Lesson Proper for Week 2

PROGRAM

Before a program can require a plan, program objectives are defined and technical and management disciplines are identified. This information defines a reasonable estimate or cause for:

- Cost evaluations
- Risk management assessments
- Defined and documented tasks
- Manageable schedules
- Progress repor

The people who work in a software design/development life cycle are expected to meet and achieve program objectives and understand the high expectations required of them. These activities begin at the systems design level of engineering and flows down to other software disciplines.

Program objectives identify goals for the program with consideration of how these goals are to be accomplished. Effective programs that perform to defined objectives and within the scope are successful due to implementing:

- Required data
- Tasks or functions
- How the work product performs

- How the work product performs
- Quantitative mechanisms

When program objectives and the scope are considered, program managers can select the best approach that would eliminate “roadblocks” imposed by scheduled delivery deadlines, budget concerns, and people issues.

Framework Established

Software processes provide the framework and effective planning when it is time for deliveries to software and systems integration facilities and the customer. Activities related to the framework are required for all programs. For effective planning, there are multiple tasks, scheduled milestones, and quality aspects necessary to ensure the framework is established not only for managers but also for the people who work. The quality assurance team, which is independent at times, and configuration management personnel monitor the framework processes.

PROJECT

The main reason that software projects are planned and controlled is to eliminate any confusion that could occur. The teams that are expected to provide work products struggle if projects are not planned, and control is not even an option. Studies showed that when schedule, cost, and quality objectives are not a top priority, and the project is not successful. Although project success rates are improving, the failure rate can be high when an objective such as quality attributes is not implemented. To avoid failure, the project manager and a team of systems and software engineers who build work products must develop an approach for project planning, oversee activities, and ensure configuration control is in place.

Software projects get in trouble when uncertainty and confusion come into play. There are times when systems and software designers do not communicate, so defined requirements are not discussed in relation to the developed work product. To eliminate this lack of communication, guidelines must be established, such as:

- Structure daily meetings
- Share ideas
- Inform project managers of problems occurring
- Listen and try to resolve complaints

Projects can have a “daily standup” meeting. One a day can get to the critical points or problems and resolve them that day. If there are concerns, discuss issues with a project manager, then you do not waste other team members’ time. In the past and currently, these meetings have an impact on hours of work that could be accomplish. My philosophy is that project managers need to have the confidence that their people can take care of the daily

routines, so the project managers do not need to attend meetings for hours and hours. Time is lost; then, people

routines, so the project managers do not need to attend meetings for hours and hours. Time is lost, then, people are ready to go home for the day knowing they have time sitting around listening to people who have no impacts on what they are trying to accomplish that day. Stop this right now. Let us go to work.

PLANNING

Communication planning principals define goals and objectives during the course of program and project planning. The planning aspects require a set of managers to understand not only their position but also the technical practices that support systems and software engineering and to define the course that lies ahead. There are many planning ideas and decisions by managers that are not accepted by team members due to the complexity of change. What should you do? Under planning, the program and project should consider eliminating chaos. The pressure on teams can be enormous, and useful guidance can be provided, such as:

- Providing a scope for the team to know what is ahead
- Involving systems and software teams to help with delivery schedules
- Planning to adjust and accommodate change
- Identifying risks that could have an impact on program and project planning
- Defining and understanding quality
- Tracking the progress daily and adjusting if needed

SENIOR MANAGEMENT

At the senior management level, program and project managers are required to provide effective planning and focus so teams can be effective during software design/development activities. Failure in planning is not an option and does jeopardize the success in achieving sound practices in program and project execution. Communication early in the process is the key to eliminate risks and the ability to embark on operational deployments. The required job of a senior manager is to provide the common framework for program and project planning to address engineering tasks. Many software managers begin their careers as software designers or developers. These types of managers serve:

- The company, military, and aerospace program and projects
- Their employees
- Themselves

When a software manager's team or organization delivers software to a customer in a timely fashion, this is called execution. There are questions that involve execution, such as:

- Do you have customer requirements?
- Do you have an approved budget?

