13. How does project management differ between hardware and software projects?

Depends on:

▲ Project size and complexity

Dealing with Tangible and Intangible Deliverables

The basic principles of project management are applicable to projects of any type, but there are some key differences worth noting between hardware development projects having physical deliverables and software projects that generate less tangible results. The specifics of the project life cycle may be dissimilar, and there are often differences in processes, such as those related to testing and scope change control.

Defining Life Cycles

Software projects, especially those with very novel deliverables that are relatively small, may elect to use an ‘‘agile’’ or cyclic life cycle, as discussed in Problem 7.

Hardware projects and larger software projects generally employ a more traditional waterfall-type life cycle, but the names of the life-cycle phases may differ. For product development projects, whether hardware or software, a typical life cycle will begin with one or more phases focused on definition and analysis, with a business decision to carry the project forward at a relatively early process stage. Software development undertaken on a fee-for-service basis, on the other hand, usually has more phases on the front end related to sales and proposal activities necessary to win the business. The business decision in this case is further along the sequence of phases and represents the decision by the customer to agree to the proposal and sign a contract. There may be only one or two phases subsequent to this decision point, to execute the contract work, and then to secure approval and payment.

Establishing Processes and Roles

With a well-defined hardware project, scoping changes are expected to be rare and the process for managing those changes is usually defined quite formally. Software projects also need good scoping management processes, but changes are inevitably more common, and (whether it is actually true or not) changes are considered to be less costly and disruptive to software projects. Particularly early in a software project, the process used to manage changes to the deliverable can be relatively informal, even after the baseline plan has been set.

Testing is another area where there are often differences. Software projects may have multiple interim deliverables that need to be tested and evaluated, so testing may be necessary throughout a software project. Owing to the nature of hardware projects, most testing tends to be scheduled shortly before project closure, including the unit tests of subcomponents of a complex system deliverable. Because hardware components may come together in a testable configuration only near the end, evaluation (except for that done as part of early feasibility investigation) is mostly done fairly late in the project.

For some types of hardware projects, the rate of technological evolution is relatively slow compared to software projects. Because of this, the technical expertise of a project leader of a hardware project tends to be deeper than that of those who lead software projects. For all projects, success depends on the subject matter expertise of the project manager, but software project leaders may be much more dependent on the specialized backgrounds of their team members. Because of this, and the increasingly cross-functional nature of software projects, effective software project leaders need to have especially well-developed people leadership skills.