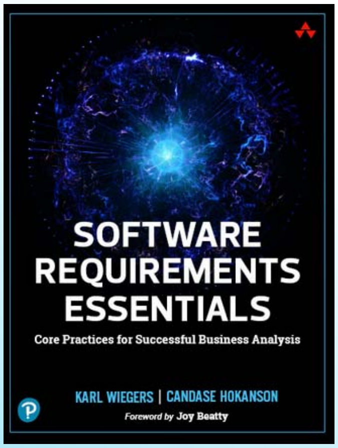

SOFTWARE REQUIREMENTS ESSENTIALS

Core Practices for Successful Business Analysis

KARL WIEGERS | CANDASE HOKANSON

Foreword by Joy Beatty

Quality Begins with Requirements

Karl Wiegers

PROCESS IMPACT

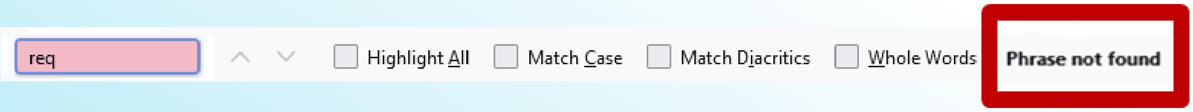
processimpact.com
softwarereqs.com

PNSQC 2023

PNSQC 2023 Themes and Tracks

Conference theme: Using technology and processes to amplify quality

- Processes and Tools
  - process & methods, tools & technologies, implementation & infrastructure
- Management, Leadership, and People
- Emerging Technologies and Concepts

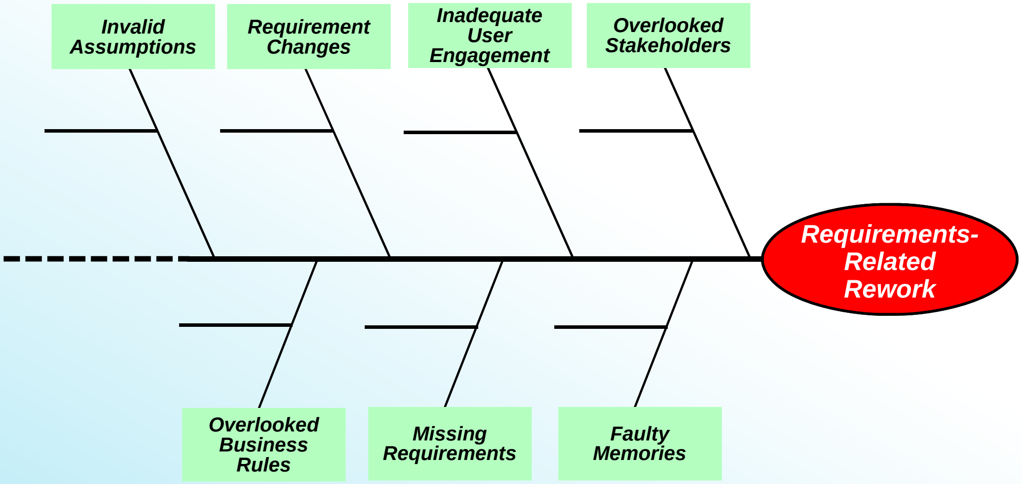


req

☐ Highlight All ☐ Match Case ☐ Match Diacritics ☐ Whole Words

Phrase not found

Consider Sources of Errors and Rework



Invalid Assumptions

Requirement Changes

Inadequate User Engagement

Overlooked Stakeholders

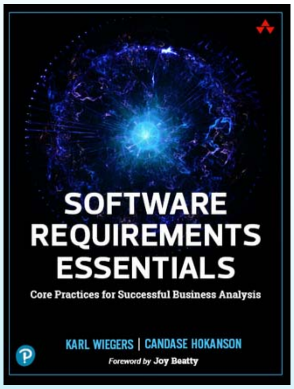
Requirements-Related Rework

Overlooked Business Rules

Missing Requirements

Faulty Memories

The 20 Core Requirements Practices


SOFTWARE REQUIREMENTS ESSENTIALS
Core Practices for Successful Business Analysis
KARL WIEGERS | CANDASE HOKANSON
Foreword by Joy Beatty

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

Laying the Foundation: Core Practices

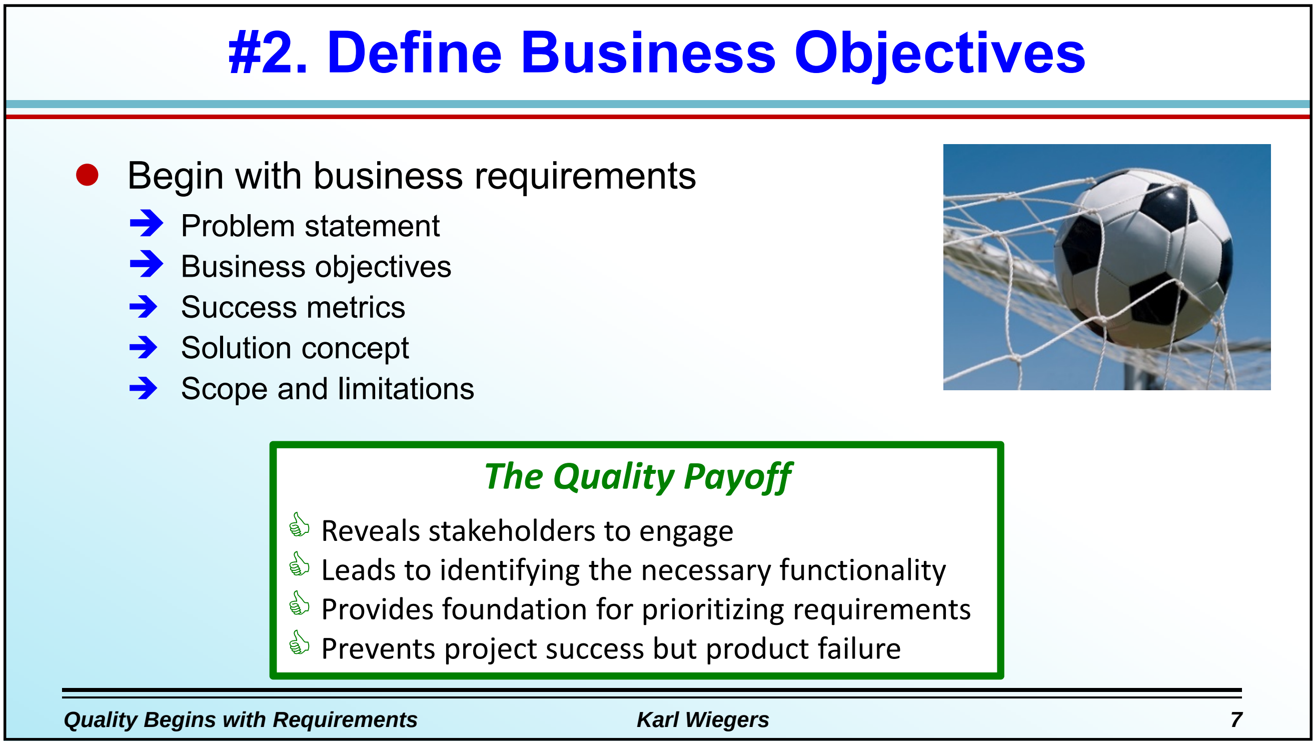
#1. Understand the problem before converging on a solution.

