CSci 463: Spring 2020

Exam I

March 12, 2020
Exam Due March 13

(Total Point: 120)

First Name_Sam___________________
Last Name_Dressler___________________

Note 1: About the take-home exam's part. The solutions must be typed and spell checked. The solution must yours and must be different from others.
Please submit your take-home solution as pdf or doc file via email by the due date (Friday March 13 by 11:59pm).

Note 2: Please use legible handwriting!

1. (20 points) The two main source of information for the requirement elicitation process are **users** and **the application domains**. These sources both presuppose that there exists something out there to start with, from which requirements can be elicited. The rows in following table lists a number of elicitation techniques discussed in the book and class. The columns indicate main source of information (Application Domain, User), and type of systems (Current system or Future system).

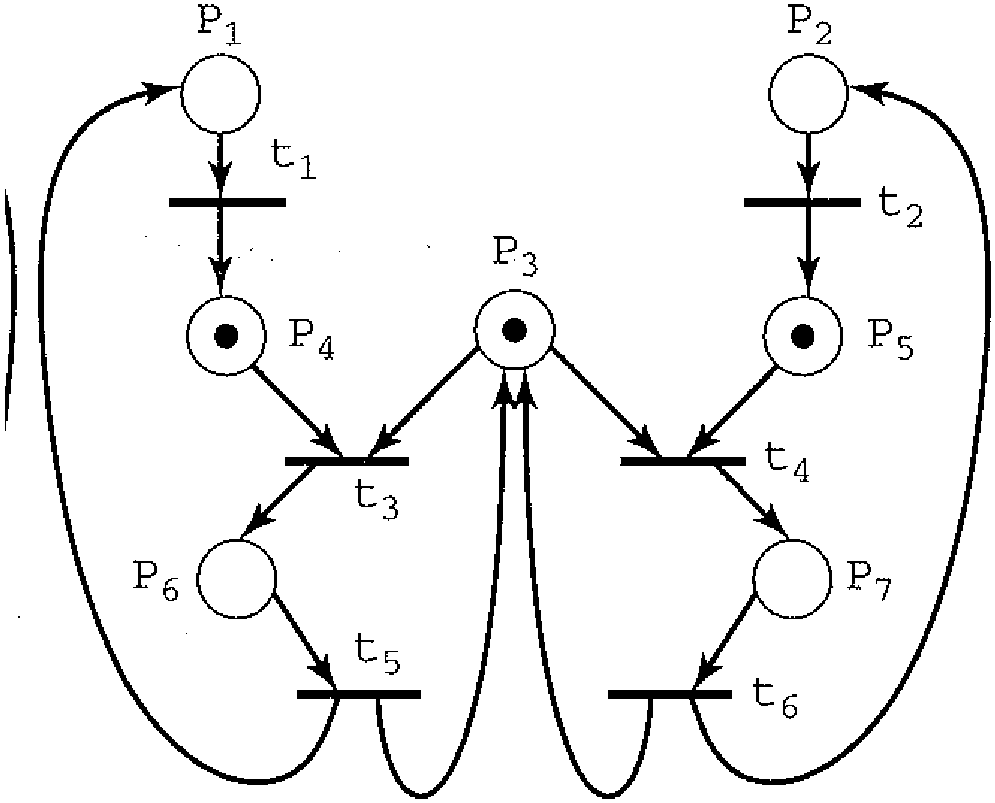
   You need to fill up the blank cells with 'X' (whenever appropriated) in the following table to indicate the applicability of techniques represented in rows with respect to columns (i.e., the source of information and type of systems). For instance, the technique, say, Y, is suitable to get information from User to build both current or future systems.

Technique	**Source of Information**		**Type of Systems**	
	Application Domain	**User Domain**	**Current**	**Future**
Interview	X	X		X
Scenario (use-case) analysis	X			X
Ethnography		X		X
Prototyping	X			X
Natural Language		X	X	X

2. (10 points) Modeling a complex system, which demands parallelism, in one, large, monolithic Finite State Machine (FSM), which pictorially are normally represented as state transition diagrams (STDs), is not to be recommended. Explain why it is not appropriate to use FMS. What other alternatives modeling would you recommend? Briefly explain why?

