Summary Lecture Requirements Engineering

Lecture 12, DAT230, Requirements Engineering
Robert Feldt, 2010-10-13

Material for written exam

- All chapters for the two books listed on home page
- All articles linked to from home page
- All lecture and exercise slides/material
- Assignment material and what you learnt from there
- Not explicitly included:
    - Chapters from books not listed on home page (although some of that material may be covered by other material above and thus might be included)

“Personality” assignment

- We have not looked at your answers yet, to ensure no effect/bias
- Will do after everything else corrected
- Email with your results + links to norms

SEMAT assignment

- Many rejects: around 45%; you have two days to update after you get comments
- Main reasons:
  - General ramblings but no answer to question(s)
  - Missing references
  - Wrong format or missing intro/abstract/conclusion
  - Calling Dr. Jacobsson: Ivar, Ivan, ...

SEMAT answer themes

- 1. What is the critique that SEMAT and Ivar Jacobson present to the current state in Software Engineering and Development?
  - SE like fashion industry, governed by hype/trends
  - Rift between academia and industry
  - Methods not really tested in industry

SEMAT answer themes

- 2. In particular how does his and the SEMAT view(s) relate to the way we work with Requirements?
  - Upfront requirements are not good, things change over time
  - Too much documentation that nobody reads

SEMAT answer themes

- 3. What alternative(s) to the current Requirements Engineering practice does the SEMAT/Jacobsson view propose?
    - Suggestions of agile approach and skinny systems

SEMAT answer themes

- 4. Based on your personal Software Engineering/Development experience and views how and why would the proposed solutions/ideas of SEMAT/Jacobson solve some of the current problems (late or unsuccessful projects)?
    - Changes to Requirements formats

SEMAT answer themes

- 5. Discuss why you think the SEMAT alternative might not have its intended effects on the practice of Software Engineering.
  - This is another trend/hype that promise to solve everything

Group assignment

- “Problematic” group members will be further investigated
- We know of 3 persons so far
  - Little to no effort on group assignment
  - Likely to be many more; report or forget
  - We will contact problematic ones in coming weeks
- If we judge that you have not contributed enough
  - No point “cushion” on written exam (even retroactively)
  - Fail group assignment - rework

Group assignment: Presentations

- Customers were impressed with many groups!
- General notes:
  - Many talks about the users as “he”; very 20th century
  - Choose few/good presenters rather than many/mediocre
  - Good to have ok clothing when selling/presenting
  - Body language is important; no hands in pockets, hold in something if nervous, don’t read from slides
  - NOT ok to be late, lack of respect for everyone

Group assignment

- Winning groups: 20, 21, 11, 2, 12, 4, 17, 16, 15, 10, 1, 6
- You will have a 3 point "cushion" on written exam to get the higher grades (NOT useable to get a pass)

NatLangFR

Advantages	Flexible, Easy to understand for everyone, Use for any type, Fallback option, Easier use during meetings, No specific knowledge reqs, Easier to version control and prioritize
Disadv.	Ambiguity, Harder to “use” in further dev, Requires language skills, Can lack structure (too flexible), Dependencies harder to track (?), Harder to get overview
Efficiency	Quick, Saves time,
Use again	
Not use	Some reqs hard in text (UI, QR, Sequences)

I*

Advantages	
Disadv.	
Efficiency	
Use again	
Not use	

Use Cases

Advantages	
Disadv.	
Efficiency	
Use again	
Not use	

BDDC

Advantages	Suited for seqs, “Executable”, Makes them structured, Aligned with UCs, Connect to testing!?, Discover alternatives/other modes, Can be read by users (with less training than other models)
Disadv.	Requires training, Can be misused, Somewhat confusing terminology, Requires BDD-focused meetings, Can’t cover all req types
Efficiency	Easy and ~quick to write
Use again	
Not use	

NatLangQR

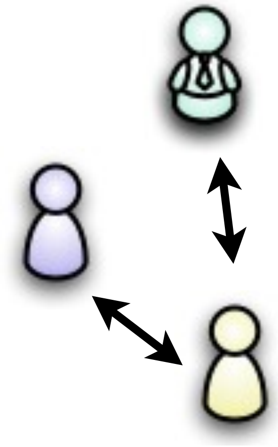
Advantages	
Disadv.	
Efficiency	
Use again	
Not use	

PLanguage

Advantages	
Disadv.	
Efficiency	
Use again	
Not use	

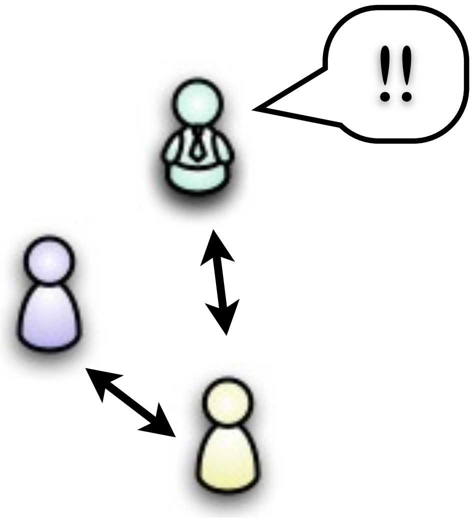
Document

Stakeholders



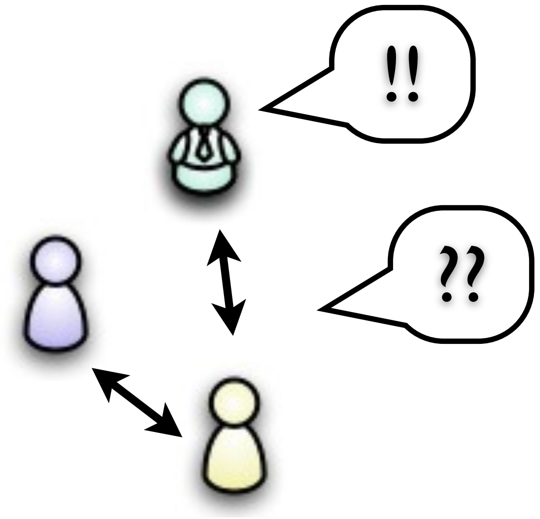
Relations

Say

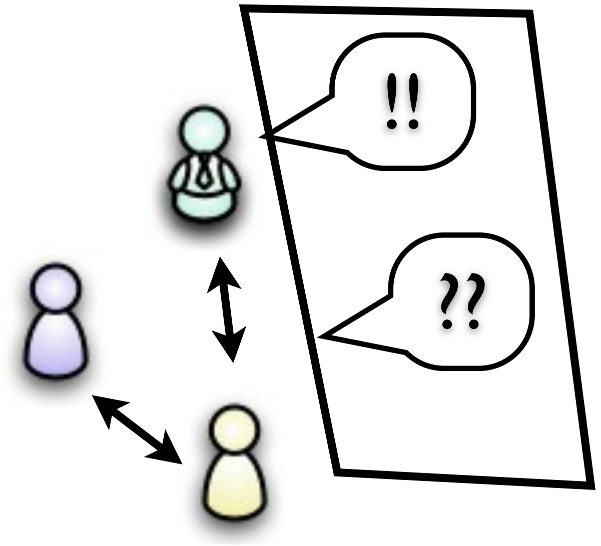


Need!

