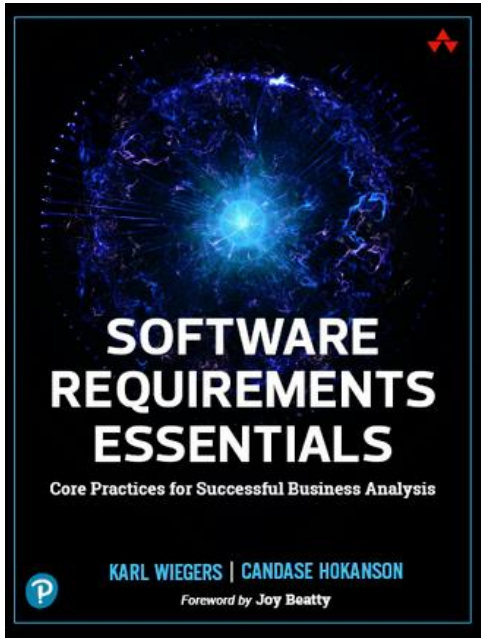


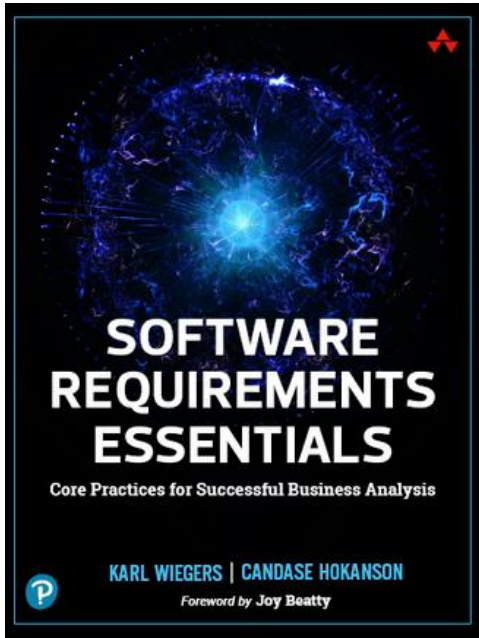
# ESSENTIAL REQUIREMENTS PRACTICES



Candase Hokanson  
ArgonDigital  
Softwarereqs.com



# Agenda



- Laying the foundation
- Requirements elicitation
- Requirements analysis
- Requirements specification
- Requirements validation
- Requirements management

# Laying the Foundation for Success

Why are we working on this?

What are we trying to build?

Who are we trying to satisfy?

What do we implement first? Next? Maybe never?

How can we tell if our solution is good enough?

How do we know when we're done?

# Laying the Foundation: Core Practices

- #1. Understand the problem before converging on a solution.
- #2. Define business objectives.
- #3. Define the solution's boundaries.
- #4. Identify and characterize stakeholders.
- #5. Identify empowered decision makers.