The reason why using a FSM to represent a parallel system is a bad idea is becasue FSMs are incapable of representing concurrent systems. In order to accuratley represent the system, the designer should use either a state chart or a petri to model the system. Both of these options are capable of showing concurrent behaviour because the model can show the system being in more than one state at a time.

3. (20 points) 1) What are the main limitations of Petri nets? 2) what are their main benefits? Furthermore, give two or more examples of firing sequences of transitions for the following net.



https://en.wikipedia.org/wiki/Petri_net

1) Some of the main limitations of Petri nets is the fact that they cannot specify time, type, or hierarchy. Not having the ability to specify means that the model cannot specify which transition will fire nor will it show exactly when it will fire. Not being able to specify hierarchy means the model is flat. Finnaly, since a Petri net does not account for time, there is no way to show if a certain system spent the majority of its time in one place and not in another. Furthermore, Petri nets can become increasingly complex at a faster rate than other models, due to the fact there is concurrent behavior being represented.

2) Some of the benefits of Petri nets are its ability to be asynchronous as well as concurrent. A model being asynchronous means that it does not reflect time. This can be seen as a limitation, however, in some cases this fact is not a problem because of the timeless nature of some systems. As long as they are represented in a sequential order the aspect of time is not important. Next,This model being concurrent means that it has the ability to show parallel activities by having an increased amount of transitions being activated at the same time.

Possible Firing Routes: P3 is a shared resource that is used in order for either side to complete its loop.There are two different firing sequences. The first is where tokens from P3 and P4 cause T3 to fire first and the second is when tokens from P3 and P5 cause T4 to fire. The exact paths for each are as follows:

P3 & P4 Firing sequence:

1) Tokens at P3 and P4 cause T3 to fire. The token is now in P6. T5 only requires one token but creates two. The first token from T5 goes to P1, through T1 and back to P4 where it will wait for another token to be in P3 so that the firing criteria for T3 is ready again. The second token from T5 returns to P3. The new token now in P3 will trigger the P3 & P5 firing sequence below.

P3 & P5 Firing sequence:

2) tokens at P3 and P5 cause T4 to fire. The token is now in P7. T6 only requires one token but creates two. The first token from T6 goes to P2, through T2, and then back to P5 where it will wait for another token to be in P3. The second token from T6 will return to P3. The new token now in P3 will trigger the P3 & P4 firing sequence above.

4. (20 points) what are the factors that require the project to be re-estimated? (Just list them.). Furthermore, Function Point offers many advantages over other software sizing techniques. What are the pros and cons of using function point to estimate the effort?

As outlined in our book, there are many reasons why an estimation for a project could need to be re-estimated. Some of these reasons are because of the developer where as some could be a result of the customers lack of understanding.
A brief list of some of the factors is as follows:
- There are frequent request for changes by users.
- Tasks have been overlooked by the developers.
- There is a lack of understanding from the customers on their own requirement specifications.
- Not enough effort was put into analysis prior to completing the estimate.
- Lack of coordination or availability between the development team and other resources i.e Simulator equipment for testing.
- The complexity of the system was greater than anticipated.
- Capabilities of the project team members were greater or less than the expectations.
- Code or documentation standards are changed or needing to be applied once development begins

