



eCOMMONS

Loyola University Chicago  
Loyola eCommons

---

Computer Science: Faculty Publications and Other  
Works

Faculty Publications

---

4-2016

# Introduction to Atomic Requirements

William L. Honig

Loyola University Chicago, [whonig@luc.edu](mailto:whonig@luc.edu)

---

## Recommended Citation

Honig, William L.. Introduction to Atomic Requirements. , , , 2016. Retrieved from Loyola eCommons, Computer Science: Faculty Publications and Other Works,

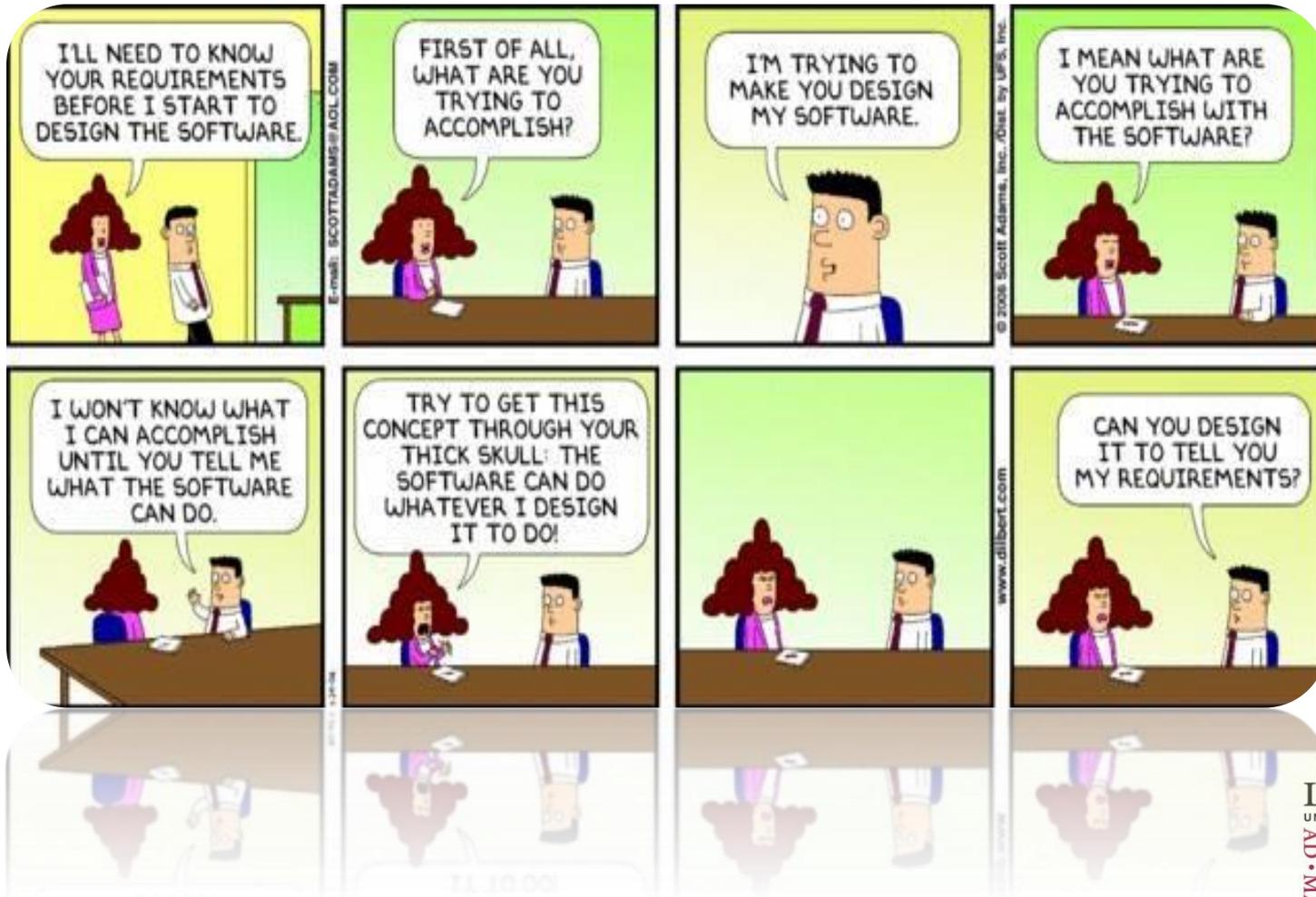
This Presentation is brought to you for free and open access by the Faculty Publications at Loyola eCommons. It has been accepted for inclusion in Computer Science: Faculty Publications and Other Works by an authorized administrator of Loyola eCommons. For more information, please contact [ecommons@luc.edu](mailto:ecommons@luc.edu).



This work is licensed under a [Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 License](https://creativecommons.org/licenses/by-nc-nd/3.0/).

© 2016 William L. Honig

# Introducing Atomic Requirements...



**Software Engineering**  
**Guest Lecture**  
**19 April 2016**

**William L. Honig, Ph.D.**  
**Associate Professor**  
**Department of Computer Science**  
**Loyola University Chicago**  
**Visiting Researcher, Keio University**  
**Tokyo, Japan**



# Requirements == Hard Work

- No single approach works all the time
- Most common cause of development failure is incorrect, incomplete, inconsistent (or non existing) requirements



- Agile developers may try to avoid creating requirements
  - If you don't know where you're going, any road will do!