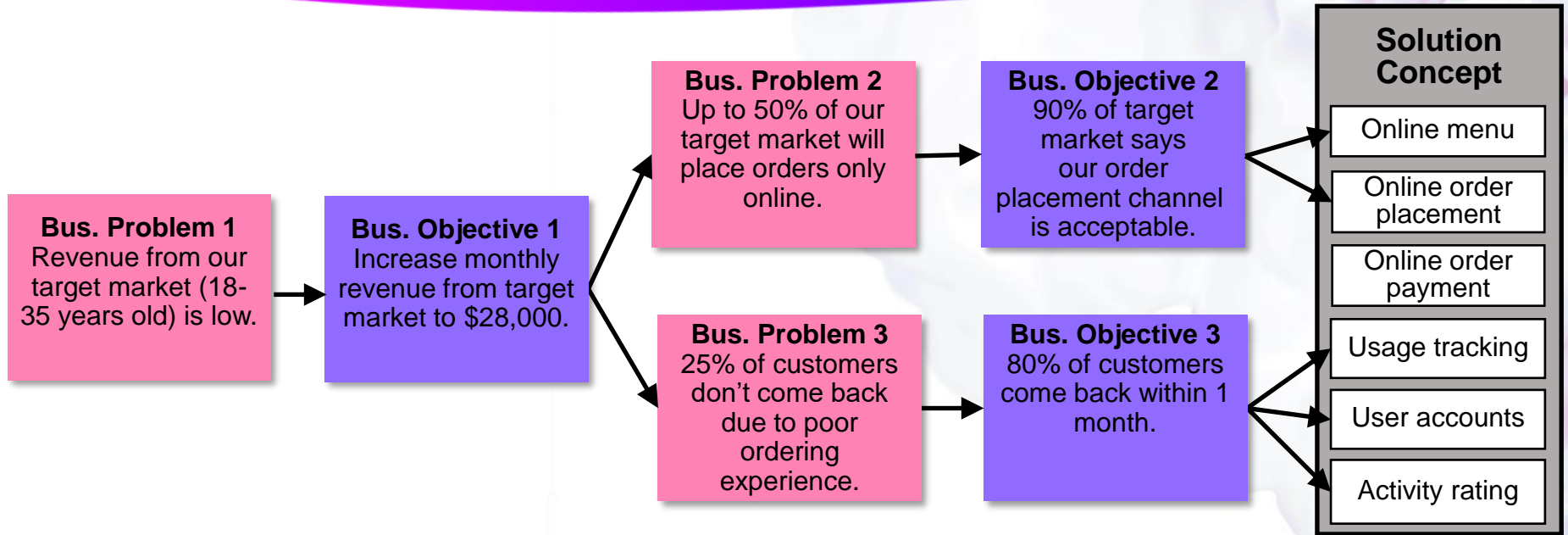


# #2. Define Business Objectives

- Begin with a problem statement
  - ➔ Business objectives
  - ➔ Success metrics
  - ➔ Solution concept
  - ➔ Scope definition
  - ➔ Solution requirements
- Setting objectives helps to
  - Identify stakeholders
  - Define all the necessary functionality
  - Prioritize requirements
  - Plan releases



# Business Objectives Model



# Requirements Elicitation

- What is requirements elicitation?
  - Involves collection, exploration, discovery, and invention
  - Many sources of requirements
  - Many elicitation techniques: interviews, workshops, observation, surveys, process modeling, prototyping, document analysis, ...
- Core elicitation practices
  - #6. Understand what users need to do with the solution.
  - #7. Identify events and responses.
  - #8. Assess data concepts and relationships.
  - #9. Elicit and evaluate quality attributes.



# #6. Understand What Users Need to Do

- Usage-centric vs product-centric elicitation approach
  - **What functionality stakeholders think the solution should have**, versus
  - **What users need to do with the solution**
- Eliciting user requirements
  - Must align with achieving business objectives
  - Explore normal, alternative, and exception scenarios
  - Reveals needed functionality
  - Avoids building unnecessary functionality





# Use Cases and User Stories

- Use cases
  - Use case name defines the user's goal: *"View article statistics"*
  - Use case specification has preconditions, postconditions, flow steps, etc.
  - Derive functional requirements and tests from the use case spec
- User stories
  - Story gives more information: *"As an author, I want to view the statistics for my articles so that I can see which topics my readers enjoy the most."*
  - Provide story details through acceptance criteria
- With both:
  - Focus on user goals, not bits of functionality

