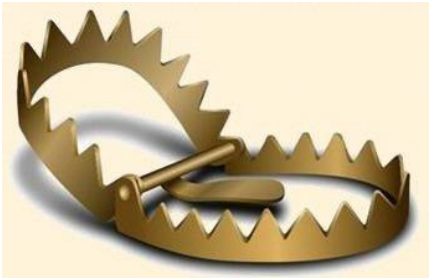
MODERN analyst Webinar Series

Software Requirements: 10 Traps to Avoid



Sponsored by:

IREB International Requirements Engineering Board

Dr. Karl Wiegers

Principal Consultant, Process Impact

www.processimpact.com

Sponsor: IREB

Enduring education concept

Renowned experts in RE share their interdisciplinary expertise to provide you with an enduring education concept!

CPRE - Certified Professional for Requirements Engineering

The three tier education concept enables you to grow in Requirements…

- Elicitation and Consolidation
- Modeling
- Management
- in Agile Environment

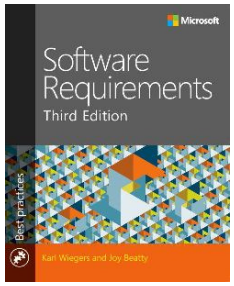
Document your competence in RE and get ready for better results in your RE projects.

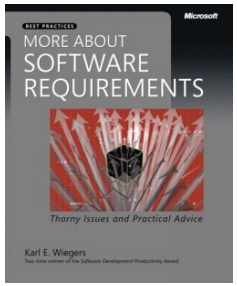
Featured Speaker

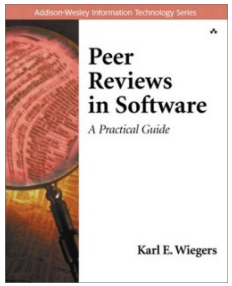
Karl Wiegers

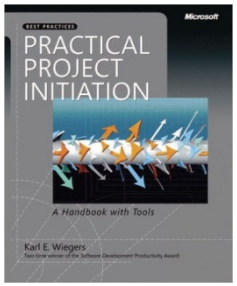
Principal Consultant, Process Impact
www.processimpact.com

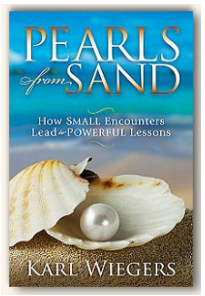
PROCESS IMPACT











Phone #: 503-698-7879

E-mail: karl@processimpact.com

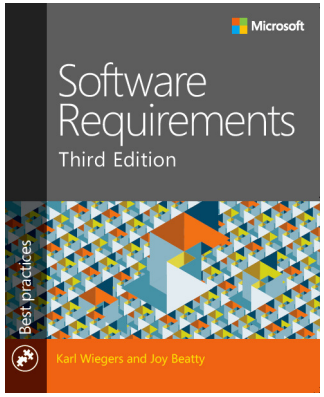
Blog: Consulting Tips & Tricks Blog: www.karlconsulting.blogspot.com

Downloadable Process Goodies: www.processimpact.com/goodies.shtml

IREB International Requirements Engineering Board

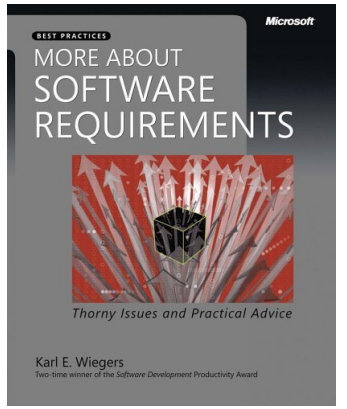
Sponsored By

Source Books



- *Software Requirements, 3rd Edition,* by Karl Wiegers and Joy Beatty (Microsoft Press, 2013)

tinyurl.com/reqs3



- *More About Software Requirements*, by Karl E. Wiegers (Microsoft Press, 2006)

IREB International Requirements Engineering Board

Sponsored By

Trap #1: Confusion Over “Requirements”

