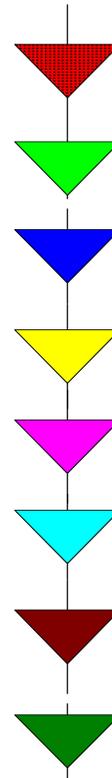


# **The 8 Characteristics of a good user requirements**

# What Are the 8 Characteristics of Good User Requirements?

**A user requirement is good if it is:**

1. Verifiable
2. Clear and concise
3. Complete
4. Consistent
5. Traceable
6. Viable
7. Necessary
8. Implementation free



**Think of these characteristics as a series of filters. A good requirement will pass through all eight filters.**

# What Makes a UR Verifiable?

## A verifiable requirement ...

- is stated in such a way that it **can be tested** by:
  - inspection,
  - analysis, or
  - demonstration.
- makes it **possible to evaluate** whether the system met the requirement, and
- is verifiable by **means that will not contaminate** the product **or compromise** the data integrity.

# Is this UR Verifiable?

- **Bad example:**

- **UR1: The system must be user friendly.**
- **How should we measure user friendliness?**

- **Good example:**

- **UR1: The user interface shall be menu driven. It shall provide dialog boxes, help screens, radio buttons, dropdown list boxes, and spin buttons for user inputs.**

# What Makes a UR Clear & Concise?

## A clear & concise requirement ...

- must consist of a **single requirement**,
- should be no more than **30-50 words** in length,
- must be **easily read and understood** by non technical people,
- must be **unambiguous** and not susceptible to multiple interpretations,
- must **not contain** definitions, descriptions of its use, or reasons for its need, and
- must **avoid** subjective or open-ended terms.

# Is this UR Clear & Concise?

- **Bad example:**

- UR2: All screens must appear on the monitor quickly.

- How long is quickly?

- **Good example:**

- UR2: When the user accesses any screen, it must appear on the monitor within 2 seconds.

# What Makes a UR Complete?

## A complete requirement ...

- contains **all the information** that is needed to define the system function,
- leaves **no one guessing** (For how long?, 50 % of what?), and
- includes **measurement units** (inches or centimeters?).

# Is this UR Complete?

- **Bad example:**

- UR3: On loss of power, the battery backup must support normal operations.

- For how long?

- **Good example:**

- UR3: On loss of power, the battery backup must support normal operations for 20 minutes.

# What Makes a UR Consistent?

## A consistent requirement ...

- **does not conflict** with other requirements in the requirement specification,
- uses the **same terminology** throughout the requirement specification, and
- **does not duplicate** other URs or pieces of other URs or create redundancy in any way.

# Is this UR Consistent?

- **Bad example:**

- UR4: The electronic batch records shall be Part 11 compliant.
- UR47: An on-going training program for 21 CFR Part 11 needs to be established at the sites.
- **Do these refer to the same regulation or different ones?**

- **Good example:**

- UR4: The electronic batch records shall be 21 CFR Part 11 compliant.
- UR47: An on-going training program for 21 CFR Part 11 needs to be established at the site.

# What Makes a UR Traceable?

## A traceable requirement ...

- has a **unique identity** or number,
- **cannot be separated** or broken into smaller requirements,
- can **easily be traced** through to specification, design, and testing.
- **Change Control** on UR level.

# Is this UR Traceable?

- **Bad example:**

- **UR: The system must generate a batch end report and a discrepancy report when a batch is aborted.**
- **How is this uniquely identified? If the requirement is changed later so that it does not require a discrepancy report, how will you trace it back so you can delete it?**

- **Good example:**

- **UR6v1: The system must generate a batch end report when a batch is aborted.**
- **UR7v2: The system must generate a discrepancy report when a batch is completed or aborted.**

# What Makes a UR Viable?

## A viable requirement ...

- can be **met using existing technology,**
- can be **achieved within the budget,**
- can be **met within the schedule,**
- is something the organization has the **necessary skills to utilize,**
- will be **used by the end users,** and
- must be **helpful to build the system.**

# Is this UR Viable or Feasible?

- **Bad example:**

- The replacement control system shall be installed with no disruption to production.

- **This is an unrealistic expectation.**

- **Good example:**

- The replacement control system shall be installed causing no more than 2 days of production disruption.

# What Makes a UR Necessary?

## A necessary requirement ...

- is one that **must be present to meet system objectives**, and
- is **absolutely critical** for the operation of the system,
- leads to a **deficiency in the system if it is removed**.

# Is this UR Necessary?

- **Bad example:**

- All desktop PCs for the project must be configured with 512MB of memory, DVD ROM/CD-RW multifunction drive and a 21-inch flat screen monitor.
- **This may not be needed for all PCs for the project.**

- **Good example:**

- The desktop PCs for the developers on the project must be configured with 512MB of memory, DVD ROM/CD-RW multifunction drive and a 21-inch flat screen monitor.

# What Makes a UR Free of Implementation Details?

**A requirement that is free of implementation details ...**

- defines **what functions are provided** by the system,
- **does NOT specify how** a function can or should be implemented, and
- allows the **system developer to decide what technology is best** suited to achieve the function.

# Is this UR Free of Implementation Details?

- **Bad example:**

- After 3 unsuccessful attempts to log on, a JavaScript routine must run and lock the user out of the system.
- **Specifying a JavaScript routine concerns how the requirement will be implemented.**

- **Good example:**

- After 3 unsuccessful attempts to log on, the user must be locked out of the system.