UNIT- IV:

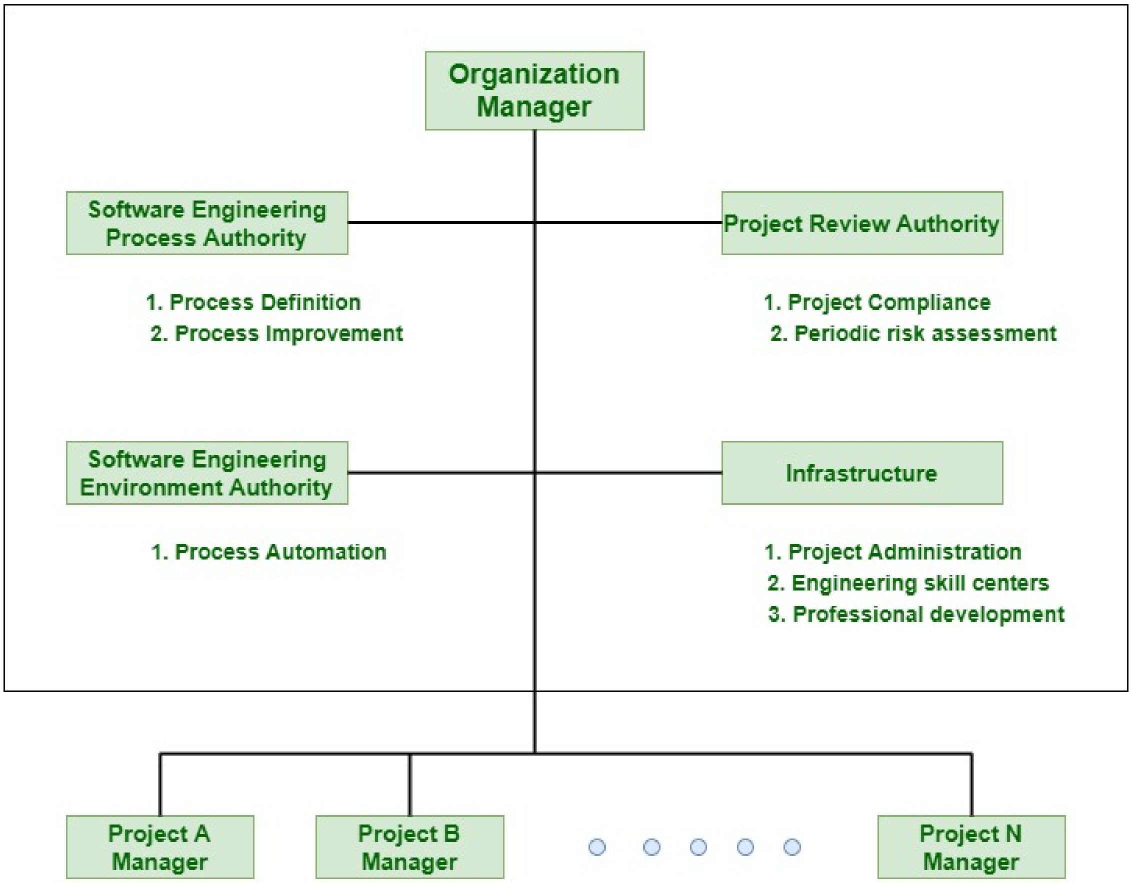
Project Organizations and Responsibilities: Line-of-Business Organizations, Project Organizations, evolution of Organizations.

Process Automation: Automation Building blocks, The Project Environment.

Project Control and Process instrumentation: The seven core Metrics, Management indicators, quality indicators, life cycle expectations, pragmatic Software Metrics, Metrics automation.

Line-of-Business Organizations:

Line business organizations need to support projects with infrastructure that are necessary and essential to make use of a common process. Line of business simply a general term that describes and explains products and services simply offered by a business or manufacturer. Software lines of the business are generally motivated and supported by Return of Investment (ROI), new business discriminators, market diversification, and profitability.



Default roles in a Software Line-of-Business Organization

Responsibility of organization:

- They are generally responsible for definition of process even maintenance of project process.
- They are also responsible for process automation. This is an organizational role and it is equally important to that of role of definition of process.
- The responsibility of organization's role or role of process automation is taken and achieved by a single individual or various other teams.

