Crib for 4C4 Design Methods 2023/2024

Version: JMC/4

Question 1

a) marks 10%

	T1	T2	T3	T4	T5	T6
T1				X		
T2	X		X			
T3		X				X
T4						
T5		X				
T6					X	

b) marks 20%

Rework can be reduced by partitioning and tearing. Both reduce later dependencies by making the dependency structure matrix closer to lower triangular form. The original dependency structure matrix is as follows:

↵	T1	T2	T3	T4	T5	T6
T1				X		
T2	X		X			
T3		X				X
T4						
T5		X				
T6					X	

First, move task T4 as it does not depend on anything:

↵	T4	T1	T2	T3	T5	T6
T4						
T1	X					
T2		X		X		
T3			X			X
T5			X			
T6					X	

Second, tear task T2 into subtasks T2a and T2b, where only task T2b depend on task T3:

↵	T4	T1	T2a	T2b	T3	T5	T6
T4							
T1	X						
T2a		X					
T2b		X	X		X		
T3			X				X
T5			X				
T6						X	

Third, reorder subtask T2b and task T3:

↵	T4	T1	T2a	T3	T2b	T5	T6
T4							
T1	X						
T2a		X					
T3		X	X				
T2b			X	X			X
T5			X				
T6						X	

c) marks 30%

↵	T1	T2	T3	T4	T5	T6	W1	W2	W3
T1				X					
T2	X		X						
T3		X				X			
T4									
T5		X							
T6					X				
W1	X	X	X						X
W2				X	X	X			X
W3			X			X	X	X	

d) marks 20%

We need to find the number of passes for T3 $\rightarrow$ T2 and T6 $\rightarrow$ T3.

T3 $\rightarrow$ T2 has a quality of 0.9 and requires two passes:

Pass 1 $0.9 \cdot 100\% = 90\%$.

Pass 2 $90\% + (0.9 \cdot 10\%) = 99\%$.

$T6 \rightarrow T3$ has a quality of 0.8 and requires three passes:

Pass 1 $0.8 \cdot 100\% = 80\%$.

Pass 2 $80\% + (0.8 \cdot 20\%) = 96\%$.

Pass 3 $96\% + (0.8 \cdot 4\%) = 99.2\%$.

Each $T3 \rightarrow T2$ rework cycle is one week with one additional pass, hence 1 week overrun per iteration.

Each $T6 \rightarrow T3$ rework cycle is two weeks with two additional passes. Each additional pass involves two weeks of rework time for $T6 \rightarrow T3$ + one additional week due to the dependence of $T3 \rightarrow T2 = 3$ weeks. Since two passes are required the project is overrun by six weeks.

e) marks 20%

Solving yields:

$$\alpha = \mu\alpha + \mu\beta$$

$$\beta = \frac{1-\mu}{\mu}\alpha$$

$$\alpha = \frac{\mu^2 - \mu^3}{\sigma^2} - \mu$$

$$\beta = \frac{\mu - 2\mu^2 + \mu^3}{\sigma^2} - (1-\mu)$$

We are given $\mu = 0.04$ and $\sigma^2 = 0.06^2 = 0.0036$.

Evaluating for α yields $\approx 0.386 \approx 0.39$.

Evaluating for β yields $\approx 9.279 \approx 9.28$.

Rework is affected by the uncertainty in monitoring the quality of rework as perceived rework will reduce if such uncertainty is high, leading to a larger undiscovered rework pool.

Question 2

a) marks 10%

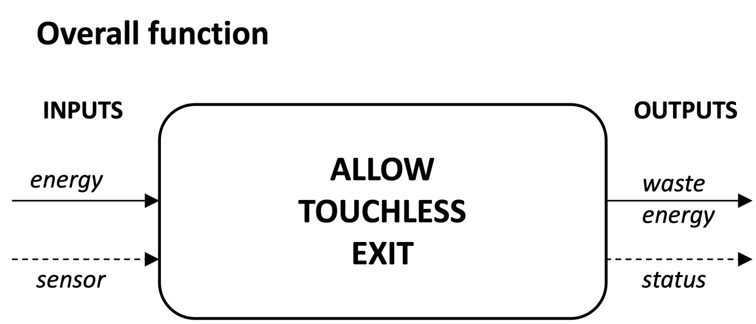
The overall function is of the form `Allow Touchless Exit` (many possible formulations are possible). Importantly it is not merely `Open Door`, as this is a sub-function.

