Date: / / Day: M T W T F S

4@

Describe general Principles of Software Engineering:

Ans:- David Hooker has purposed seven principles that focus on software engineering as a whole.

(i) **The First Principle: The Reason It All Exists:-** A software system exists for one reason to provide value to the users. All desicions should be made with this in mind. Before specifying a system requirements, before noting a price of system functionality, before determing the Hardware platforms, ask yourself some questions. If the answer is no, don't do it. All other principles support this one.

- **The Second Principle:- KISS (Keep It Simple, Stupid!)**

Software design is not haphazard process. There are many factors to consider in any design effort. All design should be as simple as possible, but no simpler. This facillities having a more easily understood and easily maintained system.

Date: __/__/____ Day: M T W T F S

• The third Principle: Maintain the vision:-

A clear vision is essential to the success of a software project. Without one, a process almost unfailingly ends up being "of two [or more] minds" about yourself. Compromising the architectural vision of the software system weakens and will eventualy break even the well-designed systems. Having the empower architect with strong vision led the software project to success.

• The Fourth Principle: (What you produce, Others will consume:-)

In some way or other, someone else will use, maintain, document or otherwise depend on being able to understand your system. So, always specify design, and implement knowing someone else will know what you are doing. The audience of software project development is potentially large. Someone may have to debug your code and that makes them a user of your code. Making thier job easier adds value to system.

Date: / /

Day: M T W T F S

• The Fifth Principle: Be open to the future:

A system with a long lifetime is more valueable. In today's computing enviroment, where specifications, change on a movement's notice and hardware platforms are just a few month's old, software lifetimes are measure in months instead of years. However, true "industrial-strength" softwares systems must endure for longer.

• The Sixth Principle: Plan Ahead for Reuse:

Reuse saves time and effort. Achieving the high level of reuse is the hardest goal to accomplish in developing a software system. The reuse of code and design have been proclaimed as a major benefit of using object-oriented technology. However, the returns on this investment is not automatic. There are many techniques to realize reuse at every level of system development process.

Date: __/__/____ Day: M T W T F S

- The Seventh Principle :- Think !

The last principle is probably the most overlooked- Placing clear, complete thought before action almost always produces better results- When you think about something, you are more likely to do it right- You also gain knowledge about how to do it right again- If you do think about something and still do it wrong, it become a valuable experience- A side effect of thinking is learning to recoganize when you don't know something.