estimation becomes easier and smaller. The t-shirt planning works very well for defining the product blueprint and Release Planning.

##### How Does T-Shirt Sizing In Agile Work?

When you go to a store to buy a t-shirt for yourself, the t-shirts are marked XS, S, M, L, XL denoting extra small, small, medium, large, and extra-large sizes which means this is a relative sizing and they can fit different shoulder sizes. You can easily select what fits you. It becomes much easier to choose from such sizing than numerical sizing like 36, 38, 40, etc.

In the same way, different items or User Stories can be put in different categories like extra small, small, medium, large, or extra-large. In this way, numerical scores linked to story points are removed by the t-shirt size estimation technique. So, as far as the effort required for a story is concerned, this gives developers the leverage to be more flexible and dynamic.

Generally, a team uses the following process for t-shirt size estimation:

1. All the members of the team join the estimation session and are given the XS, S, M, L, and XL cards.

2. The team is given a brief explanation of the User Story for which the estimation is to be done by the Product Owner. After listening to the briefing, the team members ask questions to bring more clarity to their understanding of the User Story. Usually, the questions asked are like this:

* Is it required to learn any new thing for this User Story before the design stage is started?
* Is it possible to complete this story with the existing code class library or a new one is required to be written?
* Are any special measures required for unit testing for this story?
* What main user expectations are being targeted in this story?

3. After all these questions are answered, each member of the team assigns a t-shirt size to the story.

4. All members of the team then display their cards at the same time.

5. If every member has picked the same card, then it is taken as the final estimate. In case members have picked different cards, then the team holds a discussion until a consensus is reached on one size.

6. More User Stories are taken one by one and the team repeats the above process for each story till all the stories have been estimated.

7. The team can add another step to the above process if it wishes. It can estimate the time required to complete these stories after completing the estimation of all the stories.

So, you can see it is an easy and quick technique of relative sizing but the most important aspect of t-shirt size estimation is that the team should reach a consensus.

##### What To Do And What Not To Do

There are certain dos and don'ts for t-shirt size estimation which, if followed, give the best results. Let's have a look:

* You can think of bigger things and targets while doing t-shirt size estimation because your results will be rough estimates, So you can work freely.
* Do not unnecessarily increase the number of sizes as this technique is for simplifying decision-making for you so do not make the process complex by needlessly adding sizes.
* When working in and with a team, there are numerous ideas that come from all directions from every team member which may lead to a diversion from the goal. Make sure that you confine yourself to the scope of the estimation. You should not lose sight of the goal and make sure that the estimation is taking you nearer to the goal.
* It is not necessary to stick with t-shirt labels only. If you want, you can take other labels to denote sizes like fruits with the smallest fruit denoting the smallest size and the biggest fruit signifying the largest size. You can be innovative with the sizing labels. Idea is to reach a common agreement on size.

##### Pluses And Minuses Of T-Shirt Sizing Estimation

There are many pluses of the t-shirt size estimation technique but at the same time, there are a few minus points too. Let's see the pluses first:

* More number of User Stories can be estimated in less time as it is a simpler method
* It is very useful for the teams that are just starting in Agile
* Teams can set their priorities in a better way and know the amount of effort they would be required to put with flexible deadlines
* Since the deadlines are flexible, in case there are impediments developers can reset the delivery time or their priorities
* The t-shirt size estimation works well when there are large backlogs

###### Now the minuses:

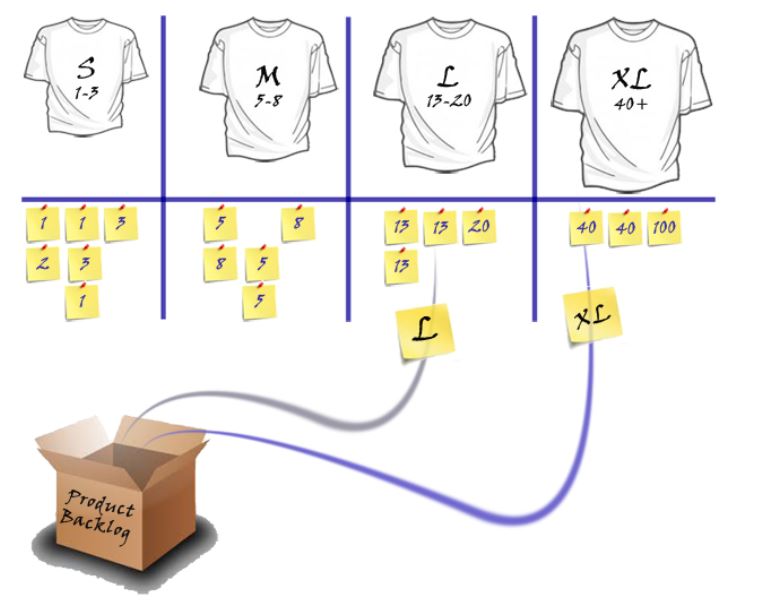
* In the t-shirt estimation technique, accuracy is compromised somewhat because the estimates achieved are on the basis of sizes rather than numerical values
* Estimation arrived at is relative
* At a later stage, if the team wants to calculate velocity, it may need to convert the sizes into numerical values