#2. Define business objectives.

#3. Define the solution's boundaries.

#4. Identify and characterize stakeholders.

#5. Identify empowered decision makers.

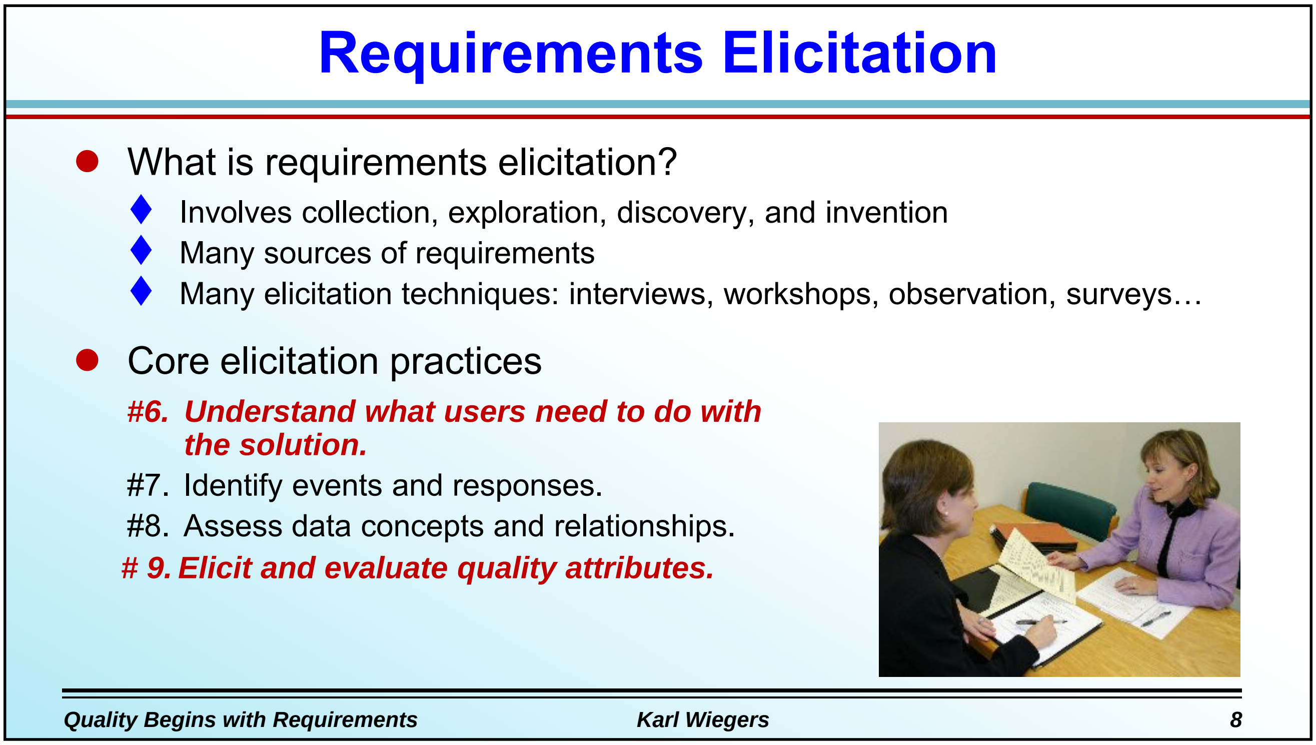


#2. Define Business Objectives

- Begin with business requirements
  - Problem statement
  - Business objectives
  - Success metrics
  - Solution concept
  - Scope and limitations

The Quality Payoff

- Reveals stakeholders to engage
- Leads to identifying the necessary functionality
- Provides foundation for prioritizing requirements
- Prevents project success but product failure



Requirements Elicitation

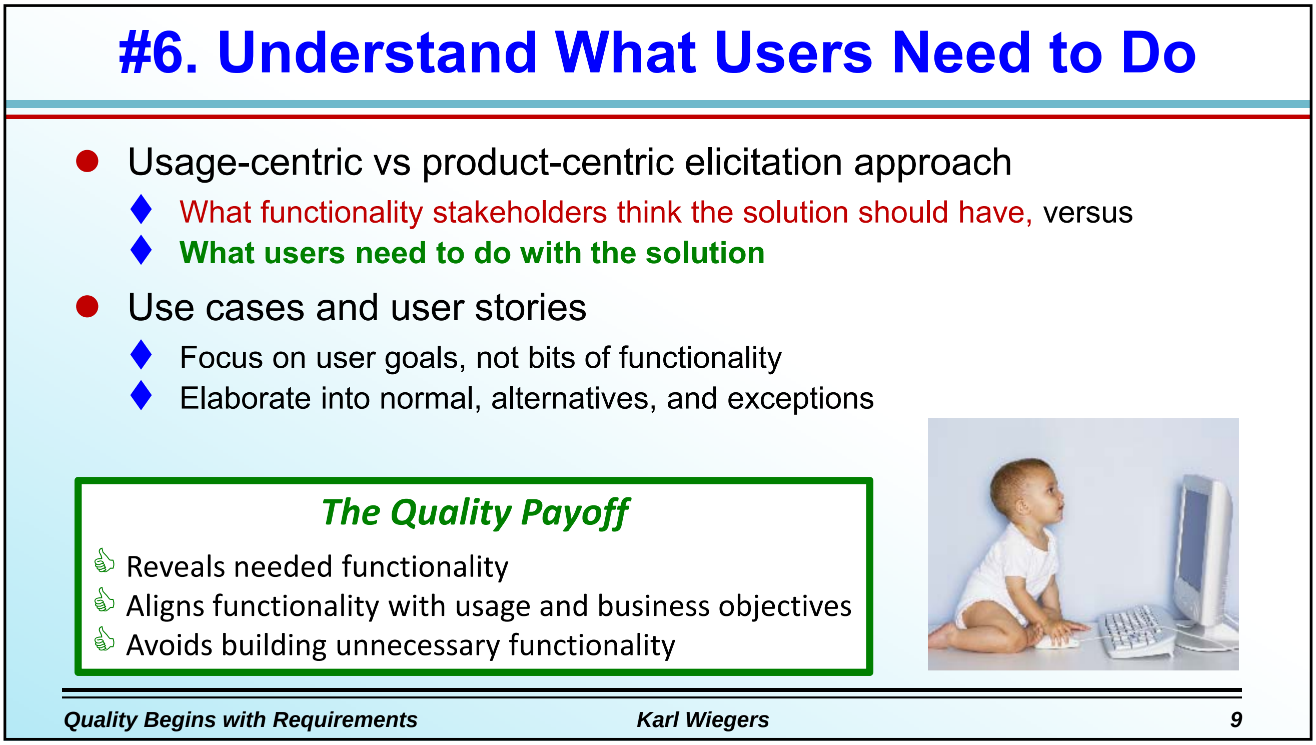
- What is requirements elicitation?
  - Involves collection, exploration, discovery, and invention
  - Many sources of requirements
  - Many elicitation techniques: interviews, workshops, observation, surveys…
- Core elicitation practices

