Unit V S/w Project Estimation and Scheduling

Vidula Meshram
vidula.meshram@viit.ac.in
Department of Computer Engineering

BRACT'S, Vishwakarma Institute of Information Technology, Pune-48

(An Autonomous Institute affiliated to Savitribai Phule Pune University)
(NBA and NAAC accredited, ISO 9001:2015 certified)

Unit 5

S/w Project Estimation and Scheduling

Course Objective and Outcome

Course Objective:

To learn to estimate cost and schedule of a software project

Course Outcome:

Estimate cost and schedule of the software project.

Unit IV Syllabus

- Project Estimation
- Different methods of estimation (COCOMO model, Delphi cost estimation etc.)
- Function point analysis
- PERT & Gantt Charts
- Introduction to Microsoft Project
- CM planning, Change Management
- Version and Release Management
- Configuration Management Tools

Project Definition

- “Unique process consisting of a set of coordinated and controlled activities with start and finish dates, undertaken to achieve an objective conforming to specific requirements, including constraints of time, cost, quality and resources”

- A Project is a planned set of activities
- A Project has a scope
- A Project has time, cost, quality and resource constraints

Project Management

The art of organising, leading, reporting and completing a project through people

A Quote on Measurement

"When you can measure what you are speaking about and express it in numbers, you know something about it; but when you cannot measure, when you cannot express it in numbers, your knowledge is of an inadequate and unsatisfactory kind; it may be the beginning of knowledge, but you have not quite in your thoughts, advanced to the stage of science."

LORD WILLIAM KELVIN (1824 – 1907)

Uses of Measurement

- Can be applied to the software process with the intent of improving it on a continuous basis
- Can be used throughout a software project to assist in estimation, quality control, productivity assessment, and project control
- Can be used to help assess the quality of software work products and to assist in strategic decision making as a project proceeds.

Software Metrics

- Standard of measure that contains many activities which involve some degree of measurement
  - Direct Metrics: Immediately measurable Eg: Line of Code
  - Indirect Metrics: Not immediately quantifiable Eg: Functionality

- **Product Metrics**
- Describes the characteristics of the product
  - size
  - Complexity
  - Design features
  - Performance
  - Quality level
  - Reliability
  - Functionalities

- **Process Metrics**
- Used to improve the development process and maintenance activities of the software
  - Effort required
  - Time to produce the product
  - No of defects found
  - Tools and tech
  - Quality
  - Efficiency

- **Project Metrics**
- Describes the project characteristics and execution
  - No of s/w developers
  - Staffing patterns
  - Cost
  - Schedule
  - Productivity
  - Quality
  - Assess status of ongoing project

https://www.youtube.com/watch?v=6DT7qyP5tK4

Size Oriented Metrics (KLOC Measurements)

Metrics for Software Cost and Effort Estimation

- Measures **size** of the software produced
- Normalized by counting in Thousand Lines Of Code (KLOC)
- Size-oriented metrics of project computed by:
  - Errors per KLOC (thousand lines of code)
  - Defects per KLOC
  - $ per KLOC
  - Pages of documentation per KLOC
- Other metrics computed are:
  - Errors per person-month
  - KLOC per person-month
  - $ per page of documentation

Advantages:

- easy to count or calculate from developed code

Issues in Estimation using Size Oriented Metrics

- Programming Language Dependent
- Penalize well-designed but short programs
- Not universally accepted as a best way to measure software process

https://www.youtube.com/watch?v=6DT7qyP5tK4

Function Oriented Metrics

- Measure of the **functionality** delivered by the application
- Most widely used metric of this type is the **Function Point**
- Using historical data it can:
  - Estimate the cost or effort required to design, code, and test the software
  - Predict the number of errors that will be encountered during testing
  - Forecast the number of components and/or the number of projected source lines in the implemented system

Advantages over Size Oriented Metrics:

- Programming language independent
- Based on data that are more likely to be known in the early stages of a project

Disadvantages

- Computation is based on subjective data
- Counts of the information domain can be difficult to collect
- Has no direct physical meaning, it's just a number

https://www.youtube.com/watch?v=6DT7qyP5tK4

Function Point - Estimation

What is function Point ?

To measure the standard worth of the software, as a unit of software worth, Function Point was developed.

Allan Albrecht of IBM in 1977.

FP measures functionality from the User's point of view like what the user receives from the software and what the user requests from the software. It focuses on what functionality is being delivered.

"A Functional Point" is a unit of measurement to express the amount of business functionality an information system provides to a user. - Wiki

Using historical data, the FP metric can then be used to

1. Estimate the cost or effort required to design, code, and test the software.

2. Predict the number of errors that will be encountered during testing

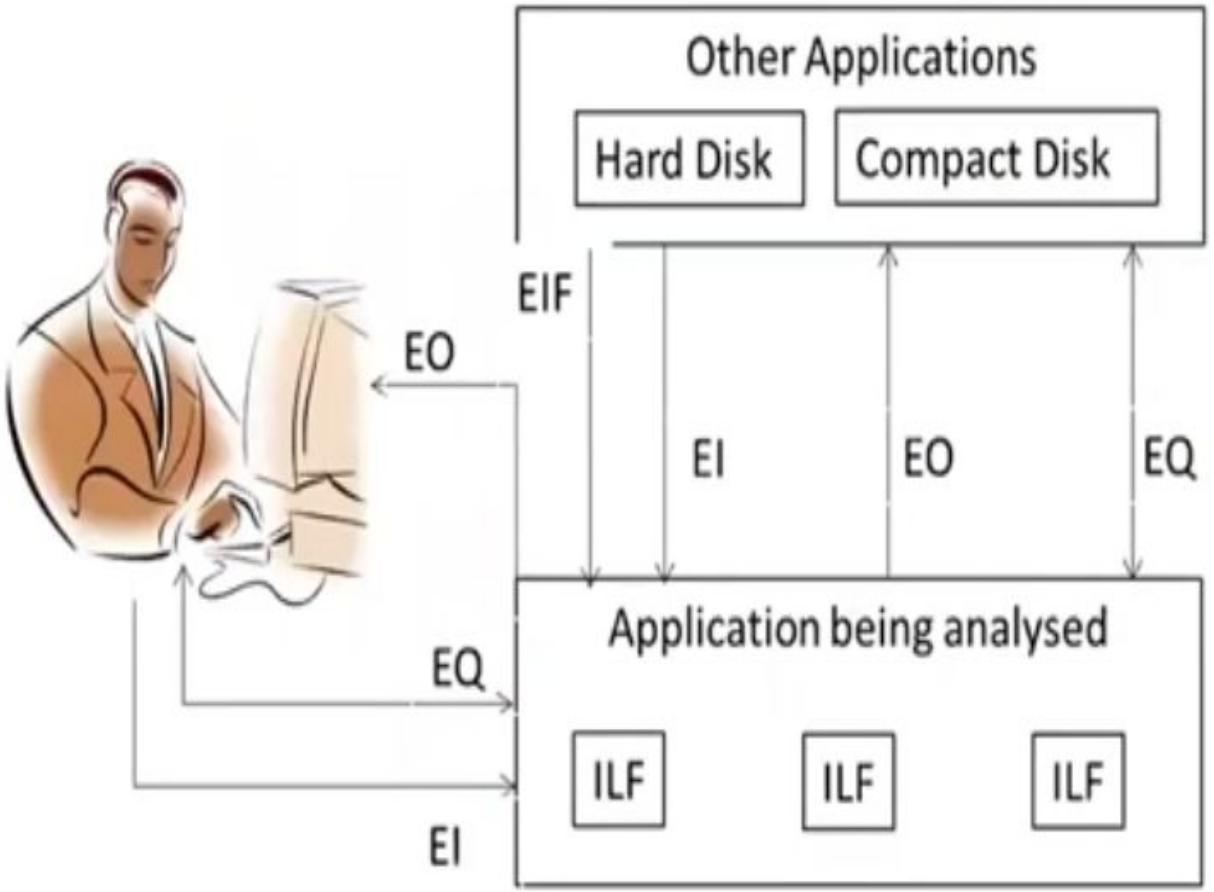
3. Forecast the number of components and/or the number of projected source lines in the implemented system.

https://www.youtube.com/watch?v=CeKP0rUIotc

Function Oriented Inputs

Principle of FPA

- System is decomposed into five functional units
  - External Inputs (EI)
  - External Outputs (EO)
  - External Enquiries (EQ)
  - Internal Logical Files (ILF)
  - External Interface Files (EIF)