The main sub-functions could be : `Sense User Input`, `Display Output`, `Unlock Door`, `Open Door`, `Close Door` and `Lock Door`. Several possible formulations are possible.

b) marks 40%

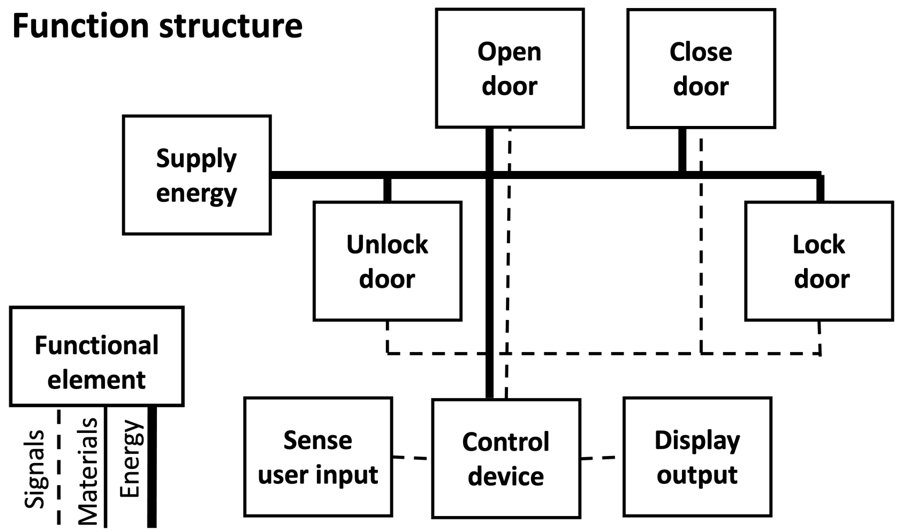
The overall function can be decomposed into function structures in several ways and individual solutions are judged on whether they correctly model the flow of signals between necessary sub-functions to carry out the overall function.

The overall function receives a *Sense User Input* signal, for example, passing a hand in front of an infrared sensor. Successful activation of the user input signal generates an *Output* status, for example, the a green indicator light. In addition, the system needs energy, to lock/unlock the door and open/close the door, which is converted to waste energy. No material flows are involved (the user moving through the door is considered outside the boundary). An example of the overall function of the system is provided in the figure below:



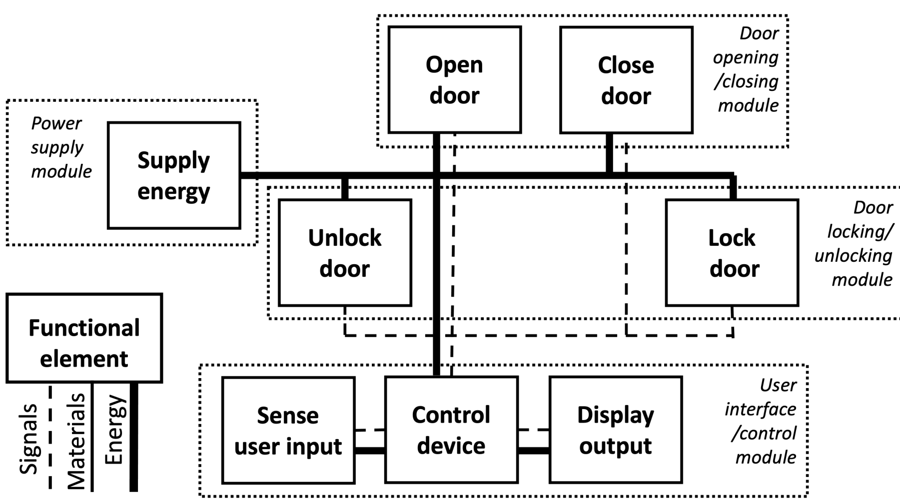
The overall function can be decomposed into the following user-centred sub-functions: `Sense User Input` function, `Display Output`, `Unlock Door`, `Open Door`, `Close Door` and `Lock Door`. A sub-function of `Supply Energy` is added and connection to all others sub-functions; the energy to the `Sense User Input` function and `Display Output` can be routed via the `Control Device`. Note, the `Close Door` sub-function might be provided by a spring, rather than an electrical power supply.

