1155116317 Ng Chi Hon CSCI3100

Q1)

System Type	Phase Costs (%)		
	Requirements/design	Implementation	Testing
Command and control Systems (b) TACCS	46	20	34
Spaceborne systems (e) Spaceborne System	34	20	46
Operating systems (IOS13) (a)	33	17	50
Scientific systems (c) MatLab	44	26	30
Business systems (d) IQMS	44	28	28

Interm of "Requirements / design", the phase cost in descending order b > d >= c > e > a. Command and Control System has highest phase cost percentage because the system usually have complicate requirement and design to meet operational environment needs. And for business system, due to different business system demand different requirement, it need to design program differently everytime to fulfill requirement. Same as Scientific System, it need to fulfill different requirement for different scientific model. And compare to above three, Operating System and Spaceborne System need less phase cost in requirement and design because it require more phase cost in testing and improve the System.

In term of "Implementation", the phase cost in descending order is d > c > b = e > a, since business System and Scietific System have different design due to different requirement everytime, it need more time in complicted Detail design and coding that increase implementation Cost. And other three is similar due to fairly standard program detail design and coding with different System.

In term of "testing", the phase cost in descending order is a > e > b > c > d. Since Operating System have to handle interactions of various application after finish system design and implementation, testing is more tedious than other. Also, spaceborne System impose stringent reliability requirement as manual repair is not available, thus they need to be thoroughly tested.

Q2) (1) j, a, e, i, b, h, g, f, d, c

2) (2) j, a belong to the phase of requirement analysis and specification

e, i belong to the phase of Design and specification

b, h belong to the phase of coding and module testing

g, f belong to the phase of integration and system testing

d, c belong to the phase of Delivery and maintenance

Q2(3) Extreme Programming.

because the food-booking System always change the menu, type of food or payment method according to customer like McDonald, With high flexiabilty, allow system to change in different phase.

) Longtime Response Performance
font & button small user friendless
login previous user Security

Q3 (1)

→ Problem 1: Longtime Response. It is lacking of Performance

→ Problem 2: font and button are extremely small.
It is lacking of user friendliness. If font and button too small make user difficult to use.

→ Problem 3 login previous user on same PC,
It is lacking of security. The last system user should be carefully protected and not being accessed by irrelevant user. The user privacy can't be protect.

→ Problem 4, bug take long time fixed and stilly poorly fixed
It is lacking of maintainability, Getting feedback from user and fix bug is common to maintain system. The system need to fix bug correctly and faster.

(2) I don't think it is easy to apply algorithm sort because the quicksort function can only apply for integer data type vector variable.

(3) I don't think it is a good measurement for employee because this kind of measurement approach will make people more of line of code, and less in quality, like for loop logic and divide into many short statement and gain lots of no. of Lines of code.

Q4 (1)

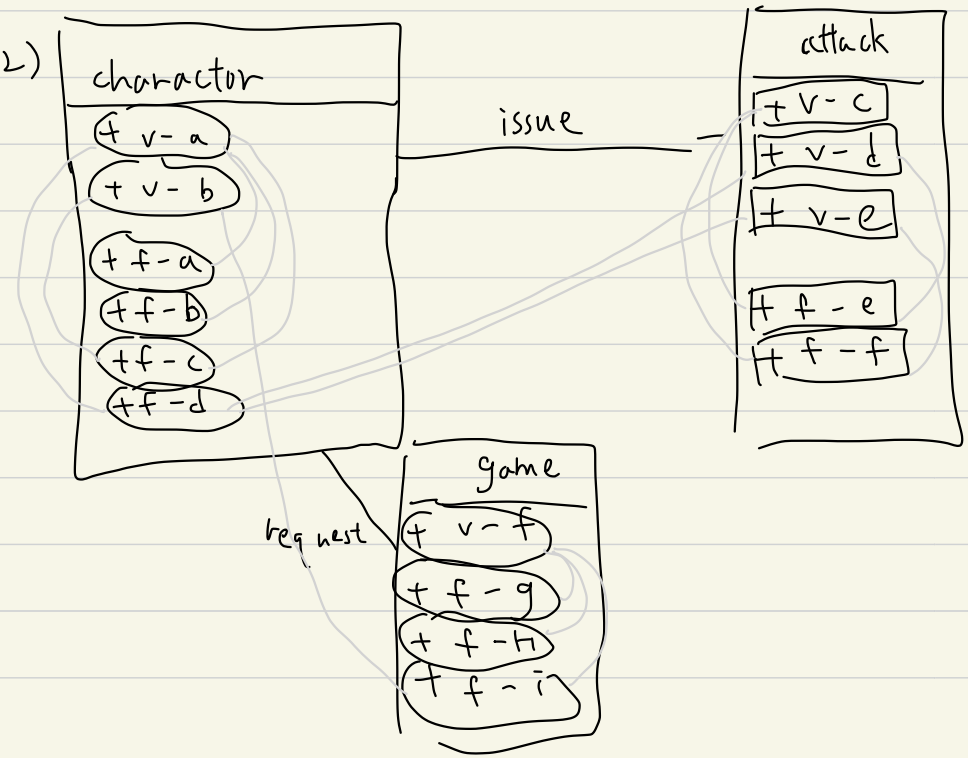
```
class character
{
    float pos;
    float health;
    void jump(float Pos)
    void move_fawar(float Pos)
    void move_back(float Pos)
}
```

```
class attack
{
    int attack_type;
    float damage;
    float range;
    float get_damage(int attacktype)
    float get_range(int attacktype)
    void hurt(float health, float pos, float damage, float range);
}
```

```
class Game
{
    vector<int> character_id;
    vector<int> choose_character(vector<int> character_id
    void reorder_team(vector<int> character_id);
    bool check_wining(vector<int> character_id,
                      float health);
```

I would make variable and function in each class related to each other inside the class. At same time, variable and function in different class be unrelated as most as possible.

(2)



(3) one more class charactor_status that include void hurt()
and float health.