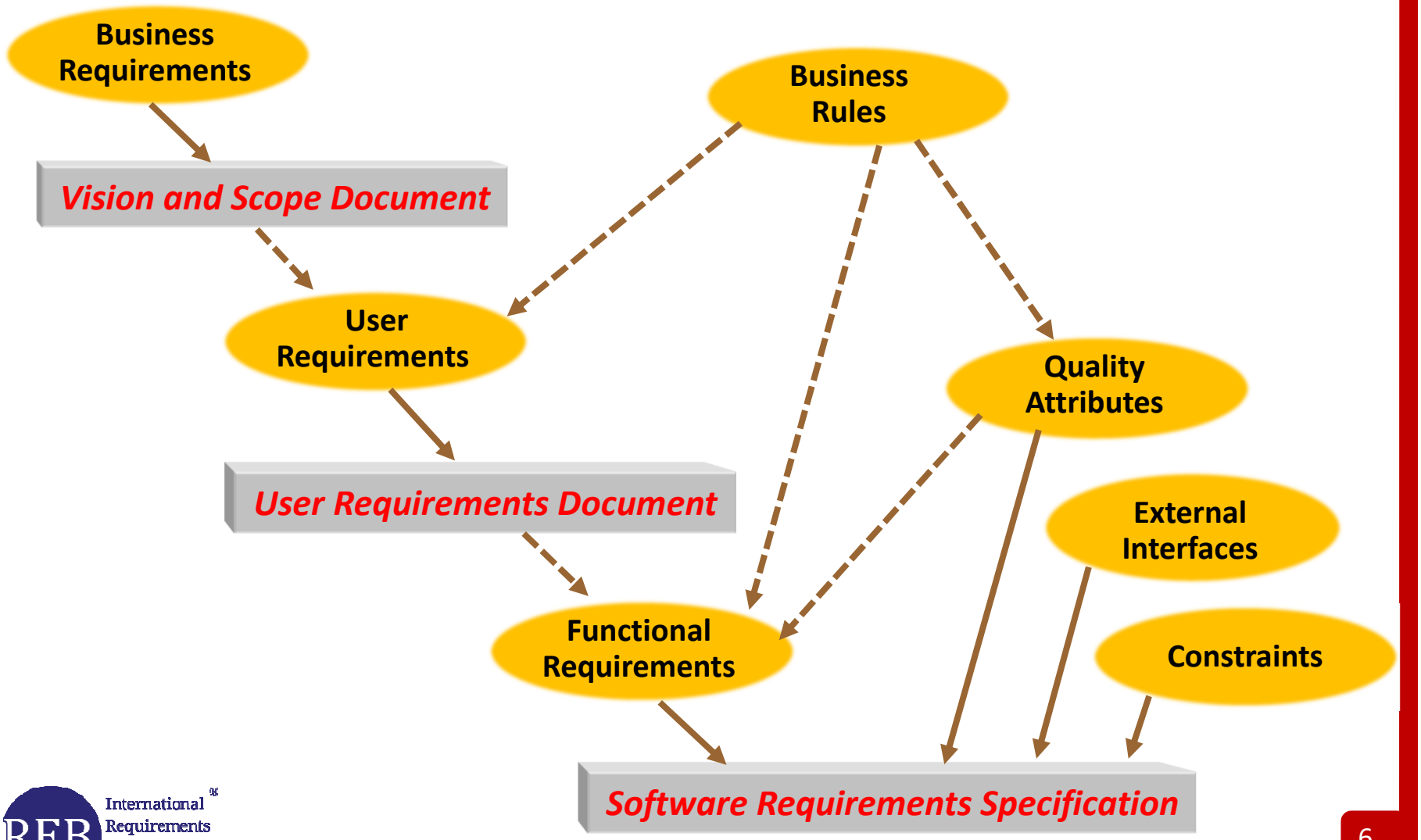
Symptoms

- **Stakeholders discuss “requirements” with no adjectives in front.**
- **Project sponsor presents a high-level concept as “the requirements”.**
- **User interface screens are viewed as “the requirements”.**
- **Users provide “the requirements,” but developers still don’t know what to build.**
- **Requirements focus just on functionality.**

IREB International Requirements Engineering Board

Sponsored By

Three Levels of Software Requirements



IREB International Requirements Engineering Board

Sponsored By

Trap #1: Confusion Over "Requirements"

Solutions

- **Adopt templates for three sets of requirements.**
  - ***business*** requirements (Vision & Scope Document)
  - ***user*** requirements (e.g., Use Case Document)
  - ***software*** requirements (Software Requirements Specification)
- **Distinguish functional from nonfunctional requirements.**
  - quality attributes, constraints, external interface requirements, business rules
- **Classify customer input into different categories.**
- **Distinguish solution ideas from requirements.**

Trap #2: Inadequate Customer Involvement

Symptoms

- Some user classes are overlooked.
- Some user classes don't have a voice.
- User surrogates attempt to speak for users.
- Developers have to make many requirements decisions.
- Customers reject the product when they first see it.

Trap #2: Inadequate Customer Involvement

Solutions

- Identify your various user classes.
- Identify *product champions* as user representatives.
- Convene focus groups.
- Identify requirements decision-makers.
- Have users evaluate prototypes.
- Have user reps review the requirements.
- Have users provide acceptance criteria.