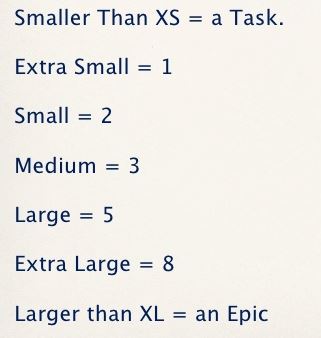
Overall, for getting preliminary estimates, the t-shirt size estimation is a very useful approach. All the stakeholders and team members get a projection of how much time and effort would be required for a particular product. As said earlier, it can be used when there is a large backlog or the team has just started working on the Agile framework. These teams which are new to the Agile framework can initially keep using the t-shirt size estimation technique and can later on move to more accurate techniques once they get a sufficient grasp of other more accurate estimation techniques. If the team is focused and follows the correct process, the t-shirt size estimation technique can increase its efficiency by giving faster and better estimates.

##### References

1. https://doasync.com/blog/what-is-t-shirt-sizing/
2. https://www.easyagile.com/blog/agile-estimation-techniques

**What is T-shirt sizing?**T-shirt Sizing is one of the Story points sizing technique to estimate user story usually used in agile projects. It's a relative Estimation Technique.

Rather than using a number of planning pokers, here, Items are classified into t-shirt sizes: XS, S, M, L, XL.  
  
The term originates from the way T-shirt sizes are indicated in the US. Rather than having T-shirts in sizes 4, 5, 6 etc, there are just a few sizes: Small (S), Medium (M), Large (L) and Extra Large (XL) and so on.  
  
With T-shirt measuring, the development team is made a request to evaluate whether they think a story is extra-small, small, medium, large, extra-large, or double extra-large. By expelling the numerical score, the development team is allowed to think in a more dynamic manner about the exertion associated with a story. The sizes can, if necessary, be given numerical value after the estimation is finished.  
  
   
Image Source: http://www.deltamatrix.com/agile-estimation/  
  
**Steps**

1. Make S, M, L, XL Cards
2. Product Owner will explain the story to be estimated and the development team will ask questions if they have any issues or unclarity. For example,  
     
   Design Related- Do we have to learn new things before starting the design/HTML/jQuery etc?  
   Coding Related- Do we have any code class library ready or we have to write it from the scratch?  
   Testing Related- Any specific setup required for Unit testing?
3. Each developer gives each story a t-shirt size.
4. All in Development team will raise their cards simultaneously.
5. The development team will discuss the differences.
6. The product owner explains the story further or clarifies misunderstanding if any.
7. The team will Go back to Step 3-Step5 until all are agree with one size.
8. Complete or place the stories in size buckets.
9. Estimate the time to complete all stories in S, M, L, XL buckets.  
     
      
   Image source :https://www.slideshare.net/agiledad/agile-estimating-planning

**Benefits**

* This is a very informal strategy and can be utilized quickly with a large number of items.
* It is a popular agile relative estimation technique.
* Forcing the estimate into one of a fixed set of sizes allows the process to go quickly
* It is a good way to Introduce terms to relative estimating.
* It is very effective for affinity estimating

T-shirt sizes can be a great way to becoming familiar with relative estimating. So, start with them if your dev team finds that easier.

But However, insignificantly put some fundamental numbers on them (e.g., Medium=5) and after that steadily shift to using the numbers directly.  
  
I hope you like this article. stay tuned for more article on Agile Development. If you have any query, please feel free to post in the comments section.   
  
**Reference**

* http://www.deltamatrix.com/agile-estimation/
* https://www.mountaingoatsoftware.com/blog/estimating-with-tee-shirt-sizes
* https://blogs.msdn.microsoft.com/oldnewthing/20090512-00/?p=18293
* https://www.slideshare.net/agiledad/agile-estimating-planning

1. Explain the concept of Risk in software engineering with the help of an example.

**RISK IN SOFTWARE ENGINEERING:**

Risk is some problem that happen in a today for feature. When unwanted and unexpected risk are happened in software which loss your software organization and these types of potential issues which harm your timeline of software and affected your software cost. In an easy word we say that unexpected problems or issues which is linked with software for feature this is called software risks.

