MODULE - IV Software Quality Management Components - Project Progress Control - components of project progress control- Progress control of internal projects and external participants Implementation of project progress control.

Software Quality Management Components

The three basic SQA management tools offered here are:
- Project progress control
- Software quality metrics
- Software quality costs.

The first managerial SQA tool, project progress control, enables management to oversee each project and initiate, when required, changes and improvements in how a project is performed. Software quality metrics and software quality costs provide overviews of the headway made in assuring software quality by identifying evolving trends.

Components of Project Progress Control

1.1 Planning and Scheduling

- Defining project objectives, deliverables, and milestones.
- Creating a Work Breakdown Structure (WBS) to divide the project into smaller tasks.
- Setting up a **project schedule** using techniques like **Gantt Charts, PERT, or Critical Path Method (CPM)**.
- Assigning resources and responsibilities for each task.

1.2 Monitoring and Tracking

- **Regular progress tracking** against planned schedules.
- Using tools like **Earned Value Management (EVM)** to measure cost and schedule performance.
- Continuous assessment of risks and dependencies.
- Identifying bottlenecks and taking corrective actions.

1.3 Performance Measurement and Evaluation

- Measuring progress using **Key Performance Indicators (KPIs)** such as:
    - Percentage of completed tasks.
    - Actual vs. planned budget usage.
    - Code quality and defect rates.
- Conducting reviews and audits to ensure compliance with quality standards.
- Using techniques like **benchmarking, variance analysis, and performance dashboards**.

1.4 Risk Management

- Identifying potential risks that may impact project progress.
- Implementing **mitigation strategies** like contingency planning, risk-sharing, and backup plans.
- Regular risk assessment meetings to update the risk register.

1.5 Communication and Reporting

- Establishing clear **communication channels** for reporting project progress.
- Using **status reports, dashboards, and meetings** to keep stakeholders informed.
- Ensuring transparency between internal teams and external participants.

Project progress control (CMM uses the term ―software project tracking‖) has one immediate objective: early detection of irregular events. Detection promotes the timely initiation of problem-solving responses.

The accumulated information on progress control as well as successes and extreme failures also serve a long-term objective: initiation of corrective actions.

The main components of project progress control are:
- Control of risk management activities
- Project schedule control
- Project resource control
- Project budget control.

Control of risk management activities

The software development team copes with software risk items by applying systematic risk management activities. Control of the progress of risk management begins with the preparation of periodic assessments about the state of software risk items and the expected outcomes of the risk management activities performed in their wake. Based on these reports, project managers are expected to intervene and help arrive at a solution in the more extreme cases. Several standards and many books and articles deal with software project risks, e.g., IEEE (2001) and Jones (1994),to mention just two.

Project schedule control

This deals with the project's compliance with its approved and contracted timetables. Follow-up is based mainly on milestones, which are set (in part) to facilitate identification of delays in completion of planned activities.

Milestones set in contracts, especially dates for delivery of specified software products to the customer or completion of a development activity, generally receive special emphasis.

Project resource control

This focuses on professional human resources but it can deal with other assets as well. For real-time software systems and firmware, software development and testing facilities resources typically demand the most exacting control. Here as well, management's control is based on periodic reports of resource use that compare actual to scheduled utilization because, it should be stressed, the true extent of deviations in resource use can be assessed only from the viewpoint of the project's progress.

Another aspect of resource control is internal composition or allocation. For example, management may find that no deviations have taken place in total man-months allocated to system analysts. However, review of itemized expenditures may disclose that instead of the 25% of man-months originally allocated to senior system analysts,50%wasactuallyspent,astepthatmayeventuallyunderminetheplannedbudget.Although **project**

budget controls also reveal deviations of this type, they do so at a much later stage of the project, a fact that impedes introduction of remedial actions. If the deviations are justified, management can intervene by increasing the resources allocated; alternatively, management can shift resources by reorganizing the project teams, revising the project's plan, and so forth.

Project budget control

This is based on the comparison of actual with scheduled expenditures. As in resource control, a more accurate picture of budget deviations requires that the associated delays in completion of activities be taken into consideration.

