Exam 1 will cover:
Software development lifecycle
Design methodologies
Software requirements engineering

To reiterate: FURPS+ is functionality, usability, reliability, performance, and supportability, plus other things like implementation, interface, operations, packaging, and legal.

Requirements estimation is an estimation for how long it will take to implement a requirement. It is also tied to the cost to implement. This estimation, however, is very difficult to do; coding a project will always take longer than expected and it is hard to account for roadblocks.
To determine how long something would take to implement, use a 1-100 scale based off of the project velocity. Don't make the highest point value more than one iteration of work.
To get a more accurate estimate, split a project up into many small features, and estimate the time it would take to create each one of those. Once an estimate is acquired for every small feature, simply add up the estimates to get the total time. In iterative design, a high-level estimate is enough.

When planning, one can use "planning poker". Everyone is given a set of cards with progressively larger numbers, with progressively larger gaps between the numbers. Everyone then discusses the work that would go into implementing a feature, selects a card to show how long they think the feature would take to develop, and then all reveal their estimate at once. Everyone then justifies their estimate, and reselections occur until everyone is in consensus as to what the estimated time should be.

Requirements specification is the last stage of requirements engineering. It is the output of the requirements engineering process. It is a semi-formal or formal description of the requirements. It could contain written documentation, graphical models, mathematical models, use cases, prototypes, etc.
Agile and iterative design methods often have less formal specification requirements.
Requirements specification must be complete, consistent, clear, and correct.

- Complete: all possible scenarios in the system need to be described.
- Consistent: no requirements can contradict another requirement.
- Unambiguous: exactly one system must be defined, with no other possible interpretations.
- Correct: must accurately describe what the customer wants and what the developers will build.

Requirements specification should be realistic, verifiable and traceable.

- Realistic: the system must actually be buildable and implementable. If it's an unrealistic requirement, this has to be explained to the customer.
- Verifiable: repeatable tests can demonstrate that the system meets the requirement. This is done with acceptance tests.
- Traceable: every requirement should be able to be traced through the entire development process. One should track dependencies so that if one part of the code is changed, changes to the other parts of the code can be tracked as well.