A sub-function of `Control Device` may be added to organise the control signals; the Control Device is connected to all the other sub-functions, to receive the user input, display the outputs and monitor the door lock/unlock and open/close sub-functions. Further details can be provided on how user input and output are communicated across the design, but this is not expected. An example of the function structure, including functional elements and flow of signals is shown in the figure below:



c) marks 30%

A modular product architecture is created by decomposing the overall function into its sub-functions, attaching any necessary flows of energy, materials and signals, and then clustering related functions into modules. While the product architecture can vary, a typical *high-level* modular architecture would contain at least four main modules: 1) power supply; 2) user interface/control; 3) door locking/unlocking; 4) door opening/closing. It is reasonable to assume the sensor, door locking mechanism and door opening mechanism are commodity components. An example of the modular architecture is shown in the figure below:



An integral product architecture might include the sensing, display, locking and opening mechanisms into an integrated door design. This could replace an existing door, as a single unit, and require only a connection to an energy supply.

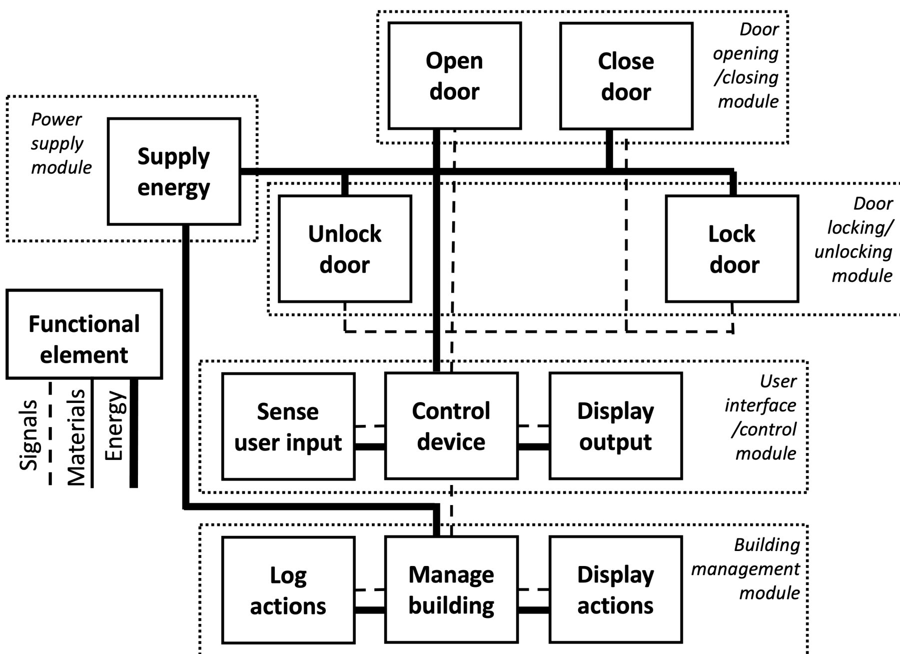
A modular product architecture allows for more flexibility, enabling more variety in the design offering and options for changing the design in the future. Each module implements one (or few) functional elements, and interactions between modules are ?de-coupled?. Modules can be designed independently and changes to one module are less likely to require changes to others.