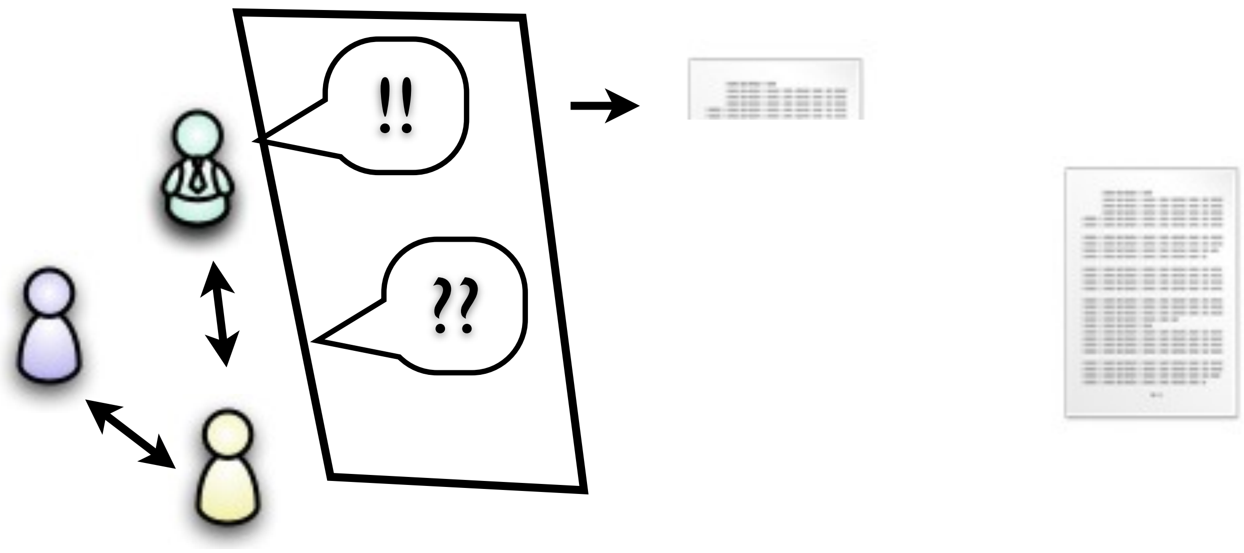
Say Think

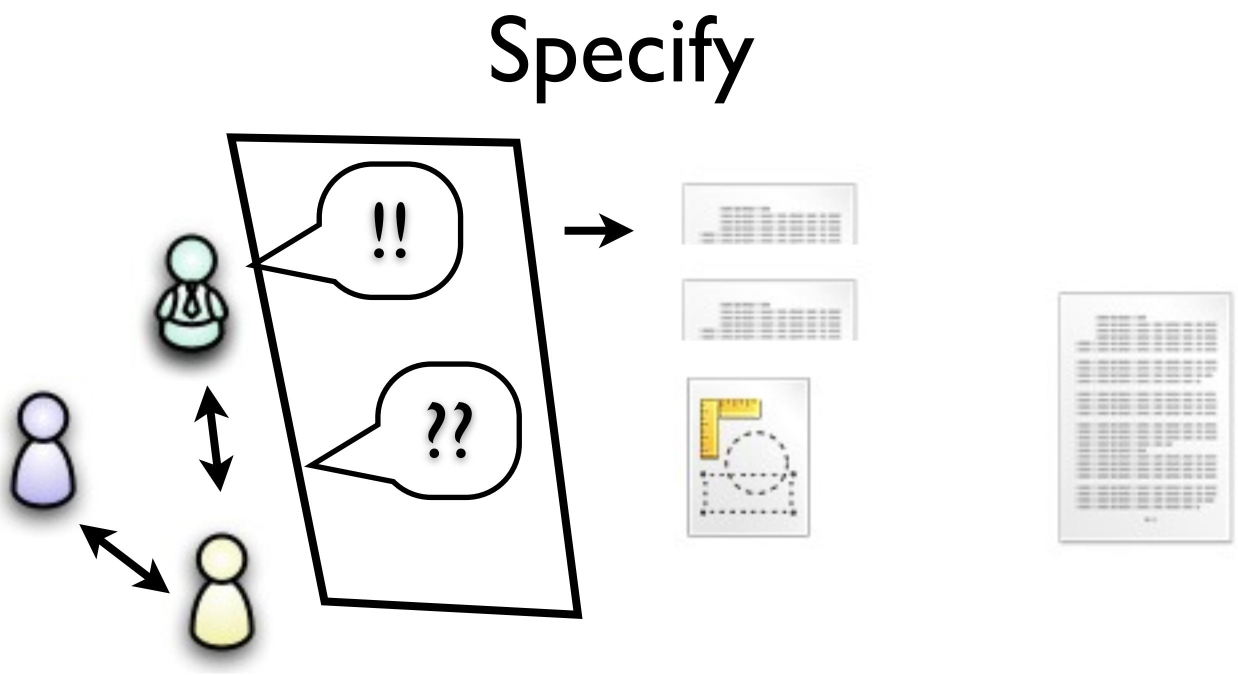