Users view of the system

Function Point - Estimation

Functional Unit	Wighting Factors		
	Low	Average	High
External Inputs (EI)	3	4	6
External Outputs (EO)	4	5	7
External Enquired (EQ)	3	4	6
Internal Logic Files (ILF)	7	10	15
External Interface Files (EIF)	5	7	10

$$F.P = UFP \ X \ CAF$$

https://www.youtube.com/watch?v=CeKP0rUIotc

Function Point - Estimation

Step 1. Calculating UFP - Unadjusted Function Point.

$$F.P = UFP \ X \ CAF$$

Measurement Parameter	Counts		Weighting Factor: Low	Average	High		
External Inputs (EI)		X	3	4	6	=	
External Outputs (EO)		X	4	5	7	=	+
External Enquired (EQ)		X	3	4	6	=	+
Internal Logic Files (ILF)		X	7	10	15	=	+
External Interface Files (EIF)		X	5	7	10	=	+

Unadjusted Function Points (UFP) = Total Count =

UFP = Sum of all the Complexities of all the EI's, EO's, EQ's, ILF's, and EIF's

https://www.youtube.com/watch?v=CeKP0rUIotc

Function Point - Estimation

Step 2: Calculating CAF - Complexity Adjustment Factor

$$F.P = UFP \ X \ CAF$$

$$CAF = 0.65 + (0.01 \times \sum Fi)$$

Where,

Fi = Value adjustment factors based on responses to the following 14 questions

https://www.youtube.com/watch?v=CeKP0rUIotc

Function Point - Estimation

Step 2. Calculating CAP – Complexity Adjustment Factor

Questionnaires for software development

1. Does the system require reliable backup and recovery? (3)
2. Are specialized data communications required to transfer information to or from the application?
3. Are there distributed processing functions?
4. Is performance critical?
5. Will the system run in an existing, heavily utilized operational environment?
6. Does the system require online data entry?
7. Does the online data entry require the input transaction to be built over multiple screens or operations?
8. Are the ILFs updated online?
9. Are the inputs, outputs, files, or inquiries complex?
10. Is the internal processing complex?
11. Is the code designed to be reusable?
12. Are conversion and installation included in the design?
13. Is the system designed for multiple installations in different organizations?
14. Is the application designed to facilitate change and ease of use by the user?

Complexity Adjustment Factor is calculated using 14 aspects of processing complexity and these 14 questions answered on a scale of 0 – 5

0 - No Influences or No Important

1 - Incidental

2 - Moderate

3 - Average

4 - Significant

5 - Essential

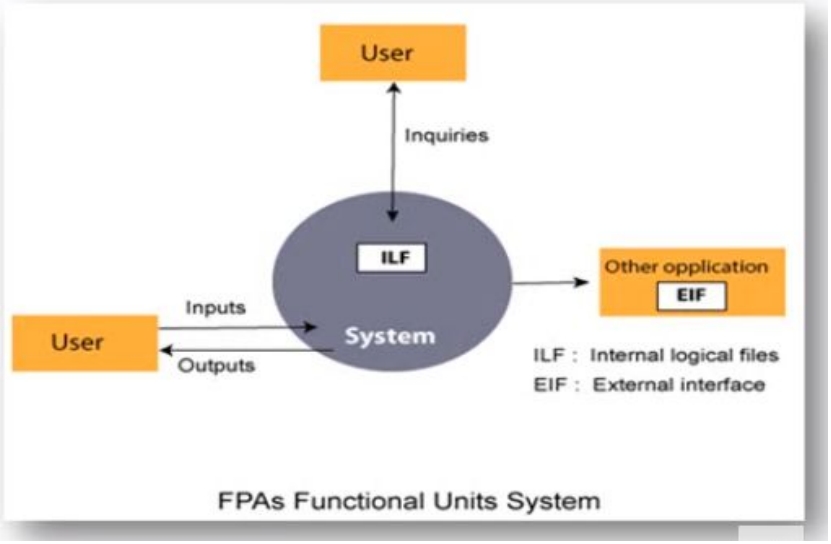
https://www.youtube.com/watch?v=CeKP0rUIotc

Function Point

- Function-oriented software metrics measure functionality, what the system performs, is the measure of the system size.
- It find out by counting the number and types of functions used in the applications

FP Attributes

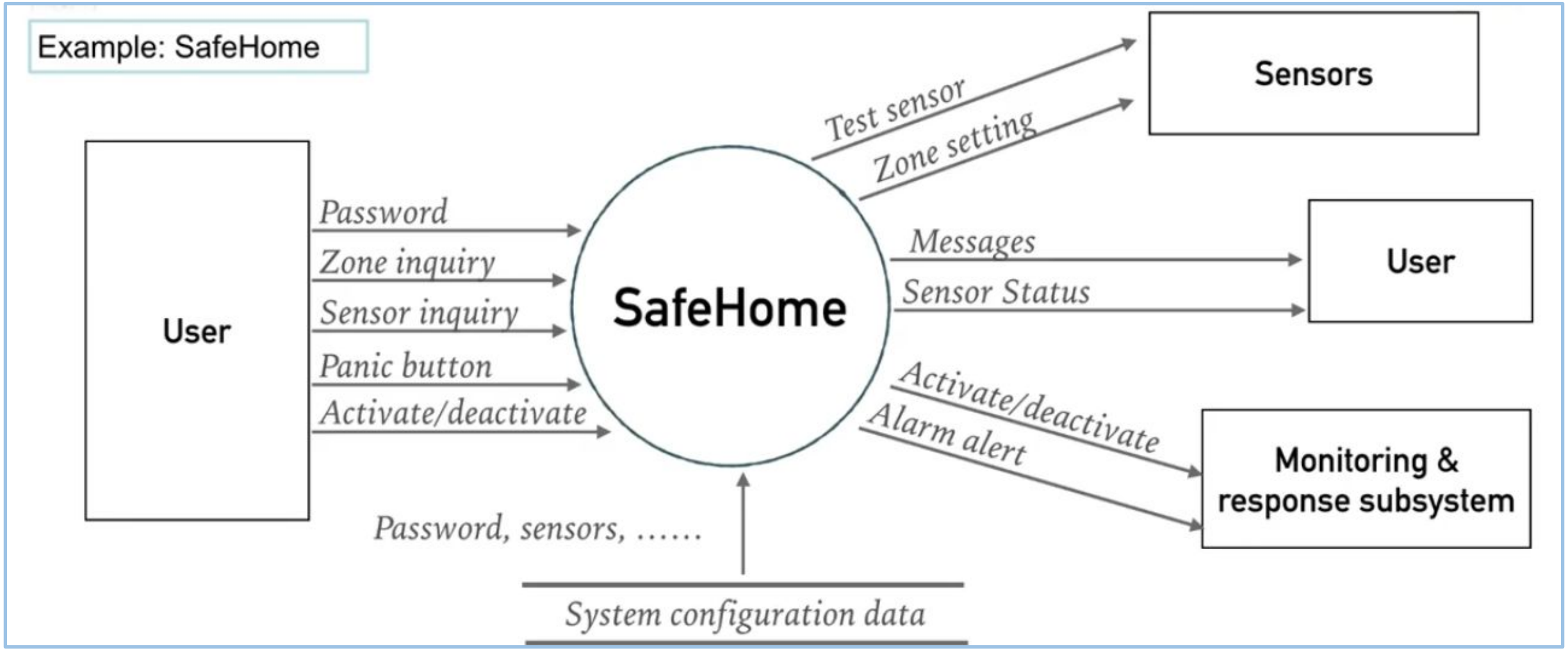
Measurements Parameters	Examples
1.Number of External Inputs(EI)	Input screen and tables
2. Number of External Output (EO)	Output screens and reports
3. Number of external inquiries (EQ)	Prompts and interrupts.
4. Number of internal files (ILF)	Databases and directories
5. Number of external interfaces (EIF)	Shared databases and shared routines.



FPAs Functional Units System

https://www.youtube.com/watch?v=CeKP0rUIotc

Function Point - Estimation



Flow Model of Safe Home User Interaction

https://www.youtube.com/watch?v=CeKP0rUIotc

Function Point Estimation

Step 1. Calculating UFP - Unadjusted Function Point.

$$F.P = UFP \ X \ CAF$$

Measurement Parameter	Counts		Weighting Factor: Low	Average	High		
External Inputs (EI)	3	X	3			=	9
External Outputs (EO)	2	X	4			=	8
External Enquired (EQ)	2	X	3			=	6
Internal Logic Files (ILF)	1	X	7			=	7
External Interface Files (EIF)	4	X	5			=	20
Unadjusted Function Points (UFP)						= Total Count =	50

UFP = Sum of all the Complexities of all the EI's, EO's, EQ's, ILF's, and EIF's

Function Point - Estimation

$$F.P = UFP \ X \ CAF$$

$$CAF = 0.65 + (0.01 \times \Sigma Fi)$$ Moderately complex product

Then,

$$\Sigma Fi = 14 \times 2 = 28$$ 2 - Moderate

$$CAF = 0.65 + (0.01 \times 28)$$

$$CAF = 0.93$$

$$F.P = UFP \ X \ CAF$$

$$F.P = 50 \ X \ 0.93 = 46.5$$

https://www.youtube.com/watch?v=CeKP0rUIotc

Function Point - Estimation

Given the Following Values, calculate the Functional Point when complexity adjustment factors are significantly complex product and weighting factors are high.

