**Atomic Requirements – First Example**

As a brief example, consider a basic system login screen. A basic initial requirement statement might be: “Login - System shall allow login with userid and password, logout, and password reset”.

As several functions or activities are included or assumed in the above, a more atomic set of statements would be those shown in Table I (only partially covering everything in the initial statement).

Intuitively, the list in Table I seems to be more singular or atomic. Requirement 4 also adds a non-functional requirement that was not explicit in the original statement.

To go beyond the intuitive feeling, metrics may be a way to assist the requirements writer. If these metrics can be used to measure the atomicity of requirements statements during requirements generation, they can become a tool for the requirements writer to use.

1. Partial Set of Atomic Requirements

| Identifier | Title | Login Requirement Statements |
| --- | --- | --- |
| Req 1 | Login Method | System allows users to login by providing **UserId** and P**assword** at the **LogInScreen**. |
| Req 2 | Password Change | Users may change their current **Password** at any time after login is successful. |
| Req 3 | Password Reset | Users who are unable to login successfully after three attempts are allowed to reset their **Password**. |
| Req 4 | Password Security | User **Password** is not stored in clear text anywhere in System. |
| Glossary |  | Terms appearing in **PascalCase** are defined in separate glossary with format, encodings, etc. |

Atomic Requirements Metrics – Working Proposal

The set of requirements metrics defined here have evolved in an academic setting and been used to teach atomic requirements to students.

# Metrics for Atomic Requirements

Given the potential represented by atomic requirements, and despite the difficulties of the imprecise definition, it is nevertheless attractive to measure and evaluate them with metrics. The metrics set described here is intended to work with any set of (supposedly) atomic requirements describing some application, system component, or subsystem. It is designed to measure the requirements quality in general (not solely if they are proper atomic requirements or not).

These metrics are divided in two categories: those applied to an individual requirement statement and those which measure the complete set of requirements. The values of each individual metric are assigned during an inspection of the full set of requirements (for an overview of the inspection process used to date, see the Section IV).

The value for each of the metrics is an integral number from 1 to 10. This range of values was picked arbitrarily to ensure adequately wide spread of metric values.

Although not covered in detail here, these metrics and the process used, assume a system or application glossary exists. Key terms used in various requirements are defined once with all details in the glossary to ensure consistent use across the separate requirements statements. The glossary includes details on length, size, capacity, units, and format when relevant to the defined terms.

## Metrics For a Single Atomic Requirement

Table II presents a working set of metrics to measure each atomic requirement individually, giving a high-level description of each. This section describes the motivation for these metrics and further details on the current methods for measuring them.

The first two metrics, Requirement Correctness (Ra1) and Requirement Unambiguity (Ra2), are like oft-used terms for describing requirement quality [5], [7]. They have their usual definitions and are evaluated in the usual way. For purposes of this paper, which chiefly looks at the issues raised with atomic requirements they will be only briefly discussed.

1. Metrics for Individual Requirements

|  | Metric | Brief Definition | Range |
| --- | --- | --- | --- |
| Ra1 | Requirement Correctness | Is individual requirement properly defining a genuine system function and need? | 1-10 |
| Ra2 | Requirement Unambiguity | Is the individual requirement clear and understandable to the expected users of the document? | 1-10 |
| Ra3 | Requirement Atomic Completeness | Does the individual requirement include everything necessary to fully understand the desired functionality? | 1-10 |
| Ra4 | Requirement Atomic Verifiability | How adequately can this individual requirement be tested with a result able to show 100% passed or failed? | 1-10 |
| Ra5 | Retirement Atomic Undecompos-ability | Would further breakup of this requirement into separate parts be extremely difficult or detract from understandability? | 1-10 |

The Requirement Correctness (Ra1) metric measures the legitimacy and genuine need for the capability or features described. For a high-quality measure this metric would use a formal requirements verification process (in a simpler, educational environment, the inspection team supplies a subjective rating). Requirements Correctness must consider whatever information the requirement provides on priority, importance, optionality, etc.

The Requirement Unambiguity (Ra2) metric evaluates the understandability of the requirement for intended audience(s) including stakeholders, developers, testers, etc. The goal is to prevent differing interpretations of the requirement by different readers; the system glossary supports this evaluation.

These two metrics are included in the individual metrics set to ensure that the overall value of the requirement is analyzed. The remaining three metrics address particularly the atomicity of the individual requirement.

### Requirement Atomic Completeness (Ra3)

The Requirement Atomic Completeness (Ra3) metric determines if everything that belongs in the requirement is present. A simple chemistry analogy with an element shows the intent of this metric – to completely describe an element includes listing things such as its symbol, atomic number, common isotopes, atomic weight, etc. It’s unlikely that an element would be completely described without each of these.

Of course, potential atomic requirements are nowhere near as indivisible as chemical elements. The goal for the requirement inspection process is to measure how well a single requirement statement meets the completeness goal; to do so the following checklist is used:

* Are all likely varieties and values of inputs covered including clear indications of which are legitimate and which are invalid?
* Does the requirement cover all possible variations and sub cases for the feature or function, and clearly specify which ones are to be handled?
* Is it clear what outputs, changes in system state, and other results must or may be expected?
* Is there nothing missing to make the requirement completely describe a single function or feature; are there any “What about this…” questions that can still be asked?
* Does the glossary include every term necessary to understand this requirement; are each of these terms defined fully in the glossary?

Currently, no individual values or weights are provided with this checklist. The reviewers determine a single metric value for the requirement statement after considering all the points in the checklist. The stronger agreement or belief that the statement is true yields a higher value for the metric.