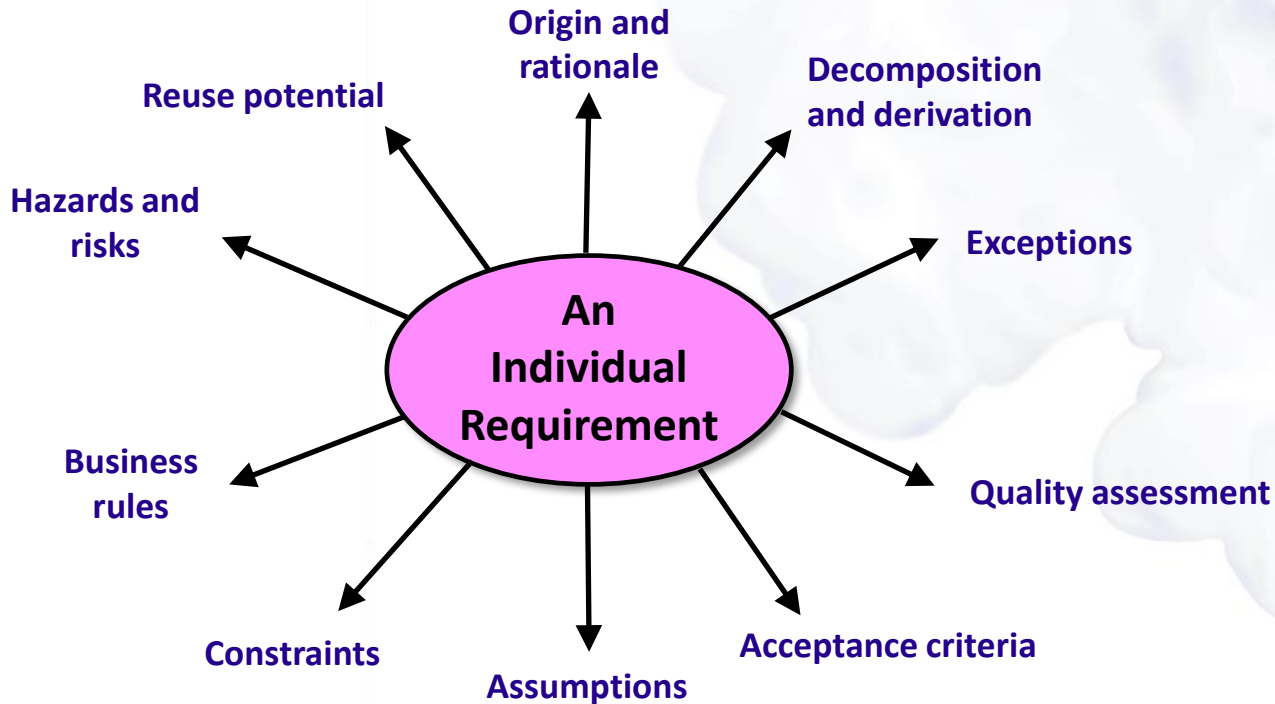


# Requirements Analysis

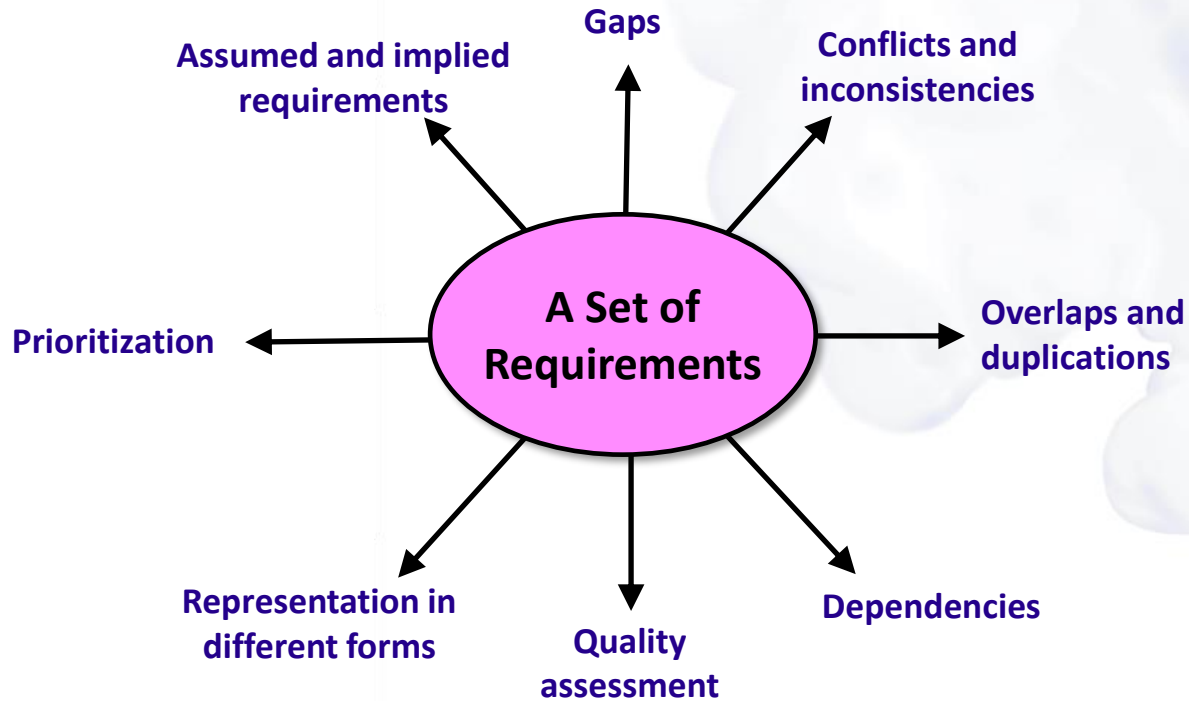
- What is requirements analysis?
  - Ensuring that all stakeholder needs are understood and recorded
  - Ensuring that a satisfactory solution can be defined, built, and tested
  - Involves questioning, learning, decomposing, comparing, filling gaps, confirming, refining, reassessing
- Core analysis practices
  - #10. Analyze requirements and requirement sets.
  - #11. Create requirements models.
  - #12. Create and evaluate prototypes.
  - #13. Prioritize the requirements.



