Lesson 1:

* **Describe the challenges in dealing with requirements for software and systems**
  + **Natural language** problems result from the ambiguities and context sensitivity of natural (human) languages.
  + One of the greatest difficulties in dealing with the requirements engineering activities, particularly elicitation and representation, for most systems is that they are **complex**.
    - Wicked Problems (Economic, political, societal)
      1. No definitive definition of a wicked problem
      2. No stopping rule
      3. Solutions are NOT true/false, but good/bad
      4. No immediate/ultimate test for a solution
      5. No way to trial-and-error, every attempt has significant impact
      6. No set of solutions or set of “well-described” steps that can be incorporated into a plan
      7. Each one is unique
      8. Each one is considered to be a symptom of another problem
      9. Discrepancies in representation can have multiple reasons and the choice of reason will determine resolution
      10. The designer has no right to be wrong
  + **Gold plating** requirements are possible to implement, but would be unlikely to use (i.e self flying helicopter) **Ridiculous Requirements** are requirements that would be difficult to deliver at the time they are proposed
  + **Obsolete requirements** are previously developed requirements that no longer applies to the system under consideration, whether it is because the requirement has changed or because the system no longer needs that requirement
    - **Dubious requirements** have no known origin or serve an unknown purpose
  + **Four Dark Corners**
    - 1. The terminology used for requirements engineering should be grounded in the reality for which the machine will be built
      2. It is unnecessary to describe the machine to built, rather how the environment will be because of the system **WHAT the system will do, not HOW**
      3. Actions should be separated into those controlled by the environment, the machine, and shared.
      4. Domain knowledge will help support the refinement of requirements into implementable specs
* **Explain the activities of the requirements engineer throughout a system’s lifecycle.**
  + **Elicitation/Discovery:** uncovering the customer’s wants and needs
  + **Analysis and Agreement:** Techniques to deal with the problems of “raw” requirements
  + **Representation:** Converting raw requirements into some model (Natural language, mathematical, visual).
    - **Informal:** nat language, sketches, diagrams
    - **Formal:** mathematically sound representations
    - **Semi Formal:** Convertible to a formal model with the addition of some semantic framework
    - **Validation:** Ensuring the requirements are a correct representation of the customer’s needs. (Am I building the right product?)
    - **Management:** Managing the realities of the changing requirements over time. Foster traceability and communicating changes those in the need-to-know
      1. Help to prevent scope creep
  + **Bodies of Knowledge:**
    - Software Engineering Body of Knowledge (SWEBOK 2014)
    - Graduate Software Engineering Reference Curriculum (GSwE 2009)
    - Principles and Practices of Software Engineering Exam Specification
* **Construct an effective system mission statement (concept of operations).**
  + Understand the motivation or objective for building teh system
  + Understand the environment in which the system is to be built, deployed, and used
* **Analyze and communicate effectively with customers and other stakeholders.**
  + Manage expectation
  + Understand the customer’s wants and what they don’t want
  + Customer’s will change their mind. They can’t see or consider every possible desired piece of the system. Sometimes the environment changes. Sometimes the ROI of the requirement changes during discovery. Customers may withhold information from you for many reasons
* **Identify hazards, relevant standards, and frameworks**

Lesson 2