MODULE 4

SYLLABUS

- Software Project management, Risk management, configuration management, version- change-release- agile s/w management – SCRUM Frame work, Kanban methodology and lean approaches-COCOMO Model

S/W Project Management

- **Software Project Management (SPM)** is a **proper way of planning and leading** software projects. It is a part of project management in which software projects are planned, implemented, monitored, and controlled.
- The goals of project management
- ■ to deliver the software to the customer at the agreed time;
- ■ to keep overall costs within budget;
- ■ to deliver software that meets the customer's expectations;
- ■ to maintain a coherent and well-functioning development team.

ACTIVITIES OF SPM

- 1. Project planning: Project managers are responsible for **planning, estimating, and scheduling project development and assigning people to tasks.** They supervise the work to ensure that it is carried out to the required standards, and they monitor progress to check that the development is on time and within budget.

- 2. Risk management: Project managers have to **assess the risks that may affect a project, monitor these risks, and take action** when problems arise.

- 3. People management: **Project managers are responsible for managing a team of people.** They have to choose people for their team and establish ways of working that lead to effective team performance.

- 4. Reporting: Project managers are usually responsible for reporting on the progress of a project to customers and to the managers of the company developing the software. They have to write concise, coherent documents that abstract critical information from detailed project reports. They must be able to present this information during progress reviews.

- 5. Proposal writing: It involve writing a proposal to win a contract to carry out an item of work. The proposal describes the objectives of the project and how it will be carried out. It usually includes cost and schedule estimates and justifies why the project contract should be awarded to a particular organization or team. Proposal writing is a critical task as the survival of many software companies depends on having enough proposals accepted and contracts awarded.

Risk Management

- It involves identifying and estimating the probability of risks with their order of impact on the project.
- There are some steps that need to be followed in order to reduce risk. These steps are as follows:
- **1. Risk Identification:**
  Risk identification involves the preparation of a risk list.
- Preparation of risk list involves identification of risks that are occurring continuously in previous software projects.

- **2. Risk Analysis and Prioritization:**
  It is a process that consists of the following steps:

  1. Identifying the problems causing risk in projects

  2.Identifying the probability of occurrence of problem

  3.Identifying the impact of problem

- Assigning values to step 2 and step 3 in the range of 1 to 10
- Calculate the risk exposure factor which is the product of values of step 2 and step 3
- Prepare a table consisting of all the values and order risk on the basis of risk exposure factor

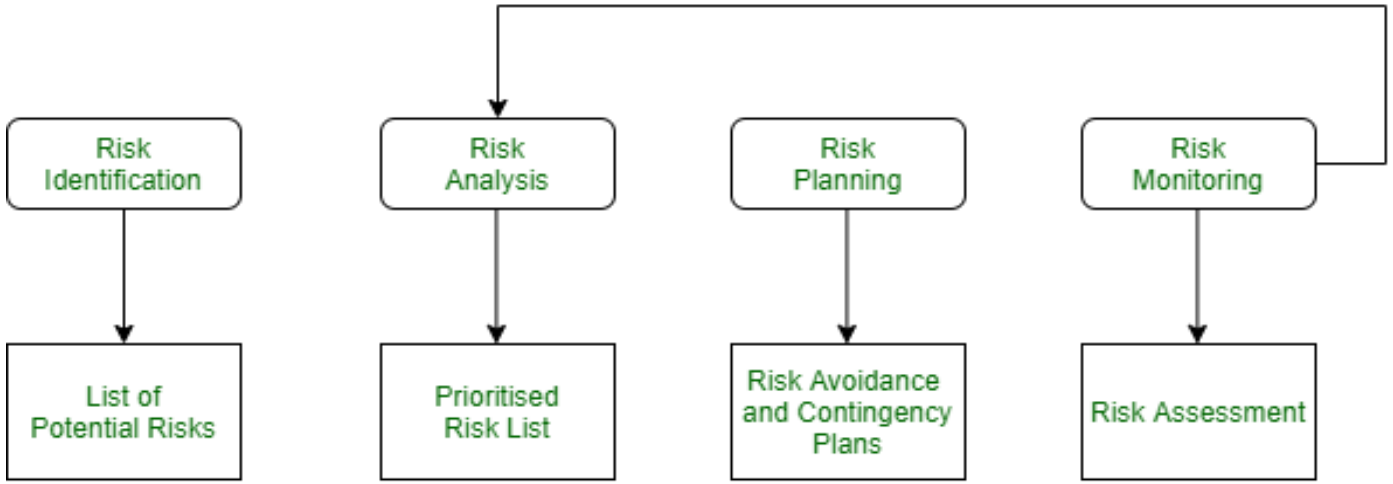
- Minimal Impact (1-3)
- Low to Moderate Impact (4-6)
- Significant Impact (7-8)
- Severe Impact (9-10)

Risk No	Problem	Probability of occurrence of problem	Impact of problem	Risk exposure	Priority
R1	Issue of incorrect password	2	2	4	10
R2	Testing reveals a lot of defects	1	9	9	7
R3	Design is not robust	2	7	14	5

- **3. Risk Avoidance and Mitigation:**
  The purpose of this technique is to altogether eliminate the occurrence of risks. so the method to avoid risks by removing non-essential requirements.