The main budget items demanding control are:
- Human resources
- Development and testing facilities
- Purchase of COTS software
- Purchase of hardware
- Payments to subcontractors.

Again, like resource control, budget control is based on milestones and the periodic reports that facilitate early identification of budget overruns. In cases of deviations by internal bodies, the menu of optional interventions is
similar to that applied in project resource control. In deviations by external participants, legal and other measures may also be applied.

Progress control of internal projects and external Participants

Project progress control is initiated in order to provide management with a comprehensive view of all the software development activities carried out in an organization. Nevertheless, in most organizations, project control provides, for different reasons, a limited view of the progress of internal software development and an even more limited view of the progress made by external participants.

Internal projects, such as those undertaken for other departments or projects dealing with software packages for the general software market, exclude, by definition, the option of external customers. These projects thus tend to occupy a lower place among management's priorities. External participants include subcontractors, suppliers of COTS software and reused software modules and, in some cases, the customer himself.

Project Progress Control applies differently to internal teams and external participants (vendors, clients, third-party contractors, etc.).

2.1 Progress Control of Internal Projects

Internal project progress control involves:

- **Daily Stand-up Meetings** (Agile methodology) or **Weekly Progress Reviews**.
- **Task Management Tools** like Jira, Trello, Asana, etc.
- **Code Reviews** and **Automated Testing** to maintain software quality.
- **Version Control** using Git, ensuring all team members track changes.
- **Change Control Mechanisms** to manage scope creep and modifications.

2.2 Progress Control of External Participants

When working with external vendors, suppliers, or offshore teams:

- **Well-defined contracts** with clear deliverables and deadlines.
- **Regular progress reports** from external participants.
- **Quality assurance checks** to validate outsourced components.
- **Periodic meetings and evaluations** to ensure compliance with project requirements.
- **Service Level Agreements (SLAs)** to define performance expectations.

---

3. Implementation of Project Progress Control

Implementing project progress control requires systematic processes and methodologies. The steps involved are:

3.1 Establishing a Progress Control Framework

- Define project milestones and check-in points.
- Assign clear responsibilities to project managers, team leads, and stakeholders.
- Choose appropriate **project management methodologies** (Agile, Waterfall, Scrum, etc.).

3.2 Data Collection and Analysis

- Use automated tools for data collection (Jenkins, SonarQube, JIRA reports, etc.).
- Analyze collected data using variance analysis and predictive analytics.
- Implement corrective actions in case of deviations.

3.3 Continuous Improvement and Adaptation

- Conduct regular **retrospective meetings** to identify areas of improvement.

- Adapt to changing requirements and market conditions.
- Implement best practices learned from previous projects.

3.4 Documentation and Reporting

- Maintain proper documentation of all progress-related activities.
- Prepare comprehensive project status reports for stakeholders.
- Use visual aids like **burn-down charts, velocity reports, and dashboards** for easy tracking.

Computerized tools for project progress control

Computerized tools for software project progress control are a clear necessity given the increasing size and complexity of projects on one hand, and the benefits they bring with them on the other. The comprehensive project management tools that have been available in the market for many years can serve most of the control components of software projects quite effectively and efficiently. The majority of these general-purpose packages apply PERT/CPM analysis so that the resulting reports take the interactions between activities and the criticality of each activity into account. These packages are usually readily adaptable to specific cases due to the great variety of options that they offer.

Examples of services that computerized tools can provide are as follows.

Control of risk management activities

■ Lists of software risk items by category and their planned solution dates.

■ Lists of exceptions of software risk items – overrun solution dates that can affect the project completion date.

Project schedule control

■ ■ Classified lists of delayed activities.

■ Classified lists of delays of critical activities – delays that can, if not corrected, affect the project's completion date.

■ Updatedactivityschedulesgeneratedaccordingtoprogressreportsandcorrectionmeasuresapplied–for teams, development units, etc.

■ Classified lists of delayed milestones.

■ Updated milestone schedules generated according to progress reports and applied correction measures – for teams, development units, etc.