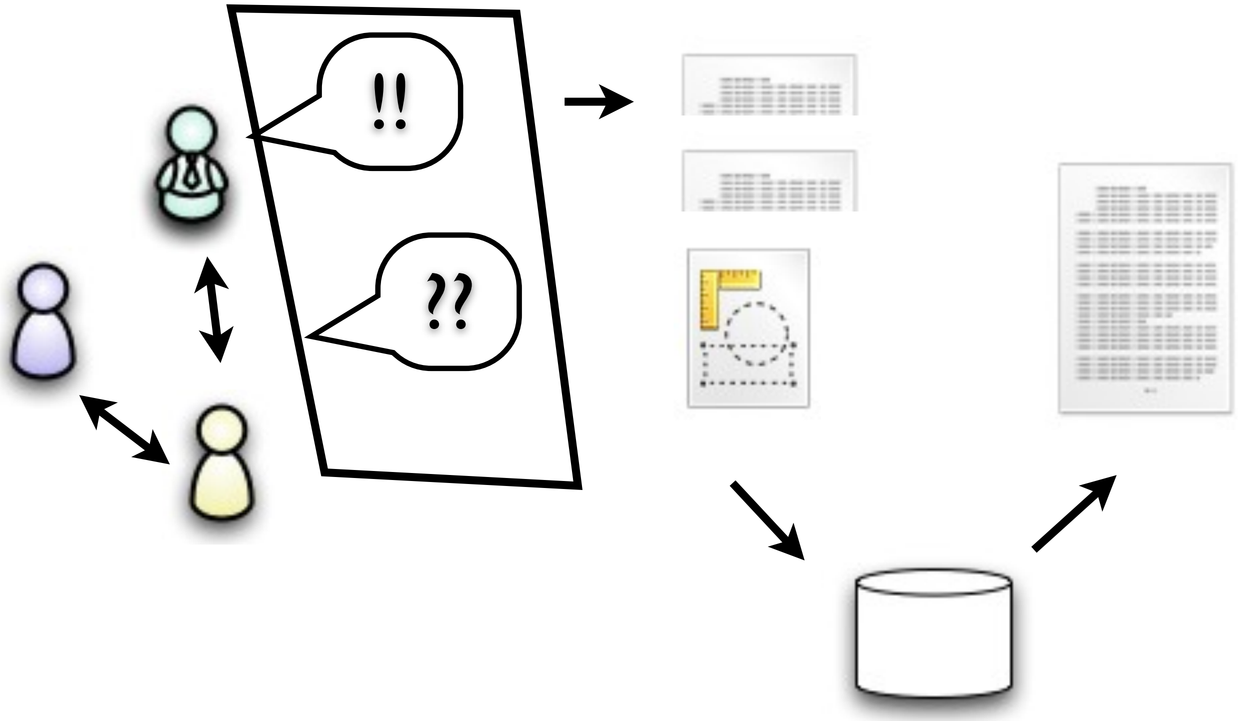
Capture



Transform

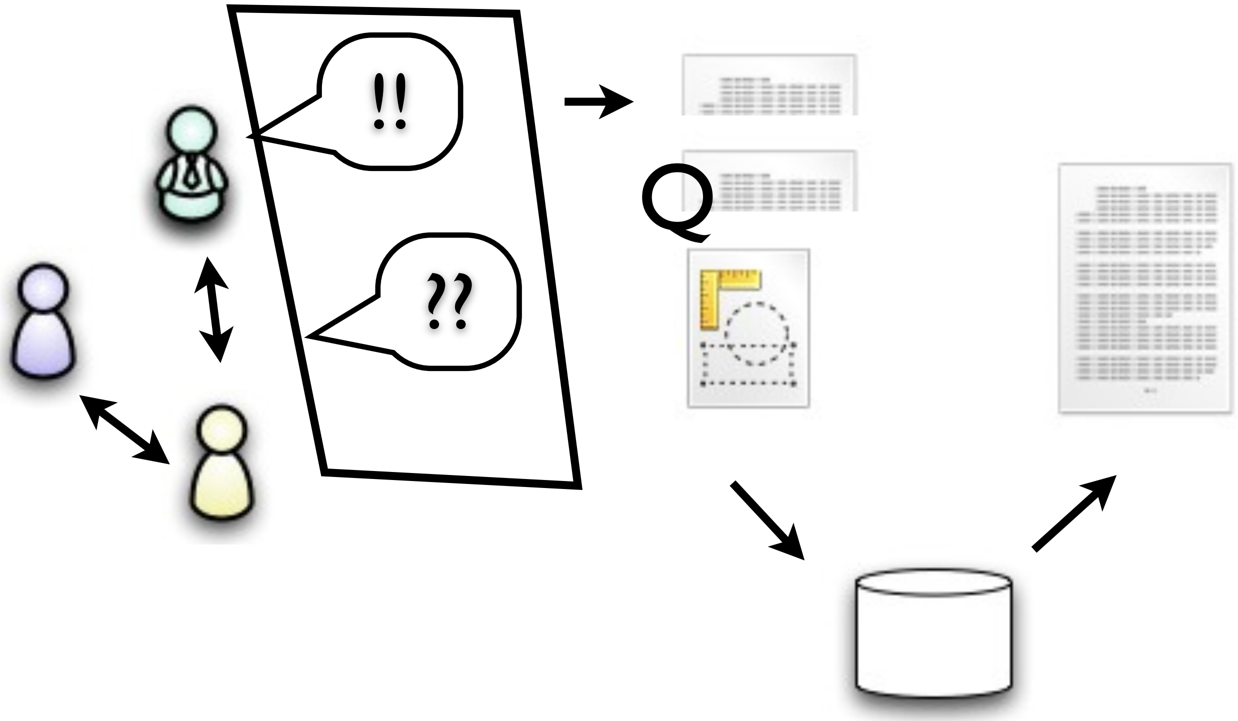



Specify
!!
??

