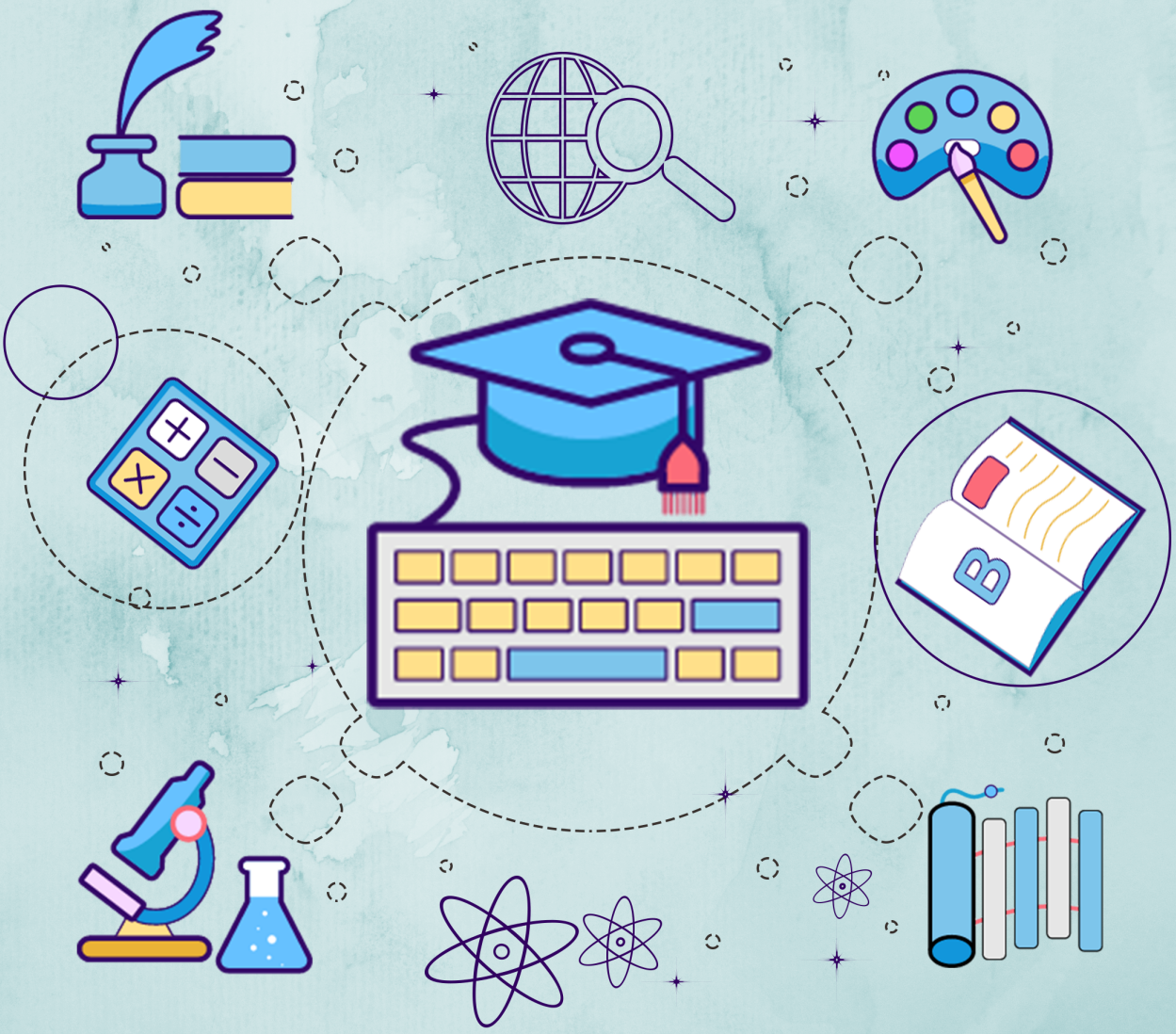
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

KeralaNotes



SYLLABUS | STUDY MATERIALS | TEXTBOOK

PDF | SOLVED QUESTION PAPERS

www.keralanotes.com

KeralaNotes

KTU STUDY MATERIALS

MANAGEMENT OF SOFTWARE SYSTEMS

CST 309

Module 4

Related Link :

- KTU S5 STUDY MATERIALS
- KTU S5 NOTES
- KTU S5 SYLLABUS
- KTU S5 TEXTBOOK PDF
- KTU S5 PREVIOUS YEAR SOLVED QUESTION PAPER

- Software project management is an essential part of software engir
- The success criteria **for project management** obviously vary from p to project, but, for most projects, **important goals are:**
  1. to deliver the software to the customer at the agreed time;
  2. to keep overall costs within budget
  3. to deliver software that meets the customer's expectations;
  4. to maintain a coherent and well-functioning development team.

KeralaNotes

- Some of the most **important factors that affect how software projects are managed are:**
  1. Company size
  2. Software customers
  3. Software size
  4. Software type
  5. Organizational culture
  6. Software development processes

For More Study Materials : https://www.keralanotes.com/

- The **fundamental project management activities** that are common organizations:
  1. **Project planning** → Project managers are responsible for planning, estima scheduling project development and assigning people to tasks.
  2. **Risk management** → Project managers have to assess the risks that may a project, monitor these risks, and take action when problems arise.
  3. **People management** → Project managers are responsible for managing a people. They have to choose people for their team and establish ways of v that lead to effective team performance.
  4. **Reporting** → Project managers are usually responsible for reporting on th of a project to customers and to the managers of the company developing software.
  5. **Proposal writing** → The first stage in a software project may involve writi proposal to win a contract to carry out an item of work. The proposal desc objectives of the project and how it will be carried out. It usually includes schedule estimates and justifies why the project contract should be award particular organization or team.

RISK MANAGEMENT

- Risk management is one of the most important jobs for a proje manager.
- Risk management involves anticipating risks that might affect th project schedule or the quality of the software being developed then taking action to avoid these risks.
- Risks can be categorized according to type of risk (technical, organizational, etc.)

- **Classification of risks according to what these risks affect:**
  1. **Project risks** → affect the project schedule or resources. An examp project risk is the loss of an experienced system architect.
  2. **Product risks** → affect the quality or performance of the software l developed. An example of a product risk is the failure of a purchase component to perform as expected.
  3. **Business risks** → affect the organization developing or procuring th software. For example, a competitor introducing a new product is a business risk.
- For large projects, you should record the results of the risk anal a risk register along with a consequence analysis. This sets out consequences of the risk for the project, product, and business

Risk	Affects	Description
Staff turnover	Project	Experienced staff will leave the project before it is finished.
Management change	Project	There will be a change of company management with different priorities.
Hardware unavailability	Project	Hardware that is essential for the project will not be delivered on schedule.
Requirements change	Project and product	There will be a larger number of changes to the requirements than anticipated.
Specification delays	Project and product	Specifications of essential interfaces are not available on schedule.
Size underestimate	Project and product	The size of the system has been underestimated.
Software tool underperformance	Product	Software tools that support the project do not perform as anticipated.
Technology change	Business	The underlying technology on which the system is built is superseded by new technology.
Product competition	Business	A competitive product is marketed before the system is completed.

