What is SPM

LiZhengYing 13211206
ZheJiang Normal University
Software Project Management

Abstract

The “software crisis” of the 1960s and 1970s was so called because of a string of high profile software project failures: over budget, overdue, etc.

SPM was proposed in 1970s. DOD(Department of Defense) of USA have been a research to find the reasons of software failure. The result showed that 70% failure projects were due to the poor management rather than technology problems.

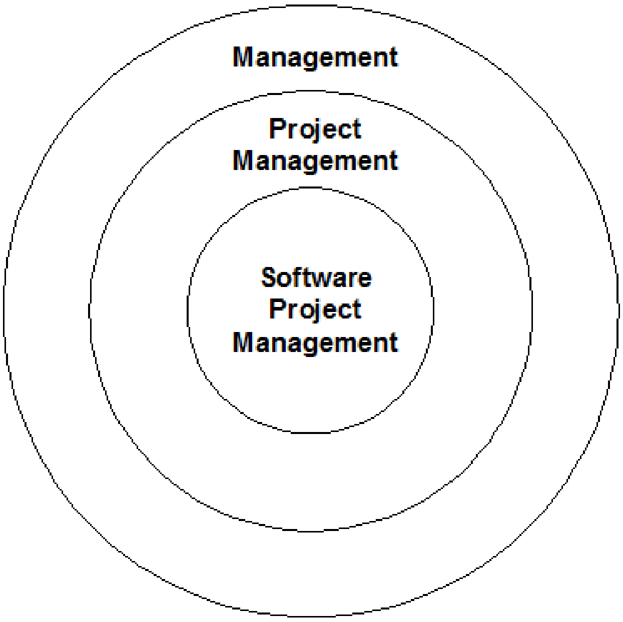
Software project management is a sub-discipline of project management , in which software projects are planned, monitored and controlled

Keywords: software, project management, planning, monitored, controlled

1 Introduction

- Software
  - a collection of computer programs and related data that provide the instructions telling a computer what to do and how to do it. (Definition from http://en.wikipedia.org/wiki/Software)
- Project
  - A project in business and science is a collaborative enterprise, frequently involving research or design, that is carefully planned to achieve a particular aim. (Wikipedia).
- Project Management
  - Project management is the discipline of planning, organizing, securing and managing resources to bring about the successful completion of specific project goals and objectives.(Wikipedia)
- Software Project Management
  - Software project management is the art and science of planning and leading software projects (Stellman & Greene, Applied Software Project Management). It is a sub-discipline of project management in which software projects are planned, monitored and controlled

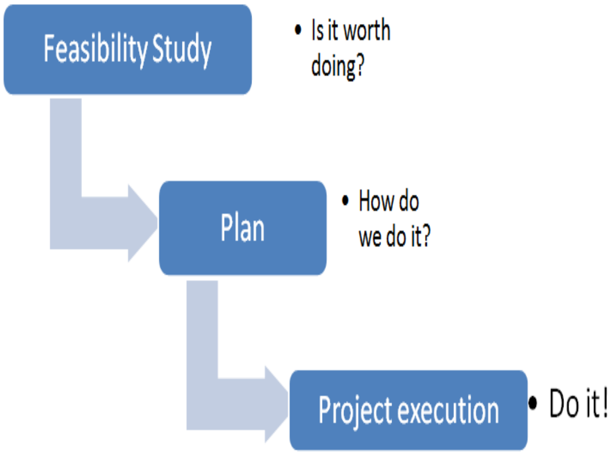
Software Project Management



2 Why is SPM important?

Good project management cannot guarantee success, but poor management on significant projects always leads to failure.

3 Activities



- The feasibility study
  - Assesses whether a project is worth starting
- Planning
  - Describe how to do it.
- Project execution
  - Do it!
  - Distinguish between planning and design
- Requirements analysis
  - Starts with requirements elicitation or requirements gathering.
  - Classification:
    - Functional requirements
    - Quality requirements
    - Resource requirements.
- Architecture design
  - Map software requirements to software components.
  - Design of the system architecture is an input to the software requirements
  - Components are not only software
- Detailed design
  - Decompose a software component into a number of software units that can be separately coded and tested.
- Code and test
  - Writing code for each software unit.
- Integration
  - The components are tested together to see if they meet the overall requirements.
  - Software, hardware
- Qualification testing
  - Verify whether fulfill all the requirements
- Installation
  - Make the new system operational.
- Acceptance support
  - Software maintenance.