CSCI3100 In-Class Practice

Software Cost Distribution

Date: ________________________________

Students Names: IDs:

____

____

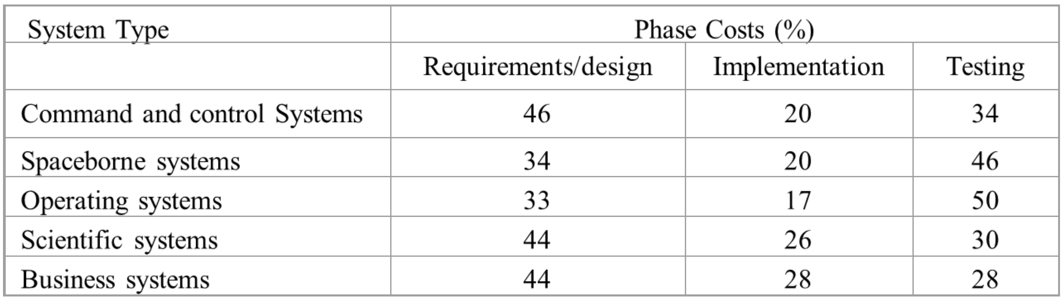
____

____

According to the software cost distribution in slide 1-10, please compare the phase costs of “Requirements/design”, “Implementation” and “Testing” in the descending order, among the following well known software systems. For example, in terms of “implementation”, one possible order is a > b > c > d > e.

a. iOS 13
b. TACCS – tactical command and control system
c. MATLAB
d. IQMS manufacturing ERP
e. Spaceborne radar systems

Meanwhile, you need to briefly explain why the cost distribution of these systems is ordered in this way.



System Type	Phase Costs (%)		
	Requirements/design	Implementation	Testing
Command and control Systems	46	20	34
Spaceborne systems	34	20	46
Operating systems	33	17	50
Scientific systems	44	26	30
Business systems	44	28	28

a. iOS 13: operating systems
b. TACCS: command and control systems
c. MATLAB: scientific systems
d. IQMS manufacturing ERP: business systems
e. Spaceborne radar systems: spaceborne systems

Consequently:

Requirements/design: b > c = d > e > a
Implementation: d > c > b = e > a
Testing: a > e > b > c > d

[Explanation] Open answers, e.g.,:

Generally, the most time-consuming part for software development is design and testing, instead of implementation. Design setup the whole framework, functionality and usage for the software. Testing is required for reliability and robustness, which takes a significant portion of efforts.

For Spaceborne systems, they are usually not very complicated, but their reliability requirement is extremely high, so testing efforts would be larger than design. For operating systems, their functionalities are generally complex but the design procedure is well known. Complexity of functionality requires large testing effort, particularly under enormous user operational scenarios, so it takes more effort for testing comparing to design.

For command and control systems, scientific systems and business systems, the software is relatively complicated while the reliability requirement is not stringent; therefore, software designs take more effort comparing to testing.