Tutorial 5: 6Qs

1. The Table gives the activities details identified in a software project. (i) Create a precedence activity network using the Table. (ii) Calculate the earliest and latest start and end dates,and the float associated with each activity. (iii) Identify the critical path.

2. Given the state transition diagram of a dual identical-components system for air traffic control. Suppose the mean-time-to-failure of either system is 8816 hours. Compute the availability of such a system. State your assumption.

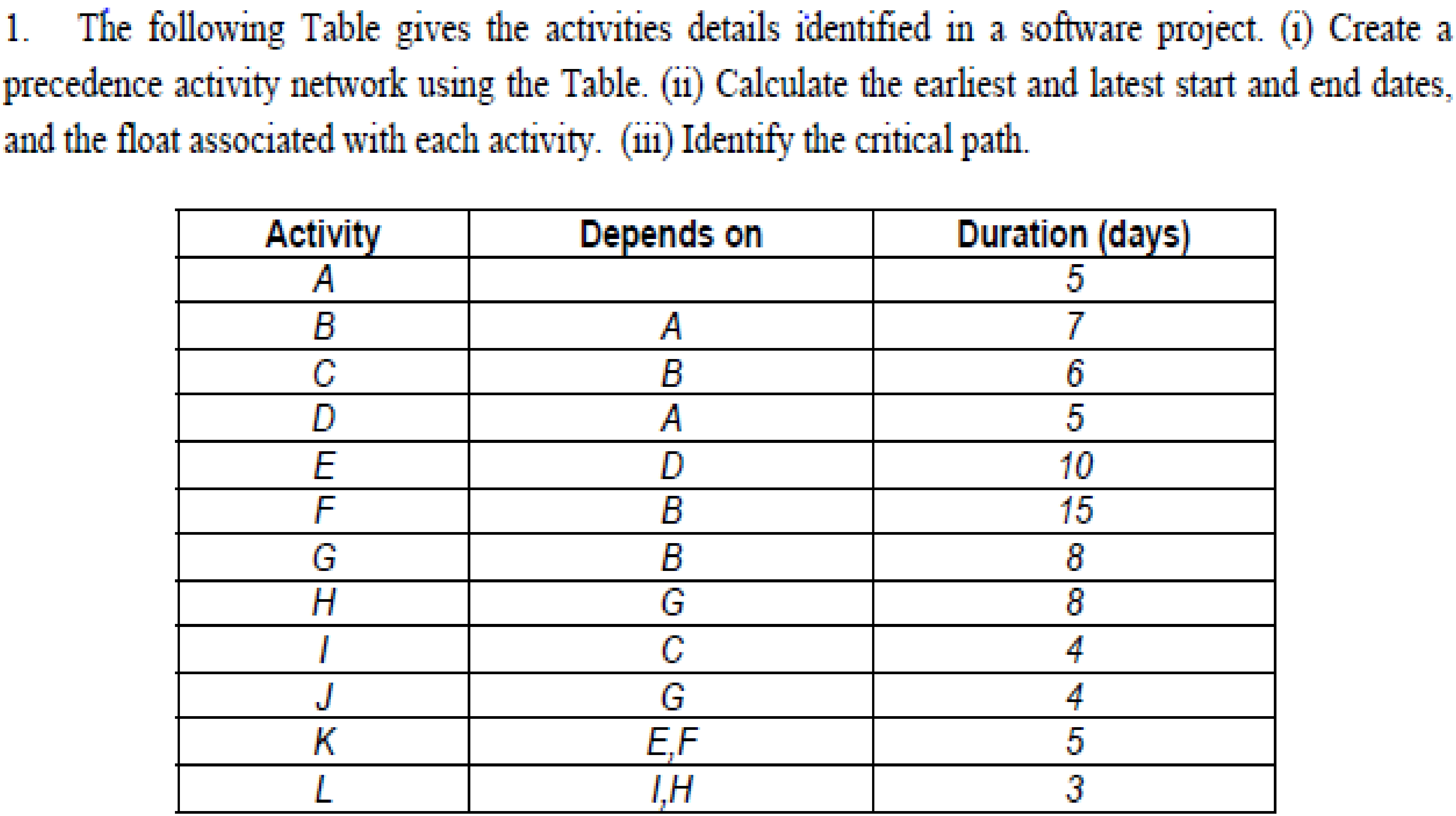
3. What is risk analysis and management in the context of software engineering? Describe key elements of the risk management.

4. Discuss the various options for using Reusable Software Resources, and briefly explain the risk level for each of the options.

5. Suppose the information below shows the financial status of a 20 weeks project at the end of the 12th week: Earned Value = S$25, 000 and Actual Cost = S$ 30,000. The budget at completion is S$50,000. Compute (i) the Planned Value as of the reporting date; (ii) the Cost Performance Index (CPI).