Fig: Examples of common project, product, and business risks

- Effective risk management makes it easier to cope with proble to ensure that these do not lead to unacceptable budget or s slippage.
- For small projects, formal risk recording may not be required, project manager should be aware of them.
- The specific risks that may affect a project depend on the proje the organizational environment in which the software is being developed.
- Software risk management is important because of the inherer uncertainties in software development.

KeralaNotes

- An outline of the process of risk management is presented Figure. It involves several **stages:**
  1. **Risk identification** → You should identify possible project, pro and business risks.
  2. **Risk analysis** → You should assess the likelihood and consequ these risks.
  3. **Risk planning** → You should make plans to address the risk, ei avoiding it or by minimizing its effects on the project.
  4. **Risk monitoring** → You should regularly assess the risk and yc plans for risk mitigation and revise these plans when you learı about the risk.
- The risk management process is an iterative process that continues throughout a project.

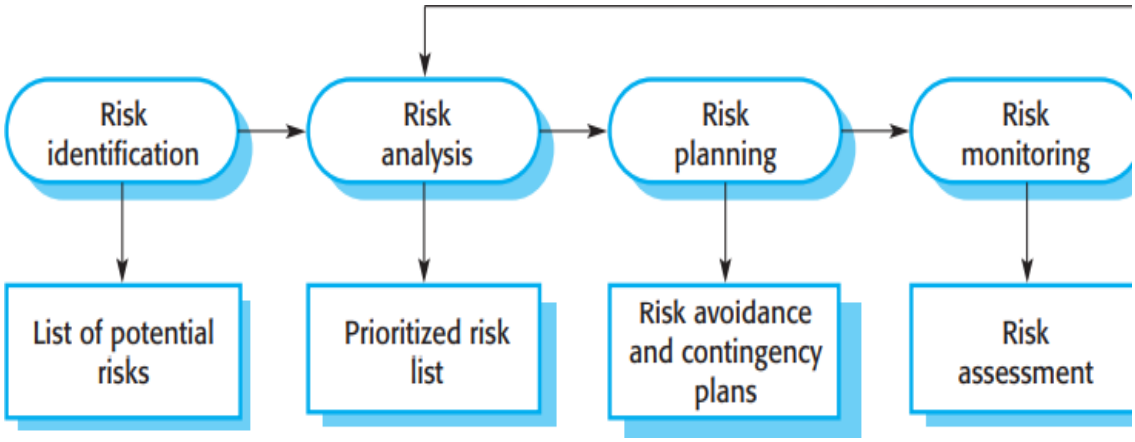


Fig: The Risk Management Process

KeralaNotes

Risk Identification:

- Risk identification is the first stage of the risk management proc
- It is concerned with identifying the risks that could pose a majo threat to the software engineering process, the software being developed, or the development organization.
- Risk identification may be a team process in which a team gets together to brainstorm possible risks.
- As a starting point for risk identification, a checklist of different risk may be used.

For More Study Materials : https://www.keralanotes.com/

- **6 types of risk** may be included in a risk checklist:
  1. **Estimation risks** → arise from the management estimates of the re required to build the system.
  2. **Organizational risks** → arise from the organizational environment v the software is being developed.
  3. **People risks** → are associated with the people in the development
  4. **Requirements risks** → come from changes to the customer require and the process of managing the requirements change.
  5. **Technology risks** → come from the software or hardware technolo are used to develop the system.
  6. **Tools risks** → come from the software tools and other support soft used to develop the system.

Risk type	Possible risks
Estimation	1. The time required to develop the software is underestimated. 2. The rate of defect repair is underestimated. 3. The size of the software is underestimated.
Organizational	4. The organization is restructured so that different management are responsible for the project. 5. Organizational financial problems force reductions in the project budget.
People	6. It is impossible to recruit staff with the skills required. 7. Key staff are ill and unavailable at critical times. 8. Required training for staff is not available.
Requirements	9. Changes to requirements that require major design rework are proposed. 10. Customers fail to understand the impact of requirements changes.
Technology	11. The database used in the system cannot process as many transactions per second as expected. 12. Faults in reusable software components have to be repaired before these components are reused.
Tools	13. The code generated by software code generation tools is inefficient. 14. Software tools cannot work together in an integrated way.

Figure 22.3 Examples of different types of risk

Risk Analysis:

- During the risk analysis process, you have to consider each identified risk and ma judgment about the probability and seriousness of that risk.
- It is not possible to make precise, numeric assessment of the probability and ser of each risk.
- You should assign the risk to one of a number of bands:
  1. The probability of the risk might be assessed as insignificant, low, moderate, high, or v
  2. The effects of the risk might be assessed as catastrophic (threaten the survival of the serious (would cause major delays), tolerable (delays are within allowed contingency) insignificant.
- You may then tabulate the results of this analysis process using a table ordered a to the seriousness of the risk.

Risk	Probability	Effects
Organizational financial problems force reductions in the project budget (5).	Low	Catastrophic
It is impossible to recruit staff with the skills required (6).	High	Catastrophic
Key staff are ill at critical times in the project (7).	Moderate	Serious
Faults in reusable software components have to be repaired before these components are reused (12).	Moderate	Serious
Changes to requirements that require major design rework are proposed (9).	Moderate	Serious
The organization is restructured so that different managements are responsible for the project (4).	High	Serious
The database used in the system cannot process as many transactions per second as expected (11).	Moderate	Serious
The time required to develop the software is underestimated (1).	High	Serious
Software tools cannot be integrated (14).	High	Tolerable
Customers fail to understand the impact of requirements changes (10).	Moderate	Tolerable
Required training for staff is not available (8).	Moderate	Tolerable
The rate of defect repair is underestimated (2).	Moderate	Tolerable
The size of the software is underestimated (3).	High	Tolerable
Code generated by code generation tools is inefficient (13).	Moderate	Insignificant

Figure 22.4 Risk types and examples

- Both the probability and the assessment of the effects of a risk change as more information about the risk becomes available a risk management plans are implemented. You should therefore update this table during each iteration of the risk management process.
- Once the risks have been analyzed and ranked, you should asse which of these risks are most significant.
- In general, catastrophic risks should always be considered, as sł all serious risks that have more than a moderate probability of occurrence.