User input = 55

User Output = 35

User Enquires = 40

User Files = 8

External Interfaces = 5

$$F.P = UFP \ X \ CAF$$

Where,

UFP = Sum of all the Complexities of all the EI's, EO's, EQ's, ILF's, and EIF's

$CAF = 0.65 + (0.01 \times \Sigma Fi)$

Step 1. Calculating UFP - Unadjusted Function Point.

$$F.P = UFP \ X \ CAF$$

Measurement Parameter	Counts		Weighting Factor: Low	Weighting Factor: Average	Weighting Factor: High		
External Inputs (EI)	55	X			6	=	330
						+	
External Outputs (EO)	35	X			7	=	245
						+	
External Enquired (EQ)	40	X			6	=	240
						+	
Internal Logic Files (ILF)	8	X			15	=	120
						+	
External Interface Files (EIF)	5	X			10	=	50

Unadjusted Function Points (UFP) = Total Count = 985

UFP = Sum of all the Complexities of all the EI's, EO's, EQ's, ILF's, and EIF's

Function Point - Estimation

$$F.P = UFP \ X \ CAF$$

$$CAF = 0.65 + (0.01 \times \Sigma Fi)$$

Significantly complex

Hence,

$$\Sigma Fi = 14 \times 4 = 56$$

$$CAF = 0.65 + (0.01 \times 56)$$

$$CAF = 1.21$$

$$F.P = UFP \ X \ CAF$$

$$F.P = 985 \ X \ 1.21 = 1191.85$$

Complexity factors

5 questions = Average

5 questions = Moderate

4 questions = No influence

$\Sigma Fi = (5 \times 3) + (5 \times 2) + (4 \times 0) = 25$

https://www.youtube.com/watch?v=CeKP0rUIotc

Function Point - Estimation

The Steps invloved in Function Point Analysis are:

1. Determine the Five Components Value
2. Compute the Unadjusted Function Point UFP
3. Compute Complexity Adjustment Factor CAF based on 14 general system characteristics
4. Calculate the Final Function Point

https://www.youtube.com/watch?v=CeKP0rUIotc

Function Point - Estimation

Calculations

- FPA = UFP*CAF
- $UFP = \sum_{i=1\ j=1}^{i=5,j=3} W_{ij} * C_{ij}$
- $CAF=0.65+[0.01*\sum F_i]$

Function Point Analysis (FPA)
Unadjusted Function Point (UFP)
Complexity Adjustment Factor (CAF)
Weighting Factor (W)
Count (C)
Total Complexity Adjustment Value (F)

FPA Matrix

Information Domain Value	Weighting factor Simple	Average	Complex
External Inputs (EIs)	3	4	6
External Outputs (EOs)	4	5	7
External Inquiries (EQs)	3	4	6
Internal Logical Files (ILFs)	7	10	15
External Interface Files (EIFs)	5	7	10

- ***Calculate FPA if all CAF and WAF are average for the following count values EI=10, EO=30, EQ=50, EIF=10, ILF=20***

Step1: Calculate $UFP = \sum_{i=1\ j=1}^{i=5,j=3} W_{ij} * Ci_j$

UFP=(10*4)+(30*5)+(50*4)+(10*7)+(20*10)

UFP=40+150+200+70+200

UFP=660

- **Step 2: Calculate** $CAF=0.65+[0.01*\sum F_i]$

CAF=0.65+[0.01*14*3]

CAF=1.07

Step 3: Calculate FPA=UFP*CAF

FPA=660*1.07

FPA=706.2

https://www.youtube.com/watch?v=6DT7qyP5tK4

Project Estimation Techniques

Estimation of various projects parameters is an important project planning activity. The different parameters of a project that need to be estimated include–

- Project Size,
- Effort required to complete the project,
- Project Duration and
- Cost

Accurate estimation of these parameters is important for resource planning and scheduling. Estimation Techniques can be classified as :

- **Empirical Estimation Techniques**
- **Heuristic Estimation Techniques**
- **Analytical Estimation Techniques**

https://www.youtube.com/watch?v=Ir0vA_p6T7Q&t=848s

COCOMO

- Stands for Constructive Cost Model
- Introduced by Barry Boehm in 1981 in his book “Software Engineering Economics”
- Became one of the well-known and widely-used estimation models in the industry
- It has evolved into a more comprehensive estimation model called COCOMO II
- COCOMO II is actually a hierarchy of three estimation models
- As with all estimation models, it requires sizing information and accepts it in three forms: object points, function points, and lines of source code

COCOMO

- projects used in this model have following attributes :-
  1. ranging in size from 2,000 to 100,000 lines of code
  2. programming languages ranging from assembly to PL/I.
  3. These projects were based on the waterfall model of software development.
- Boehm stated that any software development project can be classified into three categories :-

1. Organic:

- If the project deals with developing a well understood application program.
- The size of development is reasonably small and experienced.
- The team members are experienced in developing similar kind of projects.

https://www.youtube.com/watch?v=lr0vA_p6T7Q&t=848s

COCOMO

2. Semidetached:

- If the development team consists of a combination of both experienced and inexperienced staff.
- Team members have limited experience about some aspects but are totally unfamiliar with some aspects of the system being developed.
- Mixed Experience.

3. Embedded:

- If the software being developed is strongly coupled to complex hardware.
- Software projects that must be developed within a set of tight software, hardware and operational constraints.

https://www.youtube.com/watch?v=IroVA_p0TYQ&t=848s

Mode	Project size	Nature of Project	Innovation	Deadline of the project	Development Environment
Organic	Typically 2-50 KLOC	Small size project, experienced developers in the familiar environment. For example, pay roll, inventory projects etc.	Little	Not tight	Familiar & In house
Semi detached	Typically 50-300 KLOC	Medium size project, Medium size team, Average previous experience on similar project. For example: Utility systems like compilers, database systems, editors etc.	Medium	Medium	Medium
Embedded	Typically over 300 KLOC	Large project, Real time systems, Complex interfaces, Very little previous experience. For example: ATMs, Air Traffic Control etc.	Significant	Tight	Complex Hardware/ customer Interfaces required

https://www.youtube.com/watch?v=lr0vA_p6T7Q&t=848s

Person Month

- The effort estimation is expressed in units of person-months (PM).
- An effort of 100 PM does not imply that 100 persons should work for 1 month nor does it imply that 1 person should be employed for 100 months, but it denotes the area under the person-month curve .
- It is the area under the person-month plot.

https://www.youtube.com/watch?v=Ir0vA_p6T7Q&t=848s

COCOMO VERSIONS

- Basic COCOMO
- Intermediate COCOMO
- Complete/Detailed COCOMO

Basic COCOMO VERSION

- Basic COCOMO model computes software development effort, time and cost as a function of program size. Program size is expressed in estimated thousands of source lines of code (SLOC, KLOC).

$$EFFORT = a_1 \times (KLOC)^{a2} \ PM$$

$$T_{dev} = b_1 \times (Effort)^{b2} \ Months$$

Where

- KLOC is the estimated number of delivered lines (expressed in thousands) of code for project, estimated size of software product.
- The coefficients a_1, a_2, b_1 and b_2 are constants for each category of software products.
- T_{dev} is the estimated time to develop the software, in months.
- Effort is the total efforts required to develop the software product, expressed in Person Months (PM)

https://www.youtube.com/watch?v=lr0vA_p6T7Q&t=848s

Estimation of Development Effort

Organic	: Effort = $2.4(KLOC)^{1.05}$	PM
Semi-detached	: Effort = $3.0(KLOC)^{1.12}$	PM
Embedded	: Effort = $3.6(KLOC)^{1.20}$	PM

https://www.youtube.com/watch?v=lr0vA_p6T7Q&t=848s

Software Projects	a_1	a_2	b_1	b_2
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 30,000 lines of code . We will obtain estimation for this project as follows:

(1)Effort estimation E= a1(KLOC)Exp(a2)person-months

E=3.0(30)Exp(1.12) where lines of code=30000=30 KLOC

E=135 person-month

(2) Duration estimation D=b1(E)Exp(b2)months =2.5(135)Exp(0.35)

D=14 months

(3) Person Estimation N = E/D

= 135/14 N = 10 Person Approximately

Calculating Effort and Estimation

When effort and development time are known, the average staff size to complete the project may be calculated as:

$$\text{Average staff size } (SS) = \frac{E}{D} \; Persons$$

When project size is known, the productivity level may be calculated as:

$$\text{Productivity } (P) = \frac{KLOC}{E} \; KLOC / PM$$

https://www.youtube.com/watch?v=lr0vA_p6T7Q&t=848s

Basic COCOMO

Merits-

- Basic COCOMO is good for quick , rough and early estimate of software costs.