Various authorities of Organization:

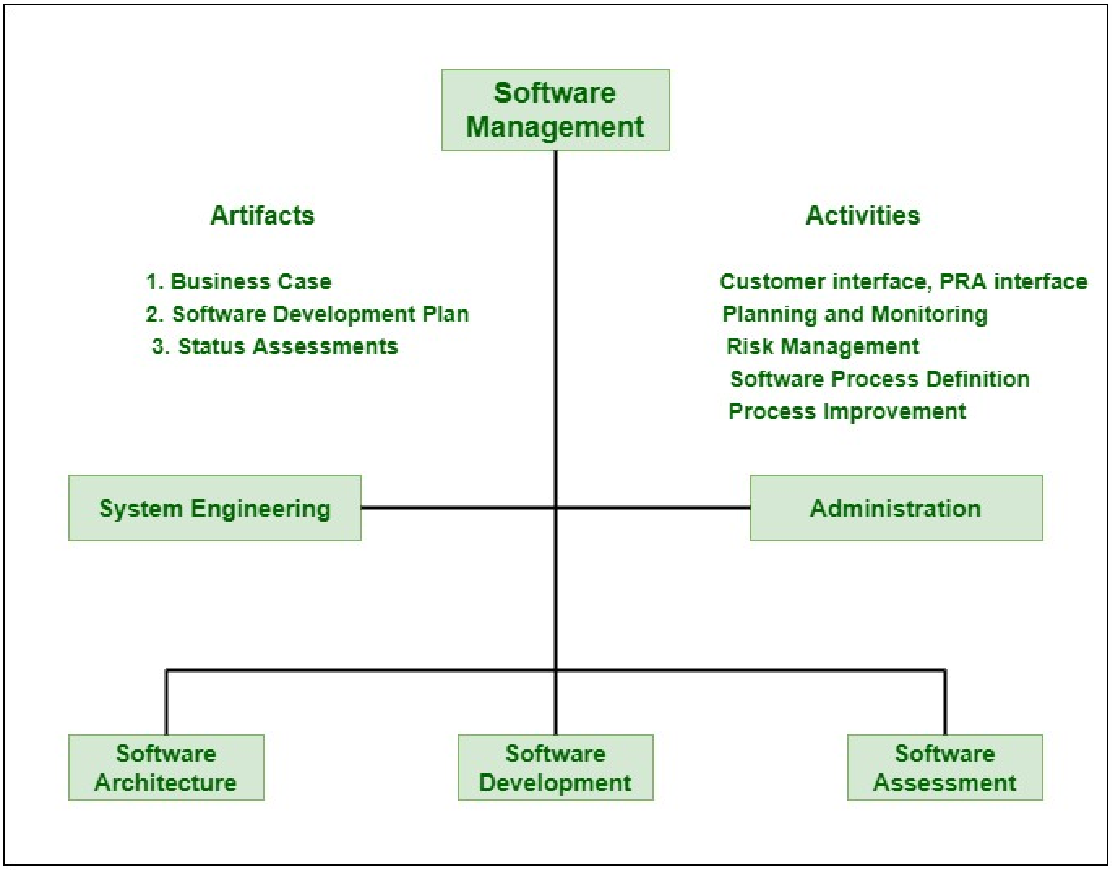
1. **Software Engineering Process Authority (SEPA) –**
   It is team that is responsible for exchanging information and guidance of project both to and from project practitioners. The project practitioners simply perform work and are usually responsible for one or more process activities. SEPA is a very important and essential role or responsibility in any of organizations.

2. **Project Review Authority (PRA) –**
   Project review is simply a scheduled status meeting that is taken on a regular basis. It includes project progress, issues, and risks. It is responsible for project review. The PRA generally reviews both conformance to contractual obligations and organizational policy obligations of project.

3. **Software Engineering Environment Authority (SEEA) –**
   SEEA is a very important role and is very much needed to achieve an ROI for a common process. It is simply responsible for supporting and managing a standard environment. Due to this, different tools, techniques, and training can be effectively amortized across all types of projects.

4. **Infrastructure –**
   Organizational infrastructure generally consists of systems, protocols, and various processes that provide structure to an organization, support human resources, supports organization in carrying out its vision, mission, goals, and values. It can range from trivial to largely entrenched bureaucracies. Various components of organizational infrastructure are Project administration, Engineering skill centres, and professional development.

Below diagram given that shows roles and responsibilities of a default project organization. Project organizations generally need to allocate artifacts and responsibilities across project team simply to ensure and confirm a balance of global (architecture) and local (component) concerns.