KeralaNotes

Risk Planning:

- The risk planning process develops strategies to manage the ke that threaten the project.
- For each risk, you have to think of actions that you might take t minimize the disruption to the project if the problem identified risk occurs.
- You should also think about the information that you need to c while monitoring the project so that emerging problems can be detected before they become serious.

For More Study Materials : https://www.keralanotes.com/

- In risk planning, you have to ask “what-if” questions that consi both individual risks, combinations of risks, and external factor affect these risks. For example, questions that you might ask ar
  1. What if several engineers are ill at the same time?
  2. What if an economic downturn leads to budget cuts of 20% for the
  3. What if the performance of open-source software is inadequate an only expert on that open-source software leaves?
  4. What if the company that supplies and maintains software compon goes out of business?
  5. What if the customer fails to deliver the revised requirements as pr
- Based on the answers to these “what-if” questions, you may d strategies for managing the risks.

KeralaNotes

- The **possible risk management strategies** fall into 3 categories:
  1. **Avoidance strategies** → Following these strategies means that the probability that the risk will arise is reduced. An example of a risk a strategy is the strategy for dealing with defective components.
  2. **Minimization strategies** → Following these strategies means that t impact of the risk is reduced. An example of a risk minimization str the strategy for staff illness.
  3. **Contingency plans** → Following these strategies means that you ar prepared for the worst and have a strategy in place to deal with it. A example of a contingency strategy is the strategy for organizational problems.
- The strategies used in critical systems ensure reliability, security safety, where you must avoid, tolerate, or recover from failures.

For More Study Materials : https://www.keralanotes.com/

- It is best to use a strategy that avoids the risk.
- If this is not possible, you should use a strategy that reduces th chances that the risk will have serious effects.
- Finally, you should have strategies in place to cope with the risk arises. These should reduce the overall impact of a risk on the p or product.

Risk	Strategy
Organizational financial problems	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business and presenting reasons why cuts to the project budget would not be cost-effective.
Recruitment problems	Alert customer to potential difficulties and the possibility of delays; investigate buying-in components.
Staff illness	Reorganize team so that there is more overlap of work and people therefore understand each other's jobs.
Defective components	Replace potentially defective components with bought-in components of known reliability.
Requirements changes	Derive traceability information to assess requirements change impact; maximize information hiding in the design.
Organizational restructuring	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Database performance	Investigate the possibility of buying a higher-performance database.
Underestimated development time	Investigate buying-in components; investigate use of automated code generation.

Figure 22.5 Strategies to help manage risk

Risk Monitoring:

- Risk monitoring is the process of checking that your assumptio about the product, process, and business risks have not change
- You should regularly assess each of the identified risks to decid whether or not that risk is becoming more or less probable.
- You should also think about whether or not the effects of the ri have changed.
- To do this, you have to look at other factors, such as the numbe requirements change requests, which give you clues about the probability and its effects. These factors are dependent on the of risk.

Risk type	Potential indicators
Estimation	Failure to meet agreed schedule; failure to clear report defects.
Organizational	Organizational gossip; lack of action by senior manage
People	Poor staff morale; poor relationships among team mer high staff turnover.
Requirements	Many requirements change requests; customer compl
Technology	Late delivery of hardware or support software; many reported technology problems.
Tools	Reluctance by team members to use tools; complaints software tools; requests for faster computers/more me and so on.

Figure 22.6 Risk indicators

- You should monitor risks regularly at all stages in a project.
- At every management review, you should consider and discuss of the key risks separately.
- You should decide if the risk is more or less likely to arise and if seriousness and consequences of the risk have changed.

KeralaNotes

MANAGING PEOPLE

- The people working in a software organization are its greatest assets.
- It is expensive to recruit and retain good people.
- Software managers have to ensure that the engineers working on a proje productive as possible.
- It is important that software project managers understand the technical i that influence the work of software development.
- Software engineers often have strong technical skills but may lack the sof that enable them to motivate and lead a project development team.
- As a project manager, you should be aware of the potential problems of p management and should try to develop people management skills.

- **4 critical factors that influence the relationship between a manager and th** that he or she manages:
    1. **Consistency** → All the people in a project team should be treated in a comparable way. No one expects all rewards to be identical, but peop not feel that their contribution to the organization is undervalued.
    2. **Respect** → Different people have different skills, and managers shoul these differences.
    3. **Inclusion** → People contribute effectively when they feel that others them and take account of their proposals. It is important to develop a environment where all views, even those of the least experienced sta considered.
    4. **Honesty** → As a manager, you should always be honest about what is well and what is going badly in the team. You should also be honest a your level of technical knowledge and be willing to defer to staff with knowledge when necessary.

Motivating People:

- As a project manager, you need to motivate the people who wo with you so that they will contribute to the best of their abilitie
- In practice, **motivation** means organizing work and its environn encourage people to work as effectively as possible.
- To provide this encouragement, you should understand a little a what motivates people.
- People are motivated by satisfying their needs. These needs are arranged in a series of levels, as shown in Figure.

Figure 22.7 Human needs hierarchy

KeralaNotes

- The lower levels of this hierarchy represent fundamental needs for f sleep, and so on, and the need to feel secure in an environment.
- **Social need** is concerned with the need to feel part of a social grou
- **Esteem need** represents the need to feel respected by others, and s **realization need** is concerned with personal development.
- People need to satisfy lower-level needs such as hunger before the abstract, higher-level needs.
- People working in software development organizations are not usu hungry, thirsty, or physically threatened by their environment. There making sure that peoples' social, esteem, and self-realization needs satisfied is most important from a management point of view.

1. **To satisfy social needs**, you need to give people time to meet the workers and provide places for them to meet. This is relatively ea all of the members of a development team work in the same pla Social networking systems and teleconferencing can be used for r communications.
2. **To satisfy esteem needs**, you need to show people that they are by the organization. Public recognition of achievements is a sim effective way of doing this.
3. Finally, **to satisfy self-realization needs**, you need to give people responsibility for their work, assign them demanding (but not impossible) tasks, and provide opportunities for training and development where people can enhance their skills. Training is ar important motivating influence as people like to gain new knowle and learn new skills.

KeralaNotes

- Maslow's model of motivation takes an exclusively personal vie on motivation.
- It does not take adequate account of the fact that people feel themselves to be part of an organization, a professional group, one or more cultures.
- Being a member of a cohesive group is highly motivating for m people.
- Therefore, as a manager, you also have to think about how a gr a whole can be motivated.

Case study: Motivation

Alice is a software project manager working in a company that develops alarm systems. This company wishes to enter the growing market of assistive technology to help elderly and disabled people live independently. Alice has been asked to lead a team of six developers that can develop new products based on the company's alarm technology.