- **4. Risk Monitoring:**
  In this technique, the risk is monitored continuously by re-evaluating the risks, the impact of risk, and the probability of occurrence of the risk.
  This ensures that:

- *Risk has been reduced*
- *New risks are discovered*
- *Impact and magnitude of risk are measured*



The Risk Management process

Managing people

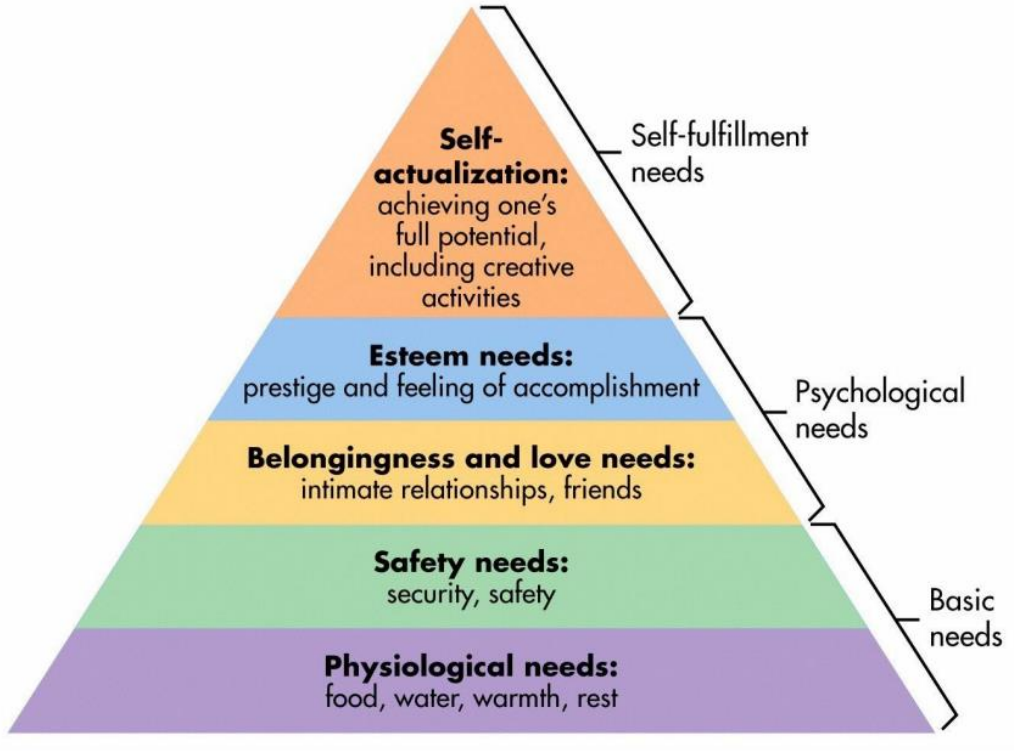
- The people working in a software organization are its greatest assets.
- It is expensive to recruit and retain good people, and it is up to software managers to ensure that the engineers working on a project are as productive as possible.
- It is important that software project managers understand the technical issues that influence the work of software development.
- As a project manager, you should be aware of the potential problems of people management and should try to develop people management skills.

- There are four critical factors that influence the relationship between a manager and the people
- 1. Consistency: All the people in a project team should be treated in a comparable way(equally and fairly)
- 2. Respect: Different people have different skills, and managers should respect these differences. All members of the team should be given an opportunity to make a contribution.
- 3. Inclusion: It is important to develop a working environment where all views, even those of the least experienced staff, are considered.
- 4. Honesty: As a manager, you should always be honest about what is going well and what is going badly in the team. You should also be honest about your level of technical knowledge and be willing to defer to staff with more knowledge when necessary.

Motivating people

- As a project manager, you need to motivate the people who work with you so that they will contribute to the best of their abilities.
- In practice, motivation means organizing work and its environment to encourage people to work as effectively as possible.
- If people are not motivated, they will be less interested in the work they are doing.