Default Project Organization and Responsibilities

Teams of Organization:

1. **Project Management Team –**
   It is an active and highly enthusiastic participant. They are responsible for producing, developing, and managing project.
2. **Architecture Team –**
   They are generally responsible for real artifacts and even for integration of components. They also find out risks of product misalignment with requirements of stakeholders and simply ensure that solution fits defined purpose.
3. **Development Team –**
   They are responsible for all work that is necessary to produce working and validated assets.
4. **Assessment Team –**
   They are responsible for assessing quality of deliverables.

Advantages of a Line Organization

- Simple to work
- Economical and effective. It also allows quick decisions and efficient coordination.
- Conforms to the scalar principle of organization. Further, it promotes the unity of command.
- In a line organization, the responsibility for the performance of tasks is fixed upon definite individuals. Therefore, there is accountability of delegated tasks.
- There is excellent discipline in a line organization due to unified control and undivided loyalties.
- The overall cost of running the organization is low due to the non-involvement of staff personnel.
- It is a stable form of organization.

Disadvantages of a Line Organization

- A line organization can suffer from a lack of specialization. This is because each department manager is concerned only with the activities of his own department. Therefore, employees are skilled in tasks pertaining to their departments alone.
- These organizations can overburden a keyman or a few key-men to the extent of their breaking point. Also, in the absence of a staff aid, if a strong man seizes the organization, he can run it arbitrarily. Such arbitrary power can lead to a considerable damage to the organization.
- Such organizations usually suffer from a lack of expert advice. If the line manager has trouble making a decision, there is no expert staff that he can turn to.
- A line organization is usually rigid and inflexible. In fact, such organizations maintain discipline so rigorously that they can rarely change.
- These organizations are based on the autocratic system of management.
- The division of work is not based on any scientific plan but on the whims of the manager.
- It might stop progress and prevent the unit to work effectively.
- Such organizations might also encourage nepotism or favouritism based on relationship or friendship.

Project Organizations:

In Software Project Management (SPM), Project Organizations are structures that facilitate and motivate coordination and implementation of activities of the project[1]. Their main purpose is to create an environment that encourages development of interactions between team members with minimal disruptions, overlaps, and conflicts.

The most important decision of a project management team is the form of organization structure that will be required and essential for the project. The organization should evolve with Work Breakdown Structure (WBS) and life-cycle concerns.

Responsibilities of these organizations include the definition and maintenance of project processes, and they may also be responsible for process automation. Various roles within these organizations include the Software Engineering Process Authority (SEPA), which is responsible for exchanging information and guidance of project both to and from project practitioners, and the Project Review Authority (PRA), which is responsible for project review. The Software Engineering Environment Authority (SEEA) is responsible for supporting and managing a standard environment.

There are many ways to organize the project team. Some important ways are as follows:

- Hierarchical team organization
- Chief-programmer team organization
- Matrix team organization
- Egoless team organization
- Democratic team organization

Each of these organizational structures has its own advantages and disadvantages, and the choice of structure depends on the specific needs and context of the project.

Evolution of Organizations:

The evolution of organizations in Software Project Management (SPM) is a dynamic process that adapts to the changing needs of the project at each stage of its life cycle. This involves adjusting the team structure, roles, and responsibilities, as well as adapting to changes in the external environment.

Inception Phase:

During the inception phase, the team moves from abstract, broad conversations into more detailed discussions. They capture fine details of what’s going to happen next and work that is needed to be completed to achieve the agreed-upon goal. This team focuses on planning, with more support from various teams to confirm and ensure that plans represent the general agreement of all perspectives.

Elaboration Phase:

In the elaboration phase, the team gains a handle on the architecture of the system. They begin setting up the environment for construction by purchasing hardware, software, and tools.

Construction Phase:

During the construction phase, the team focuses on building the product. This involves coding, testing, and debugging. The team structure during this phase typically involves developers, testers, and project managers.

Transition Phase:

In the transition phase, the team focuses on deploying the product and handing it over to the customer. This involves final testing, user training, and product support.

Adapting to External Changes:

The organization must also adapt to new environments, changing technology, and changes in other external variables so as to efficiently utilize available resources. It's important to track and examine client complaints to find recurrent problems and areas that might be improved. High staff turnover may be an indication of deeper problems.

Organizational Structures:

There are many ways to organize the project team. Some important ways are as follows:

- Hierarchical team organization
- Chief-programmer team organization
- Matrix team organization
- Egoless team organization
- Democratic team organization

Each of these organizational structures has its own advantages and disadvantages, and the choice of structure depends on the specific needs and context of the project.