Alice's assistive technology project starts well. Good working relationships develop within the team, and creative new ideas are developed. The team decides to develop a system that a user can initiate and control the alarm system from a cell phone or tablet computer. However, some months into the project, Alice notices that Dorothy, a hardware expert, starts coming into work late, that the quality of her work is deteriorating, and, increasingly, that she does not appear to be communicating with other members of the team.

Alice talks about the problem informally with other team members to try to find out if Dorothy's personal circumstances have changed and if this might be affecting her work. They don't know of anything, so Alice decides to talk with Dorothy to try to understand the problem.

After some initial denials of any problem, Dorothy admits that she has lost interest in the job. She expected that she would be able to develop and use her hardware interfacing skills. However, because of the product direction that has been chosen, she has little opportunity to use these skills. Basically, she is working as a C programmer on the alarm system software.

While she admits that the work is challenging, she is concerned that she is not developing her interfacing skills. She is worried that finding a job that involves hardware interfacing will be difficult after this project. Because she does not want to upset the team by revealing that she is thinking about the next project, she has decided that it is best to minimize conversation with them.

Figure 22.8 Individual motivation

- Psychological personality type also influences motivation.
- Bass and Dunteman (Bass and Dunteman 1963) identified **classifications for professional workers:**
  1. **Task-oriented people** → who are motivated by the work they software engineering, these are people who are motivated by intellectual challenge of software development.
  2. **Self-oriented people** → who are principally motivated by per success and recognition. They are interested in software deve as a means of achieving their own goals. They often have long goals and they wish to be successful in their work to help rea these goals.
  3. **Interaction-oriented people** → who are motivated by the pre and actions of co-workers.

- Research has shown that interaction-oriented personalities usu like to work as part of a group, whereas task-oriented and self- oriented people usually prefer to act as individuals.
- **People Capability Maturity Model (P-CMM)** → is a framewor assessing how well organizations manage the development of staff. It highlights best practice in people management and pro basis for organizations to improve their people management processes. It is best suited to large rather than small, informal companies.

KeralaNotes

TEAMWORK

- As it is impossible for everyone in a large group to work togeth single problem, large teams are usually split into a number of groups.
- Each group is responsible for developing part of the overall syst
- The best size for a software engineering group is 4 to 6 memb they should never have more than 12 members.
- When groups are small, communication problems are reduced.

- Putting together a group that has the right balance of technica experience, and personalities is a critical management task.
- A good group is cohesive and thinks of itself as a strong, single
- The people involved are motivated by the success of the group as by their own personal goals.
- In a **cohesive group**, members think of the group as more impo than the individuals who are group members.
  - They are loyal to the group.
  - They identify with group goals and other group members.
  - They attempt to protect the group, as an entity, from outside interfer This makes the group robust and able to cope with problems and une situations.

KeralaNotes

- The **benefits of creating a cohesive group** are:
  1. The group can establish its own quality standards.
  2. Individuals learn from and support each other.
  3. Knowledge is shared.
  4. Refactoring and continual improvement is encouraged.
- Good project managers should always try to encourage group cohesiveness.
- They may try to establish a sense of group identity by naming the g and establishing a group identity and territory.
- One of the most effective ways of promoting cohesion is **to be inclu** i.e., you should treat group members as responsible and trustworth make information freely available.

- An effective way of making people feel valued and part of a grc to make sure that they know what is going on.

Case study: Team spirit

Alice, an experienced project manager, understands the importance of creating a sive group. As her company is developing a new product, she takes the opportuni involve all group members in the product specification and design by getting ther discuss possible technology with elderly members of their families. She encourag them to bring these family members to meet other members of the development

Alice also arranges monthly lunches for everyone in the group. These lunche opportunity for all team members to meet informally, talk around issues of conc and get to know each other. At the lunch, Alice tells the group what she knows organizational news, policies, strategies, and so forth. Each team member then b summarizes what they have been doing, and the group discusses a general topi as new product ideas from elderly relatives.

Every few months, Alice organizes an "away day" for the group where the tea spends two days on "technology updating." Each team member prepares an up a relevant technology and presents it to the group. This is an offsite meeting, an plenty of time is scheduled for discussion and social interaction.

Figure 22.9 Group cohesion

- Given a stable organizational and project environment, the **3 fa that have the biggest effect on team working are:**
  1. The people in the group (**Selecting group members**)
  2. The way the group is organized (**Group organizations**)
  3. Technical and managerial communications (**Group communication**

Selecting Group Members:

- A manager or team leader's job is to create a cohesive group and or that group so that they work together effectively.
- This task involves selecting a group with the right balance of techni and personalities.
- Technical knowledge and ability should not be the only factor used select group members.
- People who are motivated by the work are likely to be the stronges technically.
- People who are self-oriented will probably be best at pushing the w forward to finish the job.
- People who are interaction-oriented help facilitate communications the group.

- The project manager has to control the group so that individua do not take precedence over organizational and group objectiv
- This control is easier to achieve if all group members participat each stage of the project.
- Individual initiative is most likely to develop when group memb given instructions without being aware of the part that their ta plays in the overall project.
- If all the members of the group are involved in the design from start, they are more likely to understand why design decisions h been made. They may then identify with these decisions rathe oppose them.

Case study: Group composition

In creating a group for assistive technology development, Alice is aware of
tance of selecting members with complementary personalities. When inte
potential group members, she tried to assess whether they were task-orie
oriented, or interaction-oriented. She felt that she was primarily a self-orie
because she considered the project to be a way of getting noticed by seni
ment and possibly being promoted. She therefore looked for one or perha
action-oriented personalities, with task-oriented individuals to complete th
final assessment that she arrived at was:

Alice—self-oriented
Brian—task-oriented
Chun—interaction-oriented
Dorothy—self-oriented
Ed—interaction-oriented
Fiona—task-oriented
Fred—task-oriented
Hassan—interaction-oriented

Figure 22.10 Group composition

Group Organization:

- The way a group is organized affects the group's decisions, the v information is exchanged, and the interactions between the development group and external project stakeholders.
- Project managers are often responsible for selecting the people organization who will join their software engineering team.
- Getting the best possible people in this process is very importa poor selection decisions may be a serious risk to the project.
- Key factors that should influence the selection of staff are educ and training, application domain and technology experience, communication ability, adaptability, and problem solving ability

- **Important organizational questions for project managers** inclu following:
    1. Should the project manager be the technical leader of the group?
    2. Who will be involved in making critical technical decisions, and hov these decisions be made? Will decisions be made by the system ar the project manager or by reaching consensus among a wider rang team members?
    3. How will interactions with external stakeholders and senior compa management be handled?
    4. How can groups integrate people who are not co-located?
    5. How can knowledge be shared across the group?