Demerits-

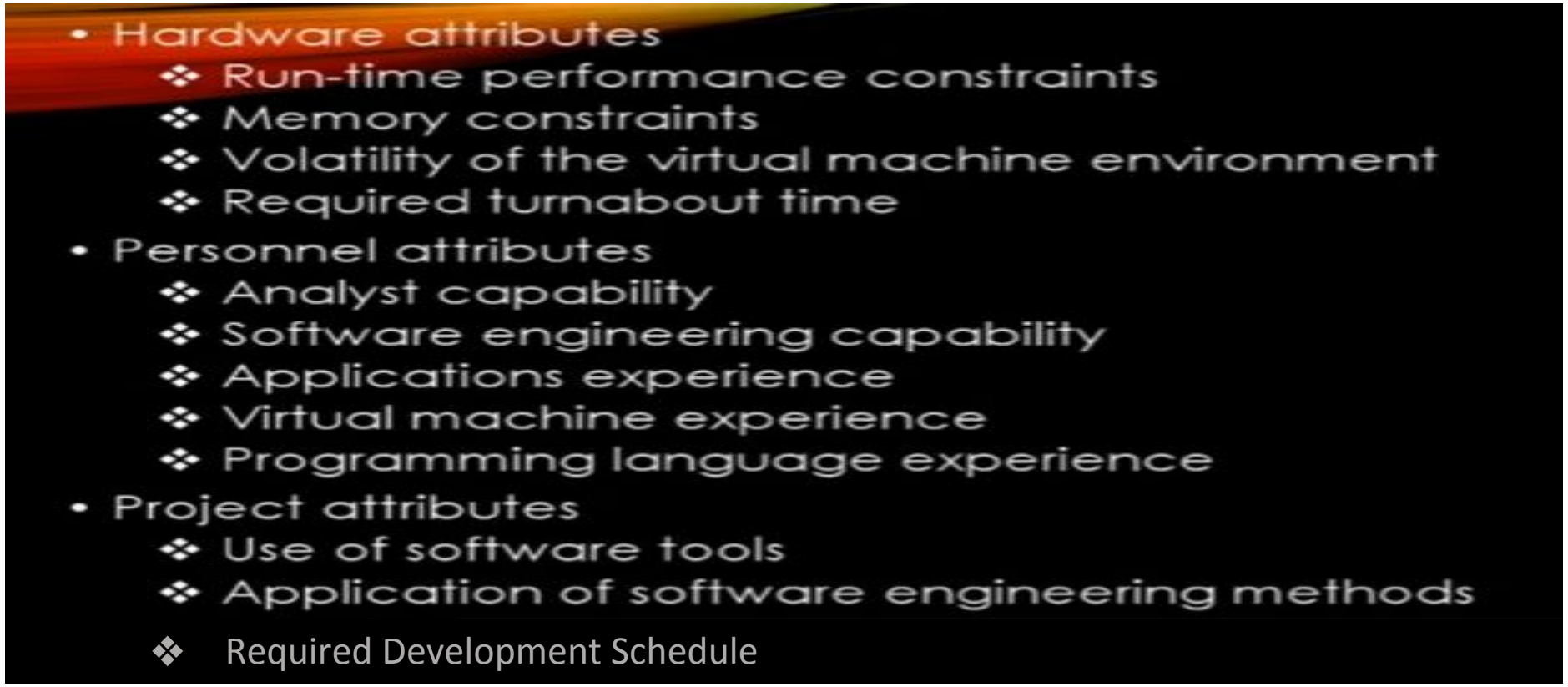
- It does not account for differences in hardware constraints, personnel quality and experience, use of modern tools and techniques, and so on.
- The accuracy of this model is limited because it does not consider certain factors for cost estimation of software.

https://www.youtube.com/watch?v=lr0vA_p6T7Q&t=848s

Intermediate COCMO Model

- *Intermediate COCOMO* computes software development effort as function of program size and a set of "cost drivers" that include subjective assessment of product, hardware, personnel and project attributes.
- The Intermediate COCOMO model refines the initial estimate obtained from the basic COCOMO by scaling the estimate up or down based on attributes of software development.
- This model uses a set of 15 cost drivers, these cost drivers are multiplied with the initial cost and effort estimates to scale the estimates up and down.
- This extension considers a set of "cost drivers", each with a number of subsidiary attributes:
- Product attributes
  - Required software reliability
  - Size of application database
  - Complexity of the product

https://www.youtube.com/watch?v=Ir0vA_p6T7Q&t=848s



https://www.youtube.com/watch?v=lr0vA_p6T7Q&t=848s

- Each of the 15 attributes receives a rating on a six-point scale that ranges from "very low" to "extra high" (in importance or value).
- formula now takes the form:

$$E=a_1(KLOC)^{(b1)} \times (EAF)$$

Where

E : Effort applied in terms of person- months

KLOC : Kilo lines of code for the project

EAF : It is the effort adjustment factor

- The value of a1 and b1 for various class of software projects:

Software Projects	a_1	b_1
Organic	3.2	1.05
Semi – Detached	3.0	1.12
Embedded	2.8	1.20

https://www.youtube.com/watch?v=lr0vA_p6T7Q&t=848s

Project Characteristics Table

Cost adjustments for computing the EAF (Effort Adjustment Factor)

	v.low	low	nominal	high	v.high	ex.high
product attributes						
required software reliability	0.75	0.88	1.00	1.15	1.40	
database size		0.94	1.00	1.08	1.16	
product complexity	0.70	0.85	1.00	1.15	1.30	1.65
computer attributes						
execution time constraints			1.00	1.11	1.30	1.66
main storage constraints			1.00	1.06	1.21	1.56
virtual machine volitility	0.87	1.00	1.15	1.30		
computer turnaround time		0.87	1.00	1.07	1.15	
personnel attributes						
analyst capability	1.46	1.19	1.00	0.86	0.71	
applications experience	1.29	1.13	1.00	0.91	0.82	
programmer capability	1.42	1.17	1.00	0.86	0.70	
virtual machine experience	1.21	1.10	1.00	0.90		
programming language experience	1.14	1.07	1.00	0.95		
project attributes						
use of modern programming practices	1.24	1.10	1.00	0.91	0.82	
use of software tools	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

https://www.youtube.com/watch?v=Ir0vA_p6T7Q&t=848s

Example:

Consider a project having 30,000 lines of code which in an embedded software with critical area hence reliability is high.The estimation can be

$$E=a_1(KLOC)^{a2} \times (EAF)$$

As reliability is high EAF=1.15(product attribute)

$a_1=2.8$

$b_1=1.20.$ for embedded software

$E=2.8(30)^{1.20} \times 1.15$

=191 person month

$D=b_1(E)^{b2}$

$=2.5(191)^{0.32}$

=13 months approximately

N=E/D =191/13

N=15 persons approx.

https://www.youtube.com/watch?v=lr0vA_p6T7Q&t=848s

Intermediate COCOMO

Merits:

- This model can be applied to almost to entire software product for easy and rough cost estimation during early stage.
- It can be applied at the software product component level for obtaining more accurate cost estimation.