# Objectives ...

1. Introduce Atomic Requirements
2. Why they are Useful
3. Current State of Understanding

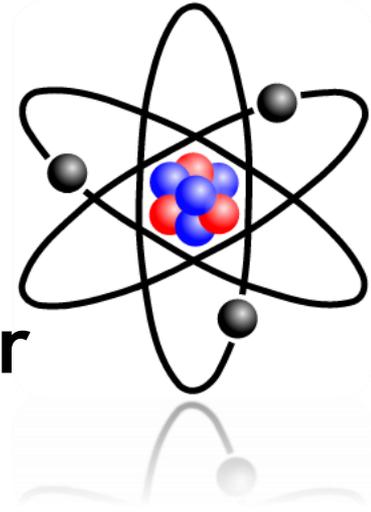


**Atomic** →  
A single complete requirement documented as a whole (i.e., indivisible or atomic)

Why? Know what “A Requirement” is?  
Then can track, count, number,....



# Initial Definition



- A **single complete** requirement documented as a whole (indivisible or **atomic**)
- IEEE Standard 29148 Systems and software engineering – Life cycle processes – Requirements
  - Requirements should be “**singular**” and include only one requirement with no use of conjunctions
- Also **Individual, Single, Complete, and Cohesive** Requirement
- No standard definition exists



# More Definition...

One atomic requirement completely describes a single function, feature, need, or capability, including all information, details, limits, and characteristics.



A single feature or function from a single Use Case

How to and hints:

1. Use judgement / common sense
2. Remember the goal is UNDERSTANDABILITY
3. When in doubt, more individual requirements instead of larger and broader statements



# What do we want?



- IEEE Standard 29148 Systems and software engineering – Life cycle processes – Requirements
- Individual Requirements (i.e. atomic) should each be (see section 5.2.3)
  - Necessary
  - Implementation Free
  - Unambiguous
  - Consistent
  - Complete
  - Singular
  - Feasible
  - Traceable
  - Verifiable

**Atomic** →

Supports many of these characteristics and other engineering processes



# Why might we want to use Atomic Requirements?

Atomic requirements support other aspects of SE especially metrics and quality

## Numbering

- Unchanging numbered requirements provide traceability, reference integrity, measure, ...

## Consistent

- Individual atomic requirement can be tested for consistency with all other requirements

## Singular

- Atomic is singular

## Complete

- Everything relevant to a single capability is there

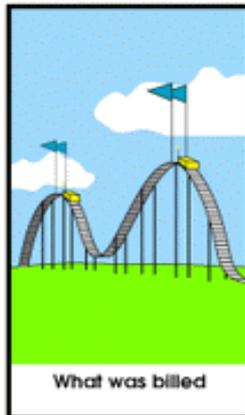
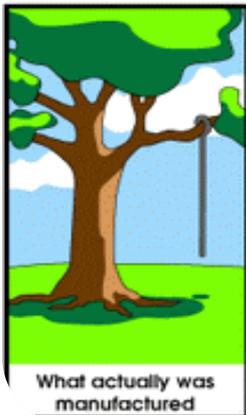
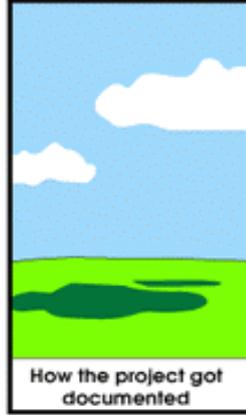
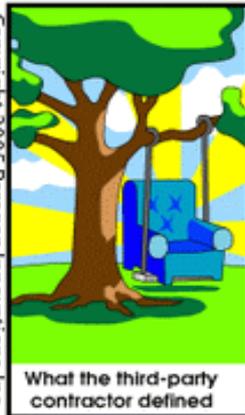
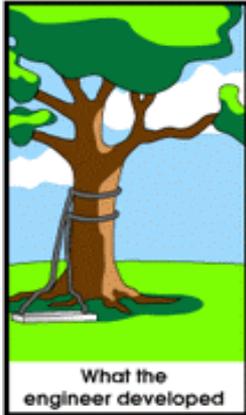
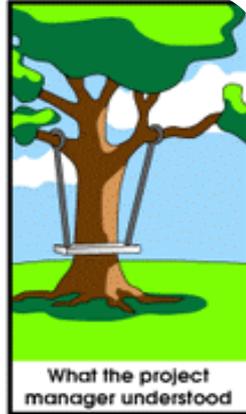
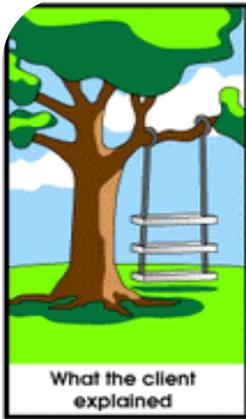
## Verifiable

- Trace and reference history of changes, tests, validations...

## Necessary

- An atomic requirement can be ranked for importance and included / removed from plan





**Remember,  
Requirements are  
Hard to Do Well!**



# Let's Try It

Write atomic  
requirement(s) for a  
typical login screen



# Summary ...

- Atomic Requirements may help improve requirements creation

- Metrics and traceability easier
- Focus your requirements thinking
- BUT, the concept is new, not completely clear

- You are **NOT REQUIRED** to use or generate atomic requirements

- It is a decision for your team

