# Important Points

* “Software requirements are not really a discrete area of software engineering. Rather, they are part of the systems engineering process known as requirements engineering (RE).” [125]
* “A requirement defines a property or capability that must be exhibited by a system in order for it to solve the business problem for which it was conceived. Thousands of requirements, of different kinds and at different levels of abstraction, may be needed to specify a system that addresses a challenging business problem.” [125]
* “One of the main tasks of requirements analysis is to elaborate the system requirements to discover more about the implications of satisfying them. This involves deriving new, lower-level requirements (derived requirements) that focus more on detailed technical issues.” [126]
* “Constraints are like negative requirements. They act to limit the set of possible solutions to the business problem.” [126]
* Steps in the requirements engineering process include: eliciting requirements [127], analyzing requirements [127], documenting requirements [127], validating requirements [127], and managing requirements [127].
* The IEEE standards IEEE Std 1362-1998, IEEE Std 1233-1998, and IEEE Std 830-1998 imply the following steps in the requirements engineering process: scoping the system [127]; system requirements elicitation, analysis, and validation [127]; creation of the systems requirements specification document [127]; system architecture development [127]; and the creation of the software requirements specification document for each software component [127].
* The requirements engineering process must “minimize the risk of unexpected requirements change” [127] due to the high risks associated with changing requirements.
* When working with an incremental development process the requirements engineering process can be adapted into an iterative process [128].
* Requirements elicitation “is not a linear process.” [129]
* “The key point is that ‘elicitation’ –the name routinely applied to this activity –is misleading. Requirements are not simply elicited fully formed from stakeholders. They have to be *discovered*.” [129]
* “Most stakeholders are busy people, so it is vital to find people who are motivated to act as *product champions* and to negotiate access to them.” [129]
* “Users’ stories or scenarios can provide a valuable tool” that is used to elicit requirements. [130]
* “Competing products and emerging trends and technologies are also likely to be important sources of requirements.” [130]
* “The analysis activity should yield a baseline set of requirements.” [131]
* “Within the system boundary are all the aspects of the problem for which the proposed system will provide a solution.” [131]
* “Models help explore the requirements by providing alternative depictions of the problem to that provided by verbal or textual descriptions.” [132] “Modeling helps uncover the detail necessary for synthesis of the derived requirements.” [132]
* Understanding the implications of system requirements is one motivation for deriving more detailed requirements. The other reason is to add specificity for the developers.” [133]
* “The process of deriving requirements is driven largely by the requirements engineers’ experience.” [133]
* “Derived requirements must have the requirements from which they are derived recorded.” [133]
* Stakeholders need to accept that some requirements will be cut from the project [134].
* “The requirements engineer should seek the most precise form of description that is consistent with readability and the overall coherence of the document.” [135]
* “Requirements validation can be crudely characterized as ensuring correctness.”
* “It is highly desirable to have a formal point in the process at which the requirements are frozen and agreement sought from all the stakeholders.” [136]
* “Weigers defines requirements management as comprising four tasks: change control, version control, requirements tracing, and status tracking.” [136]
* “Requirements creep is a well-known phenomenon of software projects. Here, uncontrolled or ad-hoc changes to the requirements make project plans impossible to manage, inevitably resulting in systems that are late and over budget.” [136]
* “Requirements must be traced. As a minimum, this means that the derivation relationships between requirements should be recorded.” [137]
* “If [requirements engineering] is neglected or done badly, it will ensure that the project subsides into a mire of misunderstood needs and uncontrolled changes.” [138]

# Disagreements

“[The system specification document] must be readable by the system stakeholders to enable them to validate the requirements and approve them (sign them off) as the basis for subsequent development. The SRS, by contrast, is primarily a technical document aimed at developers.” [134]

The system specification document is not the basis for development. The software requirements specification is the document used to form a contract for development [Cleland-Huang 119] and is the basis for development. As such, the software requirements specification is not primarily aimed at developers and must be written to be accessible to stakeholders. [Cleland-Huang 119]

# Questions/Clarifications

I don’t understand what a “black-box system” [126] is.