Demerits:

- The effort multipliers are not dependent on phases.
- A product with many components is difficult to estimate.

https://www.youtube.com/watch?v=lr0vA_p6T7Q&t=848s

SHORTCOMING OF BASIC AND INTERMEDIATE COCOMO MODELS

- Both models:
  - consider a software product as a single homogeneous entity:
  - However, most large systems are made up of several smaller sub-systems.
  - Some sub-systems may be considered as organic type, some may be considered embedded, etc.
  - For some the reliability requirements may be high, and so on.
  - So, complete COCOMO was proposed to overcome these limitations of basic and intermediate COCOMO.

https://www.youtube.com/watch?v=lr0vA_p6T7Q&t=848s

Complete COCOMO

- Incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, etc.) of the software engineering process.
- The complete COCOMO model considers the differences in characteristics of all the subsystems and estimates the effort and development time as sum of the estimates for the individual sub systems.
- Uses different effort multipliers for each cost driver attribute. These **Phase Sensitive** effort multipliers are each to determine the amount of effort required to complete each phase. In complete COCOMO, the whole software is divided in different modules and then we apply COCOMO in different modules to estimate effort and then sum the effort
- The effort is calculated as function of program size and a set of cost drivers given according to each phase of software life cycle.

https://www.youtube.com/watch?v=lr0vA_p6T7Q&t=848s

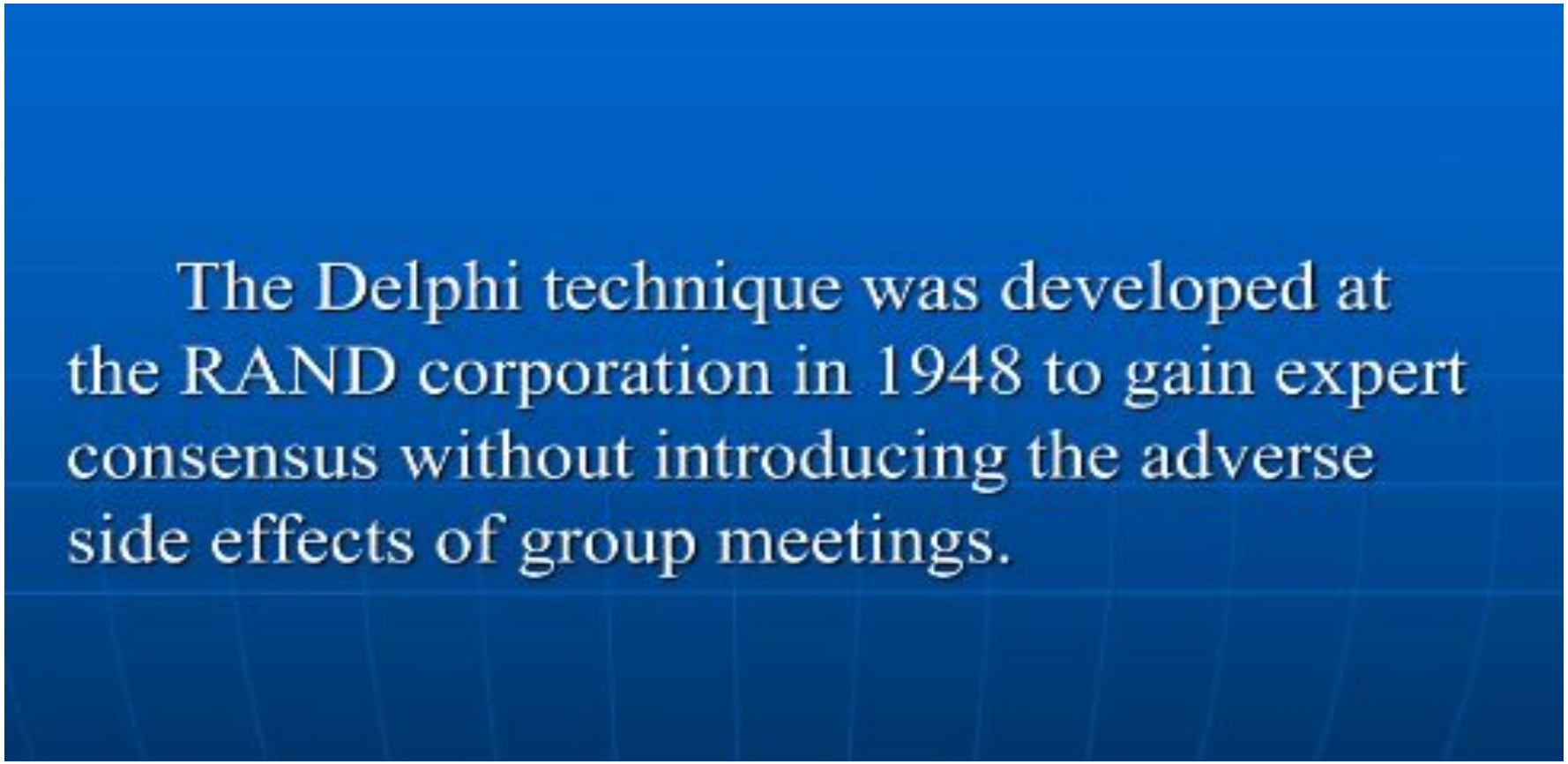
Example of Complete COCOMO

- Cost of each sub-system is estimated separately.
- Costs of the sub-systems are added to obtain total cost.
- Reduces the margin of error in the final estimate.

- **Example of complete COCOMO Model-**
- A Management Information System (MIS) for an organization having offices at several places across the country:
  - Database part (semi-detached)
  - Graphical User Interface (GUI) part (organic)
  - Communication part (embedded)
- Costs of the components are estimated separately:
  - summed up to give the overall cost of the system.

https://www.youtube.com/watch?v=lr0vA_p6T7Q&t=848s

Delphi Cost Estimation Technique



Ref : Delphi cost estimation model (slideshare.net) shashwat2010/delphi-cost-estimation-model

Delphi Cost Estimation Technique

The Delphi technique can be adapted to software cost estimation in the following manner:

- A coordinator provides each estimator with the System Definition document and form for recording a cost estimate.
- Estimators study the definition and complete their estimates anonymously. They may ask questions of the coordinator, but they do not discuss their estimates with one another.

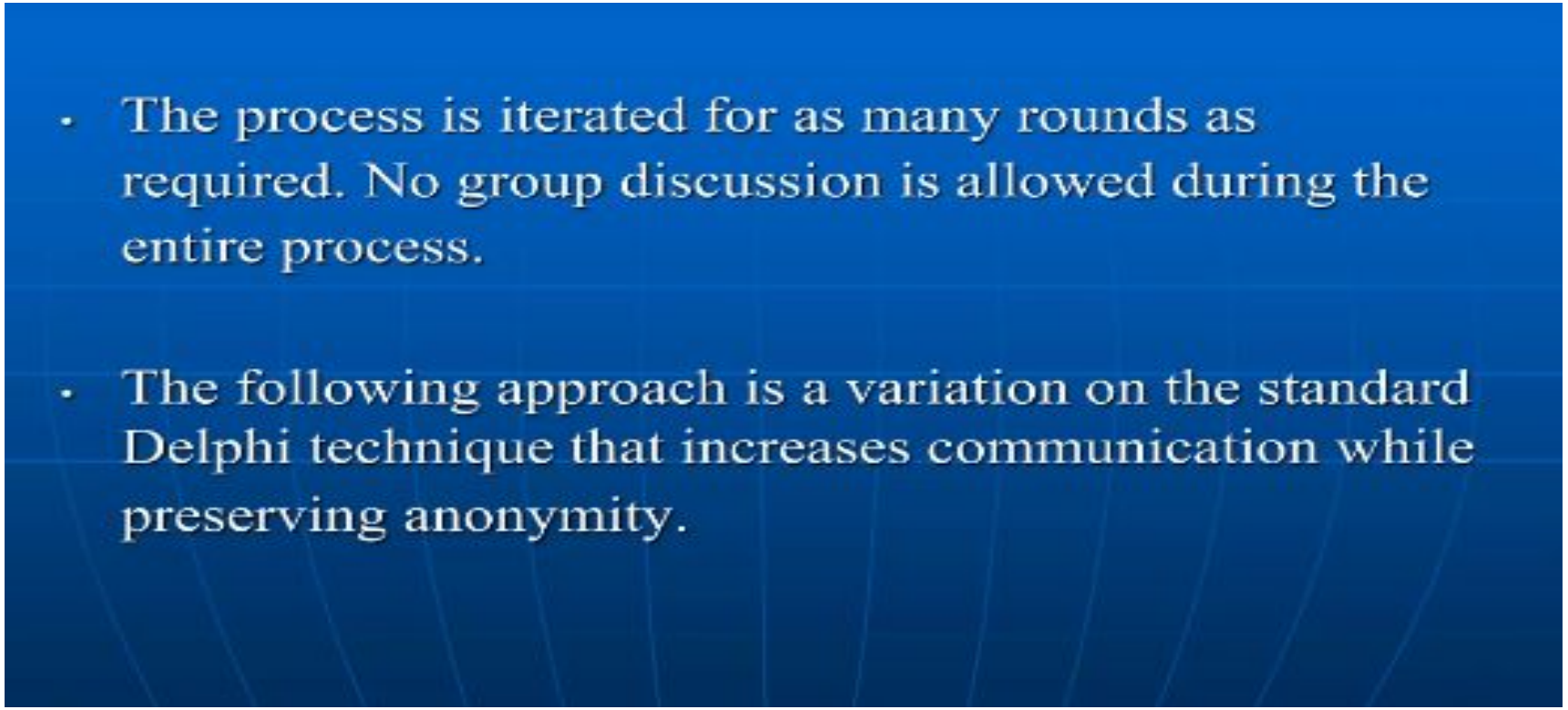
Ref : Delphi cost estimation model (slideshare.net) shashwat2010/delphi-cost-estimation-model

Delphi Cost Estimation Technique

- The coordinator prepares and distributes a summary of the estimators' responses, and includes any unusual rationales noted by the estimators.
- Estimators complete another estimate again anonymously using the results from the previous estimate. Estimators whose estimates differ sharply from the group may be asked, to provide justification for their estimates.