Informal Groups	Hierarchical Groups
1. Small programming groups are usually organized. 2. Group leader gets involved in the software development with the other group members. 3. The group as a whole discusses the work to be carried out, and tasks are allocated according to ability and experience. 4. More senior group members may be responsible for the architectural design. 5. Detailed design and implementation is the responsibility of the team member who is allocated to a particular task. 6. Groups are very successful, particularly when most group members are experienced and competent. Such a group makes decisions which improves cohesiveness and performance. 7. With no experienced engineers to direct the work, the result can be a lack of coordination between group members and, possibly, eventual project failure.	1. Group leader is at the top of the hierarch 2. Group leader has more formal authori group members and so can direct their v 3. There is a clear organizational structure. 4. Decisions are made toward the top of th and implemented by people lower down 5. Communications are primarily instruc senior staff; the people at lower le hierarchy have relatively little commun the managers at the upper levels. 6. These groups can work well whe understood problem can be easily br into software components that can be d different parts of the hierarchy. 7. This grouping allows for rapid decision m

- In software development, effective team communications at all lev essential:
  1. Changes to the software often require changes to several parts of the sys this requires discussion and negotiation at all levels in the hierarchy.
  2. Software technologies change so fast that more junior staff may know mo new technologies than experienced staff. Top-down communications may that the project manager does not find out about the opportunities of us new technologies. More junior staff may become frustrated because of w see as old-fashioned technologies being used for development.
- A major challenge facing project managers is the difference in techn ability between group members.
- i.e., adopting a group model that is based on individual experts can significant risks.

KeralaNotes

Group Communications:

- It is absolutely essential that group members communicate effe and efficiently with each other and with other project stakehol
- Good communication also helps strengthen group cohesivenes
- Group members:
  1. Exchange information on the status of their work, the design decisio have been made, and changes to previous design decisions.
  2. Resolve problems that arise with other stakeholders and inform the stakeholders of changes to the system, the group, and delivery plan
  3. Come to understand the motivations, strengths, and weaknesses of people in the group.

- The effectiveness and efficiency of communications are influen

1. **Group size** → As a group gets bigger, it gets harder for members to communicate effectively. The number of one-way communication l n * (n − 1), where n is the group size.
2. **Group structure** → People in informally structured groups commu more effectively than people in groups with a formal, hierarchical s
3. **Group composition** → People with the same personality may clash a result, communications can be inhibited.
4. **The physical work environment** → The organization of the workpl major factor in facilitating or inhibiting communications.
5. **The available communication channels** → There are many differer of communication—face to face, email messages, formal documen telephone, and technologies such as social networking and wikis.

KeralaNotes

- Effective communication is achieved when communications are two- the people involved can discuss issues and information and establish common understanding of proposals and problems.
- All this can be done through meetings, although these meetings are dominated by powerful personalities.
- Informal discussions when a manager meets with the team for coffe sometimes more effective.
- Wikis and blogs allow project members and external stakeholders to exchange information, irrespective of their location. They help mana information and keep track of discussion threads, which often becor confusing when conducted by email.
- You can also use instant messaging and teleconferences, which can b arranged, to resolve issues that need discussion.

Software Project Planning

- The ***first step*** of software ***projects management process*** is the ***project planning*** step.
- Project planning, in turn, start with ***Estimation***.
- ***Estimation*** is both an ***'art'*** and ***'science'*** ***(There are techniques for time and hu estimation)***

We have got to estimate:

- ***resources***, ***cost*** and ***schedule*** ***required*** for a Project.
- Estimation requires ***experience*** access to info on ***previous projects***.
- **Estimation carries *a risk.***
- ***If we are familiar*** with the type of projects, we are going to handle, ***estimation become***

Estimation techniques

- Concentrate **two types of technique** that can be used to do this:
  - ***Experience-based techniques*****: is based on the manager's experience of past projects and the application domain.**
  - ***Algorithmic cost modeling*****: a formulaic approach is used to compute the project effort.**
- **The PROJECT EFFORT based on estimates of product size, and process characteristics such as experience of staff involved.**

Experience-based approaches

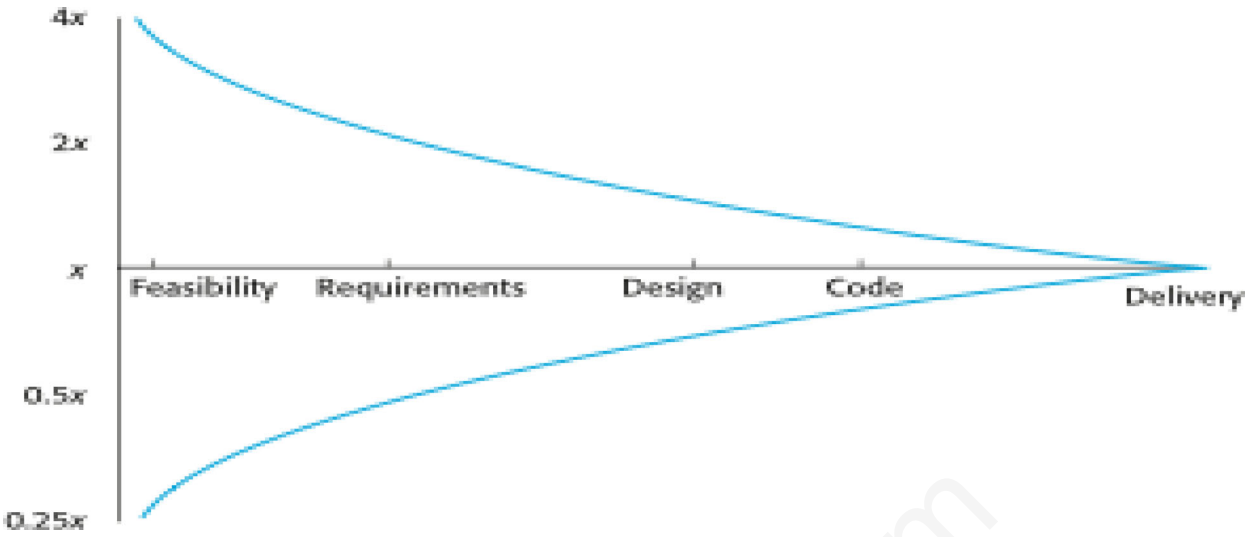
- Experience-based techniques rely on:
  - judgments based on experience of past projects
  - the effort used in these projects on software developm activities.
- Typically, you identify the deliverables to be produced i project and the different software components systems that are to be developed.
- You document these in a spreadsheet, estimate th individually and compute the total effort required.
- It usually helps to get a group of people involved in t effort estimation and to ask each member of the gro to explain their estimate.