- people are motivated by satisfying their needs. These needs are arranged in a series of levels(figure).
- The lower levels of this hierarchy represent fundamental needs for food, sleep, and so on.
- Social need/Belongingness need: is concerned with the need to feel part of a social grouping.
- Esteem need represents the need to feel respected by others, and self- realization need is concerned with personal development.


Self-actualization: achieving one's full potential, including creative activities
Esteem needs: prestige and feeling of accomplishment
Belongingness and love needs: intimate relationships, friends
Safety needs: security, safety
Physiological needs: food, water, warmth, rest
Self-fulfillment needs
Psychological needs
Basic needs

Making sure that peoples' social, esteem, and self-realization needs are satisfied is most important from a management point of view.

1. To satisfy social needs, you need to give people time to meet their co-workers and provide places for them to meet.

2. To satisfy esteem needs, you need to show people that they are valued by the organization.

3. Finally, to satisfy self-realization needs, you need to give people responsibility for their work and provide opportunities for training and development where people can enhance their skills.

Training is an important motivating influence as people like to gain new knowledge and learn new skills.

Professional workers- Team members

- Three classifications for professional workers are identified:
- 1. Task-oriented people, who are motivated by the work they do. In software engineering, these are people who are motivated by the intellectual challenge of software development.
- 2. Self-oriented people, who are principally motivated by personal success and recognition. They are interested in software development as a means of achieving their own goals. They often have longer-term goals, such as career progression, that motivate them, and they wish to be successful in their work to help realize these goals.
- 3. Interaction-oriented people, who are motivated by the presence and actions of co-workers.

TEAM WORK

For a Successful team work includes

- Selecting group members
- Group organization
- Group communications

- The people in the group: You need a mix of people in a project group as software development involves diverse activities such as negotiating with clients, programming, testing, and documentation.

- The way the group is organized: A group should be organized so that individuals can contribute to the best of their abilities and tasks can be completed as expected.

- Technical and managerial communications: Good communication between group members, and between the software engineering team and other project stakeholders, is essential.

PROJECT PLANNING or PLAN DRIVEN DEVELOPMENT

- **Project Initiation:**
  - Define the project's purpose, objectives, and scope.
  - Identify key stakeholders and their roles.
- **Requirement Gathering:**
  - Collaborate with stakeholders to collect and document detailed requirements.
  - Prioritize and validate requirements to ensure they align with project goals.
- **Project Feasibility Analysis:**
  - Assess the project's feasibility in terms of technical, financial, and operational aspects.
  - Determine if the project is viable and worth pursuing.
- **Team Formation:**
  - Assemble a project team with the necessary skills and expertise.
  - Define roles and responsibilities for team members.

- **Risk Assessment:**
  - Identify potential risks and challenges that may impact the project.
  - Develop a risk management plan to mitigate and manage these risks.
- **Project Planning:**
  - Create a project plan that outlines tasks, timelines, and resource allocation.
  - Develop a Work Breakdown Structure (WBS) to break the project into smaller, manageable tasks.
  - Define project milestones and deliverables.
- **Budgeting:**
  - Determine the project budget based on resource requirements and cost estimates.
  - Allocate resources efficiently to stay within budget constraints.

- **Scheduling:**
  - Create a project schedule, including start and end dates for each task and milestone.
  - Use project management tools to track progress and manage dependencies.
- **Quality Assurance Planning:**
  - Develop a plan for ensuring the quality of the software, including testing and review processes.
  - Define quality criteria and metrics.
- **Communication Plan:**
  - Establish a communication plan to keep stakeholders informed about project progress and changes.
  - Define reporting mechanisms and frequency.

