CS340d - Place in the Context of Systems Design, Verification and Validation

A comment in three parts.

Part 1: There is a large body of literature and practice on collecting and using requirements in systems design. Unfortunately almost all of it is process management focused and not concerned with a formal-logical approach of the type we were introduced to in this course.To see this you only have to do a little searching in the internet. Try the following.

Use your favorite Search software, and look at what comes back from from the following searches:

Search 1. Software Requirements, Software Requirements Example

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfill the needs of all stakeholders (business, users).

Software Requirements, 3rd Edition

by Karl Wiegers, Joy Beatty
Released August 2013
Publisher(s): Microsoft Press
ISBN: 9780735679658

A reference is the *IEEE std 830-1998* - IEEE Recommended Practice for Software Requirements Specifications.

Categories of requirements your SRS should have, to name a few.

Functional Requirements
User Interface Requirements
Capability Requirements
Software Inputs and Outputs and Data Requirements
Software System Interface Requirements
Security Requirements
Risk Control Requirements
Installation and Maintenance Requirements

So what makes a good SRS?

Every requirement must be unambiguous and correct. Nothing should be left to interpretation.

As you can see from the above there is plenty of descriptive and definitional information. In addition, there are textbooks, IEEE standards and so on. Now perform search 2 and search 3. This will show information on how large organizations (NASA) and even whole industries (automobile) are using requirements in their product development processes.

Search 2. NASA Software Requirements Specification

A lot of information can be found, but one item in particular caught my attention. I thought it was interesting to hear NASA's view on verifiability and testing.

Verifiability/Testability

- Can the system be tested, demonstrated, inspected, or analyzed to show that it satisfies requirements? Can this be done at the level of the system at which the requirement is stated? Does a means exist to measure the accomplishment of the requirement and verify compliance? Can the criteria for verification be stated?
- Are the requirements stated precisely to facilitate specification of system test success criteria and requirements?
- Are the requirements free of unverifiable terms (e.g., flexible, easy, sufficient, safe, ad hoc, adequate, accommodate, user-friendly, usable, when required, if required, appropriate, fast, portable, light-weight, small, large, maximize, minimize, sufficient, robust, quickly, easily, clearly, other "ly" words, other "ize" words)?

Search 3. Automotive Software Requirements Specification

Software Requirements Specification (SRS) Hands-Free ...

Michigan State University
https://cse.msu.edu › ~skidmo25 › srs-v2

The **Software Requirements Specification document** serves the purpose of guiding one through the entirety of the Hands-Free Driving System (HFDS) and all of its …

Software Requirements Specification (SRS) Pedestrian ...http://www.cse.msu.edu › Resources › Software...

http://www.cse.msu.edu › Resources › Software...

This Software Requirements Specification (SRS) document is **to produce a comprehensive description of all software required to complete the project**.

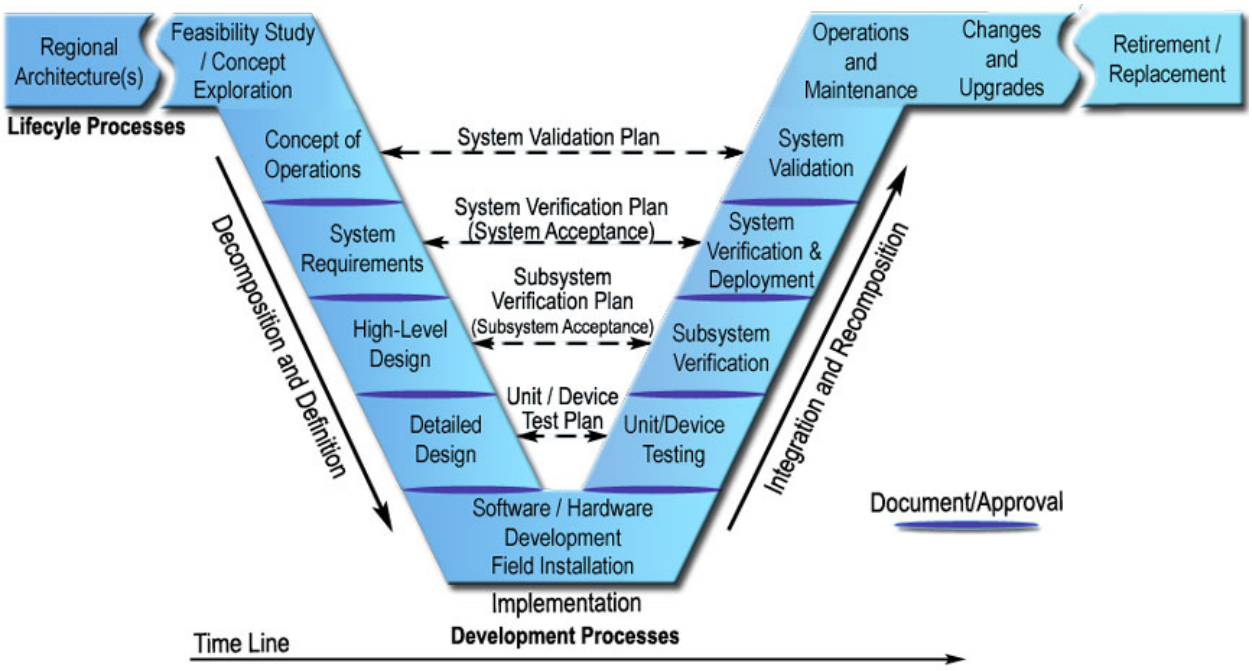
The search with the automotive keyword added yielded examples of requirements for development features such as the two above: Hands-free Driving System, and Pedestrian Safety. It is common in the automotive industry to add new software-enabled capabilities that use the existing sensor/actuator hardware on the vehicle.

Part 2: The requirements/specification process typically operates within existing system engineering processes. one such process is shown below, called the systems engineering V "Vee". This process is widely used, and for example is in use in government organizations such as NASA and Department of Transportation; in the automotive industry and others.

The following reference talks about both the use of formal methods, the requirements/specifications process and needed improvements to the system engineering V. Dr. Rozier's paper on the state of Specification in the aerospace sector is well worth a read. Also take a look at the system engineering V diagram and see if you can locate where in that process work, of the type we have been doing in CS340d, would take place. It is interesting to note that Dr. Rozier worked in formal methods while at NASA.

Kristin Yvonne Rozier. “Specification: The Biggest Bottleneck in Formal Methods and Autonomy.” In Proceedings of the 8th Working Conference on Verified Software: Theories, Tools, and Experiments (VSTTE), volume 9971 of Lecture Notes in Computer Science (LNCS), pages 1–19, Springer-Verlag, Toronto, Canada, July 17–18, 2016. (Invited)

System Engineering V



Part 3: Well it took along time to get here, but we are finally ready to put the topics of this course, debugging and verifying in the context of large-scale system design. Consider the graphic titled “Context for Debugging, Verification and Validation (V & V)”. You can see that debugging and verification are at the heart of the program. You can also see that when debugging is done you have a working object that can be verified. Verification gives you the design confirmation that the requirements are met. Once you have a debugged artifact (it runs), and have a confirmation that the working object meets the system requirements you are ready to move to validation. Validation confirms that the requirements for intended use in application are fulfilled.