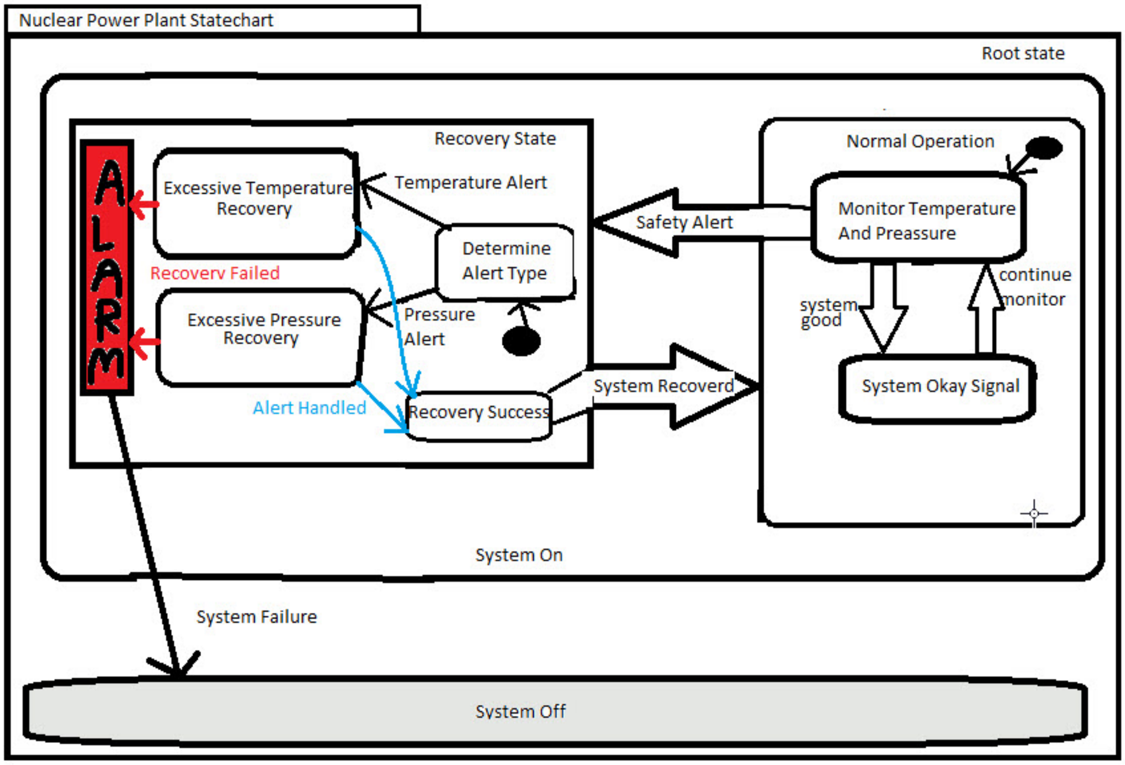
(2)Pros of using Function Points:
- Provides a clear view of the size, cost, and productivity
- Estimation that is independent of Technology or programming languages
- Good to use early on because you only need the requirements
- Inherently provides a strong base of documentation

Cons of using Function Points:
- Can be very time consuming
- Does bad in cases where the project is volatile
- To use function points, the estimator needs to know how to properly find the RCAF.

Source:
1. Software Engineering Textbook
2. http://washingtoniceaa.com/files/presentations/34_Function%20Point%20Analysis.pdf

5. (40 points) Use **Statecharts** to specify a nuclear power plant control-system. In this system, temperature and pressure levels must be constantly monitored for safety reasons. Assuming that sensors are installed to detect and generate appropriate signals when either of these level (pressure or temperature) exceeds some predefined threshold values.
The requirements and assumptions for managing the plant are the following:

    1. When one of the two signals is raised, the system must enter into a recovery state in which it tries to apply a recovery procedure.
    2. If, after a while, the recovery action succeeds, the system is automatically entering into the "normal" state, and sends an appropriate message (e.g., everything is OK) to the external environment.
    3. Otherwise, the alarm signal must be raised and the plant MUST be shut off. In this case, the system as a whole must also be switched off if it is trying to recover from one kind of anomaly (i.e., excessive pressure or temperature) once the other signal is raised.
    4. It is assumed that the two signals (excessive pressure or excessive temperature) cannot occur simultaneously.



6. (10 points) Explain how the principles underlying agile methods lead to the accelerated development and deployment of software.

The agile software development process allows for rapid development due to its basic underlying principles.
The first of these principles is strong communication. In a Agile work environment, the workers meet daily to discuss their work plan for the day as well as any roadblocks they are having. This benefits the speed at which software is developed because it reduces time spent working on the same thing as well as allows team members to address problems as they come up.

The second principle is customer interaction. Just as communication within a software team is important, the communication with the customer is crucial to developing the correct product. Agile is not alone in its use of communication with the customer, however, Agile uses sprints and increments as clear boundries for delivering features.

The final principle understands the innevitability of change. The nature of using sprints and increments as ways to divide the work being completed allows the response of change to be done with greater ease than traditional software development methods. Time is saved during the design phase because developers don't have to design the entire system all at once. They are able to design, implement, and test on a smaller scale which allows the developer to test more completley.