Second Assignment.

1.a Explain the function Point Metric of software project estimation.

Ans. Estimation of the size of the software is an essential part of software project management. It helps the project manager to further predict the effort and time which will be needed to build the project. Various measures are used in project size estimation. Some of these are:

- Lines of code.
- Number of entities in ER diagram.
- Total number of processes in detailed data flow diagram.
- Function point (FP) metric.

Function Point Metric:

This methodology is concern with the numbers of funcanalities available in the system. From this funcanality count, a metric can be calculated and hence can be used for software projects cost estimation. This approach is far more better than LOC (Lines of code) approach because at the end of the day, the feature or the funcanality matters else than the number of lines of code.

In function point metric various things are taken in consideration such as:

→ No. of external inputs: Functions related to data entering the system.

→ No. of external outputs: Functions related to data exiting the system.

→ No. of External Inquiries: They lead to data retrieval from the system but don't change the program.

→ No. of Internal Files: Functions for maintaining data and internal files.

Hence, In this way we can calculate the functional point matric for our cost estimation of software development project.

Advantages:

- It can be easily used in the early stage of project planning.
- It is independent of the programming language.
- It can be used to compare different projects even if they use different technologies (database, language, etc.)

Disadvantages:

- It is not good for real-time systems and embedded systems.
- Many cost estimation models like COCOMO uses LOC and hence FPC must be converted to LOC..

b. You are required to develop a Hotel Management system in which the estimated lines of codes (LOC) is calculated to be 76,000 and a review of the historical data reveals that the average productivity for this type of system is 250 LOC/pm and the labor rate is Rs. 6,200 per month. What would be the estimated project cost and the estimated effort for this software package?

Ans. Here;

Total line of code (LOC) = 76,000.

Average productivity = 250 LOC/pm.

Labor rate = Rs. 6,200 pm.

Now,

$$\text{Estimated effort} = \frac{\text{LOC}}{\text{Avg. Prod}} = \frac{76000}{250} = 304 \text{ person month.}$$

Estimated project cost = LOC × $\left(\frac{\text{labor rate}}{\text{Avg. productivity}}\right)$

$= 76000 \times \left(\frac{6200}{250}\right)$

$= Rs.\ 23104000.$

2. a. What do you mean by "Constructive Cost Model" in software project planning?

Ans. CoCoMo (Constructive Cost Model) is a regression model based on LOC, i.e. number of Lines of code. It is a procedural cost estimate model for software projects and is often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. It was proposed by Barry Boehm in 1981 and is based on the study of 63 projects.

The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort and schedule.

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

In CoCoMo, Boehm had categorized software model in organic, semi-detached and embedded system.

$14 \times 3 = 42$ (Avg. value)

↓ param

b. Given the data below, compute the function point value for a project with the following information domain characteristics.

Number of user inputs : 56

Number of user outputs : 58

Number of user inquiries : 35

Number of files : 15.

Number of external interfaces : 8.

Assume that all complexity adjustment values are average. The organizational average productivity for systems of this type is 6.5 FP/pm and labor rate of Rs. 8000 per month. Calculate the total cost required and effort required.

Ans. lets plot given info in the table.

Parameter	Count	Weighing factor (Avg.)	Function point count
No. of user inputs	56	4	224
No. of user outputs	58	5	290
No. of user inquiries	35.	4	140
No. of files	15.	10	150
No. of external interface	8.	7	56.
			Total = 860.

Now, $F.P = \underset{(Raw\ f.p)}{\text{Total count}} * [0.65 + 0.01 * CAV(\Sigma Fi)]$

$= 860 * [0.65 + 0.01 * (3 * 14)]$

$= 860 * [1.07]$

$= 920.2$

Effort Required = $\frac{\text{F.P}}{\text{Avg. Prod.}} = \frac{920.2}{6.5} = 141.57$ person month.

Total project cost = F.P $*$ $\left(\frac{\text{labor rate}}{\text{Avg. Productivity}}\right)$

$= 920.2 * \left(\frac{8000}{6.5}\right)$

$= \text{Rs. } 1132553.85.$

3. a. What are the software risks? explain different categories of risk associated with software projects.

