1. User Stories
2. A white sheet with blue text

   Description automatically generated
3. Functional vs non-functional
   1. Viewpoints of Requirements (?)
4. MoSCoW method
   1. Most should could wont
5. Why do requirements need tests?
   1. Vertification validation (Lack 2)
6. Agile vs. Waterfall
   1. Linear vs iteration
   2. Limitation?
      1. Agile: scope creep, documentation, fallback/rollback
      2. Waterfall: not flexible, lack testing, stakeholder may not satisfy

W4

1. Verifiable Requirements include?
   1. Clear metric, no conflict, clear scope,
2. Non-functional Metrics? Performance reliable robustness portability (Usability: ease to use and size)
3. Hard-to-test Requirements make example, but how to test it? Backup and recovery testing/mock testing
4. Domain Modelling
5. What include ? class object method
6. What relationship include? (There are 4)
   * 1. Association (normal, professor got student) aggression (has a, department got professor) composition (is a including relationship, university got department) dependency (weak, professor got salary)
7. Prototyping
8. Type of proty(2)
   1. Fully rebuild
   2. modify
9. Benefit?
   1. Stakeholder, clear view
   2. Clear scope
   3. For team clear goal

W5

* PEGS?
* Project require vs goals require? Make example?
* Analysis vs validation?
* UML class diagram 100% in exam

W6

Sequence diagrams

* Why?
* What?

W7

* **State Chart Diagram vs. Sequence Diagram?**
* When use state diagram?
* What is coupling coupling degree of dependency between the module
* What is cohesion degree of dependency inside the module
* Which is higher is better
  + Low coupling high cohesion

W8

* What architecture we have? 3 main 5 other
* Layered , pipe and filter, MVC; client service, service based (service oriented)
* What is the different between first two >
* What is MVC?

W9

* What is user interface
  + What is 3 important components for user interface?
* What are hardware interfaces
* What are software interfaces?
  + Definition? How it works? Example?
* Top-down vs. Bottom-up? How they work? Which common use in what situation and why?

W10

* Remember the design pattern and reason!! 200% exam
* Why design pattern?
* Kind of patterns? 3
* First one has?3
* Second one?1
* Facade Pattern principle and why and example
* Adapter Pattern principle and why and example
* Proxy Pattern principle and why and example
* Observer Pattern principle and why and example

W11

* What why when Refactoring?
* How
* Advantage of it?