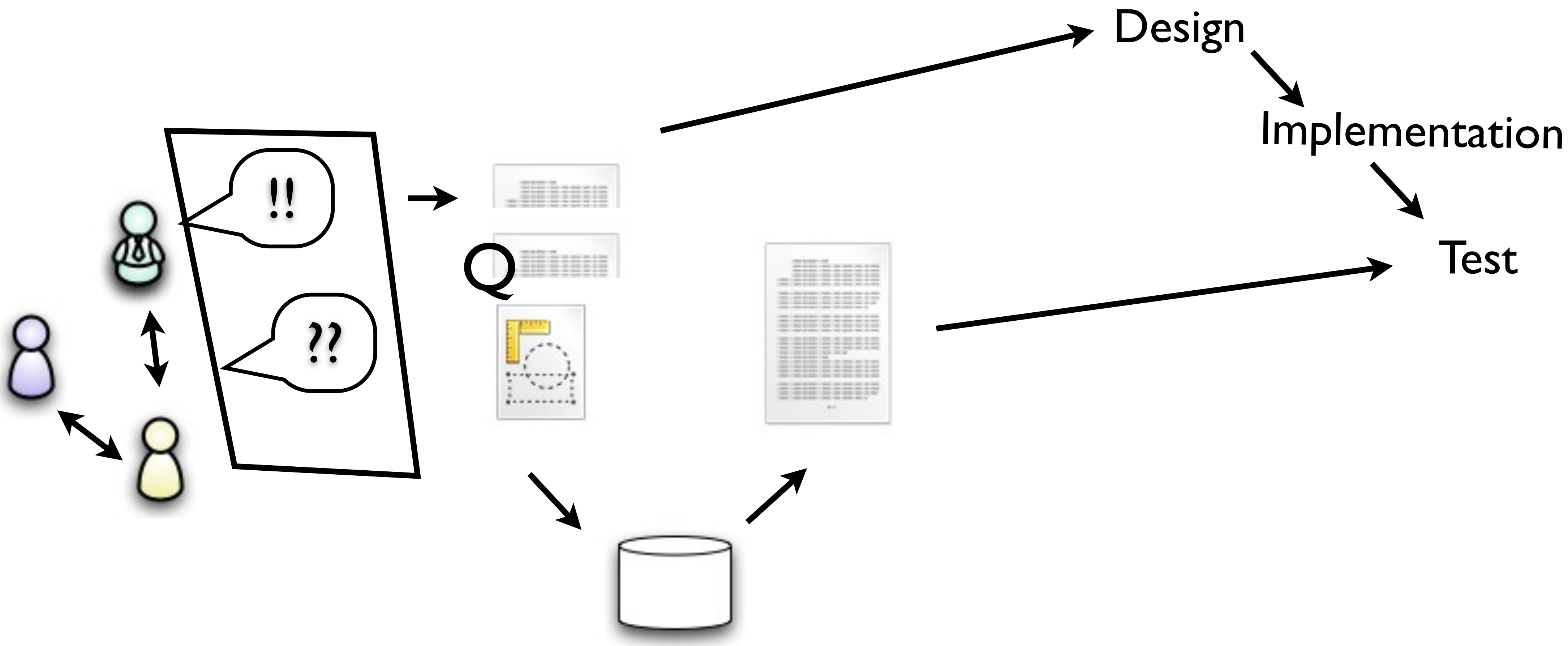


Store

Validation



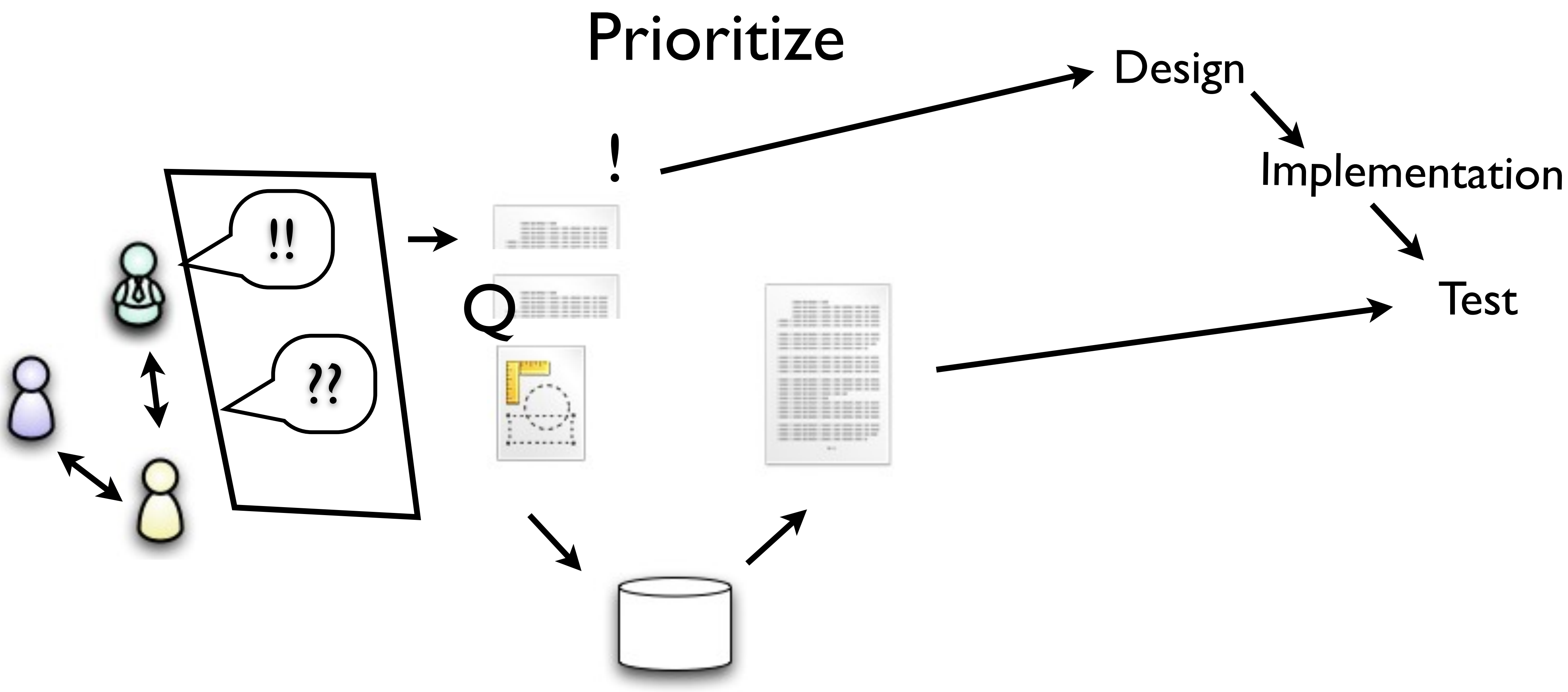
Process



Design

Implementation

Test



Prioritize

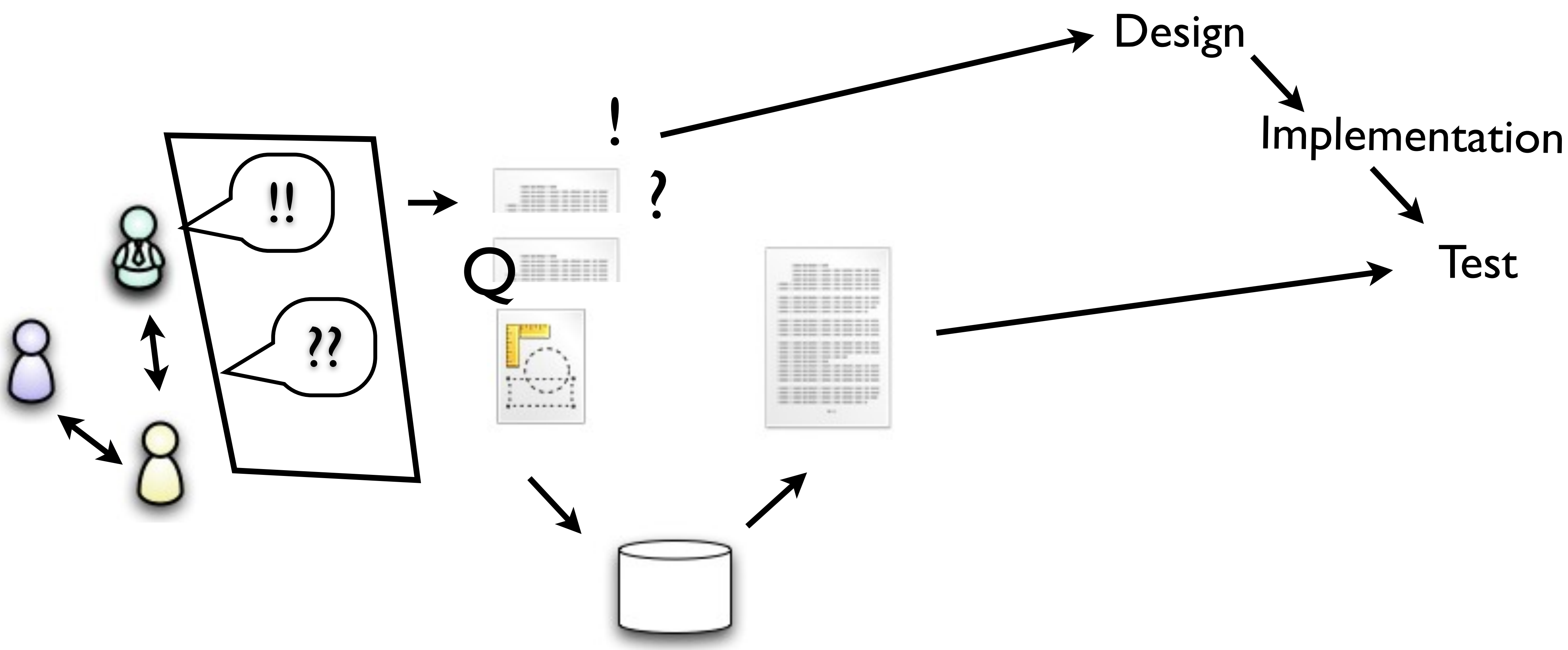
!