Overall products costs are increased.

An integral product architecture enables increased levels of optimisation. A single module implements many functions and interactions between modules ?coupled?. Modules must be designed in collaboration with modifications to one part more likely to affect other parts. Overall product costs are reduced.

d) marks 20%

The design sketched in the answer to (b) will need to be augmented to provide a signal connection between the `Control Device` and a new sub-function `Manage Building`. The `Manage Building` sub-function is connected to a new sub-functions of `Display Actions` and `Log Actions`. Energy is required for these three new sub-functions and a new Building Management module is required. An example of the modular architecture is shown in the figure below:



Signals about the state of the sensor and door are communicated to the building management module. This could be a one-way signal, allowing the building manager to see and log the activation of the sensor, unlocking/locking the door or opening/closing of the door. Alternately, the signal could be sent in the reverse direction, to allow the building manager to override the functions of the sensor, locking mechanism or opening mechanism. This would, for example, allow the system to operate differently at different times of the day, or days of the year.

Several solutions are possible and are judged on ability to model all the functionality in terms of completeness and simplicity, and by the strength of the motivation of the design and its coherence to the actual function structures.

Question 3

a) marks 10%

Quality in the context of rework means the ratio of *work really done* to the *work being done*.

b) marks 30%

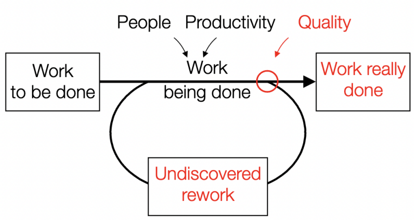
Unexpected change creates rework Rework likely as changes to components are frequent.

Rework creates rework The quality of the rework itself is affected by fabrication relying on individual workers, a complex moulding process, and frequent updates of components. A lower quality generates more rework cycles.

Freezing design components impacts rework A complex moulding process is expensive and difficult to change and hence rework needs to be minimised, which is challenging if components need to be frequently updated or changed.

c) marks 20%

The rework cycle model looks as follows:



The introduction of an individual novice worker to replace an individual skilled worker is likely to reduce quality and productivity.

The system solution may however improve quality by reducing undiscovered rework, which in this case would be process errors induced by the novice worker. The system solution forms a rework cycle of its own, assisting the novice worker in detecting work really done and undiscovered rework that needs to be redone. If effective, this "inner rework loop" can improve quality of the overall fabrication process by eliminating more costly rework discovered at late stages in fabrication (i.e. at the quality assurance stage).

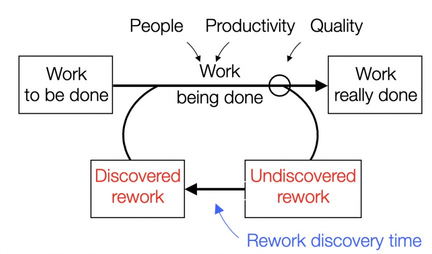
Hence there is a tradeoff between a novice worker's reduced productive and quality, and the extent the system solution can mitigate such effects.

d) marks 20%

Verification means ensuring requirements are met. Validation means ensuring the system is fit for its purpose in a deployment environment. In general, a design can pass verification by ensuring all requirements are met but fail in validation as the requirements may not accurately model the needs of the system in deployment. As an example, in this case a very likely reason why this system may succeed in verification and fail in validation is a failure to account for the operating environment of the system when verifying requirements. The operating environment of this system is a shop floor with loud machine noise, people moving around, and monitoring of operator performance, which may give rise to choking under pressure issues. Such factors are difficult to account for when verifying a system.

