ICS 3207
SOFTWARE PROJECT MANAGEMENT

Chapter 4

Project Monitoring and Control

Introduction

- Once the work schedules have been published and the project is underway, attention is focused on ensuring progress.
- This requires monitoring of what is happening, comparison of actual achievement against the schedule and, where necessary, revision of plans and schedules to bring the project as far as possible back on target.

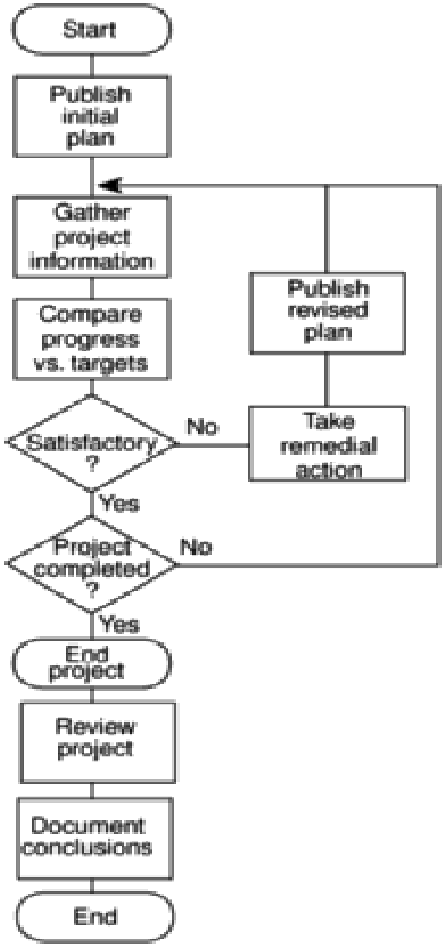
Creating a framework

- Exercising control over a project and ensuring that targets are met is a matter of regular monitoring, finding out what is happening, and comparing it with current targets.
- If there is a mismatch between the planned outcomes and the actual ones then either replanning is needed to bring the project on target or the target will have to be revised.

Creating a framework

- In practice, we are normally concerned with departures from the plan in four dimensions:
  - Delays in meeting target dates.
  - Shortfalls in quality.
  - Inadequate functionality, and
  - Costs going over target.

Project Control Cycle



Responsibility for Control

- The overall responsibility for ensuring satisfactory progress on a project is often the role of the project steering committee or project board.
- Day-to-day responsibility will rest with the project manager and, in all but the smallest of projects, aspects of this can be delegated to team members.

Responsibility for Control

- Reporting may be oral or written, formal or informal, or regular or ad hoc.
- While any effective team leader or project manager will be in touch with team members and available to discuss problems, any such informal reporting of project progress must be complemented by formal reporting procedures.

Assessing Progress

- This will normally be made on the basis of information collected and collated at regular intervals or when specific events occur.
- Whenever possible, this information will be objective and tangible.

Setting Checkpoints

- It is essential to set a series of checkpoints in the initial activity plan.
- Checkpoints may be:
  - Regular (e.g. monthly)
  - Tied to specific events such as the production of a report or other deliverable.