#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.

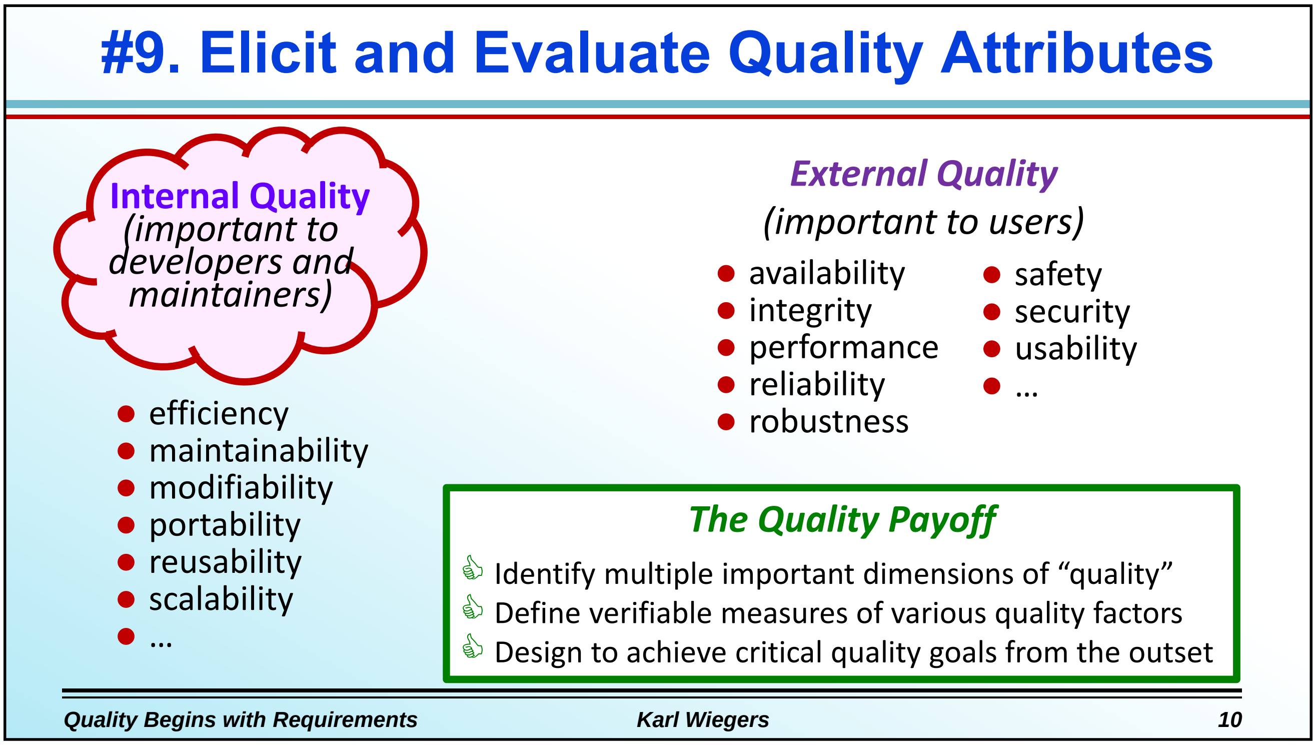


#6. Understand What Users Need to Do

- Usage-centric vs product-centric elicitation approach
  - What functionality stakeholders think the solution should have, versus
  - **What users need to do with the solution**
- Use cases and user stories
  - Focus on user goals, not bits of functionality
  - Elaborate into normal, alternatives, and exceptions

The Quality Payoff

- 👍 Reveals needed functionality
- 👍 Aligns functionality with usage and business objectives
- 👍 Avoids building unnecessary functionality



#9. Elicit and Evaluate Quality Attributes

Internal Quality
(important to developers and maintainers)

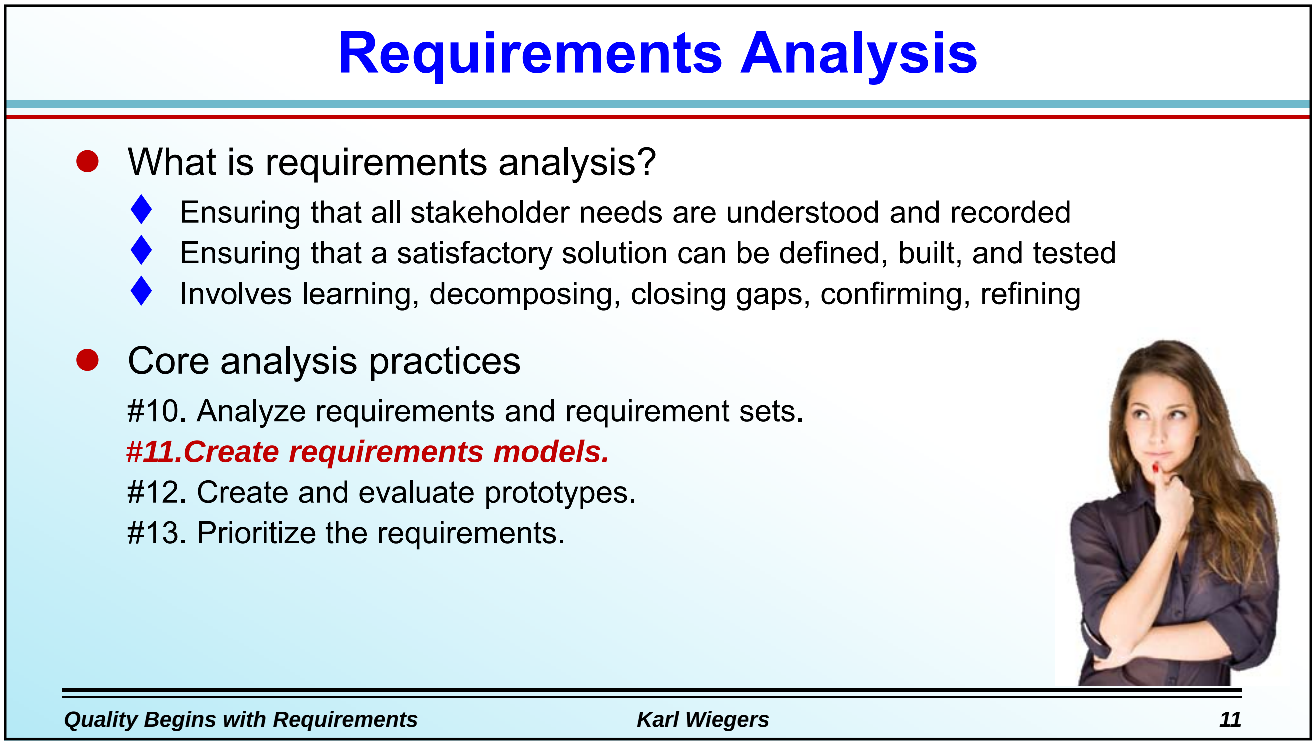
- efficiency
- maintainability
- modifiability
- portability
- reusability
- scalability
- …