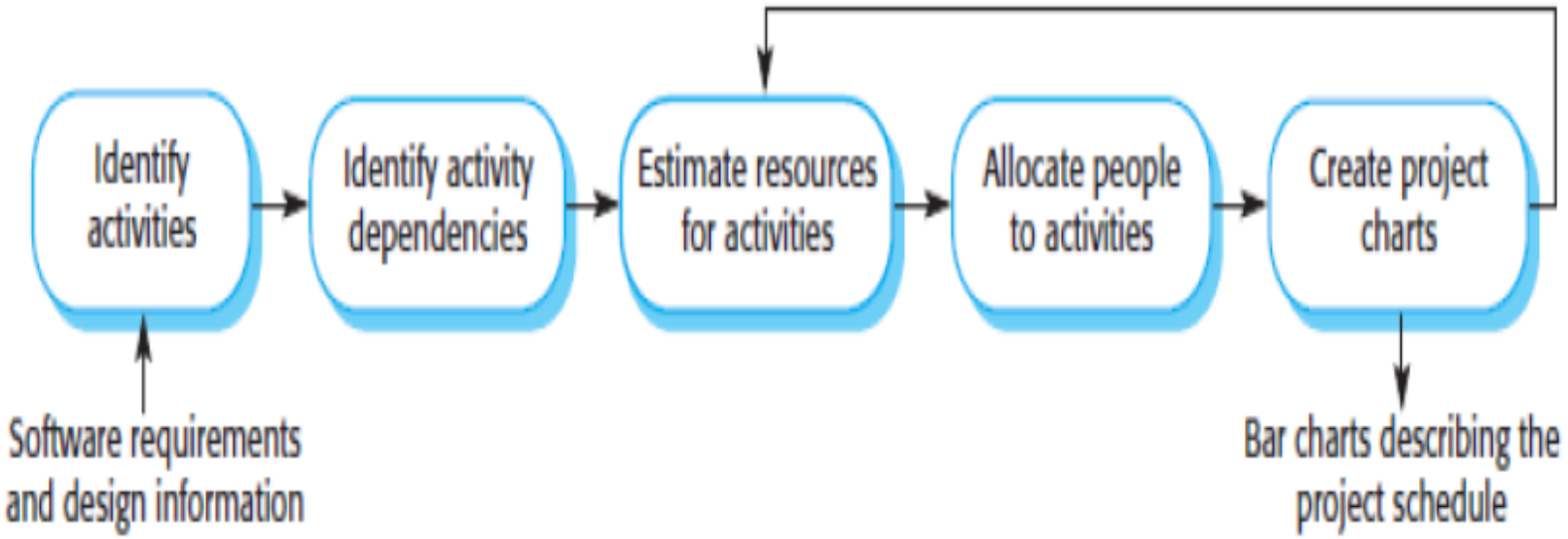
- **Change Management:**
  - Develop a process for handling change requests and modifications to project scope.
  - Ensure that changes are properly documented and approved.

- **Monitoring and Control:**
  - Implement monitoring and control mechanisms to track project progress.
  - Compare actual progress with the project plan and take corrective actions as needed.
- **Documentation:**
  - Maintain detailed project documentation, including requirements, plans, and progress reports.

Project Scheduling

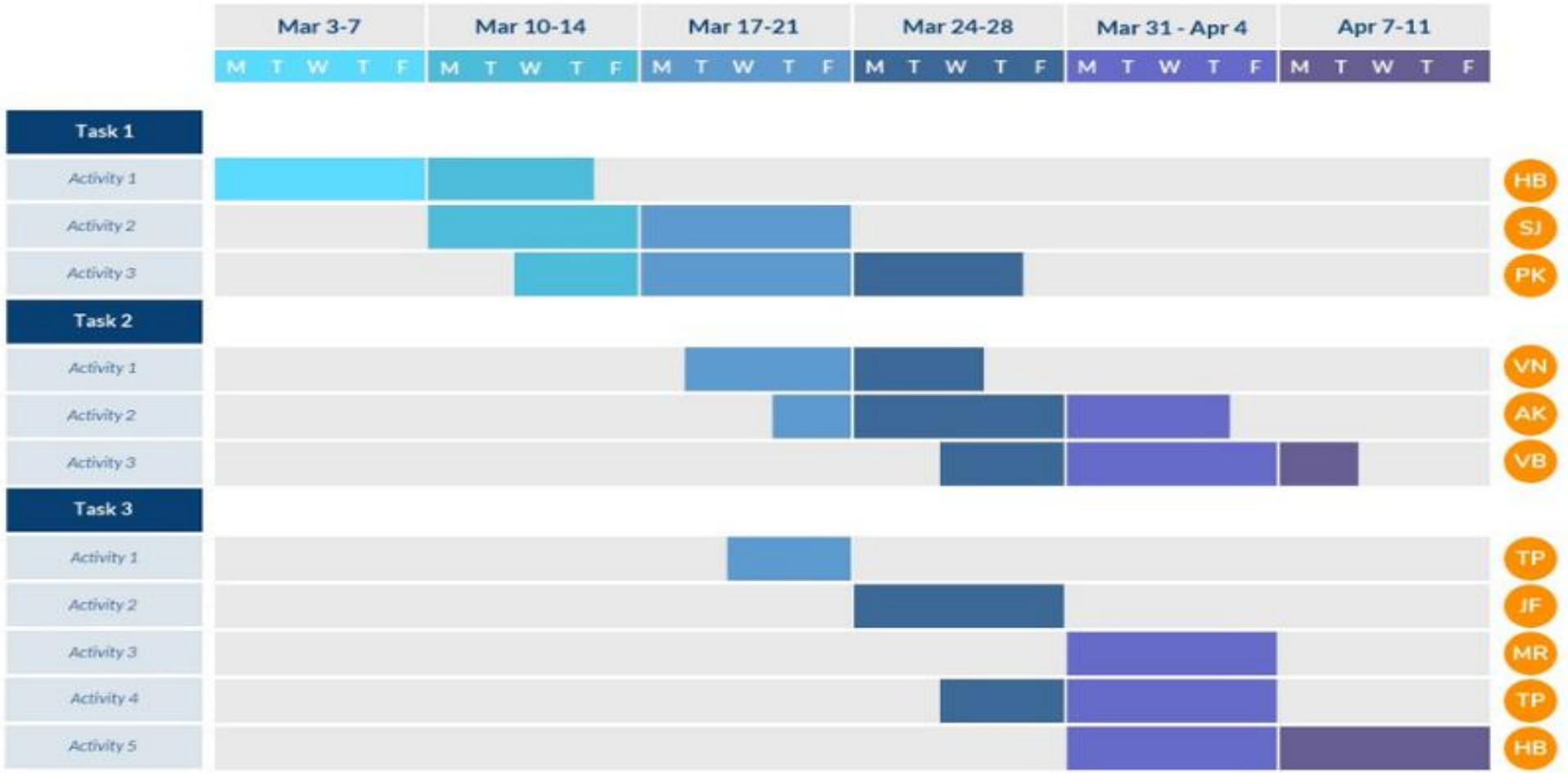
- Scheduling in plan-driven projects involves breaking down the total work involved in a project into separate tasks and estimating the time required to complete each task.