Trap #3: Vague & Ambiguous Requirements

Symptoms

- Readers interpret a requirement in various ways.
- Requirements are missing information developers need.
- Requirements are not verifiable.
- Developers have to ask many questions.
- Developers have to guess a lot.

IREB International® Requirements Engineering Board

Sponsored By

Trap #3: Vague & Ambiguous Requirements

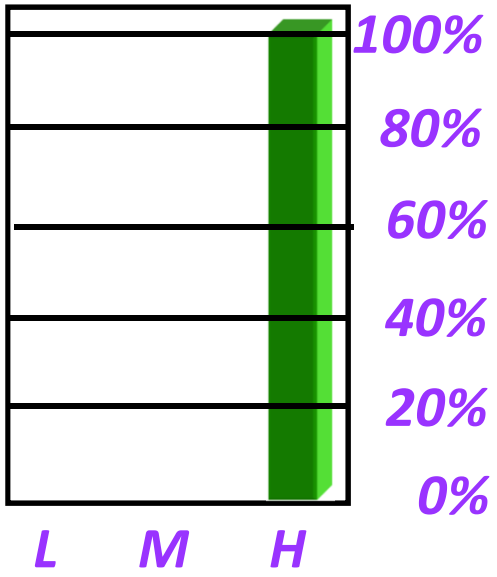
Solutions

- **Inspect requirements deliverables.**
- **Write conceptual tests against requirements.**
- **Model requirements to find knowledge gaps.**
- **Use prototypes to make requirements more tangible.**
- **Define terms in a glossary.**
- **Avoid ambiguous words:**

  minimize, maximize, optimize, rapid, user-friendly, simple, intuitive, robust, state-of-the-art, improved, efficient, ideally, flexible, several, and/or, etc., include, support, adequate

Trap #4: Unprioritized Requirements

Symptoms



- All requirements are critical!
- Different stakeholders interpret “high” priority differently.
- After prioritization, 95% are still high.
- Developers don’t want to admit they can’t do it all.
- It’s not clear which requirements to defer during the “rapid descoping phase.”

Trap #4: Unprioritized Requirements

Solutions

- Align functional requirements with high-priority user requirements.
- Align functional requirements with business requirements.
- Define priority categories unambiguously.
- Analytically prioritize discretionary requirements.
- Allocate requirements to releases or iterations.
- Dynamically re-prioritize work in the backlog.

Trap #5: Building Functionality No One Uses

Symptoms

- **Customers don't distinguish "chrome" from "steel".**
- **Users demand certain features, then no one uses them.**
- **Proposed functionality isn't related to business tasks.**
- **Developers add functions because "users will love this".**

Trap #5: Building Functionality No One Uses

Solutions

- **Derive functional requirements from user requirements.**
- **Trace every functional requirement back to its origin.**
- **Identify stakeholders who will benefit from each feature.**
- **Analytically prioritize requirements or features.**
  - have customers rate value (benefit and penalty)
  - have developers estimate cost and risk
  - avoid requirements with high cost + low value

Trap #6: Analysis Paralysis

Symptoms

- Requirements development goes on forever.
- New versions of the SRS are continually being released.
- Requirements are never baselined.
- All requirements are modeled six ways from Sunday.
- Design and coding can't start until the requirements are "done".

Trap #6: Analysis Paralysis

Solutions

- **Remember: the product is software, not requirements!**
- **Select an appropriate development life cycle.**
- **Decide when requirements are *good enough*.**
  - acceptable risk of proceeding with construction
  - reviewed by BAs, customers, developers, testers
- **Model just the complex or uncertain parts of the system.**
- **Don't include final user interface designs in SRS.**

Trap #7: Scope Creep

Symptoms

- **New requirements are continually added.**
  - schedule doesn't change
  - no more resources provided
- **Product scope is never clearly defined.**
- **Proposed requirements come, go, and return.**
- **Requirement changes sneak in through the back door.**
- **Sign-off is just a ritual.**

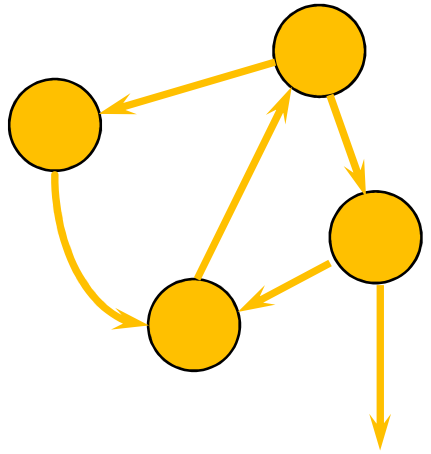
Trap #7: Scope Creep

Solutions

- Determine root causes of the scope creep.
- Document the product vision and project scope.
- Define system boundaries and interfaces.
- Follow the change control process for *all* changes.
- Improve requirements elicitation methods.
- Follow a meaningful baselining process.
- Renegotiate commitments when requirements change.