6. Describe a few risk reduction techniques for software project.

Tutorial 5: Q1



1. The following Table gives the activities details identified in a software project. (i) Create a precedence activity network using the Table. (ii) Calculate the earliest and latest start and end dates, and the float associated with each activity. (iii) Identify the critical path.

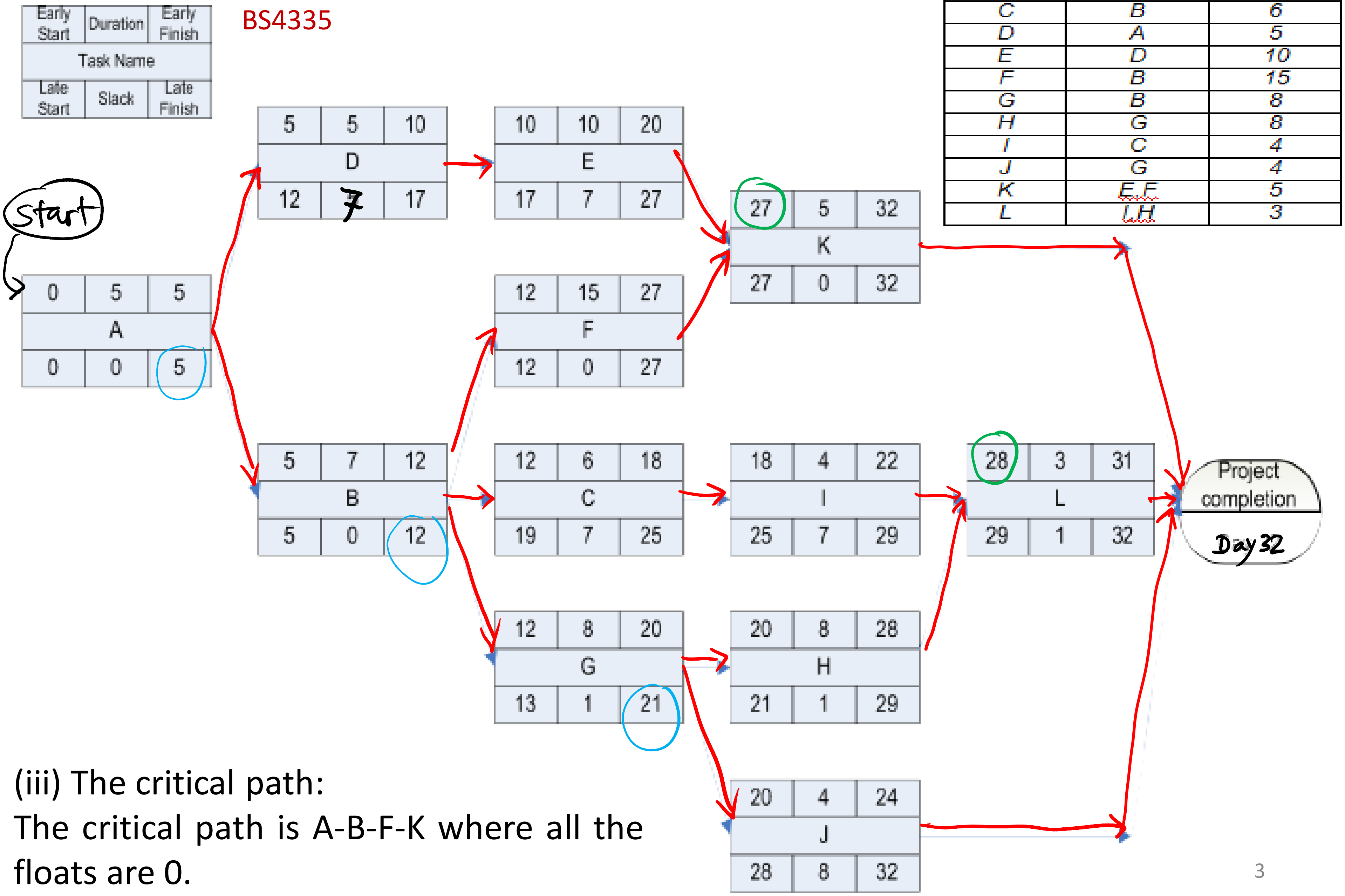
Activity	Depends on	Duration (days)
A		5
B	A	7
C	B	6
D	A	5
E	D	10
F	B	15
G	B	8
H	G	8
I	C	4
J	G	4
K	E,F	5
L	I,H	3

T5_Q1_ans:

(i)& (ii) The activity network and start/end/float:

BS4335

Activity	Depends on	Duration (days)
A		5
B	A	7
C	B	6
D	A	5
E	D	10
F	B	15
G	B	8
H	G	8
I	C	4
J	G	4
K	E,F	5
L	I,H	3