Ref : Delphi cost estimation model (slideshare.net) shashwat2010/delphi-cost-estimation-model

Delphi Cost Estimation Technique



Ref : Delphi cost estimation model (slideshare.net) shashwat2010/delphi-cost-estimation-model

Delphi Cost Estimation Technique

- It is possible that several rounds of estimates will not lead to a consensus estimate.
- In this case the coordinator must discuss the issues involved with each estimator to determine the reasons for the differences.
- The coordinator may have to gather additional information and present it to the estimators in order to resolve the differences.

Ref : Delphi cost estimation model (slideshare.net) shashwat2010/delphi-cost-estimation-model

Program (Project) Evaluation and Review Technique

INTRODUCTION

- PERT was developed in the late 1950 in the US the Polaris submarine missile programme. It has the potential to reduce both the time & cost require to complete a project.
- PERT shows the time taken by each component of project & the total time required for its completion .PERT breaks down the project into events, activity & lays down their proper sequence relationship & duration in the form of network.
- PERT is a technique of representing project plan in network. This is represented in a graphic from known as network diagram .

Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

Program (Project) Evaluation and Review Technique

DEFINITION

- The program (or project) evaluation and review technique, commonly abbreviated PERT, is a statistical tool, used in project management, which was designed to analyze and represent the tasks involved in completing a given project & to illustrate the flow of events in a project.

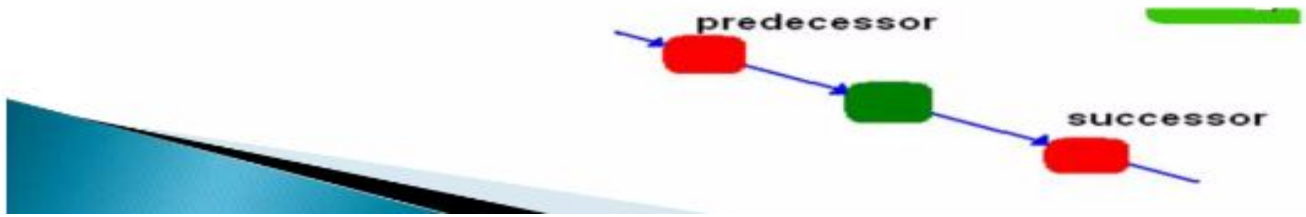
Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

Program (Project) Evaluation and Review Technique

COMPONENTS OF PERT MODEL

Every activity consume time and needs adequate resources such as manpower, material, space and machinery to change or move one event to other.

- **Predecessor event** – It is an event that precedes some other event, it can be single or multiple.
- **Successor events** – It is an event that immediately follows some other events, it can have single multiple successor events.



Seminar on pert (slideshare.net) /SantoshKumari11/seminar-on-pert

Program (Project) Evaluation and Review Technique

- **Optimistic Time (O)** – It is the minimum possible time required to complete the task anticipating that every event has occurred better than usually expected.
- **Pessimistic Time (P)** - This means the maximum possible time required to complete the given task, expecting or assuming everything goes wrong except the main catastrophes.
- **Most likely Time (M)** – The actual and the best time required to complete the task assuming everything goes in a usual way.

Seminar on pert (slideshare.net) /SantoshKumari11/seminar-on-pert

Program (Project) Evaluation and Review Technique

- **Expected time or the very best time (TH)** – The accurate or the actual time required to complete the task, it is the most reliable and valid time estimated to complete a task. It can be calculated using the following relation

$$TH = (O + 4M - P) \div 6$$

e.g. $TH = (5\text{min} + 4X\ 10\ \text{min} - 15\ \text{min}) \div 6$

$= 5\ \text{min}$

Seminar on pert (slideshare.net) /SantoshKumari11/seminar-on-pert

Program (Project) Evaluation and Review Technique

- **Float or slack time** – it is the amount of time that can be floated without causing delay in the total completion of the work.
- **Critical paths** – it is the longest possible and the actual total time required to complete the full task. It is otherwise called as total calendar time.
- **Critical total float activity** – it is an activity that has total float equal to zero. No float time is required in the critical path.

Seminar on pert (slideshare.net) /SantoshKumari11/seminar-on-pert

Program (Project) Evaluation and Review Technique

- **Lead time** – this refers to the time taken by the predecessor to complete the task. In this there is sufficient time for the subsequent activities that can follow the predecessor.

- **Lag time** – the earliest time taken by the successor event to take place, which follows the specific PERT activity .

Seminar on pert (slideshare.net) /SantoshKumari11/seminar-on-pert

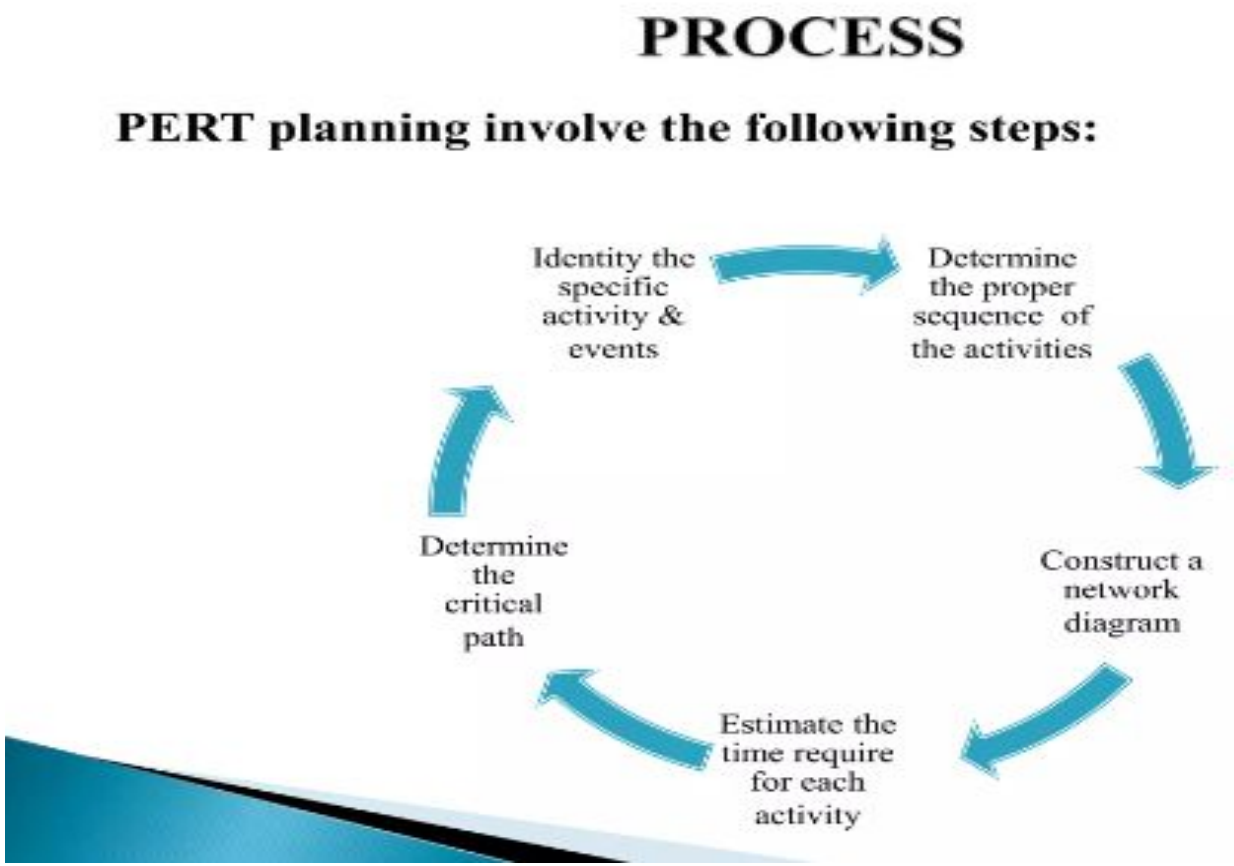
Program (Project) Evaluation and Review Technique

PURPOSES

- To schedule the project.
- To organize the project.
- To coordinate the tasks.
- To manage the time.
- To analyze the work.