- Project schedules may simply be documented in a table or spreadsheet showing the tasks, estimated effort, duration, and task dependencies.


Identify activities
Identify activity dependencies
Estimate resources for activities
Allocate people to activities
Create project charts
Software requirements and design information
Bar charts describing the project schedule

Gantt chart

- This is the most common form of project schedule. It's a horizontal bar chart that tracks activities over time and allows you to communicate the project timeline visually.


Mar 3-7
Mar 10-14
Mar 17-21
Mar 24-28
Mar 31 - Apr 4
Apr 7-11
M T W T F M T W T F M T W T F M T W T F M T W T F M T W T F
Task 1
Activity 1
Activity 2
Activity 3
Task 2
Activity 1
Activity 2
Activity 3
Task 3
Activity 1
Activity 2
Activity 3
Activity 4
Activity 5
HB
SJ
PK
VN
AK
VB
TP
JF
MR
TP
HB

Agile planning

- Agile methods of software development are iterative approaches where the software is developed and delivered to customers in increments.
- Unlike plan-driven approaches, the functionality of these increments is not planned in advance but is decided during the development.

Stages:

- 1. Release planning: The team decides what features or functionalities will be in the next release of the system. The customer has to be involved in this process.
- 2. Iteration planning: which has a shorter term outlook and focuses on planning the next increment of a system. This usually represents 2 to 4 weeks of work for the team.

