* The main advantage of the distributed version control system we can work in the network sharing the file in the very manner operating and commit it. And there is no centralized server is there so the failure of the central network will not impact this.
* The disadvantage of the distributed version control system is it is possible when two users update the same field at a time this will cause a problem also the test out is slower as compared to a centralized system.

1. A critical system is a system that must be highly reliable and retain this reliability as they evolve without incurring prohibitive costs.

There are four types of critical systems: safety-critical, mission-critical, business-critical

* Safety-critical

Safety-critical systems deal with scenarios that may lead to loss of life, serious personal injury, or damage to the natural environment. Examples of safety-critical systems are a control system for a chemical manufacturing plant, aircraft, the controller of an unmanned train metro system, a controller of a nuclear plant, etc.

* Mission-critical

Mission-critical systems are made to avoid the inability to complete the overall system, project objectives, or one of the goals for which the system was designed. Examples of mission-critical systems are a navigational system for a spacecraft, software controlling a baggage handling system of an airport, etc.

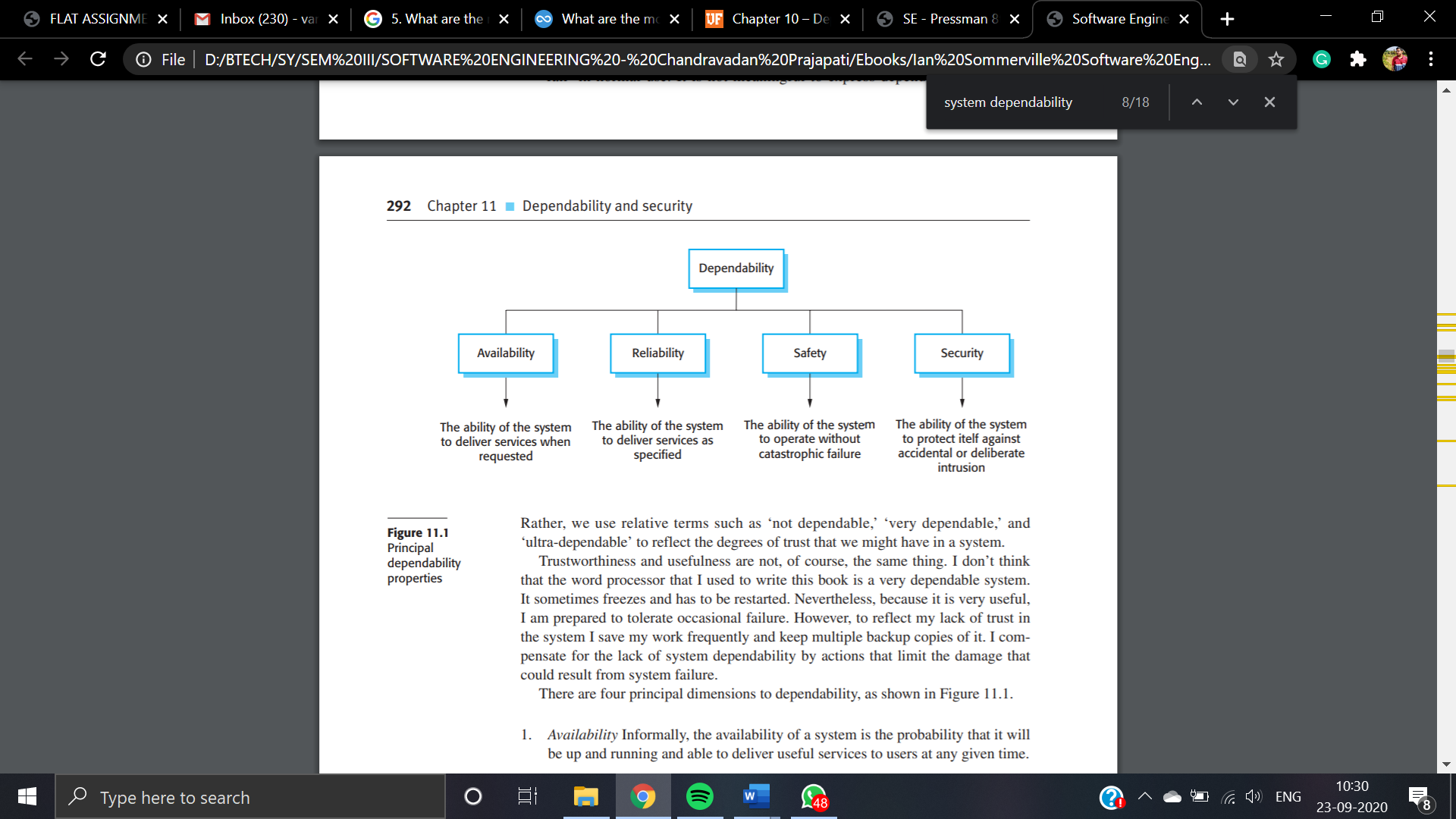
* Business-critical

Business-critical systems are programmed to avoid significant tangible or intangible economic costs; e.g., loss of business or damage to reputation. This is often due to the interruption of service caused by the system being unusable. Examples of business-critical systems are the customer accounting system in a bank, stock-trading system, ERP system of a company, Internet search engine, etc.

