20th CSEC – Past Year Paper Solution *2016-2017 Sem 1*
CZ 3002 – Advanced Software Engineering

Solver: Yong Hui Hui

1)

a) Friction between developers and customers can arise in several ways. Customers may feel that developers are not cooperative when they refuse to sign up for the development schedule that the customers want, or when they fail to deliver on their promises. Developers may feel that customers unreasonably insisting on unrealistic schedules or requirements changes after requirements have been baselined. There might simply be personality conflicts between the two groups.
To avoid this mistake, customers should provide as much detail as they can about their project requirements. Developers also have the responsibility to ask several questions for each of the requirements and to provide suggestion for bad requirements.

b) **Quality Assurance** establishes the standards and procedure that are required for an organization to follow.
Quality Control ensures and verifies that these procedures and standards are followed by the software development team.
Quality Planning selects applicable procedures and standards for a particular project and modify these as required.

c) **Length of user manual**: This is a measure of the size of a user manual. Generally, the longer the length of a user manual, the more complex and difficult it is to understand.
Fog index: This is a measure of the average length of words and sentences in documents. The higher the value for the Fog index, the more difficult it is to understand the document.

d) ① Reorganize team so that there are more overlap of work and people therefore understand each other's jobs.
② Consider adding staff or temporary worker to take up the ill employee's task.
③ Consider bringing in someone else within the company to cover the ill employee.

2)

a)

i) Functional requirements:
1: Sets current count of cars
2: Sets maximum number of cars
3: Receives notification from the sensor
4: Examines current number of cars
5: Notifies the barrier after examining the count of cars
6: Generates daily report at the end of each day
7: Generates monthly report at the end of each month

ii) [Vetter's note: I believe that there are quite a few missing details in the following table, but that might be due to the different but not incorrect interpretation. For 1 and 2, "set" should include logical files as the data is modified in the system. For 3, when a notification is received, it should also include input characteristic. While 5 should include output characteristic and not just interface. USUALLY when there is an external interface is involved, it is paired with input/output.]

Characteristics	**Low**	**Medium**	**High**
Inputs	1, 2		
Outputs		6	7
Inquiries	4		
Logical Files			
Interfaces		3, 5	

iii)

Characteristics	**Low Complexity**	**Medium Complexity**	**High Complexity**
# Inputs	2 x 3 = 6	0 x 4 = 0	0 x 6 = 0
# Outputs	0 x 4 = 0	1 x 5 = 5	1 x 7 = 7
# Inquiries	1 x 3 = 3	0 x 4 = 0	0 x 6 = 0
# Internal Files	0 x 7 = 0	0 x 10 = 0	0 x 15 = 0
# External Interfaces	0 x 5 = 0	2 x 7 = 14	0 x 10 = 0
	9	19	7

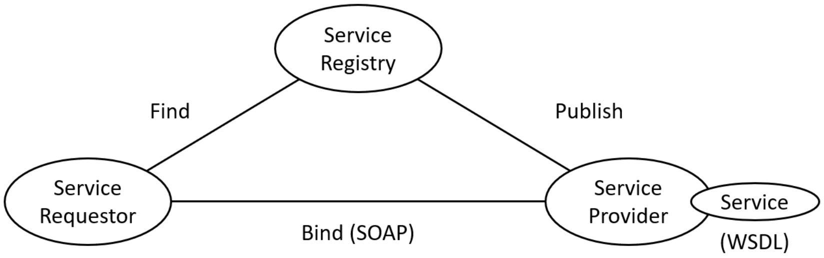
Total Unadjusted FP = 9 + 19 + 7 = **35**

b) Initial critical path: ABCEFJ (30)

Activity	**Weeks Gained**	**Total Cost of Reductions ($)**	**Critical path**
B	1	300	ABCEFJ (29)
C	1	300 + 600 = 900	ABCEFJ, ABDEFJ, AGHFJ (28)
F	1	300 + 600 + 1500 = 2400	ABCEFJ, ABDEFJ, AGHFJ, AGHIJ (27)
Total weeks can be reduced = 3			