Process Automation:

Process Automation in Software Project Management (SPM) is a crucial aspect that helps in improving efficiency, reducing errors, and freeing up time for project managers to focus on more critical tasks.

What is Process Automation?

Process Automation is the assigning of repetitive or mundane tasks to machines or software. Since machines can complete tasks with little to no assistance from people, delegating them frees up a project manager's time for other essential assignments.

Examples of Process Automation

1. Planning Phase: In the planning phase, automation can help in defining scope and deliverables, identifying key stakeholders, and developing a project plan.
2. Execution Phase: Once a project is in motion, there are many tasks that could be ripe for automation in the areas of task assignment, progress tracking, and resource allocation.
3. Monitoring Phase: During the monitoring phase, automation can help in tracking project progress, identifying issues, and generating reports.

Common Tasks for Automation

Some tasks project managers commonly automate are task reminders, job intake, and internal approvals. When deciding on processes to automate and which to leave to people, ask yourself questions like: What are our most time-consuming manual administrative tasks? Which processes are already well-documented and mapped out? Which project tasks are often repeated with little to no variation? Can those processes take place without human supervision? What would happen if one of those tasks were completed improperly? How much of an effect on the project would it have?

Limitations of Process Automation

You need a clearly defined path to make automation work, so processes that have holes or bottlenecks can’t be fixed with automation. Be sure your workflows are tightly in place before delegating them to software.

Conclusion

In conclusion, Process Automation in SPM is about replacing time-consuming administrative work to make tasks and collaboration happen more smoothly. It can keep track of tasks, deadlines, and more so you don’t have to. Just remember, you're only as good as the tools you have. So make sure to invest in quality automation software so that it can help streamline your workflow and keep you on top of all the moving parts of your projects.

Automation Building blocks:

Automation building blocks in Software Project Management (SPM) refer to the tools and technologies used to automate various aspects of the software development process. These tools are essential for improving efficiency, reducing errors, and freeing up time for project managers to focus on more critical tasks.

Macroprocess (Project)

The automation support for a project's process is called an environment. This environment provides the necessary infrastructure for the software development process, including the tools and technologies used for automation.

Microprocess (Iteration)

The automation support for generating artifacts is generally called a tool[134]. These tools can range from code editors and compilers to testing frameworks and deployment tools.

Tools: Automation Building Blocks

Many tools are available to automate the software development process. Here are some examples:

1. **Version Control:** Tools like Git are popular for version control, which is an essential step in the software development process.
2. **Static Analysis and Code Quality:** Tools like SonarQube are effective for ongoing code quality inspection.
3. **Continuous Integration and Configuration Management:** Tools such as Puppet, Chef, Ansible, etc., thrive in the space of DevOps automation, as it relates to automated deployments across different environments.
4. **Workflow Management Software:** This is an advanced platform that provides flexible tools to improve the way you work in an efficient manner.
5. **Project Management Software:** These software solutions can help you create digital workflows and manage repetitive and time-consuming processes, such as reporting status, assigning tasks, updating requests, and tracking resources.

Benefits of Automation Building Blocks

Automation building blocks can bring several benefits to the software development process:

1. **Improved Efficiency:** Automation can significantly speed up the software development process by automating repetitive tasks.
2. **Reduced Errors:** Automation can help reduce errors by ensuring that tasks are performed consistently and accurately.
3. **Freed Time:** By automating mundane tasks, project managers can free up their time to focus on more critical tasks.

In conclusion, automation building blocks play a crucial role in SPM. They help improve efficiency, reduce errors, and free up time for project managers, thereby contributing to the successful completion of software projects.

The Project Environment:

The Project Environment in Software Project Management (SPM) refers to all the internal and external factors that influence a project. It's crucial for project managers to understand and manage these factors proactively, as they can significantly impact the project's schedule, budget, team morale, and much more.

Definition

The project environment includes all factors that, directly or indirectly, externally or internally, influence your project. It represents a connection where the project is processed and impacts the project, and is, therefore, conditioned. Such an interaction is provided by numerous factors as operational, physical, ecological, social, cultural, economic, psychological, financial, organizational, etc.

Importance

Understanding the project environment is crucial for the success of a project. It helps in identifying project stakeholders and their potential to influence the project's success. This entails collaborating with others to produce the most outstanding outcomes, particularly in highly technical and complicated contexts like those seen in current building projects.

Key Elements

Two of the main elements of a project environment are internal and external factors:

1. **Internal Factors:** These include the project team, the project's objectives, scope, budget, and schedule, the organizational structure, and the organizational culture.
2. **External Factors:** These include the physical environment where the project is happening, the social environment, the local community's opinion of the project, and the demographic of the area.

