1.Why does it take so long to get software finished?

The chief reason for software takes so long to get finished is that many changes are made to a software product over its lifetime. As changes are made, defects may be inadvertently introduced to other portions of the software that interact with the portion that was changed. Beside that, software engineer must test and verified each phase of the software development process and each phase must be documented and completed before next phase can begin.

2.Why are development costs so high?

Software development cost so high because it takes skills and experience and a lots of people working to develop and test the software. It can take months even years to get all the code written and tested and checked. It takes a team of people to do this, and all of them are well educated and often have a lot of experience.

3.Why can’t we find all errors before we give the software to our customers?

Because we as a developer does not really into what customer want and request and just only thinking about to develop as the best as we can. Beside that, probably we didn’t do peer review or didn’t follow correct software development life cycle and not making case scenarios and doing the test itself.

4.Why do we spend so much time and effort maintaining existing programs?

Because once we are finished coding and completed the software that does not mean that we are finished with the project. For large project maintaining software can keep many people busy for a long time. Since there are so many factors influencing the success or failure of a project and also many competitions from others and new software.

5.Why do we continue to have difficulty in measuring progress as software is being developed and maintained?

Because basically software developed is new and fresh and different with other software so it not have example. And the error outcome is not expected as software develop goal. And of course completed the software does not mean that progress is completed and always must be maintained.

6.Why do we continue to have difficulty in software development projects?

I. ***The software industry is young***:
In comparison with the other industry ,the software industry is only about 50 years old. We still have a long way to go before we have the body of experience behind us that the construction and manufacturing industries have.

II. ***Every line of code is a potential point of failure***:
Each line of code will have dozens, even thousands, of possible inputs, outputs, states or dependencies to deal with. It can impact, or be impacted by, other lines of code or by external factors. Even if it was possible to

document every test case for a line of code you still couldn’t be sure that there wasn’t some unknown factor that could cause an error.

III. ***Lack of user input:***

Reasons for this can include:

•The system is being promoted by the management and so the business users have no buy-in

•The users are too busy and have “more important” things to do

•Relations between the user community and the I.T. team are poor

Without the involvement and input of a user representative the project is doomed to failure. This person should be a subject domain expert with the authority to make decisions and a commitment to the project timescales.

IV. ***Users don't know what they want until they see it:***

Even with good input from the users no amount of analysis of user requirements can take away an immutable fact that users only think that they know what they want. In truth, it’s not until they start seeing something, and using it, that they begin to really understand what they need. This is especially true when the software is being developed for a new idea or process that they haven’t used before.

V.***There are no barriers to entry to become a programmer:***

There is one argument that states that software development is so hard because programming is so easy. In other words it is relatively easy to learn how to write code but there is a huge gap between that and delivering great software.

VI. ***All software is affected by external factors:***

Software is “mindware” and therefore doesn’t obey physical laws but it usually must conform to external constraints such as hardware, integration with other software, government regulations, legacy data formats, performance criteria, scalability etc.