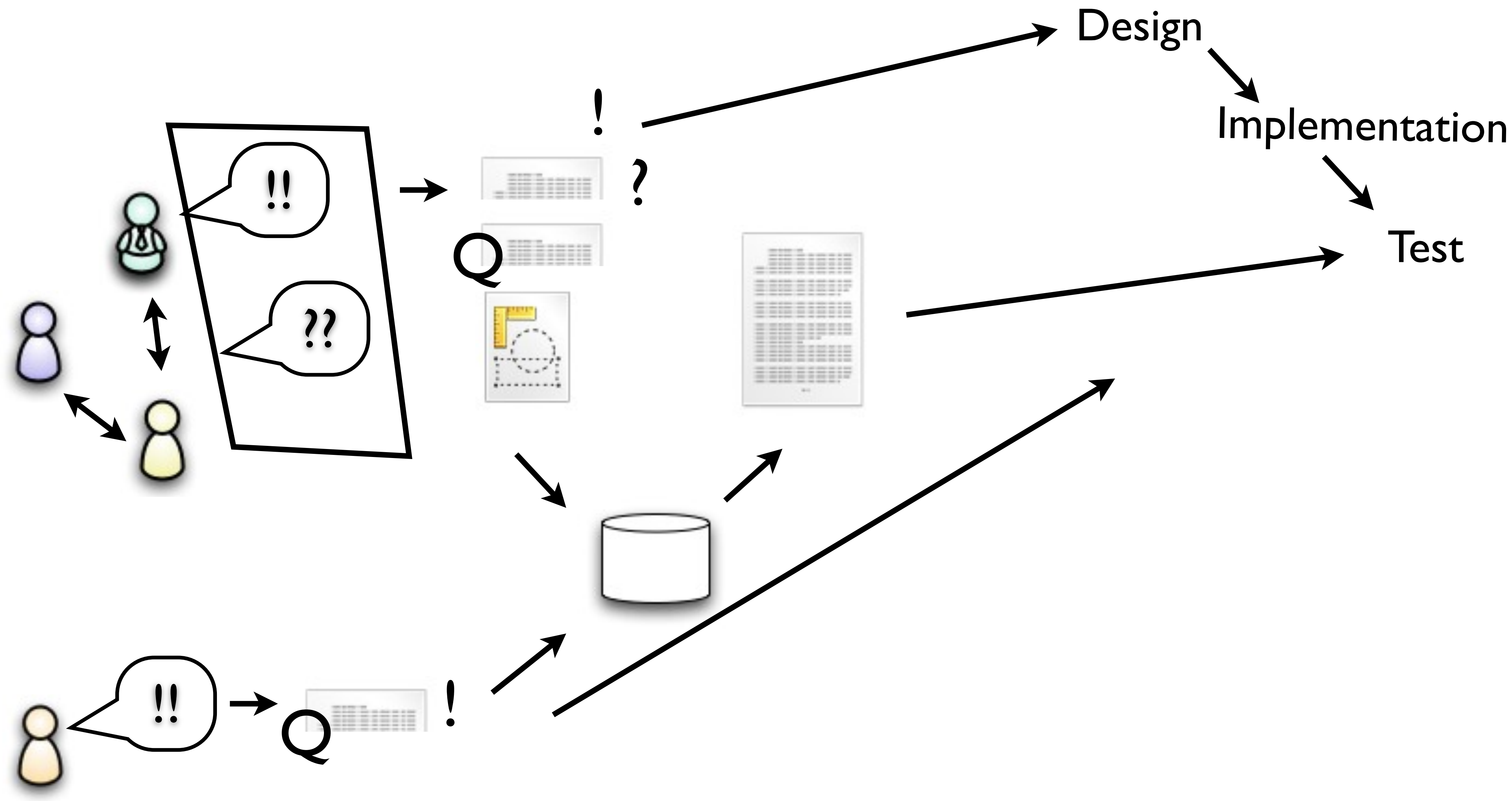
Design

Implementation

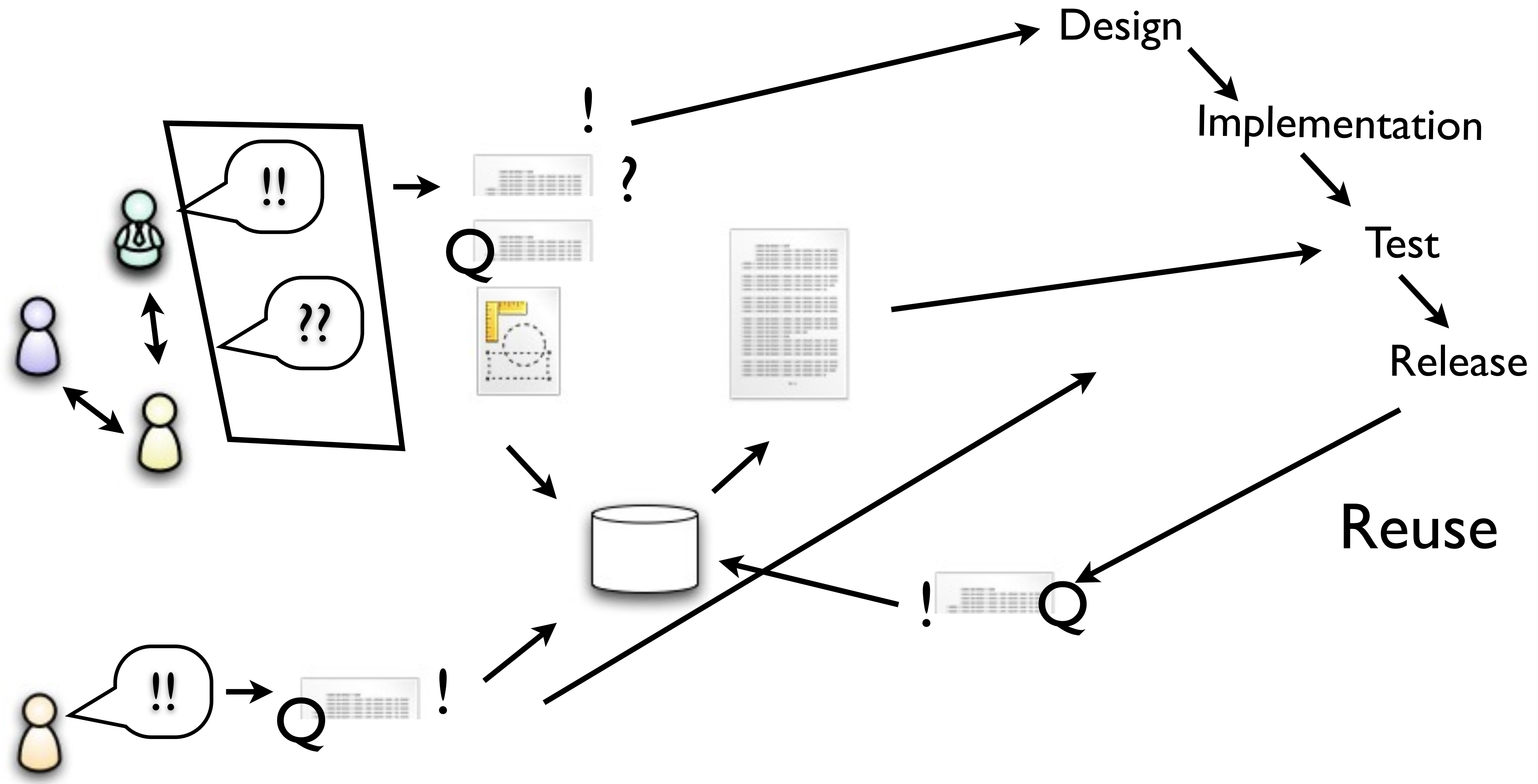
Test

Negotiate





Change



Design

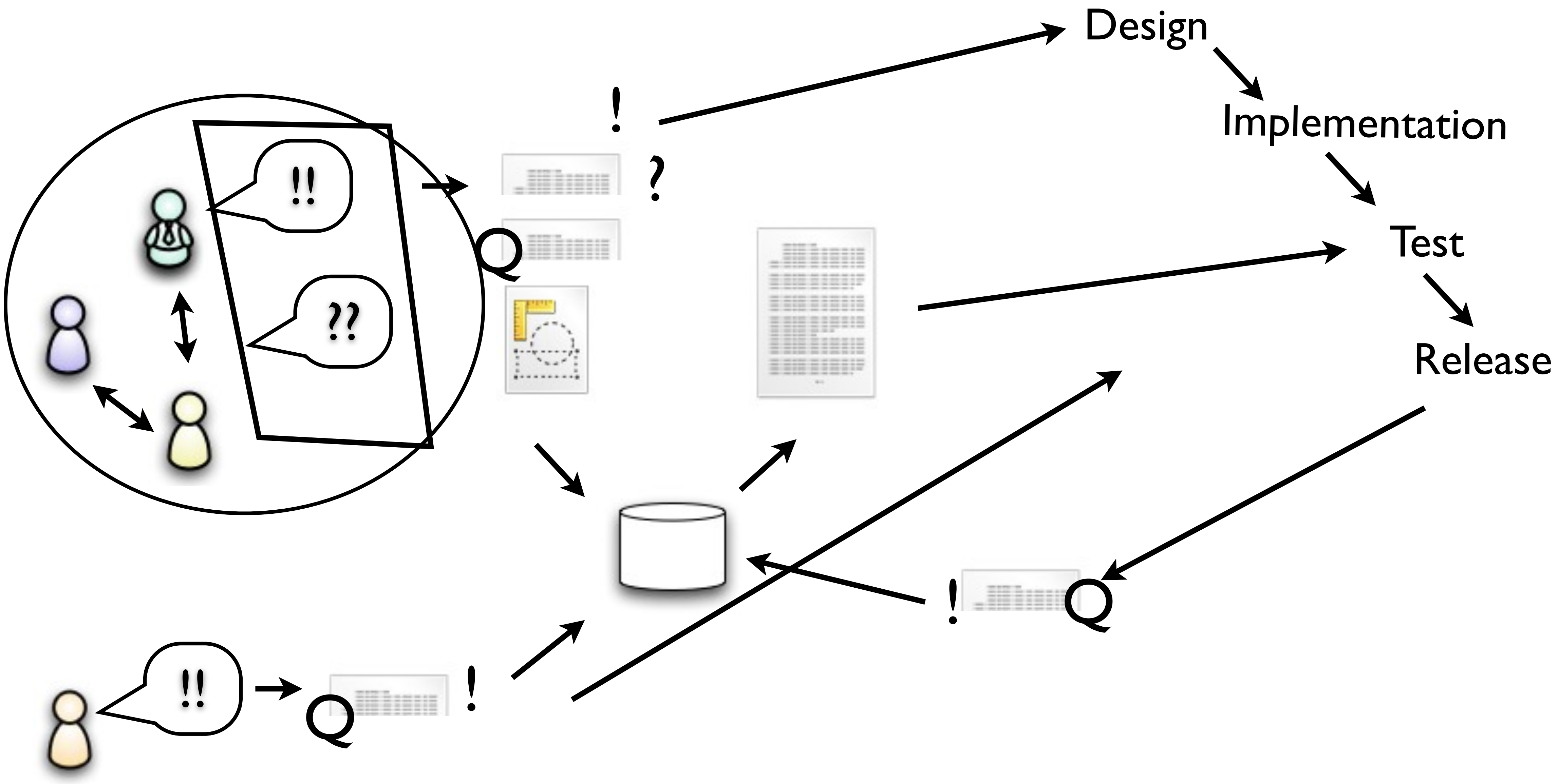
Implementation

Test

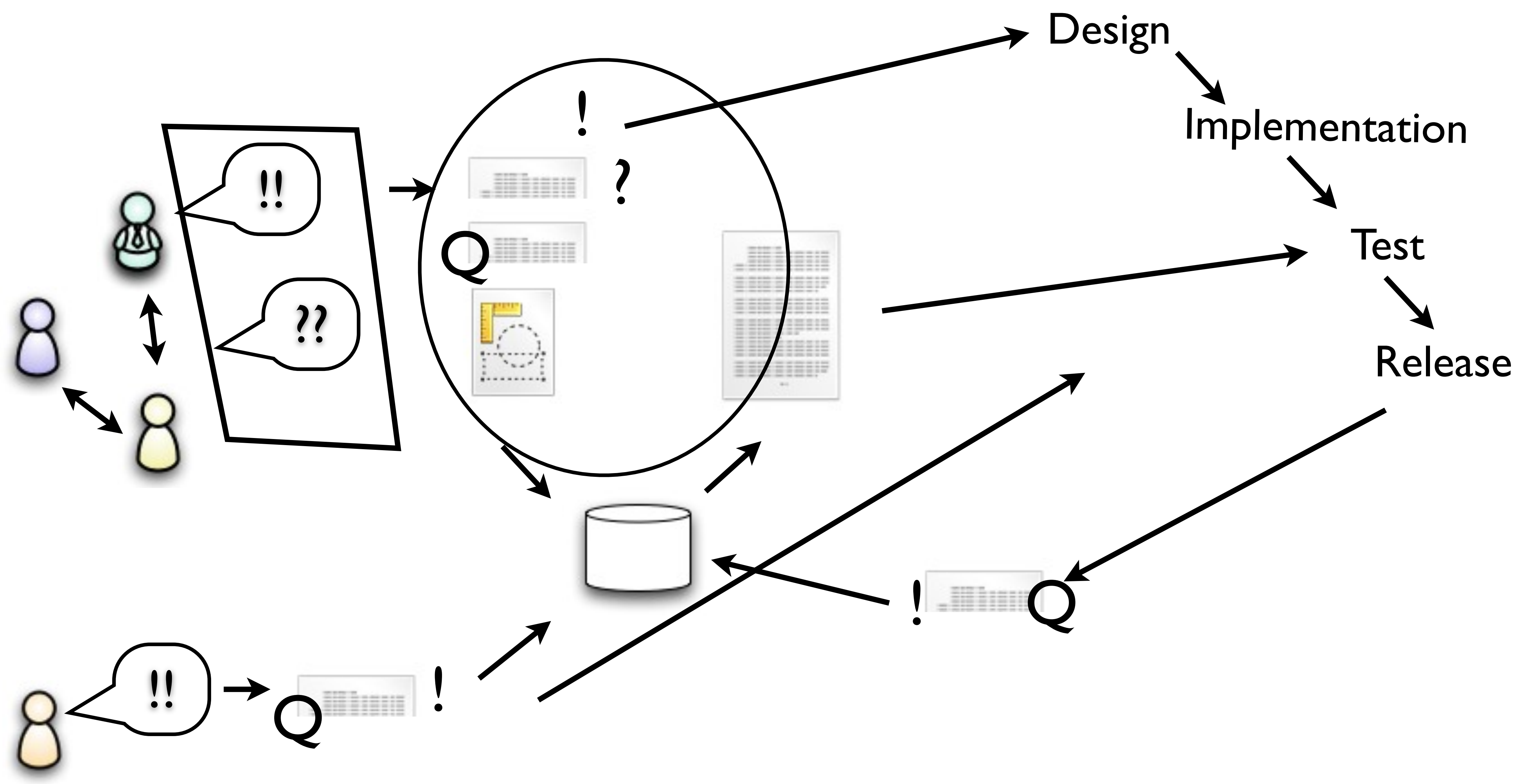
Release

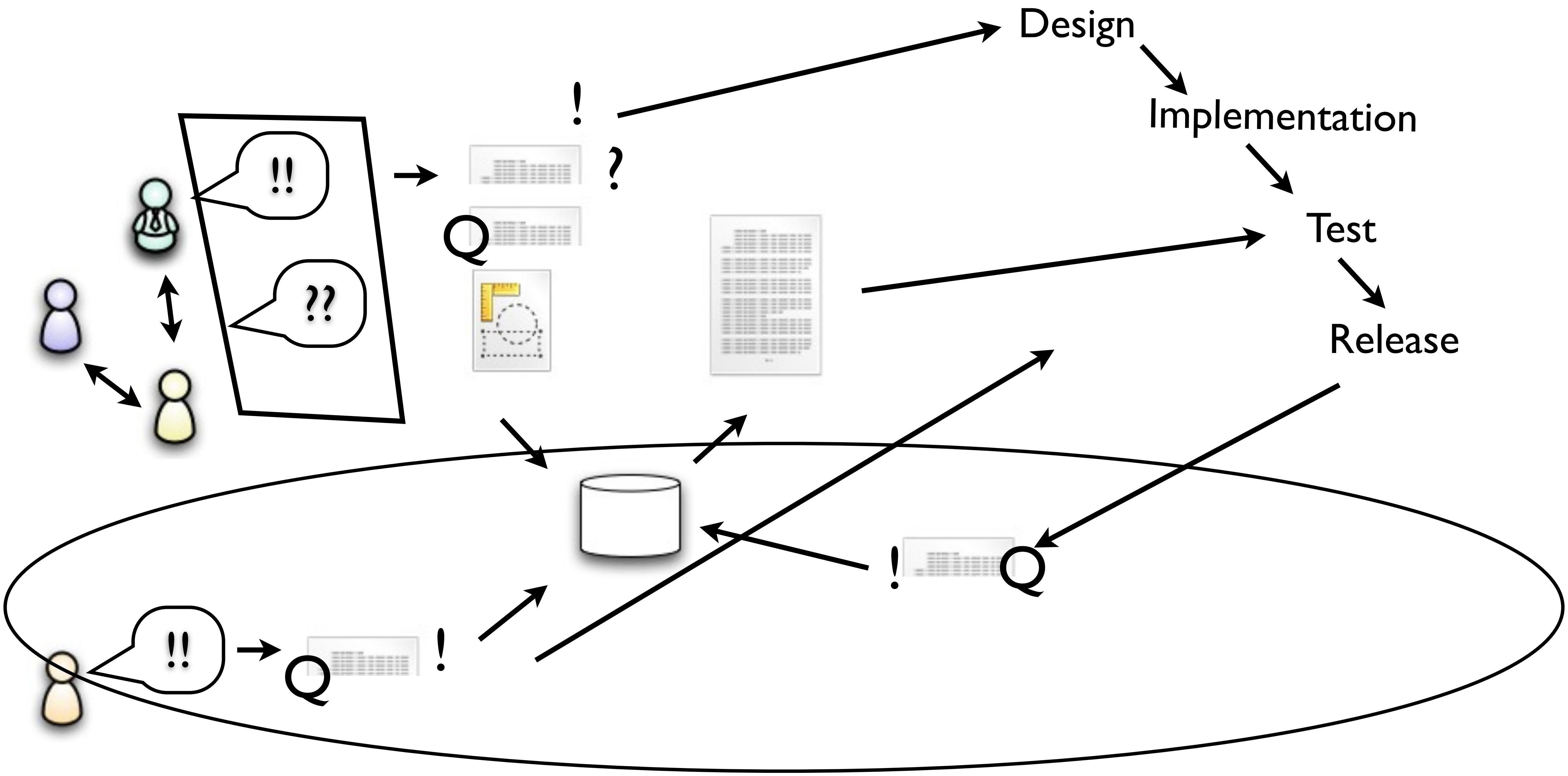
Reuse

Elicitation



Specification & Analysis





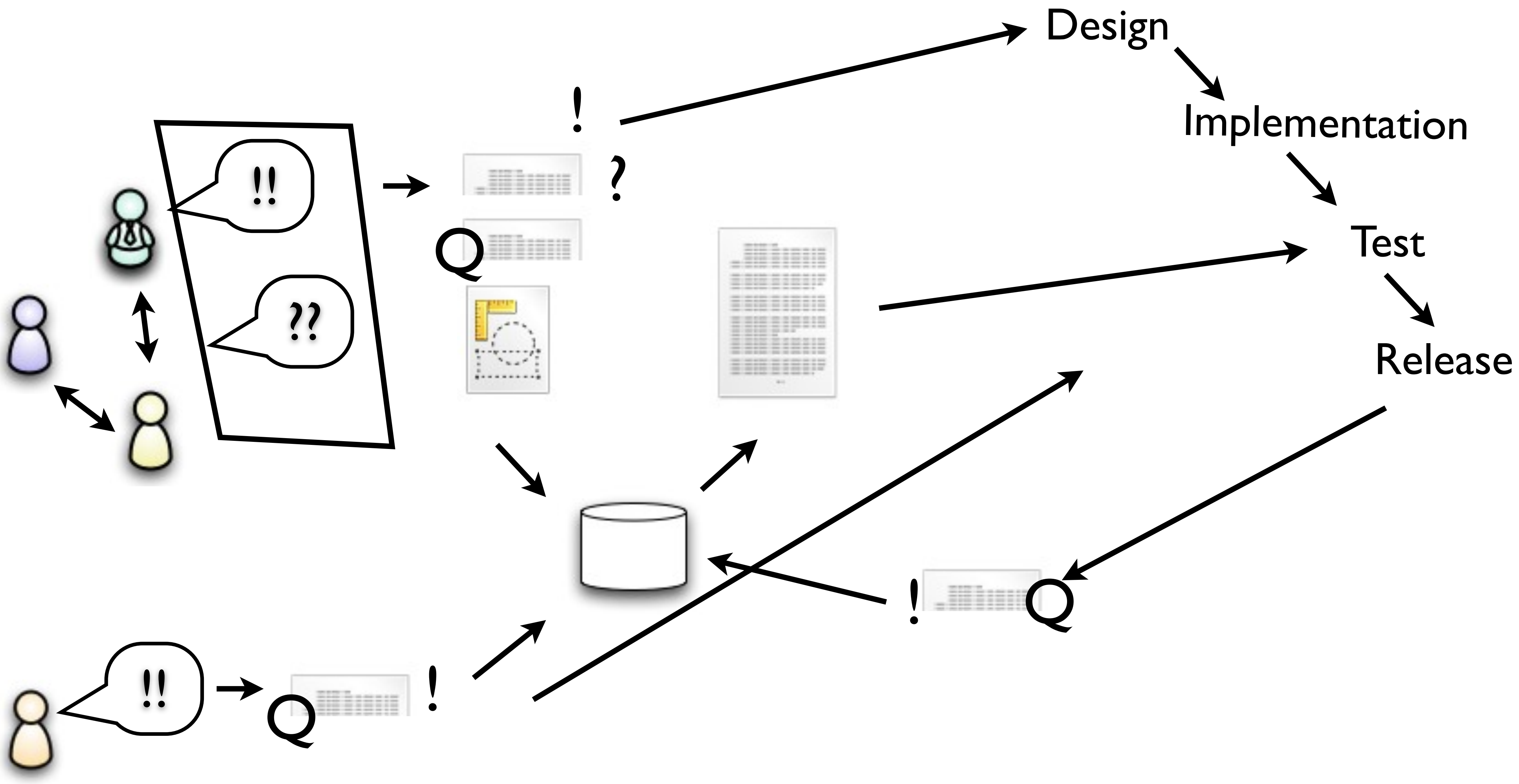
Design

Implementation

Test

Release

Management



Design

Implementation

Test

Release

Elicitation