Managing the Project Environment

Managing the project environment involves being aware of these factors and preparing for their influence on your project throughout the project management life cycle[2]. It's sort of like managing risk in that way[2]. The project manager must understand the project environment and proactively plan to manage the factors that might influence the project.

Project Control and Process instrumentation:

Project Control and Process Instrumentation are key aspects of Software Project Management (SPM). They involve the use of software metrics to implement the activities and products of the software development process[6].

Project Control

Project control involves data gathering, management, and analytical processes used to predict, understand, and constructively influence the time and cost outcomes of a project or program. It assists effective management and decision-making.

There are three management indicators used in project control:

1. **Work and Progress:** This metric measures the work performed over time.
2. **Budgeted Cost and Expenditures:** This metric measures the cost incurred over time.
3. **Staffing and Team Dynamics:** This metric measures personnel changes over time.

Process Instrumentation

Process Instrumentation involves measuring, positioning, recording, and controlling key parameters for all industrial processes. It provides an efficient means to increase plant efficiency and improve product quality.

There are four quality indicators used in process instrumentation:

1. **Change Traffic and Stability:** These metric measures change traffic over time.
2. **Breakage and Modularity:** This metric measures the average breakage per change over time.
3. **Rework and Adaptability:** This metric measures the average rework per change over time.
4. **Mean Time Between Failures (MTBF) and Maturity:** This metric measures the defect rate over time.

Project Control and Process Instrumentation are essential for managing the progress and quality of a software project. They provide valuable insights that help in making informed decisions and ensuring the successful completion of the project.

The seven core Metrics:

The seven-core metrics in Software Project Management (SPM) are essential for managing the progress and quality of a software project. They provide valuable insights that help in making informed decisions and ensuring the successful completion of the project. These seven metrics are categorized into two categories: management indicators and quality indicators.

Management Indicators

Management indicators are metrics that must be interpreted to serve as indicators. They might indicate a problem or issue, minor perturbation, or simply a change that is generally being managed. The three management indicators are:

1. **Work and Progress:** This metric measures the work performed over a given period of time.
2. **Budgeted Cost and Expenditures:** This metric measures the cost incurred over a given period of time.
3. **Staffing and Team Dynamics:** This metric measures personnel changes occurred over a given period of time.

Quality Indicators

Quality indicators are metrics that provide insight into how the process and product are developing gradually. The four quality indicators are:

1. **Change Traffic and Stability:** These metric measures change traffic over time.
2. **Breakage and Modularity:** This metric measures the average breakage per change over time.
3. **Rework and Adaptability:** This metric measures the average rework per change over time.
4. **Mean Time Between Failures (MTBF) and Maturity:** This metric measures the defect rate over time.

In conclusion, these seven-core metrics play a crucial role in SPM. They help improve efficiency, reduce errors, and free up time for project managers, thereby contributing to the successful completion of software projects.

Management indicators:

Management indicators are crucial metrics in Software Project Management (SPM) that provide insights into the project’s progress and help in decision-making. They are typically used to track and control the progress of a project. The three key management indicators are:

1. **Work and Progress:** This metric measures the work performed and the progress made over a given period of time. It helps in understanding how much work has been done and how much is left, tracking the progress of the project, and identifying any delays or issues early on.

2. **Budgeted Cost and Expenditures:** This metric measures the cost incurred over a given period of time. It helps in understanding how much money has been spent and how much is left, tracking the financial progress of the project, and identifying any cost overruns early on.

3. **Staffing and Team Dynamics:** This metric measures personnel changes occurred over a given period of time. It helps in understanding how the team is changing and evolving over time, tracking the team dynamics, and identifying any issues or conflicts early on.

Quality indicators:

Quality indicators in Software Project Management (SPM) are metrics that provide insights into the quality of the software development process and the product being developed. They are typically used to track and control the quality of a project. The four key quality indicators are:

1. **Change Traffic and Stability:** This metric measures the volume of change traffic over time. It helps in understanding the stability of the project and the impact of changes on the project's progress. A high volume of change traffic might indicate instability in the project, which could lead to delays or issues.
2. **Breakage and Modularity:** This metric measures the average breakage per change over time. It helps in understanding the modularity of the software and the impact of changes on the software's structure. A high average breakage might indicate a lack of modularity in the software, which could lead to difficulties in maintaining and extending the software.
3. **Rework and Adaptability:** This metric measures the average rework per change over time. It helps in understanding the adaptability of the software and the impact of changes on the software's functionality. A high average rework might indicate a lack of adaptability in the software, which could lead to difficulties in adapting the software to new requirements or environments.
4. **Mean Time Between Failures (MTBF) and Maturity:** This metric measures the defect rate over time. It helps in understanding the maturity of the software and the reliability of the software. A high defect rate might indicate a lack of maturity in the software, which could lead to reliability issues.

