**Ponder 01: Code and Fix Application**

**Code and Fix technique description:** This is a technique of fixing or creating a software where developer jumps into the problem and starts to tinker with the code. There is usually no design and methodology described for carrying the project.

**Scenario:** I the owner of this software company oversees creating an application for the university registration system. I have a team of 19 people including 12 software engineers. Our assignment for this project is to merge the BYUI grad plan with the registration system. We need to find an optimum schedule that will decide the number of sections needed for any specific class in that semester.

**Assignment:** For this project, we will have different meetings throughout the projected interval. And in the first meeting, we will assign different people their role and start writing documentation and checkpoint for this project.

Meetings:

* (Overall project discussion meeting) Our first meeting will be planning and drafting meeting. We will try to predict the time, resources, and profit that can be made from this project.
  + Since we are a small team of 19 people, we want all the people but custodian to attain this meeting.
  + As said earlier the main agenda of this meeting is to assign everyone their role and try to find the optimum process for doing this project.
  + We will look at the nature of the project. We will come up with different checkpoints and smalls goals that will guide us to our main goal without losing track. Checkpoints will be created for every week. We will try to get as much as information about the project if needed we will talk to Peter and Patricia. We will also plan about what we want to get done in the next meeting.   
    Along with that we will divide team of software developer into two group lead by Abe and Britney. And the tester and debugger will be an indifferent independent team.

We will also decide that the big meeting of all people, but custodial and secretary will be held twice every week on Monday and Wednesday. And the meeting of software engineer will be in the first 15 minutes of the work. The meeting time can be expanded if the plan for the whole day does not come.

* + This overall project discussion meeting will be only once and will not be over until everyone is satisfied.
* (Prototype Week Meeting). Our Third big meeting (big meetings are held twice a week) will be for the prototype shows. We will be working on the project as well as working on making prototype. Our prototype will be built by team of UX designer and they will present it in a big meeting on Monday in next week (assuming new project always starts in a new week)
  + As it will be done in Monday meeting, everyone except custodial needs to attain.
  + Planning for the next two days and talking about the checkpoint achieved (checkpoint for this week will be building a prototype and understanding the working of I-plan and registration system).
  + During this meeting, we will talk about the difficulty that might come in software development. We will do something’s if our prototype isn’t liked by the customer(school). We will ask for suggestions to the customer how they want their product. We will also discuss if we need to do any changes.
  + This prototype meeting will be held until the customer is not satisfied with the software prototype. It will be held on the first Monday after a week and will be held twice a week until the customer is not satisfied.
* (After Prototype meeting) – This is the time where we will start talking about doing things. About planning on doing the coding. Documentation, feature design, testing method, software development methodology, etc.
  + Everyone except custodian and secretary needs to attain this meeting.
  + The agenda of this meeting is to how to work. And what to do in order to achieve the checkpoint for next week. Also, who will do what and how for the next two days.
  + Accomplishment during this meeting will be to come up with an idea to solve the problem and to come up with the way of doing things that is optimum.
  + This meeting is going to happen twice a week. This meeting will be focused on day to day things for two weeks and will be done only after the prototype is done.
* Software Engineer Meeting- This is the meeting of software engineer only.
  + This meeting is only for software engineers.
  + To come up with the work plan for a day.
  + It will help in making work smooth and will direct the whole project in one single direction.
  + This meeting will be held every day.
* UX designer meeting: They will work in a group as a team of UX designer. So they won't need to have a specific meeting. They will be in meeting with engineers, technical writer, and owner/Rochak in the big meeting twice a week.

Documents: The first document we will write will be the requirement, the second will be the instruction on using those features, and third will be on how those features will be implemented.

* The main author of all three documents will be a technical writer. The reviewer and second author will be Rochak.
* The audience of the first document will be a software engineer and UX designer who will design the features and the customer will be the school. The audience of the second document will be the user of our software, and finally the audience of our third document will de software engineer as well as maintenance engineer.
* The document would be very helpful for the engineer to note what feature they have implemented and how they are going to do it. I think the biggest help will be to maintenance engineer later.
* The requirement writing will take 10% of the total time, instruction will be done at the end of the project, and documentation of features and implementation will be done in first 20%-40% time period.

Roles:

* Project manager: Project manager will be Rochak: the owner of the company. To be the project manager one mush have a good management background and business-minded. Should be able to take hard and difficult decision when needed. Rochak must do all the regular talking with customer as well as present our product to our customer. Rochak should be good at managing his workers.
* Custodian: As regular custodian will be Chrissy. She will clean the office in the evening.
* Secretaries: Stan and Sally will be secretary and will help Rochak for office work. They will be very friendly and help to fill the bridge between Rochak and his worker.
* Designer and basic front-end engineer: Ursula, Xavier, and Ingrid will be a designer and HTML coder. To be in this position one needs to have a good technical background and sense of style. They will design the front page of the software.
* Technical Writer: Teri will be a technical writer and Rochak will be the reviewer. One should have spent good enough time in tech-industry to be in this role. These people will write all the documentation need for software.
* Software engineer: These people will plan the whole software. They will architect the implementation and working of the software. Abe, Britney, and Jack will be a software engineer. To get these roles one need to have a good knowledge of software engineering as well as coding.
* Developer: Claire, Emily, Grace, and Larry are a software developer. They will write code under the design and direction of software engineers. To be in this role one needs to be good at programming. Their job is to write code.
* Tester and Debugger: Frank and Doug will be our tester and debugger. Frank will be writing our test case and Doug will try to fix bugs. To be in this role one need to have experience as a QA engineer.
* QA engineer: Holly and Keith are QA engineer. They will be trained by Abe and Britney so they do their QA job. They will check the quality of the software produced.

Checkpoints: Since we don’t want our software engineer to do nothing while technical writing is working on writing requirement we want them to work concurrently. We will follow the methodology of code & fix. We want all our worker to work from the beginning so Rochak doesn’t have to pay his employee for doing nothing.

There will be a total of 8 checkpoints and one final, which will be approximately 2 months of the project. In the first week UI engineer will work in prototype. For 2nd and 6th-week software engineer will work in architecting software, developer will program, tester will write test, and debugger will continue debugging. In 8th week all the components will be made final. Everything will be given finish touch. This method will make everyone busy and hopefully brings most (*which is not true*) out of worker.

We will have a weekly checkpoint, but we will focus mostly on 3.

1st checkpoint we will call prototype checkpoint.

2nd checkpoint we will call product checkpoint.

3rd checkpoint is a final.

1st and 3rd are done in 1 week whereas most of the time is spent in coding and making the product.

Reflection: I don’t really understand the question. It asks me to visualize things as an owner. And it is obvious jumping in the code always doesn’t work, especially in big projects. From my own experience, I can tell that when we do jump and code we will realize later that the approach we were doing is not necessarily the optimal way of doing it. Sometimes we just get stuck in a problem which could have been solved easily by careful management and planning. It is not uncommon to see problem of database when there is no planning. As we go through the project we will realize there is no way to extract the needed information with the data given. Now in this situation we have to change UI to get more data or we can just send what we need. This could have been easily solved during software design description. But since we straight jumped in the code it will be expensive to redo things. And this type of situation can happen a lot of time which will eventually cost more when we do planning and description writing before starting to code.