Seminar on pert (slideshare.net) /SantoshKumari11/seminar-on-pert

Program (Project) Evaluation and Review Technique



Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

Program (Project) Evaluation and Review Technique

1. Identity Activities & events :

The activities are tasks require to complete the project. The events marking the beginning & end of one or more activities. It is helpful to list the tasks in a table that in later steps can be expanded to include on sequence is duration .

2.Determine the proper sequence of the activities :

This steps may be combine with the activity identification steps since the activity sequence is event for some task . Other tasks may require more analysis to determine exact order in which they must be perform .

Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

Program (Project) Evaluation and Review Technique

3.Construct a network diagram :

- Using the activity sequence information , a network diagram can be drawn showing the sequence of the serial is & parallel activities .For the original activity –on-are model, the activities are depicted by arrowed lines & milestones are depicted by circles.

4.Estimate the time required for activity :

- A distinguishing feature of PERT is its ability to deal with uncertainty in activity completion times

Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

Program (Project) Evaluation and Review Technique

For each activity , the model usually includes three time estimates;-

Optimistic time:

- Generally the shortest time in which the activity can be completed .

Most likely time :

The completion time having the highest probability note that this time is different for the expected time.

Pessimistic time :

The longest time that an activity might require.

Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

Program (Project) Evaluation and Review Technique

5. Determine the critical path :

- The critical path is determine by adding the times for the activities in each sequence & determining the longest path in project. The critical path determines the total time require.

6.Update the PERT chart as the project progresses :

- Make adjustment in the PERT chart as the project progresses. As the project unfolds ,the estimated times can be replaced with actual times. The PERT chart may be modified & improved to reflect the new situation.

Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

Program (Project) Evaluation and Review Technique

ADVANTAGES

- Simple to understand and use.
- Show whether the project is on schedule; or behind /ahead of the schedule.
- Identify the activities that need closer attention
- Determine the flexibility available with activities
- Show potential risk with activities
- Provide good documentation of the project activities
- Help to set priorities among activities & resource allocation as per priority

Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

Program (Project) Evaluation and Review Technique

APPLICATION

- Used in research and development projects
- For developing, tooling and introducing a new project
- To plan and execute the acquisition and installation of an electronic system
- Development and administration of various training programmes
- For management development and organizational planning

Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

Program (Project) Evaluation and Review Technique

LIMITATION

- A major disadvantages of PERT has been its emphasis only on time, not on costs.
- The cost of setting up such system are extensive
- It is difficult to estimate accurate time & cost of various activities involved in a project Errors in estimation makes the PERT charts ,unreliable as a control aid .
- These systems will not help managers to solve all their problem.

Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

GANTT CHART

INTRODUCTION

- During the era of scientific management ,a Gantt chart is a horizontal bar chart developed as a production control tool in 1917 by Henry L. Gantt, an American engineer & social scientist
- A Gantt chart provides a graphical illustration of a schedule that helps to plan, coordinate & track specific tasks in a project.
- The horizontal axis of the Gantt chart is a time scale , expressed either in absolute time or in relative time referenced to the beginning of the project. The resolution depends on the project.

Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

GANTT CHART

DEFINITION

- A chart in which a series of horizontal lines shows the amount of work done in certain periods of time in relation to the amount planned for those periods.

Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

GANTT CHART

NEED

- Avoid Completion Confusion
- Keep Everyone on the Same Page
- Understand Task Relationships
- Effectively Allocate Resources
- Get a Handle on the Future

Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

GANTT CHART

PROCESS

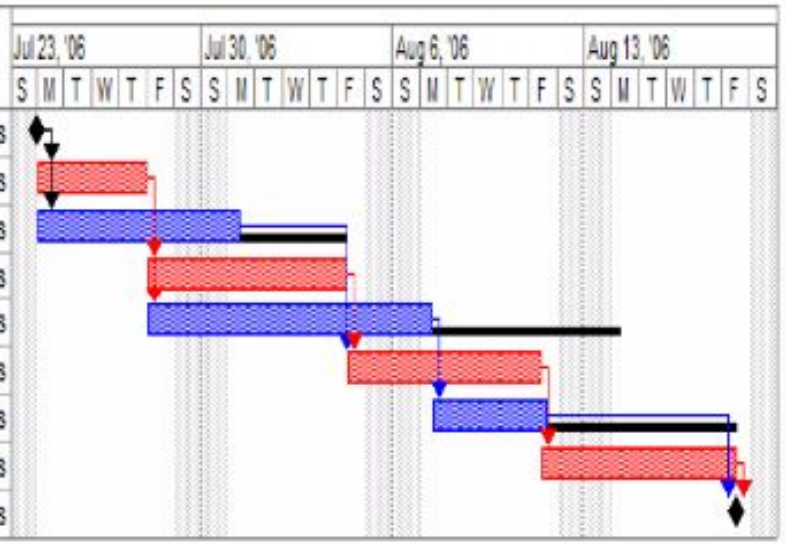
1. Identify the Purpose
2. Define the Project Timeline
3. Break the Project Down into Manageable Pieces
4. Create Progress Bars
5. Define the Critical Path
6. Add Milestone Markers

Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

GANTT CHART

MODEL GANTT CHART

ID	Task Name	Predecessors	Duration
1	Start		0 days
2	a	1	4 days
3	b	1	5.33 days
4	c	2	5.17 days
5	d	2	6.33 days
6	e	3,4	5.17 days
7	f	5	4.5 days
8	g	6	5.17 days
9	Finish	7,8	0 days



Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

GANTT CHART

ADVANTAGES

- It helps in planning and monitoring the work of project
- Time is explicitly expressed in the chart
- All tasks are visibly at a glance in relation to other
- Deadlines are depicted in the chart.

Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

GANTT CHART

LIMITATION

- Gantt charts, because of their success, form the most easy to use and the most widely used scheduling tools. But these charts are also accompanied with some limitation
- In such charts, it is very necessary to keep on updating the charts, in order to keep it in current form.
- The charts is not able to directly reveal the costs of the alternate loadings.

Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

GANTT CHART

- These charts also do not consider the varying processing times among work centers.
- Other limitations include the inability to include certain constraints like time, scope, and costs.

- These charts also do not consider the varying processing times among work centers.
- Other limitations include the inability to include certain constraints like time, scope, and costs.

Pert and gantt chart (slideshare.net) yogeshdeyogeshdengale/pert-and-gantt-chart

Introduction

- Microsoft project is a project management software product, developed and sold by Microsoft.
- It is designed to assist a project manager in developing a schedule, assigning resources to tasks, tracking progress, managing the budget, and analyzing workloads.

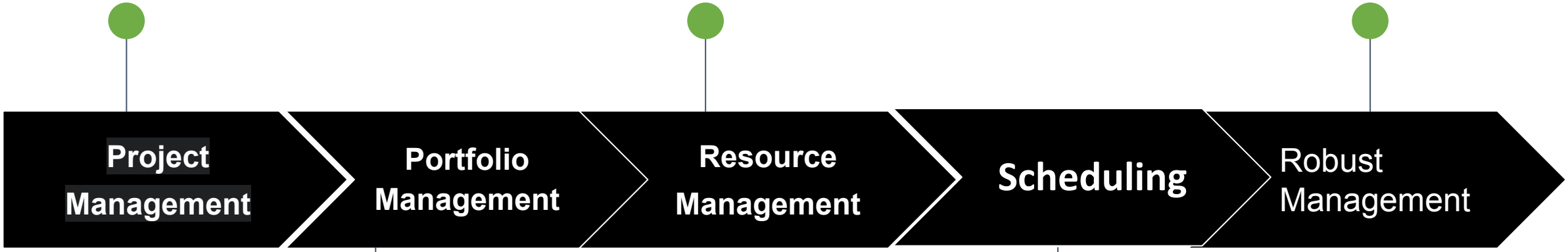
History

- The first version of Microsoft project was released for dos in 1984 by a company working for Microsoft
- Microsoft bought all rights to the software in 1985
- During the 1980's couple of versions for DOS were released
- The first windows version was released in 1990
- The latest version for windows Microsoft office project is 2021

MS Project comes with familiar scheduling tools where users can list tasks, assign them to team members, and set details such as duration and due date

MS Project lets teams forecast resource needs, predict bottlenecks early, manage utilization, and ensure timely project delivery.