Trap #8: Inadequate Change Process

Symptoms



- **No change process is defined.**
- **Some people bypass the change process.**
  - users talk to buddies on the team
  - developers implement rejected changes
  - work is done on proposed changes before they're approved
- **New functionality becomes evident during testing.**
- **Unclear change request status.**
- **Changes aren't communicated to all those affected.**
- **It's not clear who makes change decisions.**

Trap #8: Inadequate Change Process

Solutions

- **Define a practical change control process.**
- **Set up a Change Control Board.**
- **Use a tool to collect, track, and communicate changes.**
  - Remember: *A tool is not a process!*
- **Establish and enforce change control policies.**
- **Compare priorities against remaining requirements.**

Trap #9: Insufficient Change Impact Analysis

Symptoms

- People agree to make changes hastily.
- Change is more complex than anticipated.
- Change takes longer than promised.
- Change isn't technically feasible.
- Change causes project to slip.
- Developers keep finding more affected components.

Trap #9: Insufficient Change Impact Analysis

Solutions

- **Analyze the impact of each proposed change.**
  - consider implications of accepting the change
  - identify possible tasks
  - estimate effort and schedule impact
- **Use requirements traceability information.**
- **Estimate costs and benefits before making commitments.**

Trap #10: Inadequate Version Control

Symptoms

- **Accepted changes aren't incorporated into the requirements set.**
- **You can't distinguish different SRS versions.**
  - different versions have the same ID
  - identical documents have different IDs
- **People work from different SRS versions.**
  - implement canceled features
  - test against the wrong requirements
- **Change history and earlier document versions are lost.**

Trap #10: Inadequate Version Control

Solutions

- **Merge changes into the requirements set.**
- **Adopt a versioning scheme for documents.**
- **Place requirements documents under version control.**
- **Communicate revisions to all who are affected.**
- **Use a requirements management tool.**
  - record complete history of every requirement change
  - requirements spec is a report from the database

Keys to Excellent Software Requirements

- Educated developers, managers, and customers
- Skillful, trained business analysts
- A collaborative customer-developer partnership
- Understanding different kinds of requirements
- Iterative, incremental requirements development
- Standard requirements document templates
- Formal and informal requirements reviews
- Writing tests against requirements
- Thoughtful, dynamic requirements prioritization
- Practical, effective change management

IREB International® Requirements Engineering Board

Sponsored By

Software Requirements: 10 Traps to Avoid

Sponsor: IREB

IREB® International Requirements Engineering Board

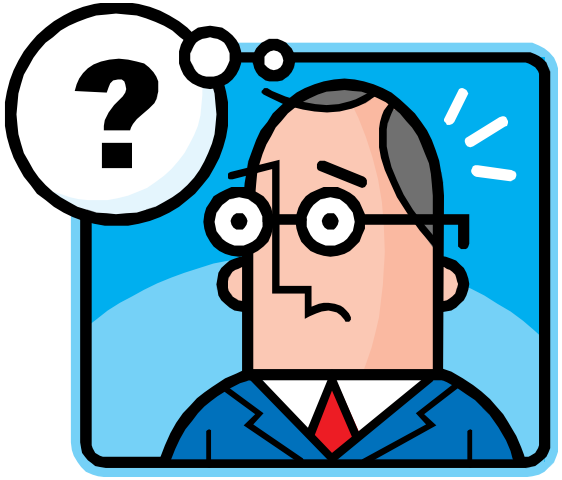
The CPRE is worldwide acknowledged – about 27,000 certified people

Gain the relevant skills, improve your professional profile and you will gain a higher market value.

www.ireb.org

- **Avoid the pitfalls Karl has talked about!**
- Work more efficiently with stakeholders during elicitation.
- Learn the appropriate methods for successful negotiations.
- Be familiar with different techniques for modeling and documenting requirements.
- Be more effective at managing requirements during the entire lifecycle of a project or product.
- Know the terminology of RE
- Become part of an internationally recognized community.

Q & A



Speaker

Karl Wiegers
Process Impact
karl@processimpact.com
www.processimpact.com

Sponsor

Stefan Sturm
IREB
info@ireb.org
www.ireb.org

International Requirements Engineering Board

Sponsored By