Ans. Risks are potential problems that might affect the successful completion of a software project. Risk is a potential problem it might happen or might not. But the regardless of the outcome it is good idea to identify it assess its probability of occurance, estimate its impact and establish contingency plan.

There are various categories of risk associated with software projects. and some of them are:

a. Project risks:

Threatens the project plan i.e. if project risks become real it is likely that the project schedule will slip and costs will increases.

Risk factors:

→ potential budgetary.

→ Schedule.
→ Personnel (staffing and organization).
→ Resource.
→ Stakeholder.
→ Requirements problem.

b. Technical risks:
- Threaten the quality and timeliness of the software to be produced.
- If a technical risks becomes a reality implementation may become difficult or impossible.
- Risk factors.
  → Potential design.
  → implementation.
  → interface.
  → verification. etc.

c. Business risks:
- Threaten the viability of the software to be produced and even the project or the product.
- Top five business risk are:
  → Market risk: building an excellent product that no one really wants.
  → Strategic risk: building a product that no longer fits into the overall business strategy for company.
  → Sales risk: building a product that the sales force doesn't understand how to sell.
  → Management risk: losing the support of senior management due to change in focus or a change in people.
  → Budget risk: losing budgetary or personnel commitment.

d. known risk.

- Those that can be uncovered after careful evaluation of the project plan, the business and technical environment in which the project is being developed, and other reliable information source.

- e.g. unrealistic delivery date, lack of documented requirements, poor development environment.

These are the risks which can occur for software development for the company itself.

3.b. Why is it necessary to do software project planning? "If you do not actively attack risk, the risk will attack you". justify. Develop a risk information sheet considering any risk that is identified.

Ans. Yes it is very necessary to do software project planning Because without proper software planning a software can't be made in totality. In tomorrow sense, the software might face lots of problems and risks because of lack of software planning.

The statement, "If you do not actively attack risk, the risk will attack you" is a very true statement. For any software project to work efficiently over the time and in any situation, the risk factors should always be find out and take possible measures before time. If the risks are not found/detected in the development time; one can always be ready for the future if possible. It is universal truth that no-one can possibly will have any risk in the software. There will be always and will remain always. So, as a system designer/developer or as a system tester, one must actively identify risk and work in healing it.

2nd part:

Let us consider the following scenario, for which we have to create Risk Information sheet.

"The project is under way and number of people announce that they are leaving if the mitigation step is followed, backup is available, information is documented and knowledge been has dispersed across the team. In addition temporarily focus resource and schedule (to readjust) to those function that are fully staffed, enabling new comers who must be added to team to get up speed."

Risk Information sheet.			
Risk ID: P02-4-32	Date: 5/22/22	Prob: 80%.	Impact: High.

Description:

Only 70 percent of the software components scheduled for reuse will, in fact, be integrated into the application. The remaining functionality will have to be custom developed.

Refinment / Context:

Subcondition 1: Certain Reusable Components were developed by a third party with no knowledge of internal design standards.

Subcondition 2: The design standard for component interfaces has not been solidified and many not conform to certain existing reusable components.

Subcondition 3: Certain reusable components have been implemented in a language that is not supported on the target environment.

Mitigation/Monitoring:

1. Contact third party to determine conformance with design standards.
2. Press for interface standards completion; consider component structure when deciding to interface protocol.
3. Check to determine number of components in subroutine 3 category; check to determine if language support can be acquired.

Management/contingency plan/trigger:

RE computed to be $20,200. Allocate this amount within project contingency cost. Develop revised schedule assuming that 18 additional components will have to be custom built; allocate staff accordingly.
Trigger: Mitigation steps unproductive as of 7/25/22.

Current status:

8/1/22: Mitigation steps initiated.

4. a. Assume that software team defines a project risk in as follows:
RISK Identification: Only 70% of the software components scheduled for reuse will, in fact, be integrated into the application. The remaining funcanality will have to be custom developed.
Risk Probability: 80% (likely).
Risk Impact: 60 reusable software components were planned. If only 70% can be used, 18 components would ~~be~~ have to be developed from scratch (in addition to other custom software that has been scheduled for development). Since the average component is 100LOC.

and local data indicate that the software engineering cost for each LOC is $14.00. Find risk exposure.

Ans. Probability of Risk (P) = 80 % = 0.8

Cost for all 18 components (C) = 18 x 100 x 14

= 25200.