3)

a) Web services technology is a collection of XML technology standards that work together to provide Web Service capabilities. With the implementation of SOA, there are 3 well defined interfaces: Service Provider, Service registry, and Service. Service provider creates web services and provides information to the registry, service registry makes web service information available to the requestor, whereas service requestor locates entries in the registry and invoke the service needed. Such implementation ensures maintainability as it provides service that are loose coupling and reusable.



b)

i) The quality of the software product of an organization is largely determined by the quality of the process used to develop and maintain it. With the use of integrated process meta-model, process capabilities and maturities of a software organization can be improved as it guides organizations through software process improvement, identifies their critical problems and establishes improvement priorities.

ii)

Name	Development Organization	Distinguishing Features
CMMI-DEV v1.3 (Staged): Maturity Level 5 (Optimizing)	Accenture Technology Solutions	Focuses on continually improving process performance. Key process areas include organizational innovation and deployment and causal analysis and resolution.
CMMI-DEV v1.3 (Staged): Maturity Level 3 (Defined)	Singapore-Housing Development Board (HDB)	Focuses on process standardization. Key process areas include requirements development, verification, validation, organizational process focus, process definition etc.
CMMI-DEV v1.3 (Staged): Maturity Level 2 (Managed)	Thales Solutions Asia	Focuses on basic project management. Key process areas include requirements management, project planning, configuration management etc.

c) (Not sure about this part)
According to Lehman's Law, as times change, a program that is used in a real-world environment necessarily must change or become progressively less useful in that environment. Also, as an evolving program changes, its structure tends to become more complex. Extra resources must be devoted to preserving and simplifying the structure. Besides, the quality of systems will appear to be declining unless they are adapted to changes in their operational environment. Furthermore, over a program's lifetime, its rate of development is approximately constant and independent of the resources devoted to system development.
Software systems is often a critical business asset of an organization, to prevent failure of software, organizations need to have a long view of the life of their software and its responsiveness to business. They should act on the implicit advice of the laws stated above and do work to maintain their system and its business value.

4)

a) Software Testing is an activity performed for evaluating product quality and for improving it, by identifying defects and problems. It consists of dynamic verification of the programme behavior on a finite set of test cases, suitably selected from the usually infinite executions domain, against the expected behavior.
The objectives of software testing are to uncover as many as errors as possible in a given timeline, to demonstrate that a given software product matches its requirement. specifications, to validate the quality of a software testing using the minimum cost and effort, to generate high quality test cases, perform effective tests, and issue correct and helpful problem reports.

b) Below are the 6 phases of SDLC:
(1) Requirement Analysis – In this phase, all requirements of the system to be developed are captured and documented in a requirement specification document. A test planning is created after the requirements analysis is completed.
(2) System Design – In this phase, system design is prepared to define the overall system

architecture. Also, the testing strategy which mentions what to test and how to test is also defined in this phase.
(3) Implementation – With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
(4) Testing – All the units developed in the implementation phase are integrated into a system. In this phase, all types of functional testing like integration testing, system testing and acceptance testing as well as non-functional testing are carried out.
(5) Deployment – Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.
(6) Maintenance – In this phase, maintenance is done to enhance the product quality. Regression Testing can be carried out to see if anything will be affected by a change.

c) 1. Git

- Clone an existing Git repo: git clone “URL”
- Add files in the local repo: git add “FILE_NAME”
- Commit the tracked changed to a remote repo: git commit -m “COMMENT”

2. SVN

- Pull a request from remote repo: svn checkout “URL”
- Commit the tracked changes to a remote repo: svn commit
- Sync local repo with the remote repo: svn update

3. CVS

- Pull a request from remote repo: cvs checkout “URL”
- Add files in the local repo: cvs add “FILE_NAME”
- Commit the tracked changes to a remote repo: cvs commit

--End of Answers--