### Requirement Atomic Verifiability (Ra4)

The Requirement Atomic Verifiability (Ra4) metric evaluates the atomicity of a single requirement by measuring the likelihood that completing suitable test cases would indicate that the requirement was fully met or not met. In other words, a truly atomic requirement can be tested as a single unit and give a definitive pass or fail result. Returning to the chemistry analogy most elements can be tested to determine what they are – it is either carbon or it’s not.

In other words, the metric measures the degree to which the tests are bound together and mutually interdependent. This checklist supports the inspection (using the same one to ten scale where ten means “most likely”, “most difficult”, etc.):

* How obvious is it what to do first in the test case (where to begin)? Are the inputs or stimulus to begin the test easy to determine?
* How well defined are the outputs or results of the feature or function? Are the values or state changes clearly determined (at least within each alternative)?
* Would it be difficult or impossible to skip one or more test cases (or steps in a test case) and still determine if the test passes (for at least some alternatives in the requirement)?
* How obvious is/are the test case(s) needed to test this requirement (if no test cases now exist)?

This checklist is, of course, easier to evaluate if the test(s) are already defined and documented and the tests are traceable back to the requirements they evaluate. Otherwise, the inspection team must use subjective evaluation to determine this metric.

### Requirement Atomic Undecomposability (Ra5)

The Requirement Atomic Undecomposability (Ra5) metric determines if anything can be removed from the requirement (and likely put elsewhere in another atomic requirement). In other words, would removing some part of the requirement statement leave an inconsistent or ill defined description of the function. In basic chemistry, elements retain their atomic number, etc., even when take part in reactions with other materials.

A general term for this characteristic might be stickiness or cohesion; a checklist to prompt thinking includes these questions:

* Is everything in the requirement necessary and important for it to be understandable? In other words, if something were to be removed, would the requirement no longer make sense or become incomplete?
* Is it likely that separating the requirement into two or more parts would be difficult, less understandable, or be likely to cause redundancy?
* Would a user or customer be able to express agreement or disagreement with the requirement with a clear “yes” or “no”?
* Would it be difficult to partially implement this requirement and achieve a working application or system?

When requirements have been defined with use cases, an atomic requirement will be tied to a single use case and a complete system event or interaction between the environment and the system. Automated processing of requirements statements to test if they are decomposable may also by possible [17].

## Metrics for a Set of Requirements

Table III lists metrics which measure a set of requirements – typically the complete requirements document for a system, subsystem, or application. These metrics are based on commonly used terms for requirements quality [7] and are not new or specific to atomic requirements; however, the methods for evaluating them are adapted and specialized to ensure atomicity is part of the metric. Each metric is valued with a single number.

Requirement Completeness (Rd1) evaluates the set of requirements statements for completeness. When used with a set of atomic requirements this evaluation offers a way to determine if any other singular requirement needs to be added. The granularity of atomic requirements may make this test easier – it becomes a simple question: shall another requirement be added to the set?

Requirement Consistency (Rd2) is also more straightforward when based on atomic requirements. Although it may be conceptually challenging, the metric can be calculated by looking at every possible pair of two requirements and asking if those two are consistent. The metric value is then reduced for any pair which raises concerns of compatibility or consistency.

Requirement Importance Ranking (Rd3) tests the existence of an importance ranking (e.g. essential, desirable, optional [18]) and subjectively evaluates the appropriateness of the rankings stated. Atomic requirements lend themselves to ordering and counting of requirements by their individual rankings; this comparison can support selecting a value for this metric (e.g. when the majority of requirements are all “essential” it may not genuinely represent what’s needed).

1. Metrics for Set of Requirements

|  | Metric | Brief Definition | Range |
| --- | --- | --- | --- |
| Rd1 | Requirement Completeness | Is this set of atomic requirements complete – does it provide a full definition of functionality for the system or subsystem? | 1-10 |
| Rd2 | Requirement Consistency | Is this set of atomic requirements internally consistent, with no contradictions, no duplication between individual requirements? | 1-10 |
| Rd3 | Requirement Importance Ranking | Are each of the atomic requirements individually assigned suitable importance categories and are the assignments appropriate? | 1-10 |
| Rd4 | Requirement Traceability | Are each of the of atomic requirements uniquely identified with unchanging identifiers? | 1-10 |
| Rd5 | Requirement Purity | Is this set of atomic requirements free from system design, project schedule, staffing, and other non-requirements material? | 1-10 |

Requirement Traceability (Rd4) evaluates the ability of the requirements to support typical approaches to traceability (linking to design, tests, and system changes). Each atomic requirement should have a unique and unchanging identification (such as a sequential number). Document section or paragraph numbers do not meet this need as they change often. When clearly delineated atomic requirements are used with a suitable naming scheme, this metric requires little subjective thought. The metric can be given the maximum value if the scheme is used consistently through the entire set (and the value reduced for omissions, duplications, or other defects). An expanded definition, and more consistent evaluation of this metric would be possible using models of the traceability [19].

Requirement Purity (Rd5) measures the appropriateness of the requirement set as pure statements of system need without inappropriate details about design, implementation, schedule, etc. While atomic requirements do support clear linkage with design choices, inclusion of design within the requirement set makes it difficult to clearly delineate separate atomic requirements. For example, many requirements may imply that information will be stored or updated in a typical relational data base to support information queries; however, the structure and keys for that data table likely would be implied across many separate requirements.

While not a metric shown in Table III, the count of individual atomic requirements is a valuable piece of information as well. Well-structured and complete atomic requirements, due to their singularity and specificity, are likely to each require a significant and separable amount of design, implementation, and testing during development. While very approximate, the total requirement count provides an indication of system size; changes in the number offer a clear insight into requirement churn.