External Quality
(important to users)

- availability
- integrity
- performance
- reliability
- robustness
- safety
- security
- usability
- …

The Quality Payoff

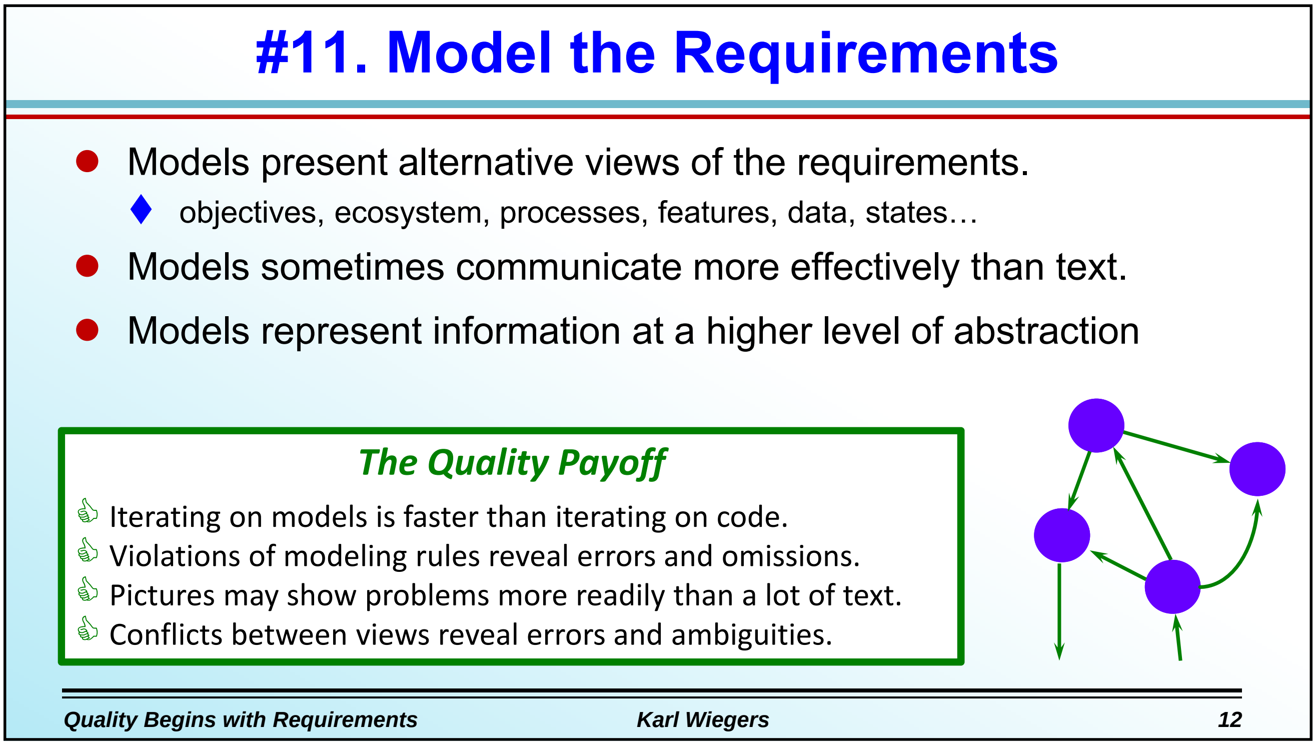
- 👍 Identify multiple important dimensions of "quality"
- 👍 Define verifiable measures of various quality factors
- 👍 Design to achieve critical quality goals from the outset



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 learning, decomposing, closing gaps, confirming, refining
- Core analysis practices

  #10. Analyze requirements and requirement sets.
  #11.Create requirements models.
  #12. Create and evaluate prototypes.
  #13. Prioritize the requirements.



#11. Model the Requirements

- Models present alternative views of the requirements.
  - objectives, ecosystem, processes, features, data, states…
- Models sometimes communicate more effectively than text.
- Models represent information at a higher level of abstraction

The Quality Payoff

- Iterating on models is faster than iterating on code.
- Violations of modeling rules reveal errors and omissions.
- Pictures may show problems more readily than a lot of text.
- Conflicts between views reveal errors and ambiguities.

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.