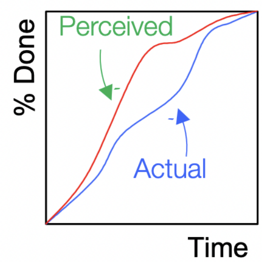
e) marks 20%

The rework cycle model can be augmented to make the transition of undiscovered rework to discovered rework explicit. The time between these two states is *rework discovery time*:



The system in (c) gives rise to a pool of undiscovered rework, hence the quality assurance problem. The monitoring by an experienced worker should improve the ability to translate undiscovered rework into discovered rework, hence reduce the rework discovery time.

Note that perceived work done usually differs from actual work done, as sketched in the figure below:



The addition of monitoring should help align these two curves, hence increase quality and reduce rework discovery time.

Question 4

a) marks 10%

A solution-neutral problem statement derivation can, for example, be of the form: “Design a process that allows designers to generate new clothing designs” → “Design a process that allows designers to generate new clothing designs for fast-moving fashion” → “Design a process that allows designers to rapidly create new clothing designs”

b) marks 20%

Key requirements should be traceable (where do they come from? e.g. scoping survey, literature review) and testable (how?), and might include:

- a collection of existing clothing designs to use as training data
- a means to learn from the training data what good design looks like
- a means to generate many possible clothing designs
- a way for the designer to visualise the generated clothing designs
- a method to choose from or rank the clothing designs
- a procedure for updating the design generator based on the selections

c) marks 30%

The six principles for creating systems that work, in integrated system design, are:

1. Pursue the purpose: the system will generate design concepts for clothing in a fast-moving fashion market. Cost, performance and make-ability are the key requirements to be defined. ”Requirements determine how the system will impact on its environment and how the environment will impact on the system”.

2. Think holistic: an integrated system needs to consider the system as a whole. Students should describe the components of the new design process (data, algorithms, visualisation, selection), the environment in which it operates (fast-moving fashion, design teams, ability to make clothes), and the tools/procedures to build it (training, commissioning, updating). ”Like a chain, a system is only as good as its weakest link”.

3. Follow a disciplined procedure: ”Systems that work do not just happen ? they have to be planned, designed and built”. The V-diagram as a methodology for partitioning (breaking the problem down into manageable parts) and integrating (bring the parts back together), would be an appropriate method to follow.

4. Be creative: in this context, the process would need to cost-effective and realisable. For example, the new design process could be sourced ‘off the shelf’ or developed ‘in-house’. Remember the aim is to deliver capability, not a technology.

5. Take account of the people: People are part of the system, and need to design, build, operate, test, maintain and decommission the new process. Therefore designers need to take account of the human aspects of the system. People will also be required to defend, challenge or tolerate the system. In this context, the designer should co-create and design the new processes with the company stakeholders, including designers, marketers, and IT support, drawing on their flexibility, inventiveness and intelligence to deliver a successful new process. ”People are at the heart of most systems; they may be both its weakest and strongest link”.

6. Manage the project and relationships: the introduction of a new process will require both the design of the new process and the design of the project to introduce the new process. The new process will have complex interactions to manage (i.e. between suppliers, designers, stakeholders, standards bodies, etc). The specific architecture for the new process will require, for example, new testing, training, monitoring, and maintenance procedures.

