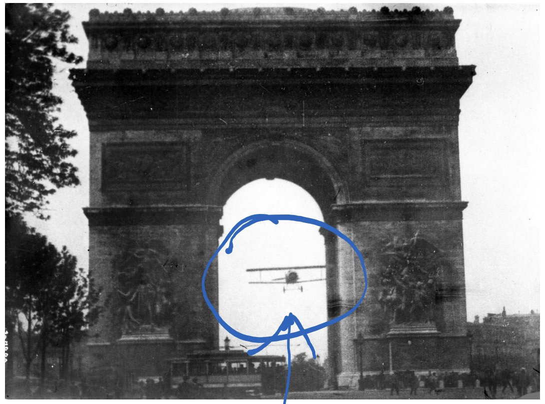
Lecture 15
Validating the Requirements: *Through the Quality Gateway*

Com S/SE 409/509

Robyn Lutz

rlutz@iastate.edu



By Jacques Mortane - http://www.histoire-image.org/site/oeuvre/full.php?oeuvre_id=479, Public Domain, https://commons.wikimedia.org/w/index.php?curid=6339842

Robyn’s Office Hours: **Tues & Thurs, 9:30-10:45**
Wandi’s Office Hours: **Mon 10 am**. & **Wed 7 pm**
Olukorede’s Office Hours--**Wed 10**

HW3 is due 10/15

Quality Gateway: Chapter 13

So far we have written *potential* requirements: some good, some bad

Problem: How to keep bad requirements out of our SRS (Software Requirements Specification)?

Solution: “Fly” each potential requirement through our **Quality Gateway**

Why worth the effort?

“Between 50-60% of errors in software development originate in the requirements & design activity” [Robertsons]

“rework needed to remove requirements errors typically accounts for as much as 50% of software development costs” [Robertsons, citing C. Jones]

- Requirements errors are the most expensive to fix
- Earlier its found, the cheaper to fix
- Avoid project failure/shelfware
- If specs are wrong, product is likely to be wrong

Goal: prevent incorrect requirements from going into the SRS

V&V

Verification: build the product right (good RE process) Volere

Validation: build the right product (accurate requirement specs in SRS)

What’s a **good** requirements specification (cont.)?

Checking the C’s

Complete

“any missing requirements?” (look to scenario steps, exceptional/failure & alternative scenarios; look to NFR types; look to startup/shutdown & boundary cases)

Clear

“can it be misunderstood?” (use EARS)

Consistent

“are descriptions and terms unambiguous?”

Conflict-free

“does satisfying it mean another requirement can’t be satisfied?”

What’s a good requirements specification (cont.)?

Checking the value

Needed

“relevant to purpose?” (look to rationale)

Valued

“does customer care?” (look to satisfaction rating)

Testable

“is fit criterion appropriate as a way to test if it’s satisfied?”

Feasible

“consistent with constraints; viable?”

Traceable

“backward to use case & provenance/forward to design?”

link

where it came from

link

where it's implemented

What’s a good requirements specification (cont.)? Sidebar: *Value ratings*

Measure value customer places on requirement (more in Chap. 16)

1. Satisfaction rating: how happy customer will be if product implements this requirement: 1-5

2. Dissastisfaction rating: how unhappy customer will be if product doesn’t: 1-5

Ex: “The product shall record changes to the road network.”

Satisfaction: 3

Dissatisfaction: 5 very unhappy!

Prioritize

Value = 8 (significant value)

What’s a good requirements specification (cont.)?