KeralaNotes

Algorithmic cost modelling

- Cost is estimated as a mathematical function product, project and process attributes whc *values are estimated by project managers:*
  - $\mathbf{Effort = A \times Size^{B} \times M}$
  - A is an organisation-dependent constant,
  - B reflects the disproportionate(unbalanced) effort for large projects
  - M is a multiplier reflecting **product, process and people** attributes.
- The most commonly used ***product attribute for c estimation*** is ***code size***.
- Most models are ***similar*** but they ***use different values*** for A and M.

Estimate uncertainty



KeralaNotes

COCOMO MODEL

- **COCOMO (Constructive Cost Model)** (Boehm, 1981)
- COCOMO is a model based on inputs relating to the size of the system and a number of cost drivers that affect productivity.
- Updated version, called **COCOMO II**, accounts for recent changes in software engineering technology, including object-oriented software, software created via spiral or evolutionary development models, software reuse and building new systems using off-the-shelf software components.

COCOMO II

KeralaNotes

The COCOMO 2 model

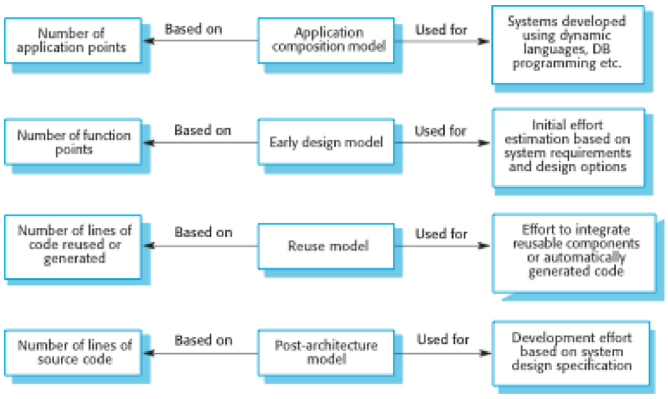
- Extendend version of (COCOMO-81) →COCOMO 2.
- COCOMO 2 recognizes different approaches to software deve such as prototyping, development by component composit use of database programming.
- COCOMO II supports a spiral model of development and several sub-models that produce increasingly detailed estimate
- These can be used in successive rounds of the development sp

COCOMO 2 models

• COCOMO 2 includes a range of sub-models. These sub-models produce increasingly d estimates. The sub-models in COCOMO 2 are:

1. **Application composition model**. This assumes that systems are created from reusable compor database programming. It is designed to make estimates of prototype development. Software based on **application points, and a simple size/productivity formula** is used to estimate the effor
2. **Early design model.** This model is used during early stages of the system design after the requir are **based on function points**, which **are then converted to number of lines of source code**.The **set of seven multipliers**.
3. **Reuse model.** This model is used to compute the effort required to integrate reusable componen code that is automatically generated by design or program translation tools. It is **usually** used ir **the post-architecture model.**
4. **Post-architecture model.** Once the system architecture has been designed, a more accurate software size can be made. However, it i**ncludes a more extensive set of 17 multipliers** re **capability and product and project characteristics.**

COCOMO estimation models



1. Application composition model

- Suitable for software built around graphical user interface (GUI) and modern GUI-bu
- Supports prototyping projects and for projects where the software is developed existing components.
- Uses **object points as a size metric extension** of function points
- **Count of the screens, reports, and 3GL Components (modules), weighted by factor (simple, medium, difficult).**

•Programmer productivity also depends on the developer's experience and capabili the capabilities of the CASE tools used to support development.

- Formula is

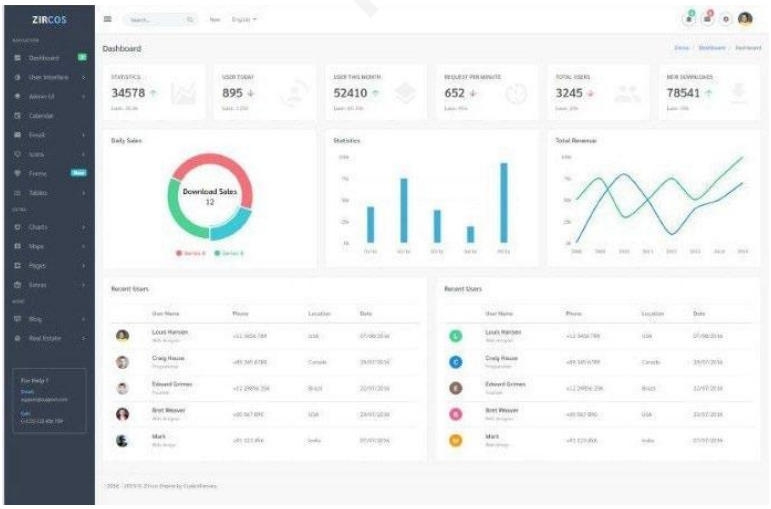
$$PM = (NAP * (1 - \%reuse/100)) / PROD$$

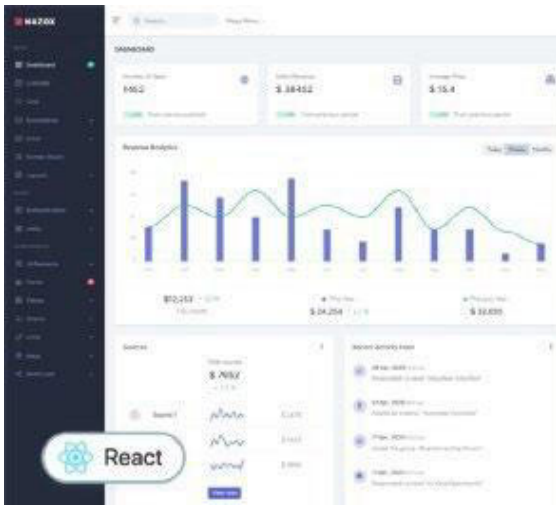
- **PM is the effort in person-months,**
- **NAP is the number of application points and**
- **PROD is the productivity.**

KeralaNotes

Screens and Views

Manager	Customer





For More Study Materials : https://www.keralanotes.com/

Example Problem

- Consider a database application project with
  1. The application has four screens with four views each and seven data tables for three servers and four clients.
  2. Application may generate two reports of six section each from seven data tables for two servers and three clients.
- 10% reuse of object points. Developer's experience and capability in similar environment is low. Calculate the object point count, New object point and effort to develop such project.