# #10. Analyze Individual Requirements...



# ...and Requirement Sets



# Requirements Specification

- What is requirements specification?
  - “Writing requirements” really means “representing requirements knowledge”
  - Specifications vary in content, structure, form, detail, and formality
  - The goal is always *clear and effective communication*
- Core specification practices
  - #14. Write requirements in consistent ways.
  - #15. Organize requirements in a structured fashion.
  - #16. Identify and document business rules.
  - #17. Create a glossary.



# #16. Identify and Document Business Rules

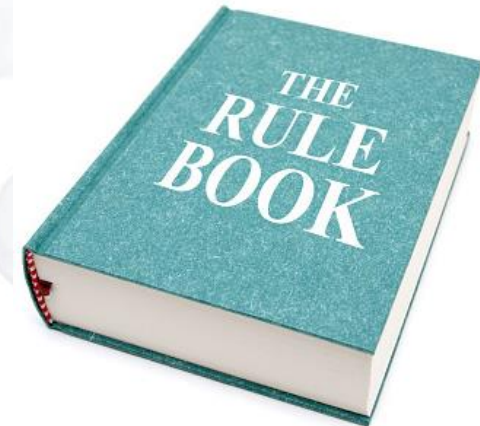
- Define or restrict an organization's operations
- Influence behaviors of people and systems
- Lead to derived functional and data requirements

- *Facts*
- *Constraints*
- *Action enablers*
- *Computations*

## Types

- *Policies*
- *Laws*
- *Regulations*
- *Industry standards*

## Sources



# Decision Tables

Rule ID	DISC-1	DISC-2	DISC-3	DISC-4	DISC-5	DISC-6
<b>Conditions</b>						
Order total	<\$50	\$50–\$100	>\$100	<\$50	\$50–\$100	>\$100
Club member	N	N	N	Y	Y	Y
<b>Action</b>						
No discount	X			X		
10% discount		X	X		X	
20% discount						X
Free shipping			X	X	X	X

# Requirements Validation

- What is requirements validation?
  - Confirm that requirements accurately describe stakeholder needs
  - Confirm that a solution would satisfy needs and achieve business objectives
  - Verification = doing the thing right  
Validation = doing the right thing
  - Interwoven with elicitation, analysis, and specification
  - Can use prototypes and early releases
- Core validation practices
  - #18. Review and test the requirements.





# #18. Review the Requirements...

1. Select the right participants
  - Author and other business analyst
  - Representatives of requirements sources
  - Consumers of requirements
2. Choose a level of formality and rigor
  - Ad hoc review, passaround, team review, inspection
  - More formal is slower but more effective
3. Use a checklist to look for common types of errors
  - Ambiguities, inconsistencies, omissions, duplications, unneeded requirements, missing information...