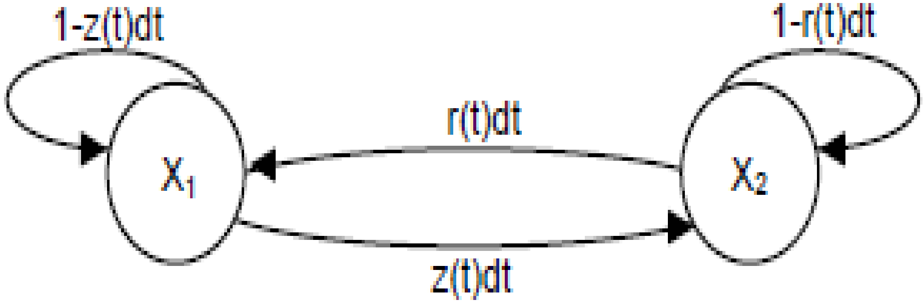


(iii) The critical path:

The critical path is A-B-F-K where all the floats are 0.

Tutorial 5_Q2

2. The following figure shows the state transition diagram of a dual identical-components system (X_1 and X_2) for air traffic control. One of the systems runs as front-end and the other is redundant. In times of failure of the front-end system, the redundant system will have to take over the control. Due to reliability requirement, a maximum of five seconds is all that is needed for the switchover. Given the mean-time-to-failure of either system is 8816 hours. Compute the availability of such a system. State your assumption.



r(t) = Repair hazard
z(t) = Failure hazard

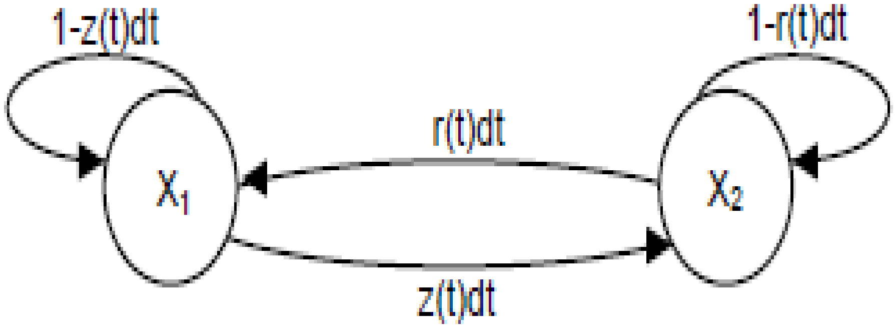
Software availability is the probability that a program is operating according to requirements at a given point in time and is defined as

Availability = [MTTF/(MTTF + MTTR)] x 100%

T5_Q2_ans: -*Availability* is the probability that the system is operational at any point in time:

Given: MeanTimeToFailure=8816 hr; MTTR=5 s.

Availability = (MTTF) / (MTTF + MTTR)



Mean time to repair = 5 / (60 x 60) hours

r(t) = Repair hazard
z(t) = Failure hazard

Availability = 8816 / (8816 + (5 / 60 x 60))
= 0.999998

Assumption: The redundant system always succeeds in gaining over control without losing any information/data.

T5_Q3: what is RM & key Elements of RM ?