- Do you have an approved budget?
- Do you have an approved plan and schedule?
- Are your program and project capable of dealing with change?
- Do you keep everyone focused?
- Do customers encounter quality issues with delivered work products?
- Do you measure work status on a regular basis?
- Do you find ways to improve?

Communication is important. A good software manager must learn to communicate in different ways, for example, providing formal presentations for upper-level management. Face-to- face communication to explain agreements with other program and project managers provides a road map and the plans for meeting goals.

E-mails work at times, but having a discussion will open up your team members to explain good and bad news. Also, communication is a positive way for team members to understand your expectations.

Program and project schedules that are not understood from the start will have an impact on resistance. To implement and use unreasonable schedules will imply that organizations and team members are not working hard. Customers are best served by creating work products that can be used over a long period of time.

Software managers must be aggressive and demand the best from designers and developers, but do not abuse them. Manage your teams wisely.

PROGRAM AND PROJECT PLANNING

The program and project planning method is well defined in the project planning process. The process area states the following:

The term “project plan” is used throughout this process area to refer to the overall plan for controlling the project. The project plan can be a stand-alone document or be distributed across multiple documents. In either case, a coherent picture of who does what should be included. Likewise, monitoring and control can be centralized or distributed, as long as at the project level a coherent picture of project status can be maintained.

The scale of numerous software design/development efforts is huge and can lead to confusion and coordination with affected teams. Internal organizations in programs and projects develop schedules and define processes and tasks. At the senior management level, managers assign responsibility, authority, and accountability to program and project managers or team leaders to define the software design/development (i.e., systems and software design, configuration management, quality engineering, etc.) to provide required support.

Planning activities include:

- Software lessons learned from previous programs and projects

- Software lessons learned from previous programs and projects
- Cost and schedule estimates and staffing plans
- Software and system requirement definitions
- Defined safety and security requirements
- Selection of appropriate software subcontractors
- Engineering documentation and historical data impacts
- Program and project objectives
- Contract understanding of required or necessary requirements

PLANNED SCHEDULES

The planned schedule defines tasks and processes to be conducted for implementation of those tasks and processes. The schedules that are planned affect team capabilities for risk assessment, configuration control, and quality. There are three critical factors in many software design/ development programs and projects (Figure 2.1). The scope, schedule, and budget combined affect the quality of work products.

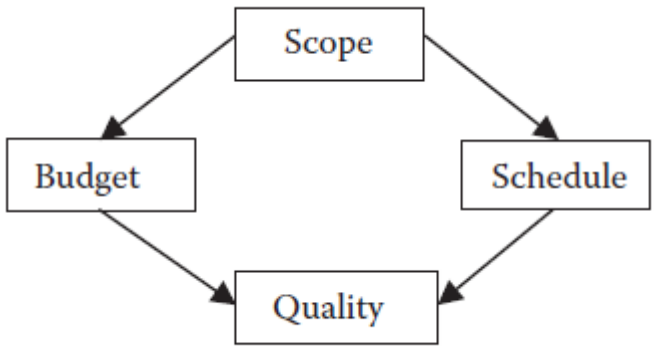

FIGURE 2.1

Planned schedules.

DEVELOPMENT PLAN

The critical items pertaining to a documented development plan consist of planned schedules and provide engineering information and direction for the production of software. It is important to know that the planning process is consistent with system-level planning. All major software design/development activities require consistency in accordance with the steps outlined in the use of development planning, including the following:

- Definition of entry and exit criteria for the software design/ development
- Review and assessment of the work product and task requirements
- Definition or updates of the process for each software activity
- Development or update of the estimating process

Development or update of the estimating process

- Development of initial cost and schedule estimation and risks
- Preparation of detailed implementation plans

TEAMWORK

An important element in all software programs and projects is teamwork, the coordination and communication within teams applied to meet work expectations. The effective methods for systems and software planning coordination provide value for a program and projects to far exceed high expectations. The software design/development energy and consistency appeal to achieve high-performance goals and aspirations. By having trust among teams, a cohesiveness is maintained in the work environment, and planning schedules becomes much easier to coordinate and implement within the team.