1. **Six reasons why dependability is important are:**

* Users may not use the system if they don't trust it.
* System failure may lead to a loss of business.
* An undependable system may lose or damage valuable data.
* An undependable system may damage its external environment.
* The reputation of the company that produced the system may be damaged hence affecting other systems.
* The system may be in breach of laws on consumer protection and the fitness of goods for purpose.

1. NA



There are four principal dimensions to dependability, as shown in Figure 11.1.

1. **Availability** Informally, the availability of a system is the probability that it will be up and running and able to deliver useful services to users at any given time.

2. **Reliability** Informally, the reliability of a system is the probability, over a given period, that the system will correctly deliver services as expected by the user.

3. **Safety** Informally, the safety of a system is a judgment of how likely it is that the system will cause damage to people or its environment.

4. **Security** Informally, the security of a system is a judgment of how likely it is that the system can resist accidental or deliberate intrusions.



* Internet server: **Availability**as the failure of availability affects a large number of people, the reputation of the supplier, and hence its current and future income.
* A computer-controlled scalpel: **Safety**as safety-related failures can cause harm to the patient.
* A directional control system: **Reliability**as mission failure could result from failure of the system to perform to specification*.*
* A personal finance management system: **Security**because of potential losses to users.

1. Possible domestic appliances that may include safety-critical software include:

* Microwave oven
* Power tools such as a drill or electric saw
* Lawnmower
* Central heating furnace
* Garbage disposal unit
* Vacuum cleaner
* Food processor or blender

1. Ensuring system reliability does not necessarily lead to system safety as reliability is concerned with meeting the system specification (the system 'shall') whereas safety is concerned with excluding the possibility of dangerous behaviour (the system 'shall not'). If the specification does not explicitly exclude dangerous behaviour then a system can be reliable but unsafe.
2. A possible hazard is the delivery of too much radiation to a patient. This can arise because of a system failure where a dose greater than the specified dose is delivered or an operator failure where the dose to be delivered is wrongly input.

Software features that may be included to guard against system failure are the delivery of radiation in increments with an operator display showing the dose delivered and the requirement that the operator confirms the delivery of the next increment. To reduce the probability of operator error, there could be a feature that requires confirmation of the dose to be delivered, and that compares this to previous doses delivered to that patient. Alternatively, two different operators could be required to independently input the dose before the machine could operate.



* **Anti-lock braking system** *This* is a safety-critical system so requires a lot of up-front analysis before implementation. It certainly needs a plan-driven approach to development with the requirements carefully analysed. A waterfall model is therefore the most appropriate approach to use, perhaps with formal transformations between the different development stages.
* **Virtual reality system** *This* is a system where the requirements will change and there will be an extensive user interface component. Incremental development with, perhaps, some UI prototyping is the most appropriate model. An agile process may be used.
* **University accounting system** *This* is a system whose requirements are fairly well-known and which will be used in an environment in conjunction with lots of other systems such as a research grant management system. Therefore, a reuse-based approach is likely to be appropriate for this.
* **Interactive travel planning system** *System* with a complex user interface but which must be stable and reliable. An incremental development approach is the most appropriate as the system requirements will change as real user experience with the system is gained.

1. The specifications of evolutionary development projects are often abstract, and as the project continues, the development and validation portions of software engineering overlap one another. This usually results in the systems being poorly constructed due to a good initial specification, and on large projects makes it more difficult to integrate new systems into the evolutionary design. Lastly, the documentation for such projects is often lacking, as the designs are constantly rebuilt to the customer’s specifications
2. The spiral model is much like the waterfall model, in that there are well-defined stages, but different in that once an initial sequence is complete, the process starts over, correcting problems and expanding ideas to better suit the customer’s needs. Also, as an iteration of the spiral may be short, a prototype can be produced that the customer can see and work with, to help guide the engineers to more accurately construct what they seek.
3. The Rational Unified Process recognizes that a single process model presents only a single view, so the RUP incorporated three perspectives, two of them being dynamic, that describes the phases of the model as time progresses, and static, that describes the activities that are enacted. The strength in using both is that phases of the development process are not tied to any one specific workflow, so the entire process can be understood.
4. NA
5. NA
6. Assignment 3 questions 1
7. NA