Solver: Jesslyn Chew

1)

a) **Noisy, crowded offices**. Workers who occupy quiet private offices tend to perform significantly better as there is less distraction when completing work. Hence, a noisy, crowded office reduces work productivity and lengthen development schedule.

b) Some major tasks include ①tracking the work against the project plan throughout the project execution phase, ②conduct reviews on risks, and ③participate in team meetings.

c) To measure the quality of a software product, the quality criteria must be identified. This quality criteria is based on the user requirements and must be measurable. Moreover, the quality assessment process must be defined to review the quality of the product. For example, the developer may want to prioritize on portability. The corresponding internal attributes of portability is number of procedure parameters and programme size in lines of code. These attributes enable the developer to quantify the software quality. By setting a limit to the number of procedure parameters and programme size in lines of code, the developer can control the quality of portability.

d) When a risk is found in the "risk monitoring" stage, and the risk is not on the list of risks registered, the project manager will go back to the risk analysis stage. He/she will assess the likelihood and consequences of the detected risk. The risk will be assigned a probability and risk effect level. From that, he/she will draw up plans to avoid or minimize the effects of the risks and carry out those plans. An avoidance strategy, minimization strategy or contingency plans can be considered to manage the risk. Afterward, he/she will continue to monitor the risks throughout the project.

e) The skills of the development team should be recognized and exploited. Team members should be left to develop their own ways of working without prescriptive processes.

2)

a)

i) ① Log in, stored locally in the system – 1 inquiry
② Register, stored locally – 1 input + 1 internal file
③ Display help information based on user's request – 1 inquiry
④ Browse doctors' information – 1 inquiry
⑤ book an appointment, stored locally – 1 input + 1 internal file
⑥ view, change or cancel appointment – 1 inquiry + 2 input + 2 internal files
⑦ send a reminding message – 2 output + 2 interface
⑧ print a report – 1 output
We assume that ⑤ and ⑦ are of medium complexity, while the rest are of low complexity.

ii)

Characteristics	Low Complexity	Medium Complexity	High Complexity
# Inputs	$3 \times 3 = 9$	$1 \times 4 = 4$	$0 \times 6 = 0$
# Outputs	$1 \times 4 = 4$	$2 \times 5 = 10$	$0 \times 7 = 0$
# Inquiries	$4 \times 3 = 12$	$0 \times 4 = 0$	$0 \times 6 = 0$
# Internal Files	$3 \times 7 = 21$	$1 \times 10 = 10$	$0 \times 15 = 0$
# External Interfaces	$0 \times 5 = 0$	$2 \times 7 = 14$	$0 \times 10 = 0$
TOTAL:	46	38	0

Total Unadjusted Points = 84

iii) Little transaction rate: 1
Little on-line data entry: 1
High end user efficiency: 4
Average on-line update: 3
High operational ease: 4
High reusability: 4
Low processing complexity: 2
Total score = 1 + 1 + 4 + 3 + 4 + 4 + 2 = 19
Influence Multiplier = (Total score) * 0.01 + 0.65
= 19 * 0.01 + 0.65
= 0.84
Total Adjusted FPs = Unadjusted FP * Influence multiplier
= 84 * 0.84
= **70.56**

iv) 1 FP = 53 lines of code
lines of code = 53 * 70.56 = 3739.68
As the development team has many years of experience in the similar system development, the team has high experience and therefore the software project mode is **Organic**.
E = a * $(KLOC)^b$
= 2.4 * $(3.73968)^{1.05}$
= 9.5871
D = c * E^d
= 2.5 * $(9.5871)^{0.38}$
= 5.9018
Team size = E / D
= 9.5871 / 5.9018

= 1.62
≈ **2**

b) Initial budget = $5,000
Critical Path = ABCFG
Activity G [Crash week available = 1] has the lowest crash cost ($700) that is in the critical path and can reduce the # week by 1.

- No Change in critical path
- Budget left = $5,000 - $700 = $ 4,300
- Crash weeks for G is now zero

Activity C [Crash week available = 1] has the lowest crash cost ($800) that is in the critical path and can reduce the # week by 1.

- No change in critical path
- Budget left = $4,300 - $800 = $3,500
- Crash week for C is now zero

Activity F [Crash week available = 1] has the lowest crash cost ($900) that is in the critical path and can reduce the # week by 1.