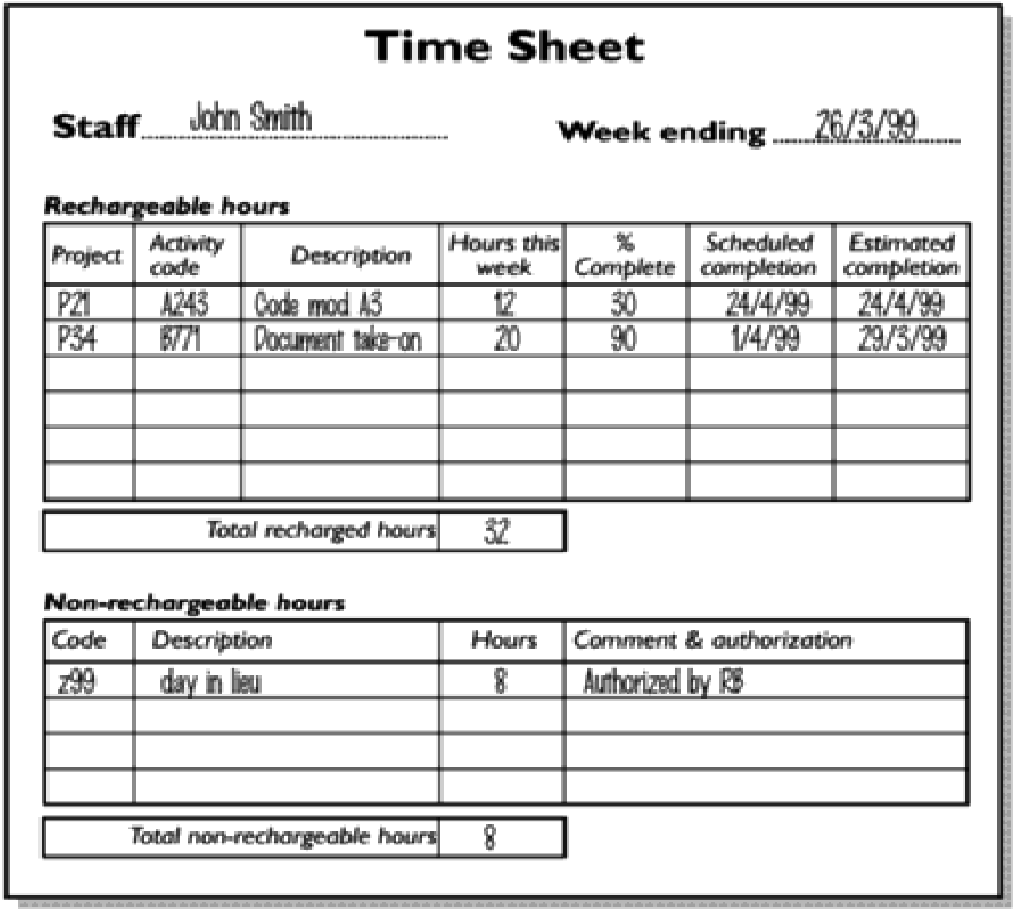
Taking Snapshots

- The frequency with which the project manager needs to receive information about progress will depend upon the size and degree of risk of the project or that part of the project under their control.
- E.g. team leaders need to assess progress daily whereas project managers may find weekly or monthly reporting appropriate.

Collecting Data

- As a rule, managers will try to break down long activities into more controllable tasks of one or two weeks duration.
- However, it will still be necessary to gather information about partially completed activities and, in particular, forecasts of how much work is left to be completed.

A Weekly time sheet and progress review form



Time Sheet

Staff John Smith **Week ending** 26/3/99

Rechargeable hours

Project	Activity code	Description	Hours this week	% Complete	Scheduled completion	Estimated completion
P21	A243	Code mod A3	12	30	24/4/99	24/4/99
P34	B771	Document take-on	20	90	1/4/99	29/3/99

Total recharged hours	32

Non-rechargeable hours

Code	Description	Hours	Comment & authorization
z99	day in lieu	8	Authorized by RB

Total non-rechargeable hours	8

Cost Monitoring

- Expenditure monitoring is an important component of project control.
- Not only in itself, but also because it provides an indication of the effort that has gone into (or at least been charged to) a project.
- A project might be on time but only because more money has been spent on activities than originally budgeted.

Cost Monitoring

- Project costs may be monitored by a company’s accounting system.

Prioritizing Monitoring

- Monitoring takes time and uses resources that might sometimes be put to better use.
- Priority that might be applied in deciding levels of monitoring are:
  - Critical path activities.
  - Activities with no free float.
  - Activities with less than a specified float.
  - High risk activities.
  - Activities using critical resources.

Critical path activities

- Any delay in an activity on the critical path will cause a delay in the completion date for the project.
- Critical path activities are therefore likely to have a very high priority for close monitoring.

Activities with no free float

- A delay in any activity with no free float will delay at least some subsequent activities even though, if the delay is less than the total float, it might not delay the project completion date.