Approaches

1. Scrum is one of the most popular Agile.

It includes

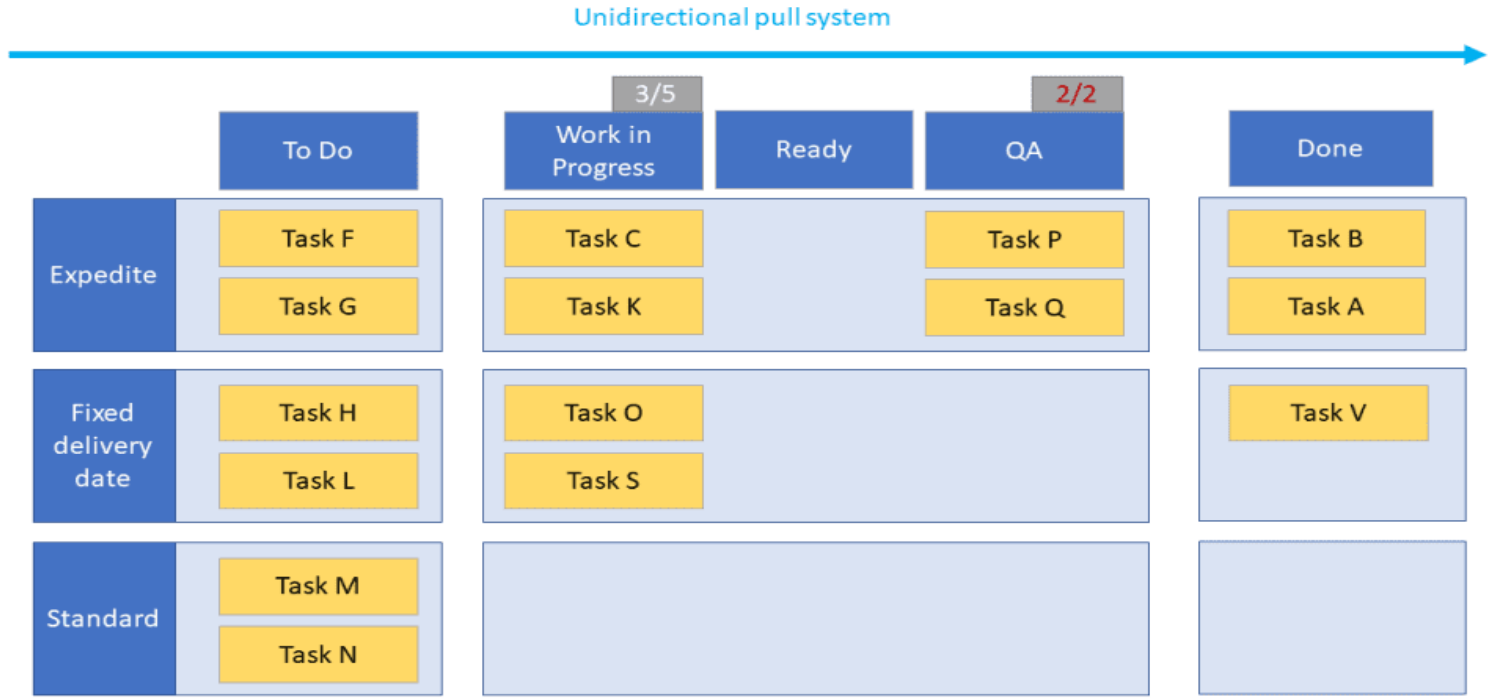
- Sprint Planning: At the start of each sprint (typically 2-4 weeks), the Scrum team selects features from the product backlog and commits to delivering them by the end of the sprint.

- Daily Standup Meetings: These short daily meetings help the team plan their day, identify any roadblocks, and ensure they are on track to meet their sprint goals

2. Kanban is a visual management method that focuses on the flow of work. While it doesn't have fixed iterations like Scrum.

- ***Work-in-Progress (WIP) :*** Kanban provide the number of work items in progress at any given time, which helps teams plan their work based on capacity and flow.

- ***Continuous Improvement:*** Teams continually analyze their workflow and make adjustments to optimize it.


Unidirectional pull system
3/5
2/2
To Do
Work in Progress
Ready
QA
Done
Expedite
Task F
Task G
Task C
Task K
Task P
Task Q
Task B
Task A
Fixed delivery date
Task H
Task L
Task O
Task S
Task V
Standard
Task M
Task N

3. The Lean approach, often associated with Lean Manufacturing, is a methodology focused on maximizing value while minimizing waste.

- Originally developed in the manufacturing industry, Lean principles have been successfully applied to various fields, including software development, project management, and business operations.

- The core concept of Lean is to deliver more value to the customer with fewer resources, less waste, and increased efficiency.

7 principles of Lean approach

- **1.Eliminate waste:** waste may include, partially done work, Extra features, Task switching, Waiting, Defects etc. In order to eliminate waste, one should be able to recognize it.

- 2. **Amplify learning:** Software design is a problem-solving process involving the developers writing the code and learned from it. Software value is measured in fitness for use and not in conformance to requirements.

- 3. **Decide as late as possible:** In software development delaying decisions as much as possible until they can be made based on facts and not on uncertain assumptions and predictions.

- 4. **Deliver as fast as possible**: In the era of rapid technology evolution, it is not the biggest that survives, but the fastest. The sooner the end product is delivered without major defects, the sooner feedback can be received, and incorporated into the next iteration.

- 5. **Empower the team**: Build projects around motivated individuals and trust them to get the job done encouraging progress, catching errors, and removing impediments.

- 6. **Build integrity in:** The customer needs to have an overall experience of the system. This is the so-called perceived integrity: how it is being advertised, delivered, deployed, accessed, how intuitive its use is, its price and how well it solves problems.

- 7. **Optimize the whole:** caring about the flow of value through the entire process from beginning to end

Estimation Techniques

Two types of techniques can be used for making estimates:

- 1. Experience-based techniques: The estimate of future effort requirements is based on the manager's experience of past projects and the application domain.