- **Given Data**
  - Screen = 4
  - Views = 4
  - Data tables = 7
  - Reports =2
  - Sections = 6
  - Reuse object Points = 10%
  - Developer's experience =
- **Calculate**
  - object point count
  - New object point (NOP)
  - Effort

Step 1

Solution:

- **Step-1: Access Object counts**
- Number of screens = 4
  Number of records = 2

Reference

- **Step-1: Access Object counts**
  - Estimate the number of sc
    reports and 3GL compone
    will comprise this applicat

Step-2: Classify complexity levels of eac object

Solution

- For screens,
  Number of views = 4
  Number of data tables = 7
  Number of servers = 3
  Number of clients = 4
- Complexity level for each screen = medium

Reference (Complexity Levels

No. of views contain	Sources of data tabl	
	Total < 4 (< 2 servers < 3 clients)	Total < 8 (2 - 3 servers 3-5 clients)
< 3	Simple	Simple
3 - 7	Simple	Medium
> 8	Medium	Difficult

For Screens

Step-2: Classify complexity levels of eac object

Solution

- For reports,
  Number of sections = 6
  Number of data tables = 7
  Number of servers = 2
  Number of clients = 3
- Complexity level for each report = difficult

Reference (Complexity Levels

No. of section contain	Sources of data tabl	
	Total < 4 (< 2 servers < 3 clients)	Total < 8 (2 - 3 servers 3-5 clients)
0 - 1	Simple	Simple
2 - 3	Simple	Medium
4 +	Medium	Difficult

For Reports

Step-3: Assign complexity weights to ea object

Solution

- By using complexity weight table we can assign complexity weight to each object instance depending upon their complexity level.
  Complexity weight for each screen = 2
  Complexity weight for each report = 8

Reference

Object Type	Complexity Weigh	
	Simple	Medium
Screen	1	2
Report	2	5
3GL Components	-	-

Complexity Weight

Step-4: Determine Object Points

Solution:

- Object point count
- = **Σ** (Number of object instances) * (its Complexity weight)
- = 4 * 2 + 2 * 8 = 24

Reference

- Add all the weighted obje instances to get one num this is known as *object po count.*
- Object Point = **Σ** (number object instances) * (Comp weight of each object ins

Step-5: Compute New Object Points (NOP)

Solution

- %reuse of object points = 10% (given)
- NOP = [object points * (100 - %reuse)]/100
- NOP= [24 * (100 -10)]/100 = 21.6

Reference

- We have to estimate the % be achieved in a project. Depending on %reuse
- NOP = [(object points) * (1(%reuse)]/100
- NOP are the object point th need to be developed and from the object point coun because there may be reus some object instance in pro

Step-6: Calculate Productivity rate (PROD)

Solution

- Developer's experience and capability is low (given)
- Using information given about developer and productivity rate table
- Productivity rate (PROD) of given project = 7

Reference

- *Productivity rate* is calculatec basis of information given ab developer's experience and c

Developers experience & capability	Proc
Very Low	
Low	
Nominal	
High	
High	

Productivity Rate

Step-7: Compute the estimated Effort

Solution:

- Effort = NOP/PROD
- Effort= 21.6/7
- Effort= 3.086 person-month

Reference

- *Effort* to develop a projec be calculated as
- Effort = NOP/PROD Effort measured in *person-mon*

Exercise

- Consider a database application project with
  1. The application has 8 screens with 8 views each and 10 data tables for 6 servers and 8 clients.
  2. Application may generate 4 reports of 6 section each from 10 data tables for 4 servers and 6 clients.
- 20% reuse of object points. Developer's experience and capability in similar environment is low. Calculate the object point count, New object point and effort to develop such project.

- **Given Data**
  - Screen = 8 (6 servers and
  - Views = 8
  - Data tables = 10
  - Reports = 4 (4 servers and
  - Sections = 6
  - Reuse object Points = 20
  - Developer's experience =
- **Calculate**
  - object point count
  - New object point (NOP)
  - Effort

Step 1

Solution:

- **Step-1: Access Object counts**

- Number of screens = 8
  Number of records = 4

Reference

- **Step-1: Access Object counts**
  - Estimate the number of sc
    reports and 3GL compone
    will comprise this applicat

KeralaNotes

Step-2: Classify complexity levels of eac object

Solution

- For screens (Given data),
  Number of views = 8
  Number of data tables = 10
  Number of servers = 6
  Number of clients = 8
- Complexity level for each screen =

Reference (Complexity Levels)

No. of views contain	Sources of data tabl	
	Total < 4 (< 2 servers < 3 clients)	Total < 8 (2 - 3 servers 3-5 clients)
< 3	Simple	Simple
3 - 7	Simple	Medium
> 8	Medium	Difficult

For Screens

Step-2: Classify complexity levels of eac object

Solution

- For screens,
  Number of views = 8
  Number of data tables = 10
  Number of servers = 6
  Number of clients = 8
- Complexity level for each screen = Difficult

Reference (Complexity Levels)

No. of views contain	Sources of data tabl	
	Total < 4 (< 2 servers < 3 clients)	Total < 8 (2 - 3 servers 3-5 clients)
< 3	Simple	Simple
3 - 7	Simple	Medium
> 8	Medium	Difficult

For Screens

KeralaNotes

Step-2: Classify complexity levels of eac object

Solution

- For reports,
  Number of sections = 6
  Number of data tables = 10
  Number of servers = 4
  Number of clients = 6
- Complexity level for each report =

Reference (Complexity Levels)

No. of section contain	Sources of data tabl	
	Total < 4 (< 2 servers < 3 clients)	Total < 8 (2 - 3 servers 3-5 clients)
0 - 1	Simple	Simple
2 - 3	Simple	Medium
4 +	Medium	Difficult

For Reports

Step-2: Classify complexity levels of eac object

Solution

- For reports,
  Number of sections = 6
  Number of data tables = 10
  Number of servers = 4
  Number of clients = 6
- Complexity level for each report = difficult

Reference (Complexity Levels)

No. of section contain	Sources of data tabl	
	Total < 4 (< 2 servers < 3 clients)	Total < 8 (2 - 3 servers 3-5 clients)
0 - 1	Simple	Simple
2 - 3	Simple	Medium
4 +	Medium	Difficult

For Reports

Step-3: Assign complexity weights to ea object

Solution

- By using complexity weight table we can assign complexity weight to each object instance depending upon their complexity level.
  Complexity weight for each screen =
  Complexity weight for each report =

Reference

Object Type	Complexity Weigh	
	Simple	Medium
Screen	1	2
Report	2	5
3GL Components	-	-

Complexity Weight

Step-3: Assign complexity weights to ea object

Solution

- By using complexity weight table we can assign complexity weight to each object instance depending upon their complexity level.
  Complexity weight for each screen = 3
  Complexity weight for each report = 8

Reference

Object Type	Complexity Weigh	
	Simple	Medium
Screen	1	2
Report	2	5
3GL Components	-	-

Complexity Weight

Step-4: Determine Object Points

Solution:

- Object point count
- = **Σ** (Number of object instances) * (its Complexity weight)
- =???

