1. Write a short paragraph to answer these three questions:

   -Two major concerts of any softwear project are managing cost and staying on schedule/time. I think personally cost is more important since there wouldnt even be a product if it isn't done with the budget given. Schedule is still important when there are a lot of people depenading on the project, but the project even being working within our budget is the first priotty.

2. In the Agile method for software development, what are the four main phases that occur in each and every iteration? Do you feel that any of them could be done at the start of the project and not be repeated in every iteration? Do you feel that would save time overall on the project? Justify your answers with a brief explanation.
   -The four main phases of the agile method for softwear development are requirements, design, code, testing, and maintenance. Some addintal phases include reflection & mcash, risk assessment, prototyping, quality assurance, production, marketing. I don't feel like any steps could be done in any other order since they build upon each other. They are all needed to ensure accuracy, speed, and sustainability. It may save time to skip review intally but it will come back ten times with errors later on.

3. In the Waterfall method for software development, what are the main phases that occur? How are they different from the phases in the Agile method? What other phases are in Waterfall that are left out of Agile? Do you think these are needed in Waterfall? Describe a situation using Agile in which one of these extra Waterfall phases might be needed.
   -The main phases that occur in the waterfall method for softwear development include requirements, design, code, testing, and maintenance. Additnal ones phases include risk assessment, prototyping, quality assurance, production, marketing. These are diffrenft from the agile method as it caters more to accuracy as well as the aftermath when it get realses. The differences include in the beginnerg snd end of the waterfall method as their phases are more spead out and done in more steps to ensure less errors. I honestly think they needed as much security and assurance as is needed in this day and age to have it meet all requirements. When you are devolooing an app, you need to take extra care and plan out all your requirements meticuolsley and then take feedback from the ussers at the end to make it for the people.

4. Write one-sentence answers to the following questions:
   -"User story" is a tool in the agile method for softwear development that gives indpeth details on the user's end view and tells us their needs so we can deliver value.
   -"Blueskying" is coming up with and bouncing off ideas with your other team members.
   -Four things that user stories should do are simple, small, customer focused, and concise.

-Three things user stories shouldnt do are being too concluded, technical, and big.

5. What is your opinion on the following statements, and why do you feel that way:

-“All assumptions are bad, and no assumption is a good assumption.”
My opinion is that not all assumptions are bad since they can actually be nefecial and optimie the project in multiple ways. Having no assumptions isn’t good, you should always be questioning, learning and growing.

-“A big user story estimate is a bad user story estimate.”
I agree most of the time of course there are some rare occurrences where someone can make all their estimates a reality. But for the common population big ideals don’t always match their estimates since so many small issues come up per every bigger one.

6. Fill in the blanks in the statements below, using the following things [you can use each thing for more than one statement]: Blueskying; Role playing; Observation; User story; Estimate; Planning poker.

- I agree with the book answers, we had similar results proving that!

  a) You can dress me up as a use case for a formal occasion: **User Story**
  b) The more of me there are, the clearer things become: **User Story**
  c) I help you capture EVERYTHING: **Blueskying**
  d) I help you get more from the customer: **Roleplaying**
  e) In court, I'd be admissible as firsthand evidence: **Observation**
  f) Some people say I'm arrogant, but really I'm just about confidence: **Estimate**
  g) Everyone's involved when it comes to me: **Blueskying**

7. Explain what is meant by a better than best-case estimate?
-It means it’s the most optimal estimate that has the minimal run time for our given inputs. For the best case, it can be the most optimal but may not be the fastest, which is what the better than best-case estimate means.

8. In your opinion, when would be the best time to tell your customer that you will NOT be able to meet her delivery schedule? Why do you feel that is the best time? Do you think that would be a difficult conversation?
-In my opinion, the best time to tell the customer I won’t be able to meet the delivery schedule is as soon as we realize we can’t meet the deadline. It’s best to do it when known to deter later issues between you and the customer. This ensures trust, which is one of the main themes in a

business.

9. Discuss why you think branching in your software configuration is bad or good. Describe a scenario to support your opinion.
-Branching in software configuration can be good when used short-lived but can be bad when used long-lived. An example where it's good to use these short-lived branches is when you and your team work in a branch for a couple of hours, which allows your merge to trunk to have the fewest issues when integrating.

10. Have you used a build tool in your development? Which tool have you used? What are its good points? What are its bad points?

    -Yes, we have used build tools in our development. We have used the build tools Next.js and Tailwind. The good points of using Tailwind are its styling features, and the good points of using Next.js are its framework and dev server. The bad points of Next.js include server constraints, and the bad points of Tailwind include customization risks.