

NONFUNCTIONAL REQUIREMENTS EXAMPLE: SATWATCH

- SatWatch is a wrist watch that:
 - uses GPS satellites to determine its location and displays the time based on its current location.
 - uses internal data structures to convert this location into a time zone.
 - never requires the owner to reset the time due to the information it stores and its accuracy.
 - has no buttons or controls available to the user since it adjusts the time and the date displayed as the watch owner crosses time zones.
 - assumes that it does not cross a time zone boundary during a GPS blackout period, but adjusts its time zone as soon as possible after the blackout period.
 - has a two-line display showing, on the top line, the time (hour, minute, second, time zone) and on the bottom line, the date (weekday, day, month, year).
 - has a readable display even under poor light conditions.
 - can have its software upgraded using the WebifyWatch device (provided with the watch) and a personal computer connected to the Internet.

NONFUNCTIONAL REQUIREMENTS EXAMPLE:

SATWATCH (cont'd)

- Any user who knows how to read a digital watch and understand international time zone abbreviations should be able to use SatWatch.

Interface
Usability

- As SatWatch has no buttons, no software faults requiring the resetting of the watch should occur.

Design Quality
Reliability

- SatWatch should accept upgrades to its onboard processor via the USB interface.

Design Quality
Supportability

NONFUNCTIONAL REQUIREMENTS EXAMPLE:

SATWATCH (cont'd)

- SatWatch should display the correct time zone within 5 minutes of the end of a GPS blackout period.
- SatWatch should measure time within 1/100th second over 5 years.
- SatWatch should display time correctly in all 24 time zones.

Performance
Response time

Performance
Accuracy

Performance
Correctness

NONFUNCTIONAL REQUIREMENTS EXAMPLE:

SATWATCH (cont'd)

- All related software associated with SatWatch will be written using Java.
- SatWatch complies with the physical, electrical, and software interfaces defined by WebifyWatch API 2.0.

**Implementation
Language**

**Interface
Format**

REQUIREMENTS VALIDATION EXAMPLE: SATWATCH

Incompleteness

Problem: The SatWatch specification does not specify the boundary behavior when the user is standing within GPS accuracy limitations of a time zone's boundary.

Solution: Add a *functional requirement* stating that the time zone depicted by SatWatch should not change more often than once every 5 minutes.

REQUIREMENTS VALIDATION EXAMPLE: **SATWATCH** (cont'd)

Inconsistent

Problem: SatWatch software should not have bugs nor need to be upgraded.

SatWatch software should be easily upgraded using the USB interface.

Solution: Revise one of the conflicting requirements.

REQUIREMENTS VALIDATION EXAMPLE: SATWATCH (cont'd)

Unclear

Problem: The SatWatch specification refers to time zones.

Does the SatWatch deal with daylight saving time or not?

Solution: Clarify the ambiguous requirement (e.g., add a requirement that SatWatch should deal with daylight saving time).

REQUIREMENTS VALIDATION EXAMPLE: **SATWATCH** (cont'd)

Incorrect

Problem: SatWatch supports supports only 24 time zones (24 hours).

There are more than 24 time zones. Several countries and territories are half an hour ahead of a neighboring time zone.

Solution: Change the requirement to support all time zones.