SEZG622 – Software Project Management
2023-24 Semester I

Individual Assignment #1 (10 marks)

Project Topic – Software Effort Estimation

I certify that this report submitted towards the Software Project Management course assignment is my own work. I also certify that I have not copied the work of other students or the material from any other sources (books, online documents, etc.) and/or other students' work.

I understand that if this report scores a similarity score above 50% then zero marks will be awarded for this assignment.

Hemant Tiwari – 2022MT93184

1 - Real-life Scenario

1.1 Company and the Project

This project is from my former company. Its name is ***Connecting People*** which is a leading is a multinational telecommunication, information technology, and consumer electronics corporation.

The project where I have been part of *Software Effort Estimation* is ***ACE Automate***. In this project a set of tools were integrated together for driving Collaboration, Customization, Automation, and Serviceability.

Major tools which were integrated together are,

1. Jira (tool for tracking product development),
2. Eclipse Che (A cloud based IED Tool),
3. Gerrit/GitLab (for Code Collaboration),
4. Jenkins (for CI/CD pipelines),
5. Jfrog Artifactiry (for storing Artifacts),
6. Internal Testing and Deployment Tools, among other internal tool chains.

These tools integrated together formed three DevOps major Component i.e.,

1. Continuous Integration (CI),
2. Continuous Testing (CT), and
3. Continuous Deployment (CD)

The team working on the project consist of 14 team members which includes, 1 Architect, 8 Developers, 3 Testers, and 2 DevOps Engineer. I was part of the project as a DevOps Engineer.

1.2 Technique used for Software Effort Estimation

Since we were following Agile Ways of Working, which includes Scrum and Scrum ceremonies, we followed ***Story Point Estimation***. We tracked all our efforts and work using the Jira tool.

Think of an Epic as a big project that can be broken down into smaller parts called User Stories and further smaller parts called Tasks. Each larger task, such as the integration of Gerrit or Jenkins, was considered an Epic.

To figure out how much work each Task required we utilized the Fibonacci series (1, 2, 3, 5, 8, 13, 21, etc.). Each number represented how many days of work we thought a Task would take.

We also organized our work into what we called Sprints, which were basically two-week periods (10 days). Our goal was to deliver working software by the end of each Sprint. Since we had 14 team members, we aimed to complete

tasks worth a total of 140 (14*10) Story Points in each Sprint. Since each Story Point was like one day of work, therefore for each task, we assigned Story Points ranging from 1 to 5 based on their complexity.
Any tasks assigned more than 8 points were further subdivided into 2 or 3 smaller tasks. Each Epic was assigned either 21 or more story points, and each User Story was assigned 8 or 13 story points.

For Story Point Estimation, everyone in the team participated. Once all the Epics, User Stories, and Tasks were created each team member was asked to anonymously decide how many Story Points each task deserved. If a majority of team members assigned the same points to a task, we used that point value. If different story points were assigned, we calculated the mean value of all the points and rounded it up to the nearest whole number in the Fibonacci series. For example, if the mean was 2.4, we rounded it up to 3.

With reference to our project ACE Automate, a simple example of an Epic, User Story, and Task is as followed,

Epic – Integration of Gerrit/GitLab for Project Creation.
- *User Story 1* – Project Setup in Gerrit/GitLab
  - *Task 1* – The name of the Project will be based on Business Unit, Geographic Location, and Client Name.
  - *Task 2* – User Should have option to select between GitLab or Gerrit.
  - *Task 3* – Creating Workspace Schema, based on the project.
- *User Story 2* – Workflow if GitLab is Selected.
  - *Task 1* – GitLab API Integration
  - *Task 2* – SQL Integration, etc.
  - *Task 3* – Any tool Dependencies.
- *User Story 3* – Workflow if Gerit is Selected.
  - *Task 1* – Gerrit API Integration
  - *Task 2* – SQL Integration, etc.
  - *Task 3* – Any tool Dependencies.
- *User Story 4* – There should be options to add users and assign them roles.
  - *Task 1* – LDAP/AD Integration, for user Database.
  - *Task 2* – Types of roles which can be assigned.
  - *Task 3* – Notifying the user
- *User Stoy 5* – Integration with other tool chains.
  - *Task 1* – Integration with Jenkins, as part of SCM Integration.
  - *Task 2* – Integration with Eclipse Che, as part of IDE and kind of workspace schema required.

Story Point Estimation wasn't just for estimating the effort required to complete a User Story or Task and planning Sprints; it also helped us estimate how long it would take to develop the whole software project. It made our project management more predictable and helped us communicate clearly with our team and stakeholders.

In summary, we followed Agile Ways of Working, and Story Point Estimation was a crucial part of Software Effort Estimation.

1.3 Typical results

Story Point Estimation helped with,

1. *Better Planning:* Story Point Estimation helps teams plan their work more effectively. It allows them to figure out how long tasks will take and how much they can get done in a certain time.

2. *Using Resources Wisely:* It helps teams assign the right people to the right tasks. If something is too hard, they can split it into smaller parts or give it to someone with more experience.

3. *Teamwork:* Teams work together to decide how hard tasks are. This brings them closer and helps them understand the project better.

4. *Clear Talking:* Teams can easily talk about how tough tasks are by using Story Points. This makes it easier to tell others how the project is going.

5. *Sprint Success:* Teams can plan sprints better and finish what they planned to do. This makes everyone in the team feel good about their work.

6. *Change Plans if Needed:* If a task is harder than they thought, they can change their plans without a big problem. They can get more help or break the task into smaller pieces.

7. *Smart Decisions:* Teams can use data from past work to make good choices about how to do future projects.
8. *Getting Better All the Time:* Teams can look back at their work and find ways to do things even better next time. This means they keep improving.

1.4 Challenges

Some common problems team faced when using Story Point Estimation, and how we deal with them:

1. *Bias:* People may be influenced by personal biases, which can lead to inaccurate story estimations. Therefore, team members were asked to assign story points anonymously.

2. *Different Ideas:* People in the team might think a task is easy or hard in different ways. To fix this, we took help from mean and median.

3. *Task Changes:* Sometimes, a task may seem easy but becomes hard later, or the other way around. To fix this, we had sprint planning before the beginning of any sprints and sprint retrospective after the end of a sprint. We also had Standup meetings everyday to discuss any potential challenges or impediments.

4. *Extra Work:* New tasks might come up during a sprint, which can change your plans. To handle this, make a separate list of new tasks and decide when to do them.

5. *Scope Creep:* Estimations may not account for potential scope changes during the project, leading to delays or overruns. To overcome this, we used sprint planning.

In short, while Story Point Estimation is a useful tool, it can have problems. To fix them, talk openly, learn from your mistakes, and be flexible. With time and practice, your team can get better at estimating and working together.

References

1. Lecture Slides
2. https://adaptmethodology.com/epic-user-story-task/
3. https://en.wikipedia.org/wiki/Fibonacci_sequence
4. https://medium.com/ducalis-blog/story-points-how-fibonacci-sequence-works-for-agile-estimation-4c7e0081f4d8
5. https://www.linkedin.com/advice/3/what-benefits-challenges-using-story-points-sprint-estimation
6. https://www.easyagile.com/blog/10-reasons-to-use-story-points/
7. https://reqtest.com/en/knowledgebase/agile-estimation-techniques/
8. https://www.scrum.org/resources/blog/why-do-we-use-story-points-estimating
9. https://www.linkedin.com/pulse/advantages-disadvantages-using-story-points-anshika-misra/