Activities with less than a specified float

- If any activity has very little float, it might use up this float before the regular activity monitoring brings the problem to the project manager’s attention.
- It is common practice to monitor closely those activities with less than , say, one week free float.

High risk activities

- A set of high risk activities should have been identified as part of the initial risk profiling exercise.
- If PERT three-estimate approach is being used, a high risk activities are those that have a high estimated duration variance.
- These activities will be given close attention because they are most likely to overrun or overspend.

Getting Project back on target

- Almost any project will, at one time or another, be subject to delays and unexpected events.
- One of the tasks of the project manager is to recognize when this is happening (or, if possible, about to happen) and, with the minimum delay and disruption to the project team, attempt to mitigate the effects of the problem.

Getting Project back on target

- In most cases, the project manager tries to ensure that the scheduled project end date remains unaffected.
- This can be done by shortening:
  - Remaining activity durations or
  - The overall duration of the remaining project.

Shortening Critical Path

- The overall duration of a project is determined by the current critical path, so speeding up non-critical path activities will not bring forward completion date.

Reconsider the precedence requirements

- If attempting to shorten critical activities proves insufficient, the next step is to consider the constraints by which some activities have to be deferred pending completion of others.
- The original project network was most probably drawn up assumming "ideal" conditions and "normal" working practices.

Reconsider the precedence requirements

- One way to overcome precedence constraints is to subdivide an activity into a component that can start immediately and one that is till constrained as before.

Getting Project back on target

- Remember that shortening activity durations might not always be the most appropriate response to disruptions to a plan.
- There is little point in spending considerable sums in overtime payments in order to speed up a project if the customer is not overly concerned with the delivery date and there is no other valuable work for the team members once this project is completed.

Changes in Projects

- Change is inevitable when computer software is built.
- This increases the level of confusion among software engineers who are working on a project.

Changes in Projects

- Confusion arises when changes are not:
  - Analyzed before they are made.
  - Recorded before they are implemented.
  - Reported to those with a need to know.
  - Controlled in a manner that will improve quality and reduce error.

Sources of Change

- New business or market conditions that dictate changes in product requirements or business rules.
- New customer needs that demand:
  - Modification of data produced by information systems.
  - Functionality delivered by products.
  - Service delivered by a computer-based system.

Sources of Change

- Reorganization and/or business downsizing that causes changes in project priorities or software engineering team structure.

Sources of Change

- Budgetary or scheduling constraints what cause a redefinition of the system or product.

What is configuration Management?

- The process of identifying and defining the items in the system, controlling the changes to these items throughout their life cycle, recording and reporting the status of items and change requests, and verifying the completeness and correctness of items.

What is Configuration Management?

- The art of identifying, organizing, and controlling modifications to the software being built by a programming team.
- The goal is to maximize productivity by minimizing mistakes.

Functions of configuration Management

- Identification
  - Unique naming of the software objects to be managed.
- Control
  - Controlling the release of a product and changes to it throughout the software life cycle.

Functions of CM

- Status accounting
  - Recording and reporting the status of components and change requests.
- Audit and review
  - Validating the completeness of a product and maintaining consistency among the components.

Functions of configuration Management

- Manufacture:
  - Managing the construction and building of a product.

Functions of CM

- Process management:
  - Ensuring the carrying out of the organization's procedures, policies, and life cycle model.

Functions of CM

- Team work:
  - Controlling the work and interactions between multiple users on a product.

Software Maintenance and Software Configuration Management

- Software Maintenance:
  - A set of software engineering activities that occur after software has been delivered to the customer and put into operation.

Software Maintenance and Software Configuration Management

- Software Configuration Management:
  - A set of tracking and control activities that begin when a software project begins and terminate only when the software is taken out of operation.

Purpose of CM

- Change to an existing software system are inevitable. Configuration management ensures that these changes:
  - Take place in an identifiable and controlled environment.
  - Do not adversely affect any properties of the system.
  - Do not adversely affect the implementation of the security policy.

Purpose of CM

- It provides assurance that additions, deletions or changes made do not compromise the trust of the originally evaluated system.
- This identifies the components of the design and implementation of a system.