#14. Write Requirements in Consistent Ways

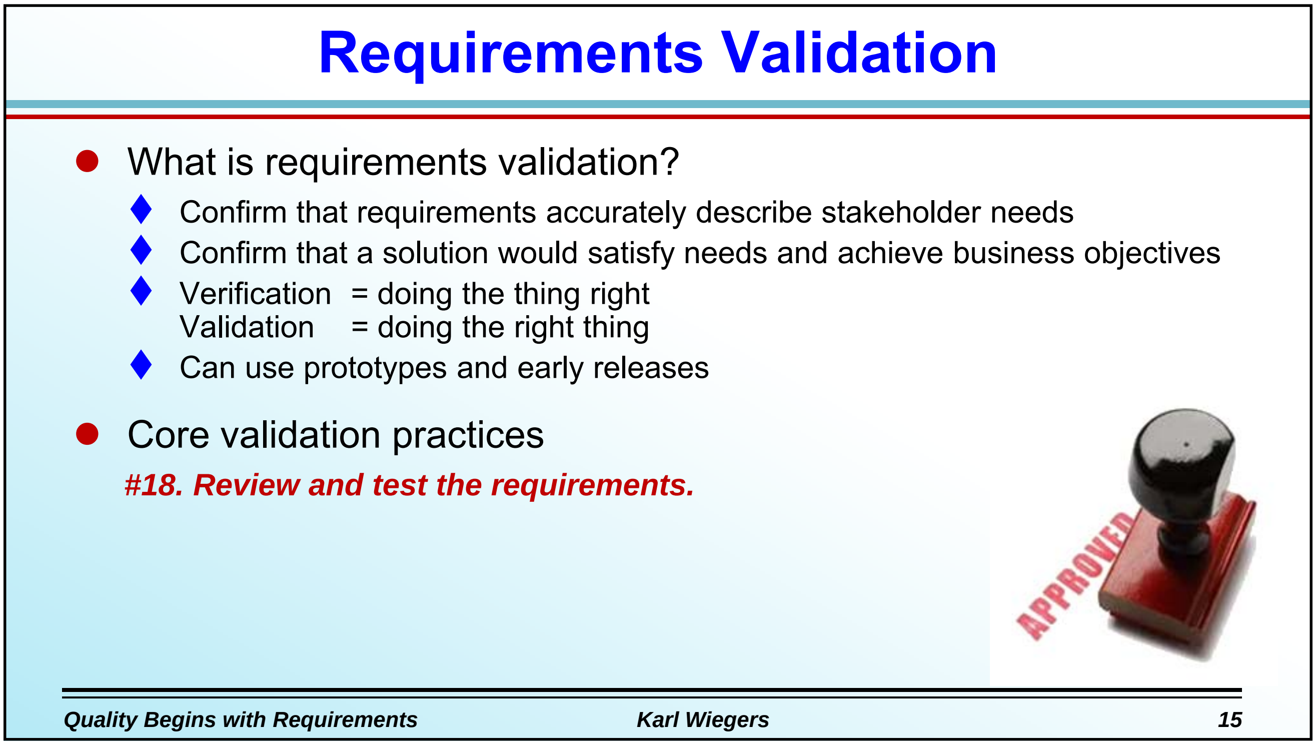
- Select techniques to communicate most effectively
  - text, tables, models, screen sketches, mathematical formulas…
- Follow patterns for writing requirements
  - user’s perspective vs system’s perspective for functionality
  - Planguage for nonfunctional requirements
- Favor clarity over purity of style or convention

The Quality Payoff

👍 Different communication forms speak to different people and needs.

👍 Consistent writing styles help readers find what they need.

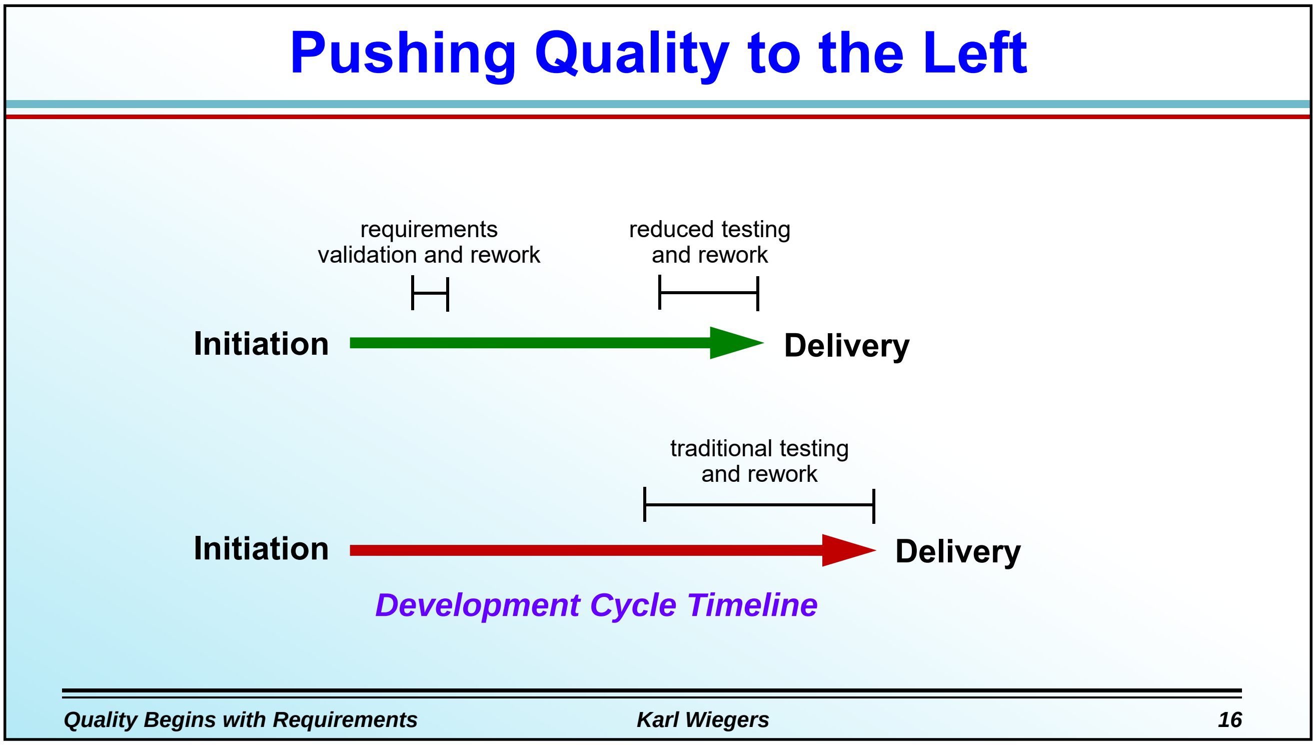
👍 Precise quality specifications lead to achievable, cost-effective designs.



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
  - Can use prototypes and early releases
- Core validation practices

  #18. Review and test the requirements.