A plan developed is correct or successful when the team delivers a high-quality work product on time to meet the schedule and works within the budget. Remember that senior managers must encourage the program and project managers to work together with their teams to become effective, respond to customer expectations, and ensure quality.

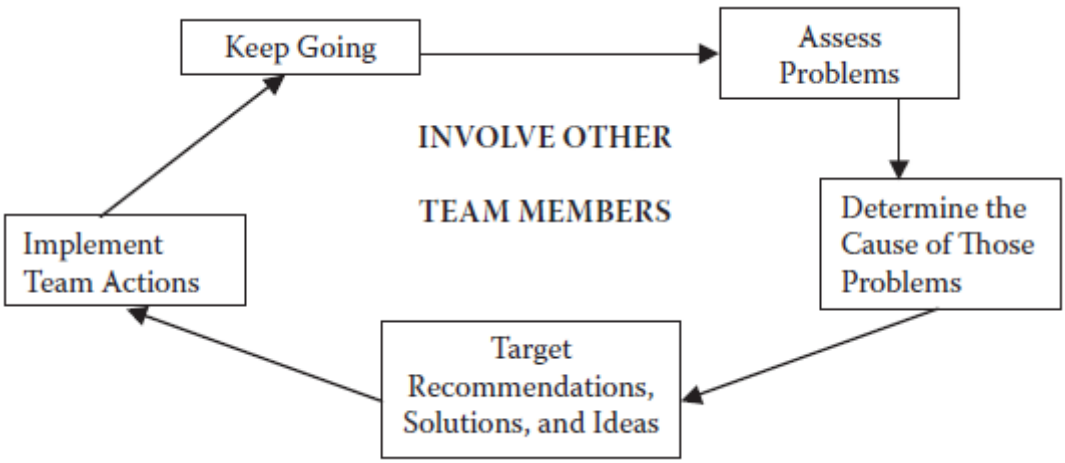

FIGURE 2.2

Team action cycle.

Managers do not control change but manage change.

As teams inside programs and projects become autonomous, they run the risk of pulling in different directions. One team that establishes goals to improve its own processes could subvert the efforts of other teams. When there is a face-to-face meeting as one group, teams are able to agree on proposed planning and project schedules and quality goals or expectations. In meeting as one group, the team will accomplish the following:

- Meet and achieve team objectives
- Resolve conflicts and issues
- Satisfy customer requirements

Satisfy customer requirements

When struggles with everyday challenges and problems are ignored, a team may use the required team action cycle shown in Figure 2.2.

TEAM CODE OF CONDUCT

It is okay for a team to fail but to be right at least 80% of the time. Teams that have the privilege and are able to provide clear communication and their own opinions seem to be successful. When one person speaks, listen and treat that person with respect. Once you help each other, you will:

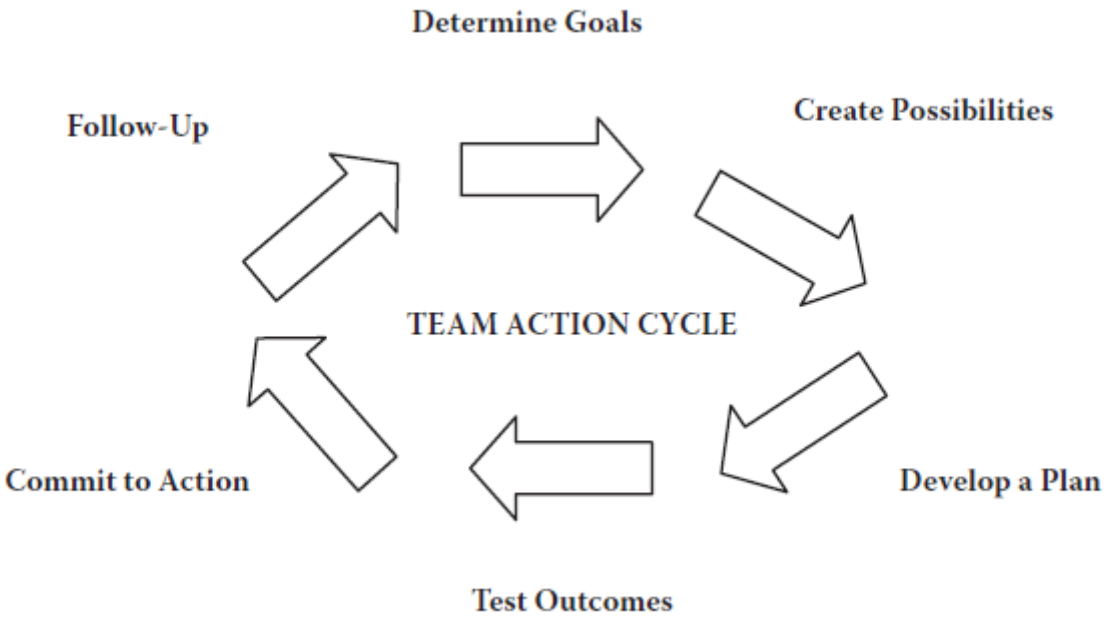

FIGURE 2.3

Team development life cycle.

- Show trust in every individual
- Be honest with your team
- Have ideas that show value
- Stop whining or crying

When teams are expected to attend meetings, be prepared and ensure that action items received are understood in connection with expected goals to be completed. Work together and do not be lazy. Many software designers get themselves in a mode of wanting to be left alone when coding. They get in a zone, so be polite, and do not interrupt, and show respect for other software designers.

The team process includes meetings; promise to honor meeting start and end times. Finally, bring your sense of

The team process includes meetings, promise to honor meeting start and end times. Finally, bring your sense of humor, be friendly and flexible, and always keep a positive attitude. As a software designer, I know the frustrations that could have an impact on jobs and careers in software design/development. Change from an individual to become a team player as shown in Figure 2.3.

CONCLUSION

Teams should not assume that being knowledgeable would offend others or expect other team members to understand what offends you. The team needs to recognize the relationship between the intent and impacts and stay away from misunderstandings and the scenario of assigning blame. Effective teams need to learn to manage their own reactivity and to be curious about what caused the blame. Practice letting members of a team know how something has an impact on you and rely on others' experience and expertise. There is no "I" in team.

◀ Preliminary Activity for Week 2 Jump to...

Analysis, Application, and Exploration for Week 2 ▶

Navigation

Home
Dashboard
Site pages
My courses
Capstone Project 1
Multimedia
Ojt/Practicum 1
Social And Professional Issues
System Integration And Architecture 2
Participants
General
01 Introduction To Software And Systems Integration
02 Program And Project Planning
Preliminary Activity for Week 2
Lesson Proper for Week 2
Analysis, Application, and Exploration for Week 2
Generalization for Week 2
Evaluation for Week 2
Assignment for Week 2
Courses

Fair Warning

NOTICE: Please be reminded that it has come to the attention of the Publishing Team of eLearning Commons that learning materials published and intended for ***free use only by students and faculty members within the eLearning Commons network were UNLAWFULLY uploaded in other sites without due and proper permission***.

PROSECUTION: Under Philippine law (Republic Act No. 8293), copyright infringement is punishable by the following: Imprisonment of between 1 to 3 years and a fine of between 50,000 to 150,000 pesos for the first offense. Imprisonment of 3 years and 1 day to six years plus a fine of between 150,000 to 500,000 pesos for the second offense.

COURSE OF ACTION: Whoever has maliciously uploaded these concerned materials are hereby given an ultimatum to take it down within 24-hours. Beyond the 24-hour grace period, our Legal Department shall initiate the proceedings in coordination with the National Bureau of Investigation for IP Address tracking, account owner identification, and filing of cases for prosecution.

Activities

- Assignments
- Forums
- Quizzes
- Resources