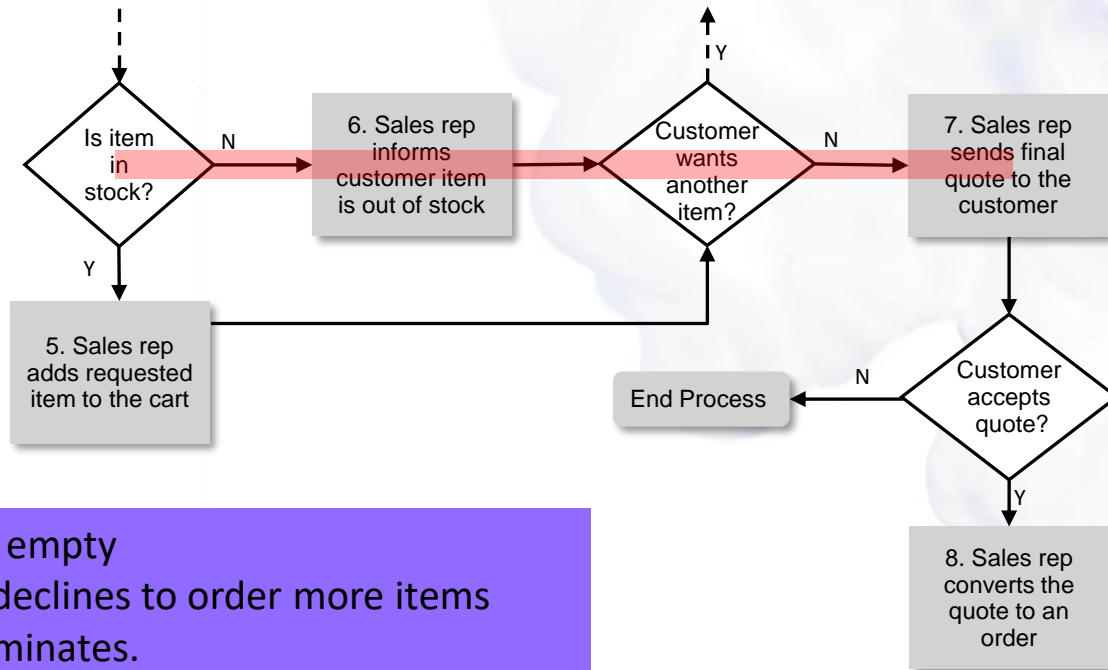


# ...and Test the Requirements

- Start testing after writing your first requirement!
- **Requirements** ← *complementary thought processes* → **Tests**
- Acceptance criteria on agile projects: *Given–When–Then*

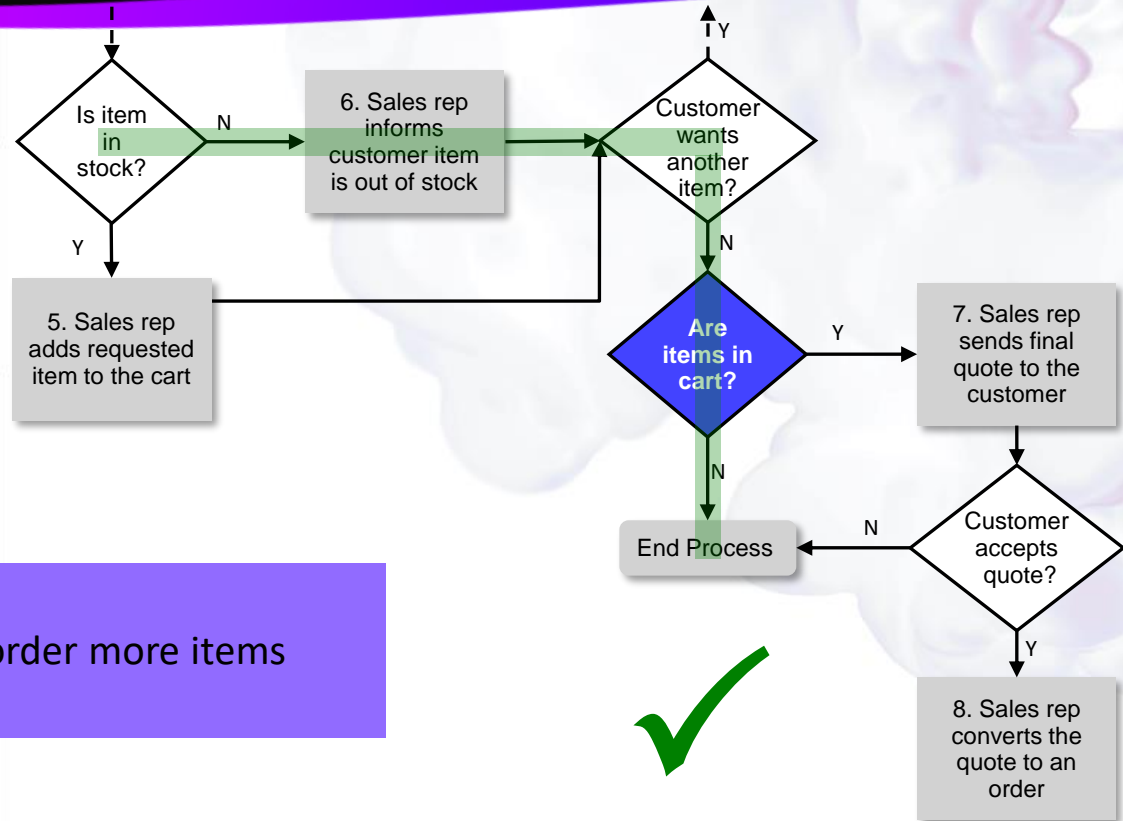
<i>ID</i>	<i>Given</i>	<i>When</i>	<i>Then</i>
AT-1	I am logged into the platform AND I have articles published	I request to view statistics	A graph of my total article view statistics from the past 30 days is displayed AND a list of statistics (views, reads, and likes) for individual articles is displayed in reverse chronological order by publication date