- “understand customers’/users’ view of their problems/ opportunities
  - understand enough to proceed with ________ ____
  - never think ___ understand better than ___
  - never assume one __________ can speak for all ________
  - Maintain a _____ of terms
  - Prepare for ____ even after elicitation
  - Stakeholders have the right to _____ their mind

Prioritization / Triage

- “address only problems/opportunities we have time and resources for”
    - accept that there is no such thing as a ______ solution
    - record ___________ between reqs
    - plan more than one _______ ahead
    - plan to ________ before each release
    - goal is to select subset to product can be delivered on _____ and to ______
    - triage participants must see themselves as a _____ and not as part of separate _______
    - both marketing & dev should avoid absolute ______

Specification

- “record understandings so all parties see up front what to expect at the end”
    - goal is to spec to enough _____ so different stakeholders are ____ in their interpretation
    - select spec notations that customers ________
    - construct _____ where nat lang introduces high risk
    - use right ____ for the right job
    - customers want their problem solved, not to learn new ________

Change/Management

- “remain flexible as customer and user needs evolve”
  - changes to reqs are ____ not ___
  - do not try to limit the ____ of changes, ____ it
  - meet regularly to decide which reqs are in next _____
  - don’t accept more than ___ change per ___, or you is likely to fail

Course feedback

- Good?
- Bad?
- Ideas/improvements?
    -