Configuration Identification

- This task may be accomplished through the use of:
  - Identifiers.
  - Baselines.

Baseline

- This is a Software Configuration Management concept that helps to control change without seriously impeding justifiable change.

Baseline

- It is a specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.

Baselining

1. Software engineering tasks produce one or more Software Configuration Items (SCIs).
2. SCI are reviewed and approved, SCIs placed in a project database (also called project library/software repository).

Baselining

3. When a member of a software engineering team wants to make modifications to a baselined SCI, it is copied from the project database into the engineer’s workspace.
4. The SCI can only be modified if SCM controls are followed.

The SCM tasks

- Identification.
- Version Control.
- Change Control.
- Configuration Auditing.
- Reporting.

Identification

- To control and manage software configuration items, each must be separately named and then organized.

identification

- Two types of objects can be identified:
  - Base objects: A ‘unit of text’ created by a software engineer during analysis, design, coding, or testing. E.g. a section of a requirements specification.
  - An aggregate object: a collection of basic objects and other aggregate objects. E.g. design specification.

Identification

- Each object has a set of distinct features that identify it uniquely:
  - Name
  - Description
  - Resources.
  - A “realization”: a pointer to the ‘unit of text’ for a basic object or null for an aggregate object.

Identification

- Identification must also consider the relationships that exist between named objects.

Configuration Control

- Involves a systematic:
  - Evaluation.
  - Coordination.
  - Approval or disapproval.

Configuration Control

- It begins in the earliest stages of the design and development of the system and extends over the full life of the configuration items included in the design and development stages.

Configuration Control

- Early initiation of configuration control procedures provides increased accountability for the system by making its development more traceable.

Configuration Control

- This early initiation serves a dual purpose:
  - It makes it possible to evaluate the impact of a change to the system and controls the change as it is being made.
  - There is less chance of making undesirable changes to a system that may later adversely affect the security of the system.

Configuration Control

- It specifies procedures to ensure that all documentation is updated properly and presents an accurate description of the system and software configuration.
- It covers a broader area than just documentation.

Configuration Control

- It maintains control of design data, source code, the running version of the object code, and test fixtures.
- Changes to any of these are subject to review and approval by an authorized authority.

Configuration Status Accounting

- Used in reporting on the progress of the development in very specific ways.
- It accomplishes this task through:
  - Data recording.
  - Data storing.
  - Data reporting.

Configuration Status Accounting

- Objective is to record and report all information that is of significance to the configuration management process.
- It establishes records and reports which enable proper logistics support to be established.

Configuration Status Accounting

- The records and reports produced through configuration status accounting should include:
  - A current configuration list.
  - A historical change list.
  - The original designs.
  - The status of change requests and their implementation.
  - The ability to trace all changes.

Configuration Audit

- Configuration Auditing:
  - This is the checking for top to bottom completeness of the configuration accounting information to ascertain that only the authorized changes have been made.

Configuration Audit

- A change should be reviewed and audited for its effect on the rest of the system.
- Configuration audits should be performed periodically.

Configuration Audit

- It should verify that:
  - The architectural design satisfies the requirements.
  - The detailed design satisfies the architectural design.
  - The code implements the detailed design.
  - The item/product performs per the requirements.
  - The configuration documentation and the item/product match.

Configuration Management Plan

- Procedures:
  - Procedures to ensure that both user and design documentation are updated in synchrony with all changes to the system.
  - Guidelines for creating and maintaining functional tests and documentation throughout the life of the system.
  - Procedures for how the design and implementation of changes are proposed, evaluated, and approved or disapproved.
  - Steps to take to ensure that only those approved changes are actually included and that the changes are included in all of the necessary areas.
  - Emergency procedures.

Configuration Management Plan

- Defines:
  - Types of documents to managed and a document naming scheme.
  - Who takes responsibility for CM procedures and creation of baselines.
  - Policies for change control and version management.
  - Tools which can be used to assist the CM process and their limitations.
  - The CM database used to record configuration information.

End