Reference

- Add all the weighted obje instances to get one num this is known as *object pc count.*
- Object Point = **Σ** (number object instances) * (Comp weight of each object ins

Step-4: Determine Object Points

Solution:

- Object point count
- = **Σ** (Number of object instances) * (its Complexity weight)
- = 8 * 3 + 4 * 8 = 56

Reference

- Add all the weighted obje instances to get one num this is known as *object po count.*
- Object Point = **Σ** (number object instances) * (Comp weight of each object ins

Step-5: Compute New Object Points (NOP)

Solution

- %reuse of object points = 10% (given)
- NOP = [object points * (100 - %reuse)]/100
- NOP= ???

Reference

- We have to estimate the % be achieved in a project. Depending on %reuse
- NOP = [(object points) * (1(%reuse)]/100
- NOP are the object point th need to be developed and from the object point coun because there may be reus some object instance in pro

Step-5: Compute New Object Points (NOP)

Solution

- %reuse of object points = 10% (given)
- NOP = [object points * (100 - %reuse)]/100
- NOP= [56 * (100 -20)]/100 = 44.8

Reference

- We have to estimate the % be achieved in a project. Depending on %reuse
- NOP = [(object points) * (1(%reuse)]/100
- NOP are the object point th need to be developed and from the object point coun because there may be reus some object instance in pro

Step-6: Calculate Productivity rate (PROD)

Solution

- Developer's experience and capability is low (given)
- Using information given about developer and productivity rate table
- Productivity rate (PROD) of given project =

Reference

- *Productivity rate* is calculatec basis of information given ab developer's experience and c

Developers experience & capability	Proc
Very Low	
Low	
Nominal	
High	
High	

Productivity Rate

Step-6: Calculate Productivity rate (PROD)

Solution

- Developer's experience and capability is low (given)
- Using information given about developer and productivity rate table
- Productivity rate (PROD) of given project = 7

Reference

- *Productivity rate* is calculatec basis of information given ab developer's experience and c

Developers experience & capability	Proc
Very Low	
Low	
Nominal	
High	
High	

Productivity Rate

Step-7: Compute the estimated Effort

Solution:

- Effort = NOP/PROD
- Effort=
- Effort=

Reference

- *Effort* to develop a projec be calculated as
- Effort = NOP/PROD Effort measured in *person-mon*

Step-7: Compute the estimated Effort

Solution:

- Effort = NOP/PROD
- Effort= 44.8/7
- Effort= 6.4 person-month

Reference

- *Effort* to develop a projec be calculated as
- Effort = NOP/PROD Effort measured in *person-mon*

KeralaNotes

Early Design Model

https://www.keralanotes.com/

Early Design Model

- Estimates can be made after the requirements have been agreed.
- Based on a standard formula for algorithmic models
  - PM = A × Size^{B} × M where
  - M = PERS × RCPX × RUSE × PDIF × PREX × FCIL × SCED;
  - A = 2.94 in initial calibration, Size in KLOC, B varies from 1.1 to 1.24 depending on novelty of the project, development flexibility, risk management approaches and the process maturity.

Multipliers

KeralaNotes

- Multipliers reflect the capability of the developers, the non-functional requirements, the familiarity with the development platform, etc.
  - RCPX - product reliability and complexity;
  - RUSE - the reuse required;
  - PDIF - platform difficulty;
  - PREX - personnel experience;
  - PERS - personnel capability;
  - SCED - required schedule;
  - FCIL - the team support facilities.

Early Design	Very l
RCPX	
RUSE	
PDIF	
PERS	
PREX	
FCIL	
SCED	

The reuse model

- Takes into account black-box code that is r without change and code that has to be adapted to integrate it with new code.
- There are two versions:
  - Black-box reuse where code is not modified. Ar estimate (PM) is computed.
  - White-box reuse where code is modified. A size estimate equivalent to the number of lines of n source code is computed. This then adjusts the estimate for new code.

Reuse model estimates 1 (Black b

- For generated code:
  - PM = (ASLOC * AT/100)/ATPROD
  - ASLOC is the number of lines of generated cc
  - AT is the percentage of code automatically generated.
  - ATPROD is the productivity of engineers in int this code.

Reuse model estimates 2 (W

- When code has to be understood and integrated:
  - ESLOC = ASLOC * (1-AT/100) * AAM.
  - ASLOC and AT as before.
  - AAM is the adaptation adjustment multiplier computed from the costs of changing the reused code, the costs of understanding how to integrate t code and the costs of reuse decision making.

ESLOC means equivalent number of lines of new source code.

ASLOC means the source lines of code in the component that has to be adapted.

AAM is adaptation Adjustment multiplier. This factor is used to take into account the efforts required to reuse the code.

Reuse model estimation 2 (White box) – Cont….

- Simplistically, AAM is the sum of three components:
  - 1. An **assessment factor** (referred to as AA) that represents the effort involved deciding whether or not to reuse components. AA varies from 0 to 8 dependi amount of time you need to spend looking for and assessing potential candida reuse.
  - 2. An understanding component (referred to as SU) that represents the costs o understanding the code to be reused and the familiarity of the engineer with that is being reused. SU ranges from 50 for complex, unstructured code to 10 f written, object-oriented code.
  - 3. An adaptation component (referred to as AAF) that represents the costs of changes to the reused code. These include design, code, and integration chang
- Use Early estimation model to compute the Effort
  - $PM=A \times Size^{B} \times M$

Post-architecture level

- Uses the same formula as the early desig model but with 17 rather than 7 associat multipliers.
- The code size is estimated as:
  - Number of lines of new code to be develope
  - Estimate of equivalent number of lines of ne computed using the reuse model;
  - An estimate of the number of lines of code t to be modified according to requirements ch

4. Post architecture model

- This model is a detailed model used to compute the efforts. formula used in this model is

$$\text{Effort} = A \times \text{Size}^B \times M$$

- In this model efforts should be estimated more accurately. In formula **A** is the amount of code. This code size estimate is mad help of three components –
  1. The estimate about **new** lines of code that is added in the progr
  2. Equivalent number of source lines of code (ESLOC) used in **reu**
  3. Due to changes in requirements the lines of code get mo estimate of amount of **code** being **modified**.
- The exponent term **B** has three possible values that are related to of project complexity. The values of B are continuous rather than depends upon the five scale factors. These scale factors vary from to extra high (i.e. from 5 to 0).