Pushing Quality to the Left

requirements validation and rework

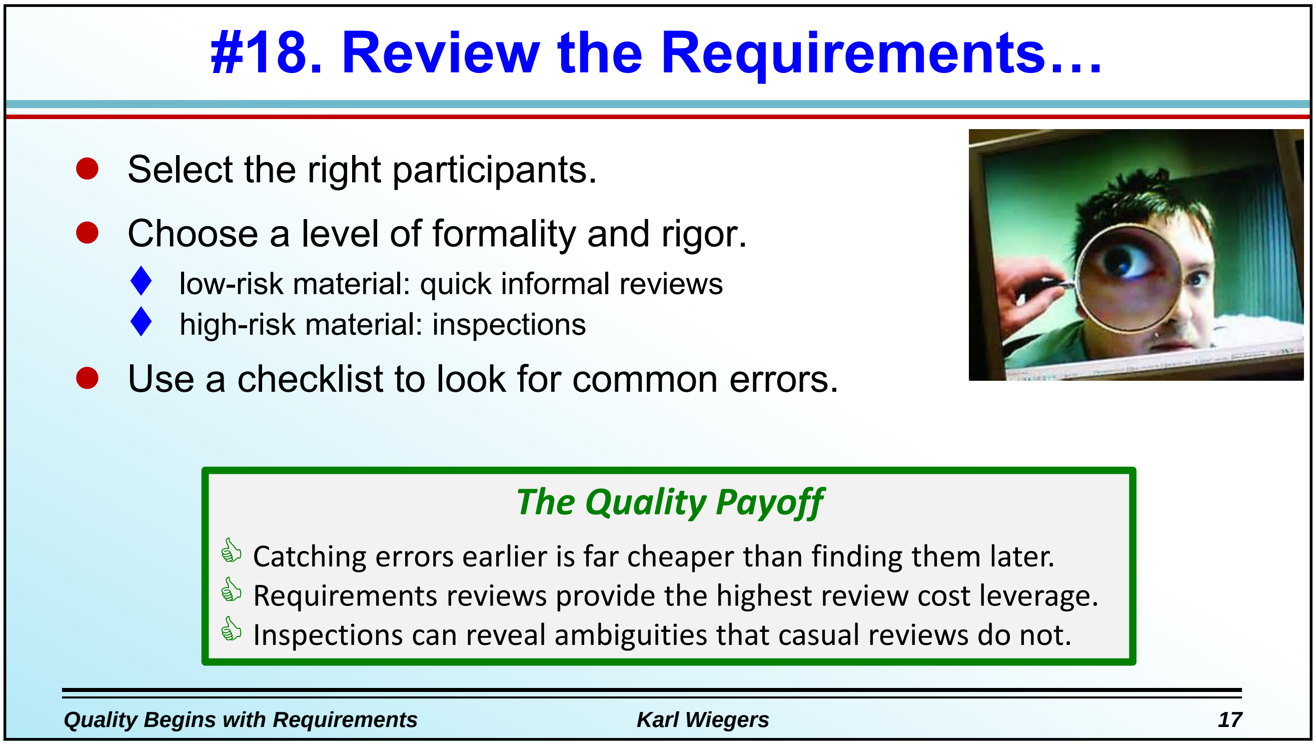
reduced testing and rework

Initiation **Delivery**

traditional testing and rework

Initiation **Delivery**

Development Cycle Timeline



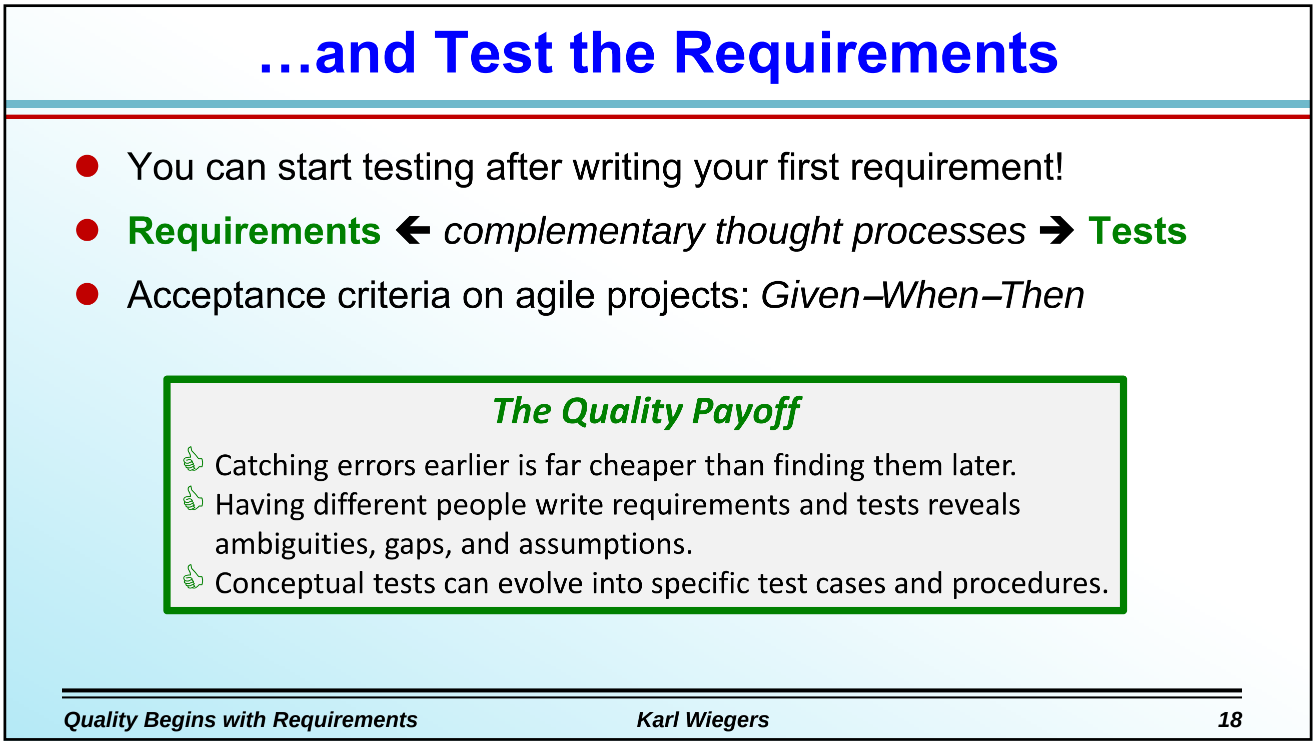
#18. Review the Requirements…

- Select the right participants.
- Choose a level of formality and rigor.
  - low-risk material: quick informal reviews
  - high-risk material: inspections
- Use a checklist to look for common errors.

The Quality Payoff

- Catching errors earlier is far cheaper than finding them later.
- Requirements reviews provide the highest review cost leverage.
- Inspections can reveal ambiguities that casual reviews do not.

