TOPICS

The Nature and History of Software Development

Problems with Software Development

Software Engineering Paradigms and Technology

PROBLEMS WITH SOFTWARE DEVELOPMENT

- Problems
- Causes

Problems

1. We have little data on the software development process.
2. Customers are often dissatisfied with the software they get.
3. Software quality is hard to define and measure.
4. Existing software is often very difficult to maintain.

Can these problems be overcome?

Causes

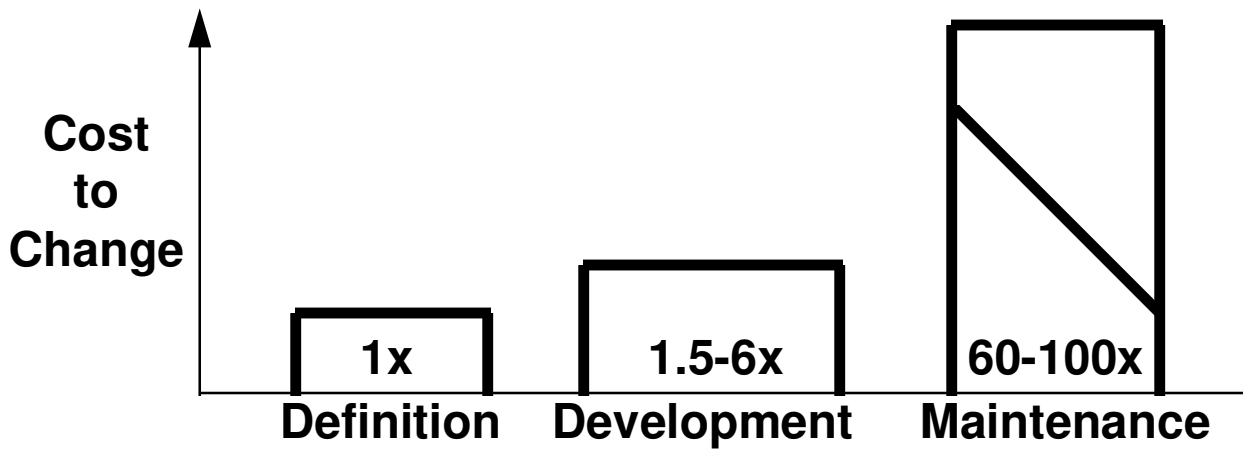
- No spare parts to replace, so an error in the original software is also in every copy.
- Software quality is a human problem.
- Project managers often have no software development experience.
- Software developers often have little or no formal training in engineering the development of the software product.
- Resistance to change from programming as an art to programming as an engineering task can be significant.

SOFTWARE MYTHS

- Customer Myths
- Developer Myths
- Management Myths

Customer Myths

Myth	*Reality*
• A general statement of objectives is enough to get going. Fill in the details later.	• Poor up-front definition of the requirements is *THE* major cause of poor and late software.
• Project requirements continually change, but change can be easily accommodated because software is flexible.	• Cost of the change to software in order to fix an error increases dramatically in later phases of the life of the software.



Developer Myths

Myth	*Reality*
Once a program is written and works, the developer's job is done.	50%-70% of the effort expended on a program occurs after it is delivered to the customer.
Until a program is running, there is no way to assess its quality.	Software reviews can be more effective in finding errors than testing for certain classes of errors.
The only deliverable for a successful project is a working program.	A software configuration includes documentation, regeneration files, test input data, and test results data.

Management Myths

Myth	*Reality*
• Books of standards exist in -house so software will be developed satisfactorily.	• Books may exist, but they are usually not up to date and not used.
• Computers and software tools that are available in-house are sufficient.	• CASE tools are needed but are not usually obtained or used.
• We can always add more programmers if the project gets behind.	• "Adding people to a late software project makes it later." -- *Brooks*