- 2. Algorithmic cost modeling: In this approach, a formulaic approach is used to compute the project effort based on estimates of product attributes, such as size, process characteristics, and experience of staff involved.

Eg:COCOMO Model

Experience based techniques

- • Experience-based techniques rely on the manager’s experience of past projects and the actual effort expended in these projects on activities that are related to software development.
- • Typically, you identify the deliverables to be produced in a project and the different software components or systems that are to be developed.
- • You document these in a spreadsheet, estimate them individually, and compute the total effort required.

- •The difficulty with experience-based techniques is that a new software project may not have much in common with previous projects.

- Software development changes very quickly, and a project will often use unfamiliar techniques such as web services, application system configuration, etc

- If you have not worked with these techniques, your previous experience may not help you to estimate the effort required, making it more difficult to produce accurate costs and schedule estimates

Algorithmic cost modelling:COCOMO Model

- Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e **number of Lines of Code**.
- The key parameters which define the quality of any software products, which are also an outcome of the COCOMO(constructive cost model) are primarily Effort & Schedule

- **Effort:** Amount of labor that will be required to complete a task. It is measured in person-months units.
- **Schedule:** Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put in. It is measured in the units of time such as weeks, and months.

$$E = a(KLOC)^b$$

$$time = c(Effort)^d$$

$$Personrequired = Effort/time$$

Software Projects	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Example

- Consider a software project using semi-detached mode with 300 Kloc .find out effort estimation, development time, and person estimation.

- Ans: **Effort (E) =** a*(KLOC)b = 3.0*(300)1.12 = 1784.42 PM
  Development Time (D) = c*(E)d* = *2.5*(1784.42)0.35 = 34.35 Months(M)
  Person Required (P) = E/D = 1784.42/34.35 = 51.9481 Persons ~52 Persons

Configuration Management

- • Configuration management (CM) is concerned with the policies, processes, and tools for managing ,changing software systems.

The configuration management of a software system product involves four closely related activities:

- 1. Version control: This involves keeping track of the multiple versions of system components and ensuring that changes made to components by different developers do not interfere with each other.

- 2. System building: This is the process of assembling program components, data, and libraries, then compiling and linking these to create an executable system.

- 3. Change management: This involves keeping track of requests for changes to delivered software from customers and developers, working out the costs and impact of making these changes, and deciding if and when the changes should be implemented.

- 4. Release management: This involves preparing software for external release and keeping track of the system versions that have been released for customer use.

Version Management

- Version management is the process of keeping track of different versions of software components and the systems in which these components are used.

- It also involves ensuring that changes made by different developers to these versions do not interfere with each other.

Version Control Systems

Version control (VC) systems identify, store, and control access to the different versions of components.

There are two types of modern version control system:

- 1. Centralized systems: Centralized version control systems contain just one repository globally and every user need to commit for reflecting one's changes in the repository. It is possible for others to see your changes by updating.

- 2. Distributed system: Distributed version control systems contain multiple repositories. Each user has their own repository and working copy. Just committing your changes will not give others access to your changes. This is because commit will reflect those changes in your local repository and you need to push them in order to make them visible on the central repository.

Key features of Version Control Systems

- 1. Version and release identification: Managed versions of a component are assigned unique identifiers when they are submitted to the system. These identifiers allow different versions of the same component to be managed, without changing the component name.

