Software Myths

**Definition:** Beliefs about software and the process used to build it.

Myths have number of attributes that have made them insidious (i.e. dangerous).

- Misleading Attitudes - caused serious problem for managers and technical people.
Management myths
Managers in most disciplines, are often under pressure to maintain budgets, keep schedules on time, and improve quality.

Myth1: We already have a book that's full of standards and procedures for building software, it does provide my people with everything they need to know.

Reality:
• Are software practitioners aware of existence standards?
• Does it reflect modern software engineering practice?
• Is it complete? Is it streamlined to improve time to delivery while still maintaining a focus on quality?

• In many cases, the answer to all of these questions is "no."
• **Myth**: My people have state-of-the-art software development tools, after all, we buy them the newest computers.

• **Reality**: It takes much more than the latest model mainframe, workstation, or PC to do high-quality software development. (E.g.=1. Parent-student-facility
  
    2.5 star facility in education-best engg.)

• Computer-aided software engineering (CASE) tools are more important than hardware for achieving good quality and productivity, yet the majority of software developers still do not use them effectively.
• **Myth**: If we get behind schedule, we can add more programmers and catch up (sometimes called the Mongolian horde concept).

• **Reality** Software development is not a mechanistic process like manufacturing.

• In the words: "adding people to a late software project makes it later." At first, this statement may seem counterintuitive.

• However, as new people are added, people who were working must spend time educating the newcomers, thereby reducing the amount of time spent on productive development effort.

• People can be added but only in a planned and well-coordinated manner.
• **Myth:** If I decide to outsource the software project to a third party, I can just relax and let that firm build it.

• **Reality:** If an organization does not understand how to manage and control software projects internally, it will invariably struggle when it outsources software projects.
Customer Myths

• **Myth:** A general statement of objectives is sufficient to begin writing programs— we can fill in the details later. (E.g. only def. is der clg mgmt sys.)

• **Reality:** A poor up-front definition is the major cause of failed software efforts.

• A formal and detailed description of the information domain, function, behavior, performance, interfaces, design constraints, and validation criteria is essential. Characteristics can be determined only after thorough communication between customer and developer.
• **Myth:** Project requirements continually change, but change can be easily accommodated because software is flexible. (E.g. Wipro-IPCA exp.)

• **Reality:** It is true that software requirements change, but the impact of change varies with the time at which it is introduced.
  • If serious attention is given to up-front definition, early requests for change can be accommodated easily with relatively little impact on cost.

When changes are requested during software design, during implementation (code and test) the cost impact grows rapidly.
Practitioner's myths

• **Myth:** Once we write the program and get it to work, our job is done.

• **Reality:** Expert said "the sooner you begin 'writing code', the longer it'll take you to get done."

• Industry data indicate that between 60 and 80 percent of all effort expended on software will be expended after it is delivered to the customer for the first time.
• **Myth:** Until I get the program "running" I have no way of assessing its quality.

• **Reality:** One of the most effective software quality assurance mechanisms can be applied from the inception of a project—the formal technical review.

• Software reviews are a "quality filter" that have been found to be more effective than testing for finding certain classes of software defects.
• **Myth**: The only deliverable work product for a successful project is the working program.

• **Reality**: A working program is only one part of a software configuration that includes many elements.

• **Documentation** provides a foundation for successful engineering and, more important, guidance for software support.
• **Myth:** Software engineering will make us create voluminous and unnecessary documentation and will invariably slow us down.

• **Reality:** Software engineering is not about creating documents. It is about creating quality

• Better quality leads to reduced rework. And reduced rework results in faster delivery times.