**CONCEPT OF RISK IN SOFTWARE ENGINEERING:**

When you start a project and during project you have some risk which lose your project and organization the success of project is depends upon quite heavily on the amount of risk that corresponds to each project activity. Any software project executive will agree that the pursuit of such opportunities cannot move forward without risk. “A **risk** is a potential problem. It’s an activity or event that may compromise the success of a software development project. Risk is the possibility of suffering loss, and total risk exposure to a specific project will account for both the probability and the size of the potential loss.” Identifying and aggregating risks is the only predictive method for capturing the probability that a software development project will experience unplanned or inadmissible events. These include terminations, discontinuities, schedule delays, cost underestimation, and overrun of project resources.

### **EXAMPLE**

To ensure that risks remain in the forefront of project management activities, it’s best to keep the risk management plan as simple as possible. For both conventional and agile software project management methodologies, a **risk register** is a proven tool for organizing and referring to known projects risks.

1. What is RMMM and its three basic issues? Explain all the concepts.

**RISK MITIGATION, MONITORING, MANAGEMENT(RMMM):**

RMMM Stands for Risk Mitigation, Monitoring, Management. Mitigation means proactive approach; Monitoring means project manager and Management means Reactive approach It is consider in software engineering for security. If you’re a project manager and controlling all things in a project off course you want to provide all benefits of your customer and you want that your project is free from any type of risk so you will be used RMMM system for making a project.

**RMMM PLAN:**

Risk Mitigation, Monitoring and Management Plan (RMMM) work on documents. Documents is the part of risk analysis, and it is used by the project manager, team of project and all stakeholders maintained this plan by using database system create, searching information and entry, priority ordering and other analysis may be easy by using this plan. Risk monitoring is a project tracking activity.

**ISSUES OF RMMM:**

There are three effective strategy issues:

* Risk avoidance
* Risk monitoring
* Risk management and contingency planning’s.
* **RISK AVOIDANCE:**

If someone is afraid or worry about risk so best way to deal with risk is risk avoidance.

Risk avoidance helps in software engineering to avoid uncertain risks.

* **RISK MONITORING:**

Not all risks are able to monitor two example of risk which is always need to monitored market and environmental risk If any factor or risk changes, it is immediately visible to everyone. Computers are also much better at continuously monitoring risks than people.

* **RISK MANAGEMENT:**

The systematic process of identifying analyzing and respond to project risk and also it includes positive events called risk management.

3.List down and explain the categories of risks. Also mention risk categories as suggested by Charette. [See Risk Management Chapter in Pressman book.]

**CATEGORIES OF RISKS IN SE:**

When you are dealing with any software project if you that project will execute you have faced some risks in software project. Commonly risk that happened in any project are following:

* **SCHEDULE RISK:**

When you away from your timeline of project due to late identification of complex functionalities or suddenly risk in software process this type of risk is directly related to schedule of project which you give your customer, and it is highly affected on your reputation and also affected on company reputation and economy.

* **BUDGET RISK:**

When you slip away from your budget schedule because of wrong budget estimation, cost overruns and project scope expansion.

* **TECHNICAL RISK:**

The Technical risk are directly related to the functions and performance of system. Problem of technical risk your program smoothly down due to complex implementation and module integration.

* **PROGRAMATIC RISKS:**

These are external risks, these risks affect the software project externally because of running out of funds, changing the customer priorities and government rules changes.

* **PROJECT RISK:**

Project risks threaten the project plan. That is, if project risks become real, it is likely that the project Schedule will slip and that costs will increase. Project risks identify potential budgetary, schedule, personal (staffing and organization), resource, stakeholders, and requirements problems and their impact on a software project.

* **BUSINESS RISK:**

Business risk threaten the viability of the software to be built and often jeopardize the project or the product. top five business risks are building a excellent product that wants called as market risk, building a product no longer fits in to the overall business strategy for the company (strategic risk), building a product that the sales force doesn’t understand how to sell(sales risk), losing the support of senior management due to change in focus or a change in people (management risk) and losing budgetary and commitment (budget risk).

## **RISK CATEGORIES SUGGESTED BY CHARETTE**

These following categories are suggested by **CHARETTE**.

### **Known Risks:**

Risks that can be uncovered after careful evaluation of the project plan are called known risks, the business and technical environment in which the project is being developed, and other reliable information sources e.g., unrealistic delivery date.

### **Predictable Risks:**

Those risks that are extrapolated from past project experience e.g., past turnover.

### **Unpredictable Risks**:

Those risks that can and do occur, but extremely difficult to identify in advance.

4.List down the task that are carried out as Risk Management activities.

**TASK OF RISK MANAGEMENT IN SOFTWARE ENGINEERING:**

Risk management means risk controlling and maintain. Firstly, you got to identify the plan and ready to handle risk by knowledge of the entire team of your project for the dealing with risk you have some task in risk management software. Risk Management includes the following tasks.

* Identify Risks
* Classify Risks
* Planning which linked risk mitigation
* Monitoring for risk triggers during the project
* Implementation on the mitigation action
* Communicate about risks status with stakeholders

Diagram

Description automatically generated