

Cosmic Truths About Software Requirements



Karl E. Wiegiers
PROCESS IMPACT 
www.processimpact.com

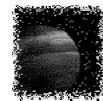
Sponsored by Brewer Consulting Services, Developer
of the CARDS Requirements Repository



Copyright © 2005 by Karl E. Wiegiers

Cosmic Truth #1

*If you don't get the
requirements right, it
doesn't matter how
well you execute the
rest of the project.*



Cosmic Truth #2

Requirements development is a discovery and invention process, not a collection process.



Cosmic Truth #3

Nowhere more than in the requirements process do the interests of all the project stakeholders intersect.



Requirements Stakeholders



Cosmic Truth #4

Customer involvement is the most critical factor in achieving software quality.



Obtaining Customer Involvement

- ◆ Identify user classes
- ◆ Select “product champions”
- ◆ Employ the use case technique
- ◆ Understand how product will be used
 - ✓ scenarios
 - ✓ operational profile
- ◆ Build prototypes
- ◆ Agree on customer rights and responsibilities
- ◆ Have on-site customer



Cosmic Truth #5

*The customer is
not always right.
But the customer
always has a point.*



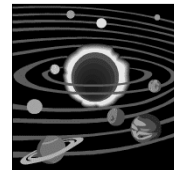
How Can the Customer Not Be Right??

- ◆ Presenting solutions in the guise of requirements
- ◆ Not prioritizing requirements
- ◆ Not communicating or respecting business rules or constraints
- ◆ Expecting a new system to drive business change
- ◆ Not making decisions
- ◆ Not accepting the need for tradeoffs
- ◆ Demanding impossible commitments
- ◆ Not accepting the cost of change



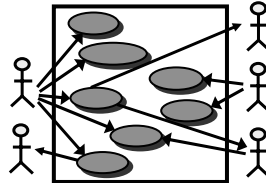
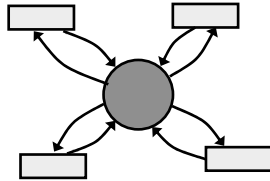
Cosmic Truth #6

The first question an analyst should ask about a proposed new requirement is, "Is this in scope?"



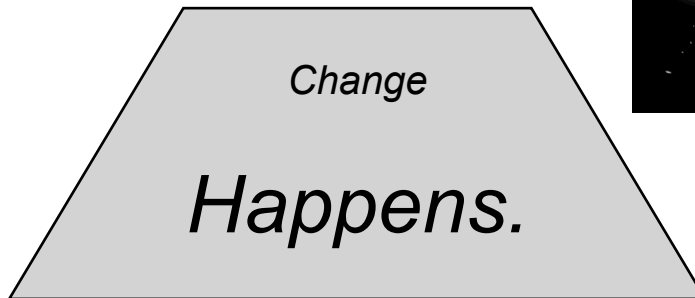
Defining Scope

- ◆ Write Vision and Scope Document or Project Charter
- ◆ Draw context diagram or use-case diagram



- ◆ Align requirements with business objectives
- ◆ Establish feedback loops between business and user requirements

Cosmic Truth #7



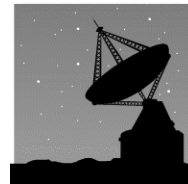
Helping Change Happen

- ◆ Establish a change control process.
- ◆ Establish a change control board.
- ◆ Incorporate contingency buffers into plans.
- ◆ Use an iterative or incremental life cycle.
- ◆ Recognize that change always has a price.



Cosmic Truth #8

*If it's not in
the requirements
specification, don't
expect to find it
in the product.*



How Much Detail Do You Need?

- ◆ Users are closely engaged throughout the project.
- ◆ Application domain is well-understood.
- ◆ Developers have extensive domain knowledge.

**Less
Detail**



**More
Detail**

- ◆ Product is complex.
- ◆ Product has severe failure modes.
- ◆ Detailed test cases will be based on requirements.
- ◆ Collaborating remote teams need a detailed “group memory.”

Two Widely Used Requirements Methods



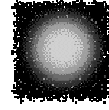
Clairvoyance



Telepathy

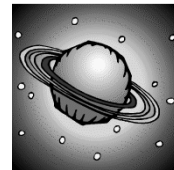
Cosmic Truth #9

*Even the best
requirements document
cannot replace
human dialog.*



Cosmic Truth #10

*The requirements
might be vague,
but the product
will be specific.*



Bonus Cosmic Truth

*You're never
going to
have perfect
requirements.*



A Practical Goal

*The objective of requirements
development is to get requirements
that are good enough to allow you
to proceed with construction at an
acceptable level of risk.*

For Further Reading

Beyer, Hugh, and Karen Holtzblatt. *Contextual Design: Defining Customer-Centered Systems*. San Francisco: Morgan Kaufmann Publishers, 1998.

Gottesdiener, Ellen. *Requirements by Collaboration: Workshops for Defining Needs*. Boston: Addison-Wesley, 2002.

Kulak, Daryl, and Eamonn Guiney. *Use Cases: Requirements in Context, 2nd Edition*. Boston: Addison-Wesley, 2003.

Lauesen, Soren. *Software Requirements: Styles and Techniques*. London: Addison-Wesley, 2002.

McGraw, Karen L., and Karan Harbison. *User-Centered Requirements: The Scenario-Based Engineering Process*. Mahwah, N.J.: Lawrence Erlbaum Associates, 1997.

Wiegiers, Karl E. *Software Requirements, 2d Ed.* Redmond, Wash.: Microsoft Press, 2003.

Wiegiers, Karl. "Karl Wiegiers Describes 10 Requirements Traps to Avoid," *Software Testing & Quality Engineering*, vol. 2, no. 1 (January/February 2000), pp. 34-40.