∴ Risk Exposure (E) = P x C

= 0.8 x 25200

= 20160.

b. Suppose that project was estimate to be 400kLOC. calculate effort and time for each of 3 models; Organic, semi-detached and embedded.

Ans.

we have,

Effort	a_1	a_2
Organic	2.4	1.05
semidetached	3.0	1.12
Embedded	3.6	1.20

Effort	b_1	b_2
Organic	2.5	0.38
semidetached	2.5	0.35
embedded	2.5	0.32

KLOC = 400

For Organic:

Effort = $a_1 (KLOC)^{a_2}$ person month

= $2.4 \times (400)^{1.05}$

= 1295.31 ≈ 1295

Time = b_1 (Effort)^{b_2} months.

= 2.5 × (1295.31)$^{0.38}$

= 38.07 months.

For Semi-detached:

Effort = a_1 (KLOC)^{a_2}

= 3 × (400)$^{1.12}$

= 2462.79

≈ 2463 person month.

Time = b_1 (Effort)^{b_2}

= 2.5 × (2463)$^{0.35}$

= 38.45 months.

For Embedded model:

Effort = a_1 (KLOC)^{a_2}

= 3.6 (400)$^{1.20}$

= 4772.81

≈ 4773 person month.

Time = b_1 (Effort)^{b_2}

= 2.5 × (4773)$^{0.32}$

= 2.5 × 15.04

= 37.59 months.

5.a. Given the data below, compute the function point value, productivity, documentation, cost per function for the project with the following information domain characteristics.

Number of user inputs : 24.

Number of user outputs : 46.

Number of user inquiries : 8.

Number of files : 4

Number of external interfaces : 2

and, Effort = 36.9 PM, Technical document = 265 pages, user document = 122 pages, cost = $7744 per month.

Complexity adjustment values are 4, 1, 0, 3, 3, 5, 4, 4, 3, 3, 3, 3, 4, 5.

Ans. Let's plot the given data in the table:

Parameters	Count	Weighing factor (Avg).	Function point Count.
No. of user inputs	24	4	96
No. of user output	46	5	230
No. of user inquiries	8	4	32
No. of files	4	10	40
No. of external interfaces	2	4	14
			Raw F.P = 412.

Now, F.P = Raw F.P * [0.65 + 0.01 * CA (ΣFi)]

= 412 * [0.65 + 0.01 * (4+1+0+3+3+5+4+4+3+3+3+3+4+5)]

= 412 * [0.65 + 0.01 * 45]

= 412 * 1.1

∴ F.P = 453.2.

since

→ Effort = F.P / Avg. Productivity

or, 36.9 = 453.2 / Avg. Productivity

∴ Avg. Productivity = 12.28

Total page of Documentation = Technical document + User documents.

~~(or Documentation)~~ (PD) = 265 + 122

= 387 pages

Documentation (D) = PD / FP.

= 387 / 453.2

= ~~175388.4~~ = 0.85.

~~≈175388 pages.~~

Cost of each funcanality = Cost × productivity

= 7744 × 12.28

= $95096.32.

7. Write short notes on (Any Two):

a. Software quality metrics:

Software quality metrics are a subset of software metrics that focuses on the quality aspects of the product, process and project. These are more closely associated with process and product metrics than with project metrics.

Software quality metrics can be further divided into three categories:

→ Product quality metrics:

→ In-process quality metrics.

→ Maintenance quality metrics

On the different factor which the software quality assurance depends is not only the thing, SQA are to be continously applied throughout the software process. In more further categories, there are things which we can find out their metric to ensure the quality of software and they are:

- Code quality.
- Reliability.
- Performance.
- Usability.
- Correctness.
- Maintainability.
- Integrity.
- Security etc.

b. Metrics for small organization:

For small organizations, the only thing that matters is the time for the project and the final output of the product with quality. In small organization they have small number of employees so it is very difficult to follow the metrics such as LOC (line of code) or Function point, as they have low budget for most of the projects. So, they are mainly concerned for final output rather than those metrics. whereas for the quality of the product is determined by the number of defects over the time. So, the less number of defects, the more is the quality of the product.

Hence for a small organization those things do matter, their always main concern remain for the final output rather than calculating other metrices.