# Testing Requirements Models - 1



**Given** that the cart is empty  
**When** the customer declines to order more items  
**Then** the process terminates.

# Testing Requirements Models - 2



**Given** that the cart is empty  
**When** the customer declines to order more items  
**Then** the process terminates.



# Requirements Management

- What is requirements management?
  - Dealing with requirements after they've been specified
  - Requirements version control
  - Tracking requirements status
  - Requirements tracing
- Core requirements management practices
  - #19. Establish and manage requirements baselines.
  - #20. Manage changes to requirements effectively.

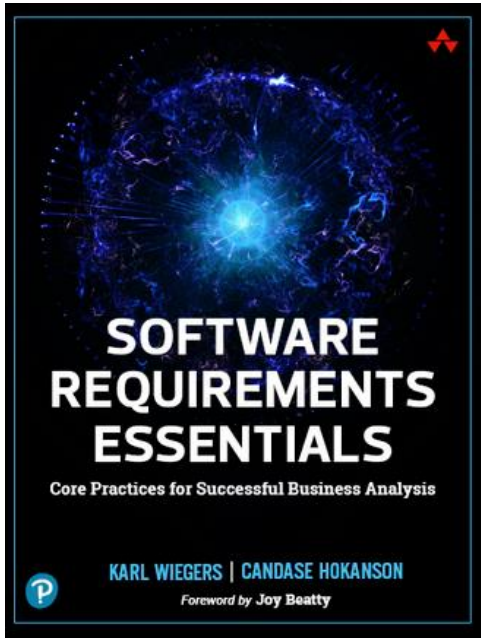


# #19. Establish Requirements Baselines

- **Baseline:** An agreed-upon set of requirements for a specific development cycle
- A baseline can be:
  - **Time-bound:** whatever requirements fits in the schedule box
  - **Scope-bound:** work until the allocated requirements are done
- Changes are made against a specific baseline
  - Follow your change control process
  - Require scope, time, and resource negotiations
  - Contingency buffers provide some slack



# ESSENTIAL REQUIREMENTS PRACTICES

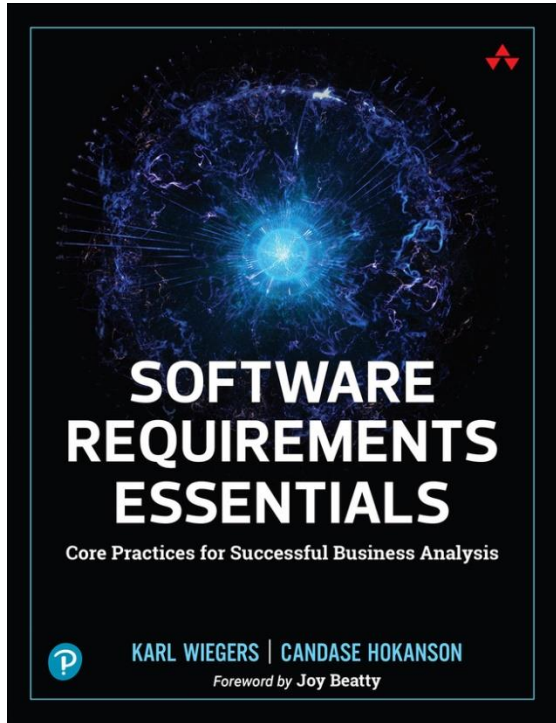


Candase Hokanson  
ArgonDigital  
[Softwarereqs.com](http://Softwarereqs.com)





# Order & Save 35%\* at [informit.com/softwarereqs](http://informit.com/softwarereqs)



- Use code **SOFTWAREREQS** during checkout
- Offer only good at [informit.com](http://informit.com)
- Print book – free U.S. shipping\*\*
- eBook – DRM-Free PDF & EPUB

### Also Available

Booksellers including Amazon and Barnes & Noble and in O'Reilly's Online Learning Service (Safari).

\*Discount code SOFTWAREREQS is only good at [informit.com](http://informit.com) and cannot be used on the already discounted book + eBook bundle or combined with any other offer.

\*\*If your order is shipping outside of the U.S., it may be subject to import duties and taxes, which are levied once the package reaches the destination country.