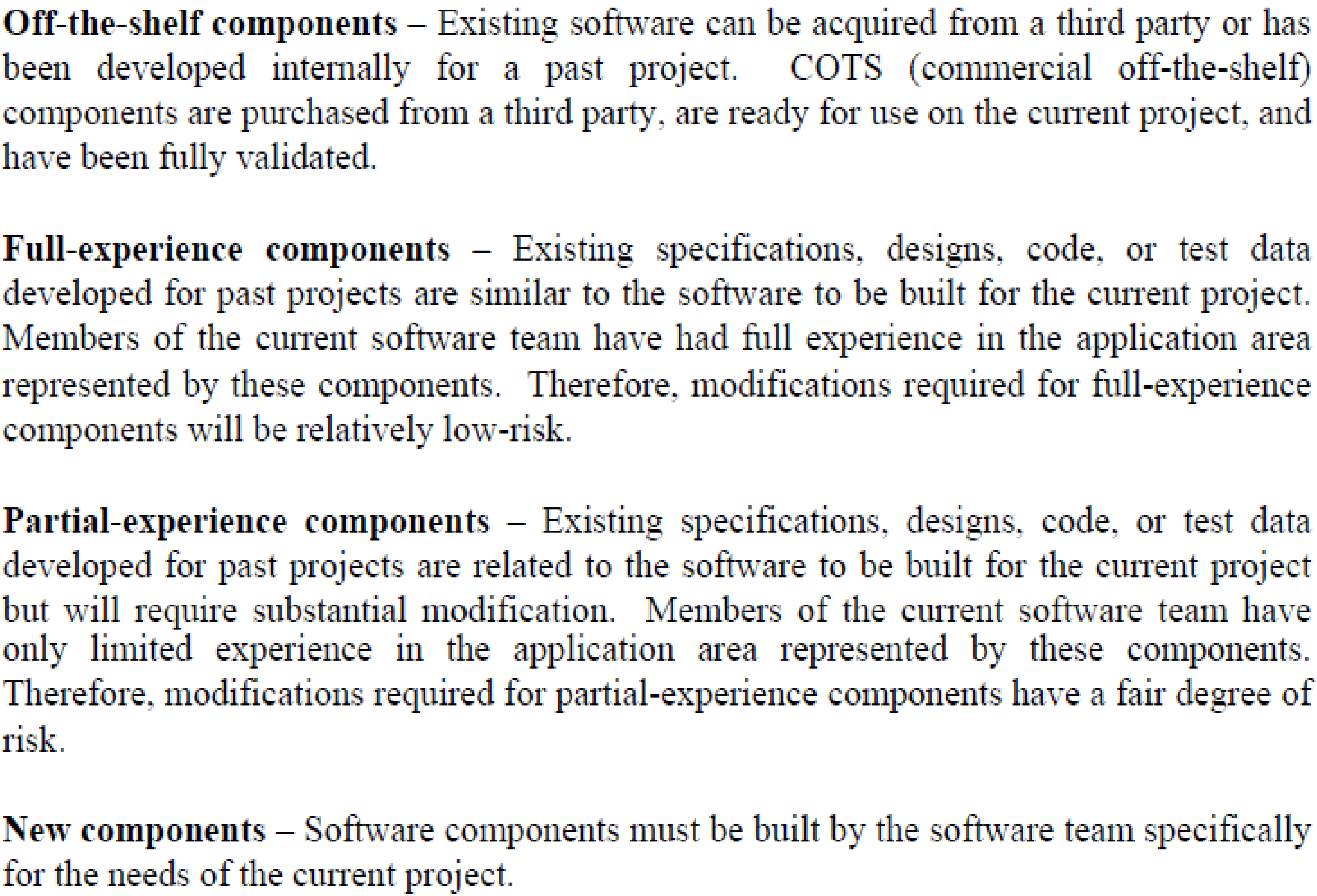
T5_Q3_Ans

Risk analysis & management in the context of SEng:
It is about actions that help a software team to understand and manage uncertainty.
It is to identify, address, and eliminate sources of risk before they become threats to successful completion of a project.

Key elements of risk management :
identification – what are the risks to a project?
analysis & Prioritization – which ones are really serious?
planning – what shall we do?
monitoring – has the planning worked?

T5_Q4: Options for using Reusable Software Resources

T5_Q4_Ans



Off-the-shelf components – Existing software can be acquired from a third party or has been developed internally for a past project. COTS (commercial off-the-shelf) components are purchased from a third party, are ready for use on the current project, and have been fully validated.

Full-experience components – Existing specifications, designs, code, or test data developed for past projects are similar to the software to be built for the current project. Members of the current software team have had full experience in the application area represented by these components. Therefore, modifications required for full-experience components will be relatively low-risk.

Partial-experience components – Existing specifications, designs, code, or test data developed for past projects are related to the software to be built for the current project but will require substantial modification. Members of the current software team have only limited experience in the application area represented by these components. Therefore, modifications required for partial-experience components have a fair degree of risk.

New components – Software components must be built by the software team specifically for the needs of the current project.

Tutorial 5_Q5

Suppose the information below shows the financial status of a 20 weeks project at the end of the 12th week:

Earned Value = S$25, 000 and

Actual Cost = S$ 30,000

The budget at completion is S$50,000.

(i) Compute the Planned Value as of the reporting date.

(ii) Compute the Cost Performance Index (CPI).

T5_Q5_ans:

What do we know: **BAC**=\$50K; Project **Duration** = 20 wks; **time elapsed**= 12 wks; EV=\$25K; Actual Cost=\$30K

- The Planned Value (at end of 12th week):
  PV= 12/20 * 50, 000 = S\$30,000

- The Cost Performance Indicator:
  CPI = EV/AC
  = S\$25,000/S\$30,000 = 0.833 < 1

If AC=35K; CPI=25K/35K = 0.714 < 1;
SPI = 25K/30K =0.833 < 1
if CPI is fixed, EAC = BAC/CPI = 50K/25K *35K = 70K

Possible techniques of risk reduction include:

T5_Q6_ans:

- Obtain a number of independent estimates using different estimation techniques. If these are widely divergent, generate more costing information iterate until the estimates converge.

- For those parts of the system which are hard to estimate, develop a prototype to find out what problems are likely to arise.

- Reuse software to reduce the amount of estimation required and to reduce overall costs.

- Adopt a design to cost approach to development where the system functionality is adapted to a fixed cost.

- Partition software requirements into critical, desirable and 'gold plating'. Eliminate 'gold-plating' if necessary.