Introduction to Plan-Driven Requirements

CS 2720

Requirement Types

We can divide requirements into one of two categories: *functional* and *nonfunctional*.
A *functional* requirement:

- specifies *what* the system should do
- "...statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations." (Sommerville, Ch. 4)
- "...defines a function of a system or its component." (Wikipedia)

Requirement Types

A *nonfunctional* requirement:

- specifies/constraints *how* the system should do it
- "...constraints on the services or functions offered by the system." (Sommerville, Ch. 4)
- "...a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors." (Wikipedia)

Nonfunctional requirements can come from *product requirements*, *organizational requirements*, or *external requirements*.

Example Requirements

- A user shall be able to search the appointments list for all clinics.
- Medical staff shall be able to use all the system functions after two hours of training. After this training, the average number of errors made by experienced users shall not exceed two per hour of system use.
- The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.

The Software Requirements Specification

Recall our waterfall process saves the requirements in the *software requirements specification* (SRS).

Several outlines ("templates") for the SRS are given in the IEEE 830-1998.

The updated IEEE standard (ISO/IEC/IEEE 29148) contains a sample template as well.

Review

According to the ISO/IEC/IEEE 29148:2011, a good requirement (or *set of requirements*) should be:

- Necessary
- Implementation Free
- Unambiguous
- Consistent
- Complete
- Singular
- Feasible
- Traceable
- Verifiable
- *Complete*
- *Consistent*
- *Affordable*
- *Bounded*