Quality Begins with Requirements ***Karl Wiegers*** ***17***



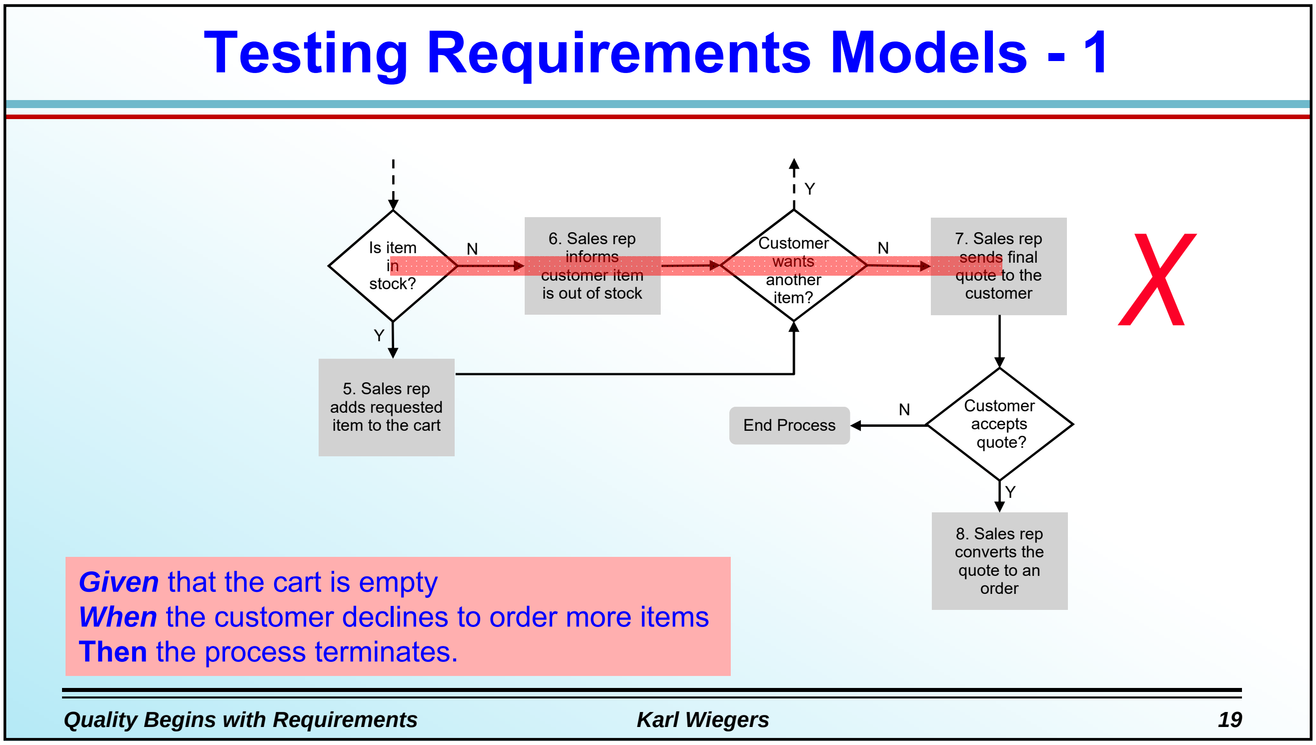
…and Test the Requirements

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

The Quality Payoff

- Catching errors earlier is far cheaper than finding them later.
- Having different people write requirements and tests reveals ambiguities, gaps, and assumptions.
- Conceptual tests can evolve into specific test cases and procedures.

Quality Begins with Requirements ***Karl Wiegers*** ***18***



Testing Requirements Models - 1

Is item in stock?

N

6. Sales rep informs customer item is out of stock

Customer wants another item?

Y

N

7. Sales rep sends final quote to the customer

X

Y

5. Sales rep adds requested item to the cart

End Process

N

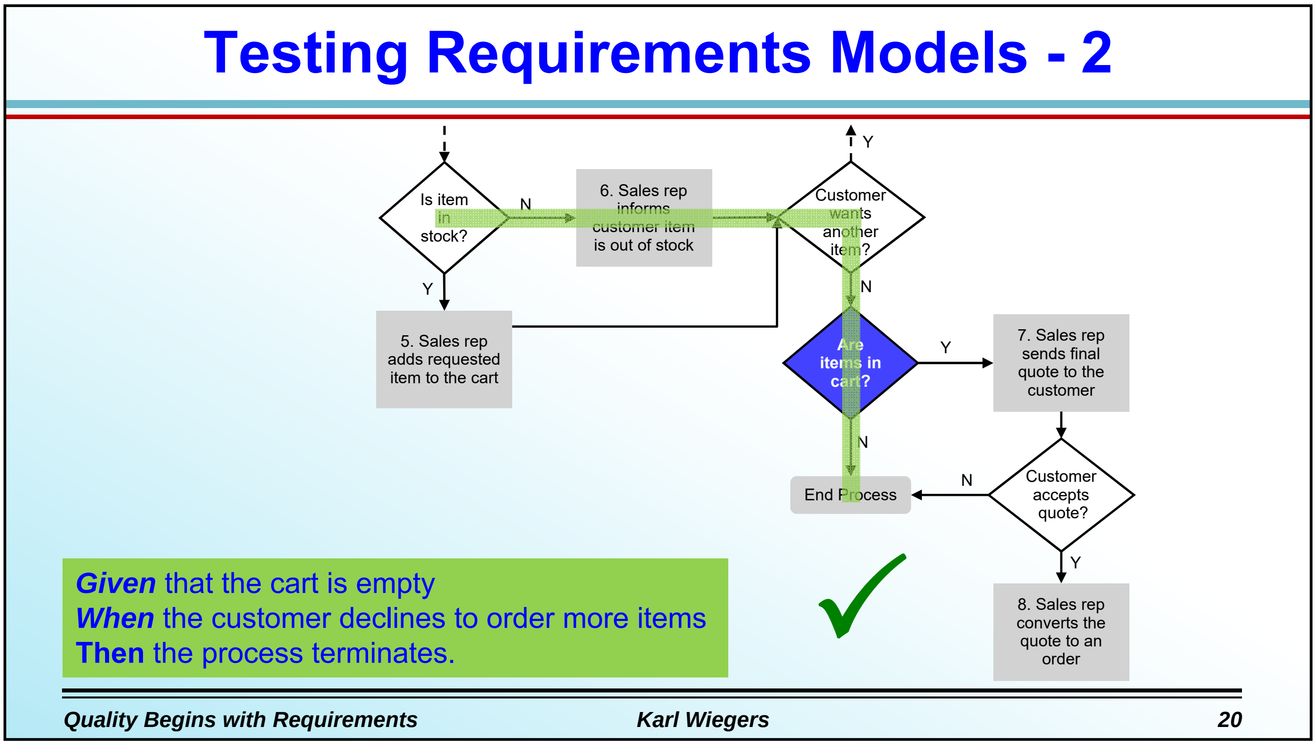
Customer accepts quote?

Y

8. Sales rep converts the quote to an order

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

Quality Begins with Requirements ***Karl Wiegers*** ***19***



Testing Requirements Models - 2

Is item in stock?

N

6. Sales rep informs customer item is out of stock

Y

Customer wants another item?

Y

5. Sales rep adds requested item to the cart

N

Are items in cart?

Y

7. Sales rep sends final quote to the customer

N

End Process

N

Customer accepts quote?