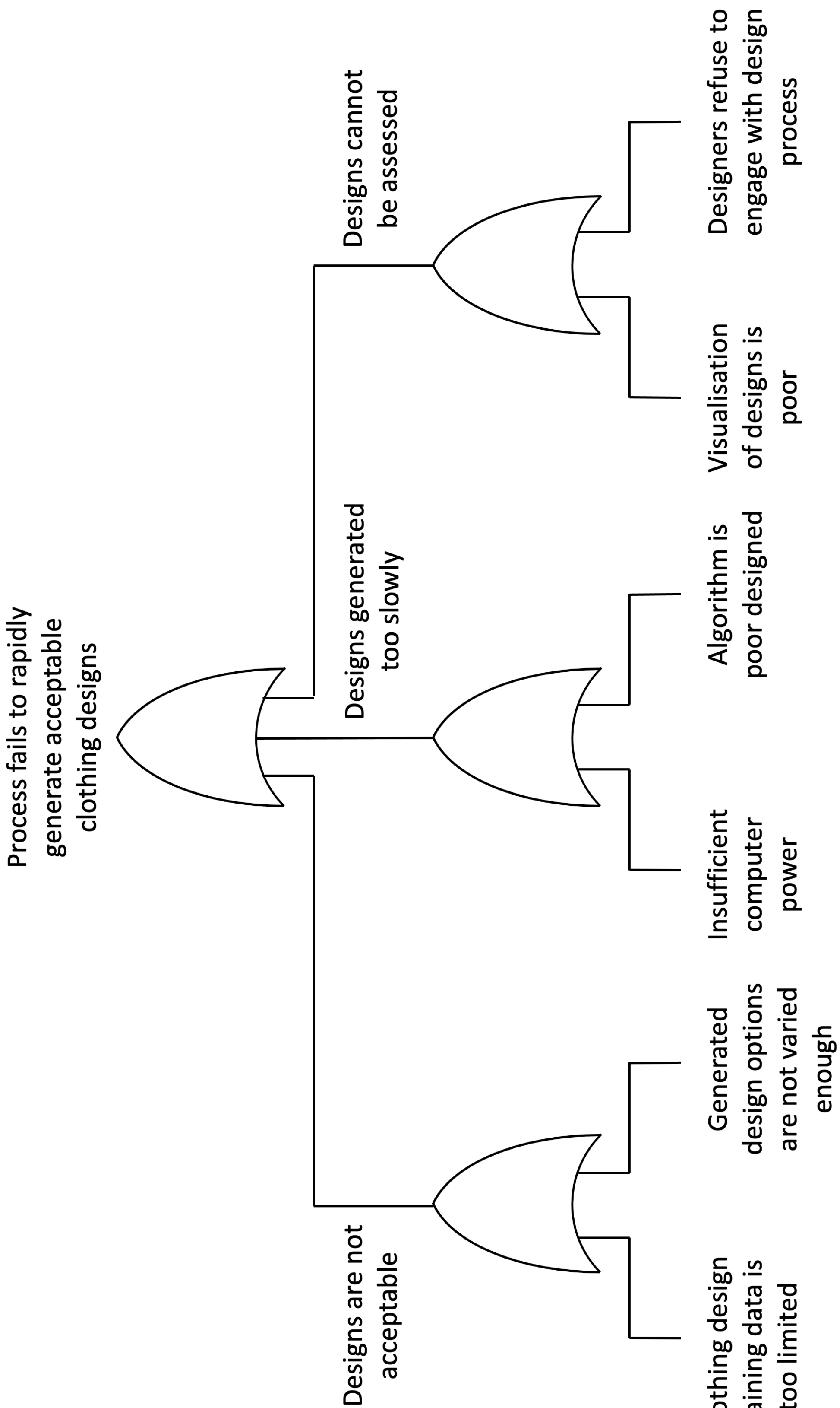
d) marks 20%

Key factors of good risk management might include, but are not limited to:

- careful assessment of the requirements for the new design process;
- assessment of the appropriateness of the training data
- consideration of the costs and timeline for introducing the new design process
- testing to ensure the AI tool generates acceptable clothing designs;
- consideration of the reliability and security of the AI tool;
- appropriate training provision for the design, marketing and IT teams;
- clear definition of the system performance requirements, and how these can be tested;
- communication of the system changes to the wider company and potential customers;

e) marks 20%

'Gates' in the fault tree are likely to be OR gates. The fault tree can take a number of different forms, but should address issues across the entire system, and relate to the key requirements identified in (b). An example is provided in the figure below.


Process fails to rapidly generate acceptable clothing designs
Designs are not acceptable
Designs generated too slowly
Designs cannot be assessed
Clothing design training data is too limited
Generated design options are not varied enough
Insufficient computer power
Algorithm is poor designed
Visualisation of designs is poor
Designers refuse to engage with design process