The quality indicators play a crucial role in SPM. They provide valuable insights that help in making informed decisions, ensuring the successful completion of the project, and contributing to the overall success of the software development process.

Life cycle expectations:

In Software Project Management (SPM), the life cycle expectations refer to the various stages a project goes through from its inception to its completion. These stages provide a structured plan for project managers to guide their projects to successful completion. The life cycle typically includes the following phases:

1. **Initiation:** This is the first phase of the project where you define the goals, scope, budget, and timeline. It involves identifying the project's purpose, stakeholders, and desired outcomes.
2. **Planning:** This phase involves creating a detailed action plan for the project. It includes establishing budgets, timelines, and milestones, sourcing materials and necessary documents, calculating and predicting risk, putting change processes into place, and outlining communication protocols.
3. **Execution:** In this phase, the plans are carried out to deliver the product. This involves coding, testing, and debugging. The team structure during this phase typically involves developers, testers, and project managers[12].
4. **Monitoring and Controlling:** Once the project begins, it must be monitored and controlled for any deviations from the plan. This involves tracking project progress, identifying issues, and generating reports[12].
5. **Closure:** This is the final phase of the project life cycle. It involves wrapping up tasks, obtaining project acceptance, and archiving records.

Each of these phases has specific expectations and deliverables associated with it. By understanding these expectations, project managers can ensure that they are meeting the goals of each phase and moving the project towards successful completion.

Pragmatic Software Metrics:

Pragmatic Software Metrics in Software Project Management (SPM) are practical, actionable metrics that provide insights into the software development process and its products. They are used to track and control the progress and quality of a project.

Types of Software Metrics

There are two types of software metrics:

1. **Product Metrics:** These are measures of various characteristics of the software product. They can be computed for different stages of the Software Development Life Cycle (SDLC).
2. **Process Metrics:** These are measures of various characteristics of the software development process. They are used to measure the characteristics of methods, techniques, and tools that are used for developing software.

Different Types of Metrics Based on What They Measure

There are also different types of metrics based on what they measure:

1. **Internal Metrics:** These metrics are used for measuring properties that are viewed to be of greater importance to a software developer.
2. **External Metrics:** These metrics are used for measuring properties that are viewed to be of greater importance to the user.
3. **Hybrid Metrics:** These metrics combine product, process, and resource metrics.
4. **Project Metrics:** These metrics are used by the project manager to check the project's progress.

Importance of Pragmatic Software Metrics

Pragmatic Software Metrics play a crucial role in SPM. They provide valuable insights that help in making informed decisions, ensuring the successful completion of the project, and contributing to the overall success of the software development process. They help improve efficiency, reduce errors, and free up time for project managers.

Pragmatic Software Metrics are essential for managing the progress and quality of a software project. They provide valuable insights that help in making informed decisions and ensuring the successful completion of the project.

Metrics automation:

Metrics automation in Software Project Management (SPM) refers to the use of automated tools and techniques to collect, analyse, and report on software metrics. This can significantly improve the efficiency and accuracy of the project management process.

Opportunities for Automation

There are many opportunities available to automate the project control activities of a software project. For instance, a Software Project Control Panel (SPCP) is essential for managing against a plan. This panel integrates data from multiple sources to show the current status of some aspect of the project.

Benefits of Metrics Automation

Metrics automation can provide several benefits to the software development process:

1. **Efficiency:** Automated tools can collect and analyse data much faster than humans, thereby improving the efficiency of the project management process.
2. **Accuracy:** Automated tools can reduce the risk of human error, thereby improving the accuracy of the data collected and the analyses performed.
3. **Consistency:** Automated tools can ensure that data is collected and analysed in a consistent manner, thereby improving the reliability of the metrics.

4. **Timeliness:** Automated tools can provide real-time updates on the status of the project, thereby enabling project managers to make timely decisions.

The metrics automation plays a crucial role in SPM. It provides valuable insights that help in making informed decisions, ensuring the successful completion of the project, and contributing to the overall success of the software development process.