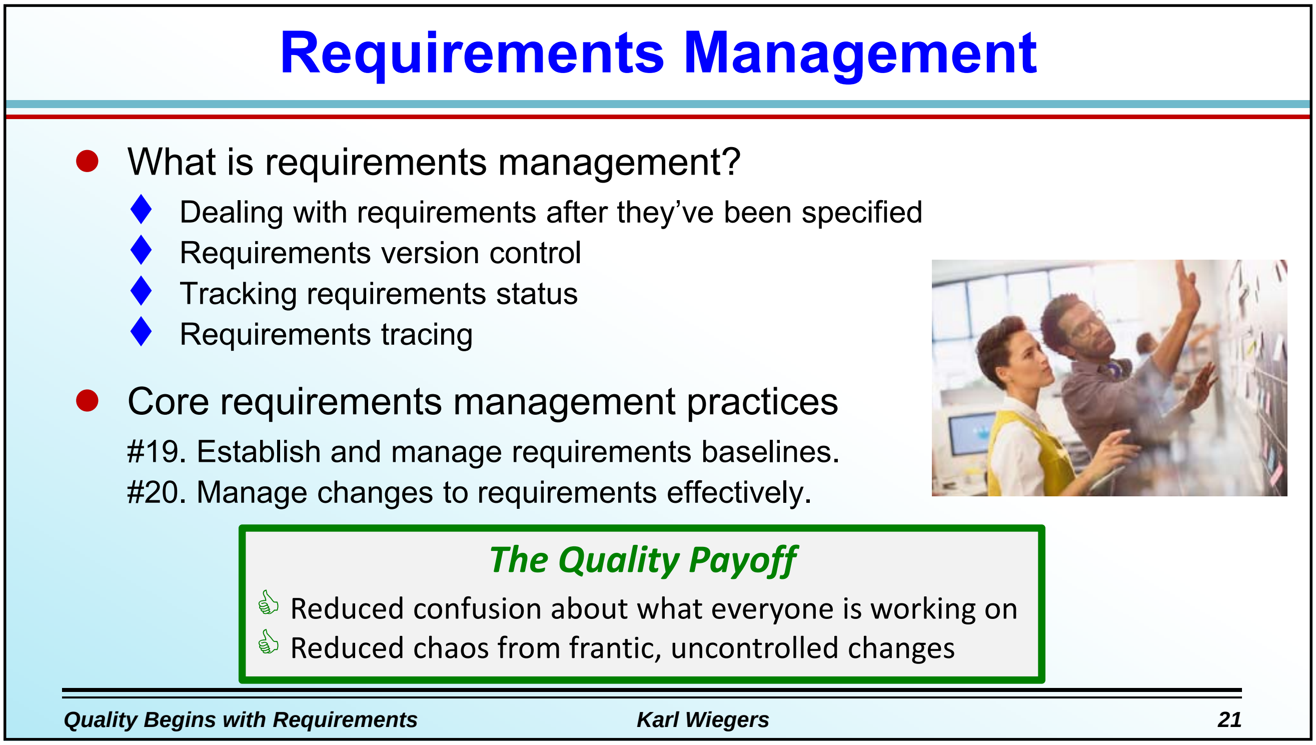
✓

Y

8. Sales rep converts the quote to an order

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

Quality Begins with Requirements ***Karl Wiegers*** ***20***



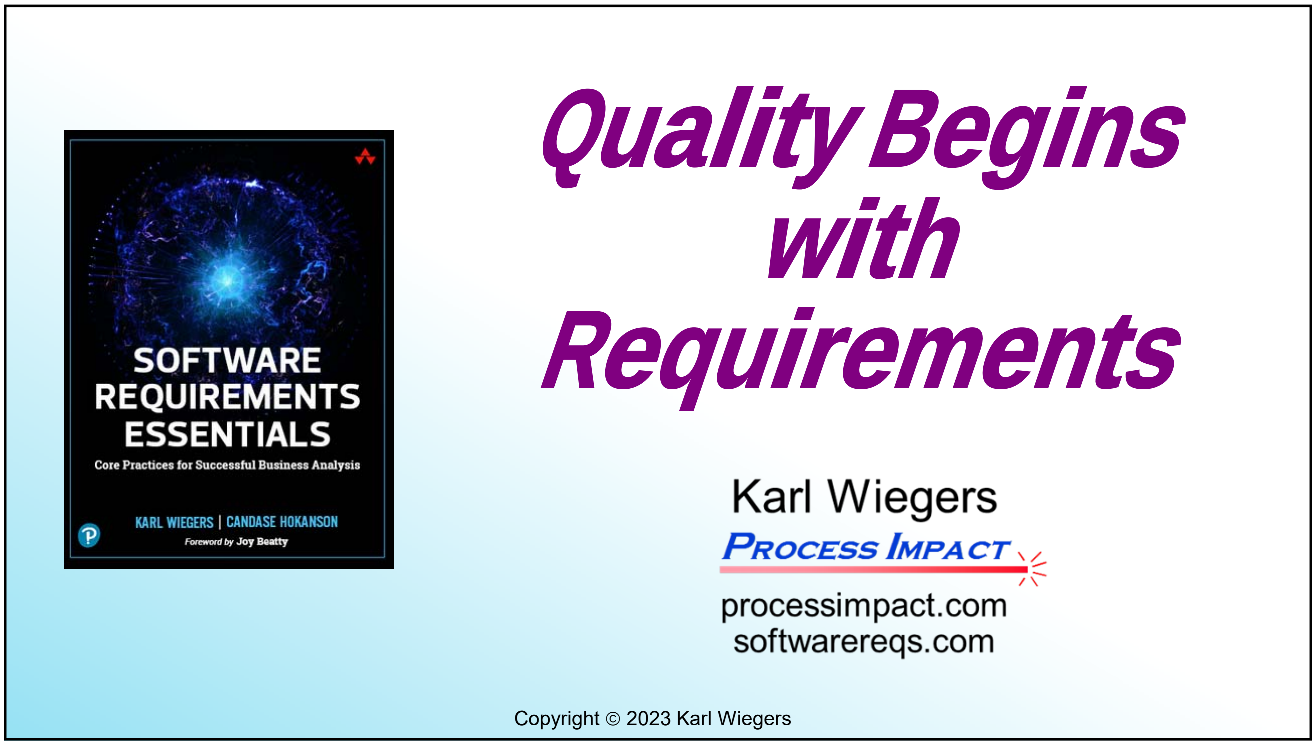
Requirements Management

- 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.

The Quality Payoff

👍 Reduced confusion about what everyone is working on
👍 Reduced chaos from frantic, uncontrolled changes



SOFTWARE REQUIREMENTS ESSENTIALS
Core Practices for Successful Business Analysis
KARL WIEGERS | CANDASE HOKANSON
Foreword by Joy Beatty

Quality Begins with Requirements

Karl Wiegers
PROCESS IMPACT
processimpact.com
softwarereqs.com