

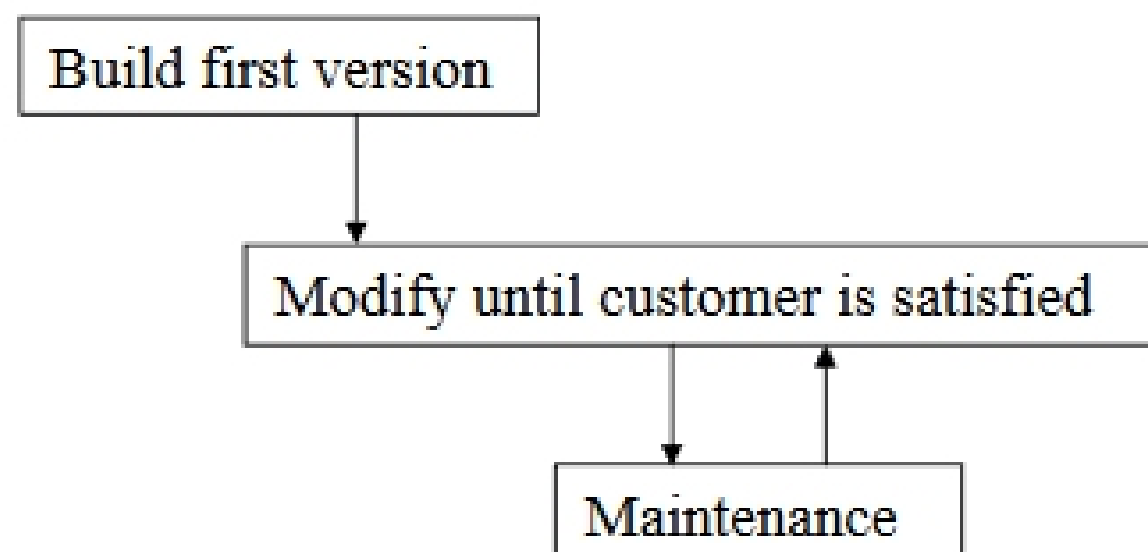
Software Life Cycle Processes

↳ Objective:

- Establish a work plan to coordinate effectively a set of tasks.
- Improves software quality.
- Allows us to manage projects more easily.
- Status of projects is more easily tracked.
- Provides a baseline for improvement and measurement.

↳ Build-and-Fix Model:

- Build a product without specs or any attempt at design
- Rework the product to satisfy the client
- It is not practical for product of reasonable size.
- Approach:



↳ Definition:

“A life cycle is the course of development changes through which a project passes from its inception as a project request to the mature state as characterized by a stable production environment.”

↳ According to IEEE Std 12207 (ISO/IEC 12207), Standard for Information Technology -- Software life cycle processes:

- Establishes a common framework for software life cycle processes that can be referenced by the software industry.
- Lists processes that can be applied during the acquisition, supply, development, operation and maintenance of software and systems containing software.

↳ Development Process

- Note that development often accounts for less than a third of the total project!
- Once a system has been built, the time spent on maintenance will be greater than the time spent on all other phases
 - Maintenance takes around 60% of total cost.
 - But note that for a very successful system (say, Unix, or Microsoft Word), maintenance can continue for a very long time.
 - In fact, some describe developing software as the process of maintaining a blank sheet of paper!

↳ Different Processes:

- Problem identification: someone identifies a problem, which needs addressing and might be susceptible to a computerized solution.
- Requirements analysis:
 - It is much harder than people think.
 - Required features that should be included in a computerized solution to provide a cost-effective solution to the problem.
 - Capture what the customer wants.
- Develop test plan: Develop test procedures to determine if a completed product correctly provides all of required features.
- Develop software design:
 - Analyze the required features to identify existing software that can be reused in this product.
 - Changes needed in reused Components.
 - Create high level design for new components
 - For each component in high level design, develop detailed design.
 - Also develop unit test plan for each component.