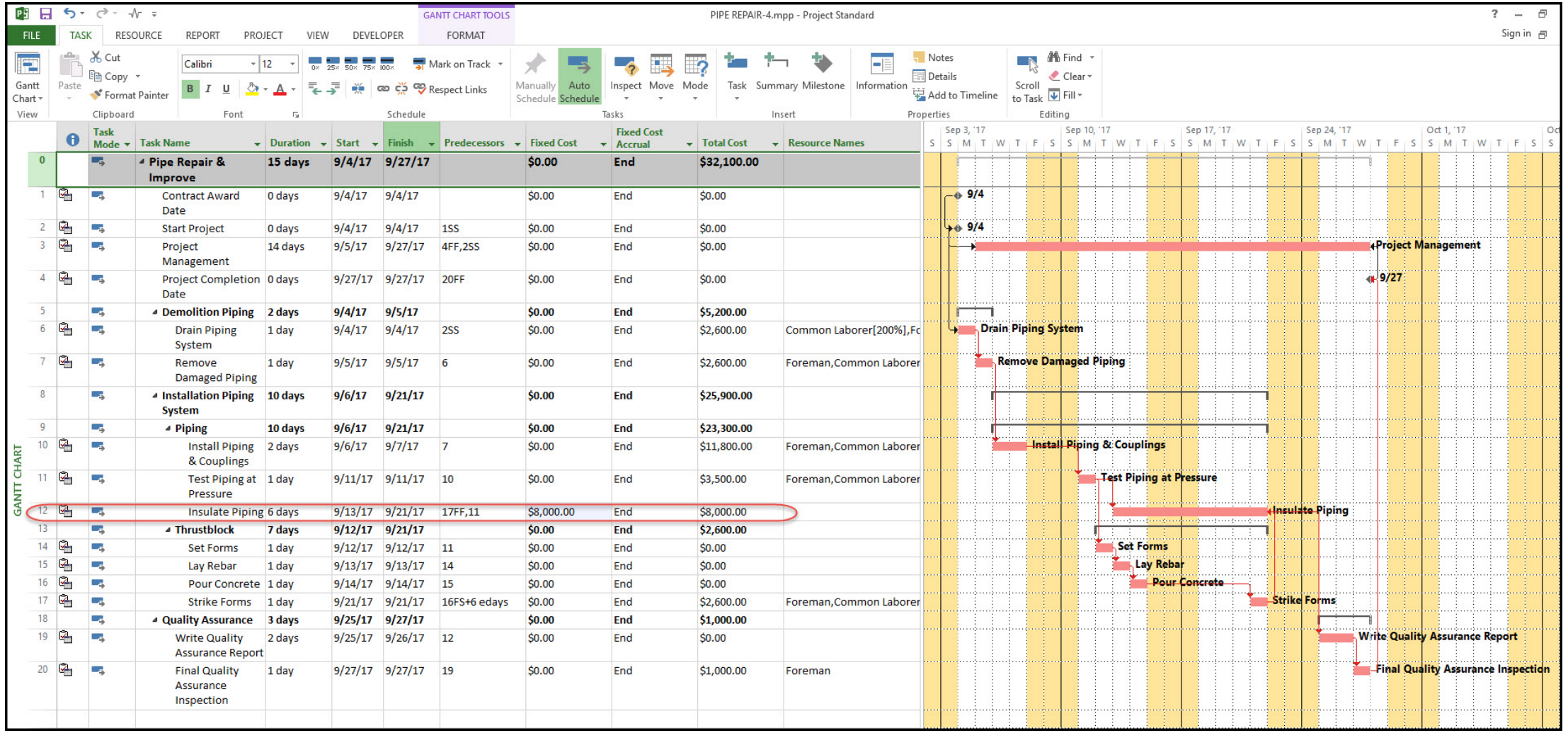
Right blend of usability , power , flexibility



MS Project allows users to manage multiple projects and programs of a company and provide information as to which initiatives should have priority

MS project is used to schedule projects

Interface of MS Project Management



	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Fixed Cost	Fixed Cost Accrual	Total Cost	Resource Names
0		**Pipe Repair & Improve**	**15 days**	**9/4/17**	**9/27/17**		**$0.00**	**End**	**$32,100.00**	
1		Contract Award Date	0 days	9/4/17	9/4/17		$0.00	End	$0.00	
2		Start Project	0 days	9/4/17	9/4/17	1SS	$0.00	End	$0.00	
3		Project Management	14 days	9/5/17	9/27/17	4FF,2SS	$0.00	End	$0.00	
4		Project Completion Date	0 days	9/27/17	9/27/17	20FF	$0.00	End	$0.00	
5		**Demolition Piping**	**2 days**	**9/4/17**	**9/5/17**		**$0.00**	**End**	**$5,200.00**	
6		Drain Piping System	1 day	9/4/17	9/4/17	2SS	$0.00	End	$2,600.00	Common Laborer[200%],Fc
7		Remove Damaged Piping	1 day	9/5/17	9/5/17	6	$0.00	End	$2,600.00	Foreman,Common Laborer
8		**Installation Piping System**	**10 days**	**9/6/17**	**9/21/17**		**$0.00**	**End**	**$25,900.00**	
9		**Piping**	**10 days**	**9/6/17**	**9/21/17**		**$0.00**	**End**	**$23,300.00**	
10		Install Piping & Couplings	2 days	9/6/17	9/7/17	7	$0.00	End	$11,800.00	Foreman,Common Laborer
11		Test Piping at Pressure	1 day	9/11/17	9/11/17	10	$0.00	End	$3,500.00	Foreman,Common Laborer
12		Insulate Piping	6 days	9/13/17	9/21/17	17FF,11	$8,000.00	End	$8,000.00	
13		**Thrustblock**	**7 days**	**9/12/17**	**9/21/17**		**$0.00**	**End**	**$2,600.00**	
14		Set Forms	1 day	9/12/17	9/12/17	11	$0.00	End	$0.00	
15		Lay Rebar	1 day	9/13/17	9/13/17	14	$0.00	End	$0.00	
16		Pour Concrete	1 day	9/14/17	9/14/17	15	$0.00	End	$0.00	
17		Strike Forms	1 day	9/21/17	9/21/17	16FS+6 edays	$0.00	End	$2,600.00	Foreman,Common Laborer
18		**Quality Assurance**	**3 days**	**9/25/17**	**9/27/17**		**$0.00**	**End**	**$1,000.00**	
19		Write Quality Assurance Report	2 days	9/25/17	9/26/17	12	$0.00	End	$0.00	
20		Final Quality Assurance Inspection	1 day	9/27/17	9/27/17	19	$0.00	End	$1,000.00	Foreman

Advantages

- **Integration**

The project management software works seamlessly with Microsoft Teams, Skype, Power BI, and Microsoft SharePoint etc

- **Dependability**

Due to its stable code base and software continually get enhanced for more relevant features

- **Customer Support**

Microsoft provides reliable support to its users

- **Flexibility**

The software is flexible enough for other purposes such as roadmapping or financial management

Disadvantages

- **Requires Training**

MS Project has become more intuitive over the years but among its cons is that it can still be overwhelming to new users.

- **Cost Constraints**

Microsoft Project management software is available as an on-premise or cloud-based solution. It can be prohibitive for small businesses to purchase many licenses for an on-premise solution

- **File Compatibility Issues**

MS Project saves files in a proprietary format, so PCs that do not use the same program cannot open them

What is Change Management?

Change management is a process of overseeing and facilitating change at any level where it occurs. It is up to management teams to decide exactly how this change will be addressed, develop the process and how to best execute and apply.

Change Management Learning Center defines change management as “the process, tools, and techniques to manage the people-side of business change to achieve the required business outcome, and to realize that business change effectively within the social infrastructure of the workplace.”

Ref: What is Change Management? - Definition and Principles | MSU Online (michiganstateuniversityonline.com)

Characteristics of Change management

- Management as a continuous process.
- Management as a Discipline.
- Management as a Career.
- Management as an applied services.
- Guidance.
- Management is a Goal Oriented.
- Management as a Leaderships.
- Management is a Human activity..

Ref: Presentation on Change Management (slideshare.net)-81547775

Who is involved in Change Management

Change management requires each of the 'gears' in the picture to fulfill their specific role. A change manager can facilitate assessments, create a change management strategy and develop change management plans, but they are not the only ones involved in managed change. The other groups involved in managing change include:

- Project team
- Senior leaders
- Managers and supervisors
- Employees

Ref: Presentation on Change Management (slideshare.net)-81547775