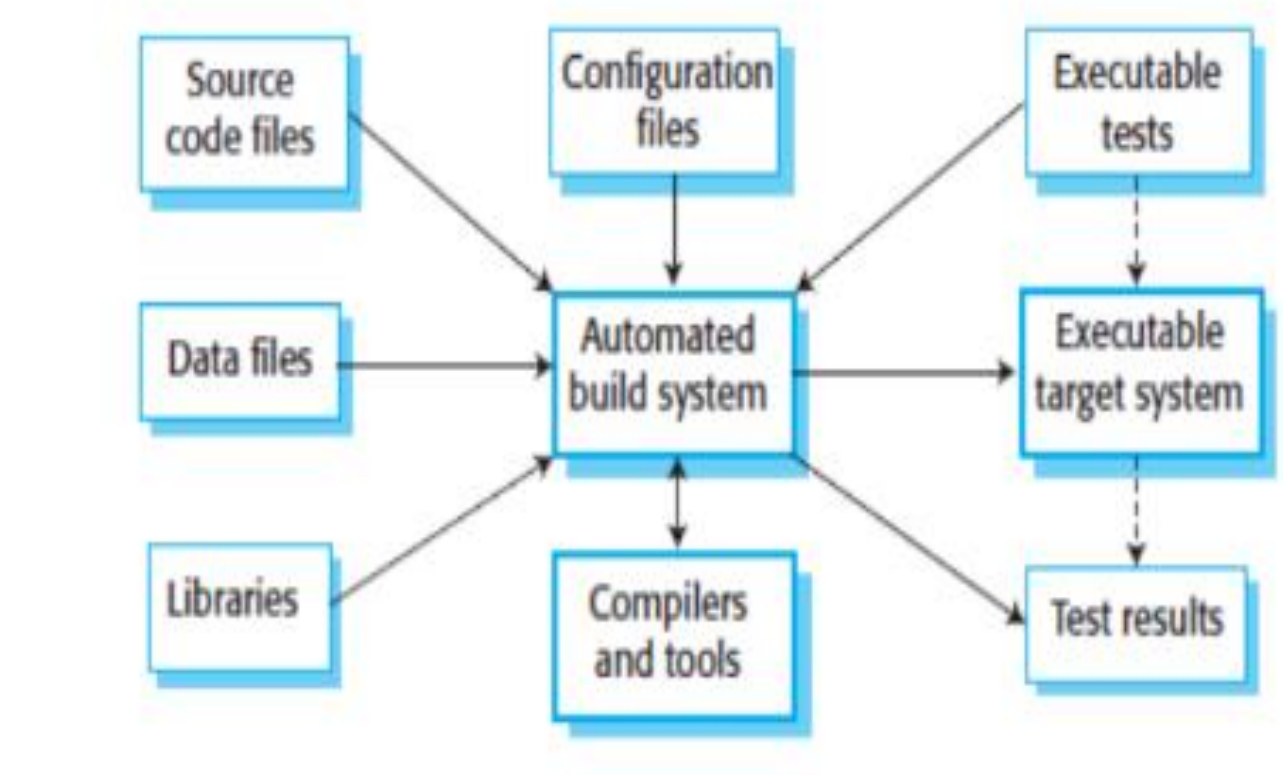
- 2. Change history recording: The VC system keeps records of the changes that have been made to create a new version of a component from an earlier version.

- 3. Independent development: Different developers may be working on the same component at the same time. The version control system keeps track of components that have been checked out for editing and ensures that changes made to a component by different developers do not interfere.
- 4. Project support: A version control system may support the development of several projects, which share components.
- 5. Storage management: Rather than maintain separate copies of all versions of a component, the version control system may use efficient mechanisms to ensure that duplicate copies of identical files are not maintained.

System Building

- System building is the process of creating a complete, executable system by compiling and linking the system components, external libraries, configuration files, and other information.

- System building involves assembling a large amount of information about the s/w and its operating environment.


Source code files
Configuration files
Executable tests
Data files
Automated build system
Executable target system
Libraries
Compilers and tools
Test results

Change Management

- Organizational needs and requirements change during the lifetime of a system, bugs have to be repaired, and systems have to adapt to changes in their environment.

- • Change management is intended to ensure that the evolution of the system is controlled and that the most urgent and cost-effective changes are prioritized.

- • Change management is the process of analyzing the costs and benefits of proposed changes, approving those changes that are cost-effective, and tracking which components in the system have been changed

Factors in change analysis:

- • The consequences of not making the change: What are the consequences if we don't make this change, especially if it's related to a known system issue?" This helps prioritize changes based on their impact and urgency.

- • The benefits of the change: Will the change benefit many users of the system, or will it only benefit the change proposer?

- • The number of users affected by the change: If only a few users are affected, then the change may be assigned a low priority.

- • The costs of making the change: If making the change affects many system components and/or takes a lot of time to implement, then the change may be rejected

Release Management

- • A system release is a version of a software system that is distributed to customers

Two types of release:

- major releases: which deliver significant new functionality.
- minor releases: which repair bugs and fix customer problems that have been reported.

Release components

A software product release is not just the executable code of the system. The release may also include:

- configuration files defining how the release should be configured for particular installations
- data files, such as files of error messages in different languages, that are needed successful system operation
- an installation program that is used to help install the system on target hardware
- electronic and paper documentation describing the system

Release creation

Process of creating the collection of files and documentation that include all components of the system release. This process involves several steps:

1. The executable code of the programs and all associated data files.

2. Configuration descriptions may have to be written for different hardware and operating systems.

3. Updated instructions may have to be written for customers who need to configure their own systems.

4. Scripts for the installation program may have to be written.

5. Web pages have to be created describing the release, with links to system documentation.

6. Finally, when all information is available, an executable master image of the software must be prepared and handed over for distribution to customers or sales outlets.