- Critical path: ABCFG and ABEG
- Budget left = $3,500 - $900 = $2,600
- Crash week for F is now zero

Activity B [Crash week available = 1] has the lowest crash cost ($1200) that is in the critical path and can reduce the # week by 1.

- Critical path: ABCFG and ABEG and ADFG
- Budget left = $2,600 – 1,200 = $1,400
- Crash week for B is now zero

No possible reduction is the weeks. Hence, **4 weeks** can be reduced with an additional budget $5,000.

3)

a)

i) Enhancement, specifically adaptive maintenance

ii) In order to determine how to evolve the system, we need to identify the business value and the quality of the system so that we can take the appropriate strategies. Hence, we can investigate the defect rate and mean time to change of the system. In this situation, the defect rate can be the percentage the system takes more than x seconds to serve a

customer, where x is determined to be too long.

iii) From the information provided, we can establish that the business value is high since the system is still required. However, the quality is low as it is mentioned that it is very slow and unable to meet the current demands. The age of the system may also cause the system to be harder to be maintained and have a higher mean time to change. Hence, with high business value and low quality, the possible options are re-engineering or replacing the current system.

b)

i) Git

ii) COMMIT: Firstly, you have to update the changes in your local repository.
PUSH: Once the changes are updated in the local repository, the "push" function will be able to update the changes to the central repository.
LOG: This function shows the commits made. The developer can specify to display the latest commit, to double check the current revision number.

iii) [Note: Should do "git add ." to include all the changes made into the commit]
git commit -m "message"
git push
git log -1

4)

a)

i) Capability Maturity Model Integration

ii) This model helps to gain closer estimation of actual cost and schedule. Moreover, moving up the levels of the CMMI can result in productivity improvements of 100-200% and reduce post-release defects by 10-94%. The software processes will be more structured and clearly defined in terms of roles and responsibilities. It will also be easier for managers to monitor the quality and analyze problems.

iii) Given that it is a start-up, let's assume that the maturity level is 1. Hence, we need to move it up to level 2 (Repeatable). The following key process areas need to be defined: Software configuration management, software quality assurance, software project tracking & oversight, software subcontract management, software project planning, and requirements management. For each of the key process areas, the startup must describe in detail and identify the goals to achieve. Each of them is organized by common features that address the implementation and contains key practices that describes activities to achieve the goal. Hence, fulfilling the key process areas and attaining level 2.

b)

i) [*Not too confident in this section.*]
Let the scenario be that there are 1 module that receives user input through a graphical user interface and another module that encodes information and stores it in the database. Let the encode function just be a +1 to the ASCII value.

Module A:
GET username from input
GET password from input
ENCODE password
RETURN username and password (To Module B via API)

Module B:
START database server
CALL Module A
OBTAIN input from Module A via API
ENCODE password
STORE key and encoded value into database

ii) The test case can be based on an input entered in Module A (e.g. username: "name", password: "pw"), the expected result should be a new entry in the database that has username as name and password as "qx".

iii) Let the regression test be an end-to-end test of storing data into the database. Before the module was integrated, Module B obtain the input values to store into the database via command line by the user. After integrating with Module A, the regression testing will identify a bug. Based on the input from Module A, the system did not get expected value that is stored in the database. This is because Module A has also included a encode function, which is a redundant feature in the system.

--End of Answers--

Solver: Jesslyn Chew

Tips

Function Points

- Login – 1 inquiry
- Registration – 1 input, 1 logical file
- Data retrieval – 1 inquiry
- Data update & delete – 2 input, 2 logical files
- Print report – 1 output
- Input: includes messages or transaction files from other application/modules
- If an inquiry is done by 3 different user types but all use the same page/feature, it is **1 inquiry** and not x3 inquiry

Crash Cost

- The activity(s) must be in every critical path. If it is only found in one of the critical paths when there are multiple present, cutting that activity will not guarantee a reduction in week(s) for the development. This means either 1 activity in all critical path, or a combination of activities where in every critical path there is at least 1 activity present from the combination.
- You may have to consider comparing cost against a combination of activity. For example, let the critical path be AC and BC.

Activity	**Crash cost per week**
A	$300
B	$400
C	$1000

The possible choices are crash weeks using activity A+B or crash weeks using activity C. Since A+B: $700 and C: $1000, the best choice is activity A+B as the cost is lower.

Version Control System

- Try to remember some of the commands for the VCS, and the differences between different VCS.
- CVS (Concurrent Versions System), Git, SVN (Subversion)