Avoiding the risks

Gold-plating

“fancy features that don’t add value?” (look for low dissatisfaction rating)

Requirements creep/leakage → water leaking into a boat

“is newly added requirement needed?” (look to scope; satisfaction rating)

“each requirement has a cost attached” [Robertsons]

creep can add ~30% of total cost [Robertsons]

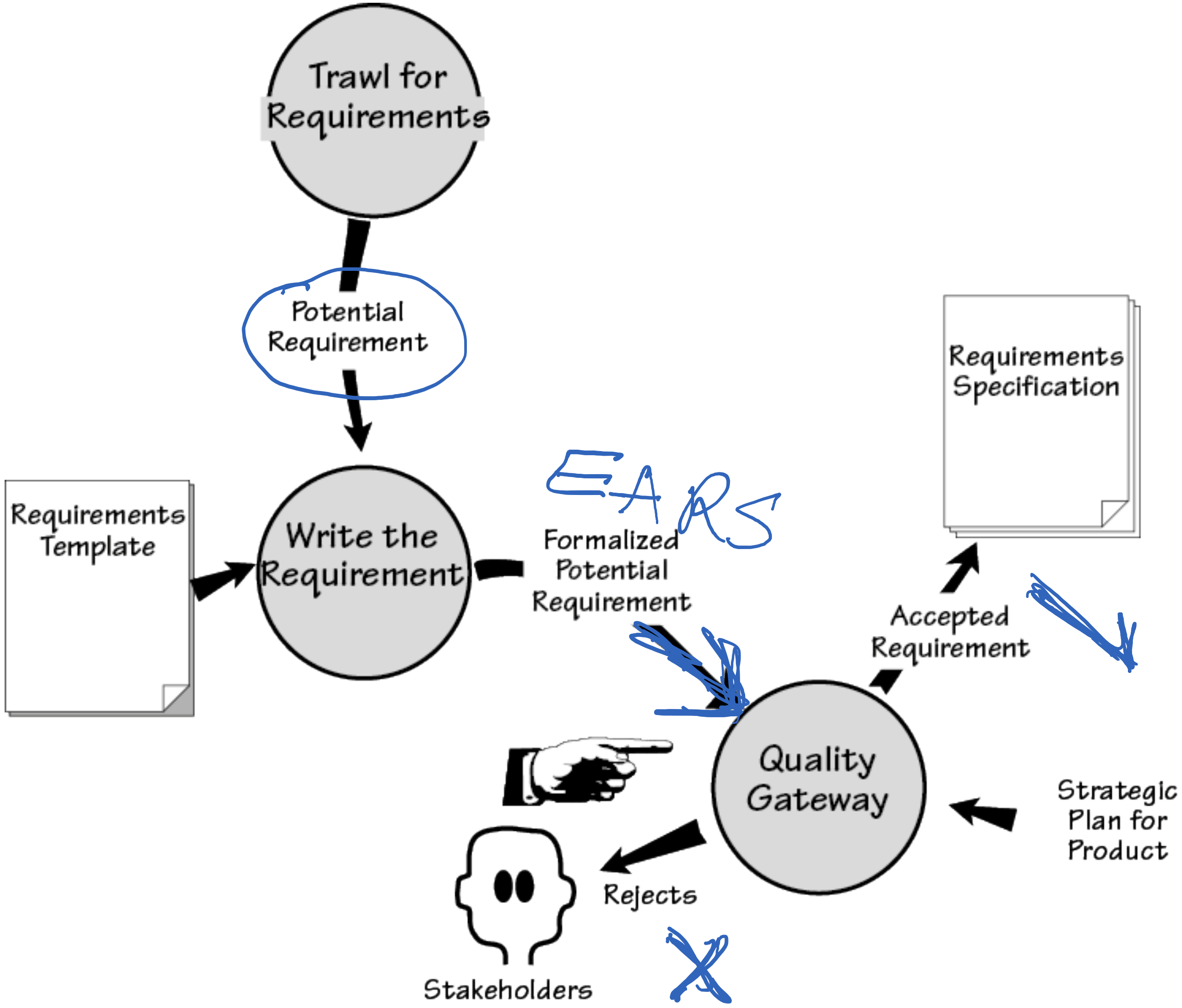
✱ requirements often do change during development

No unique identifier SRS :

“every requirement has its own identifier? (look to automated support)

Figure 13.1

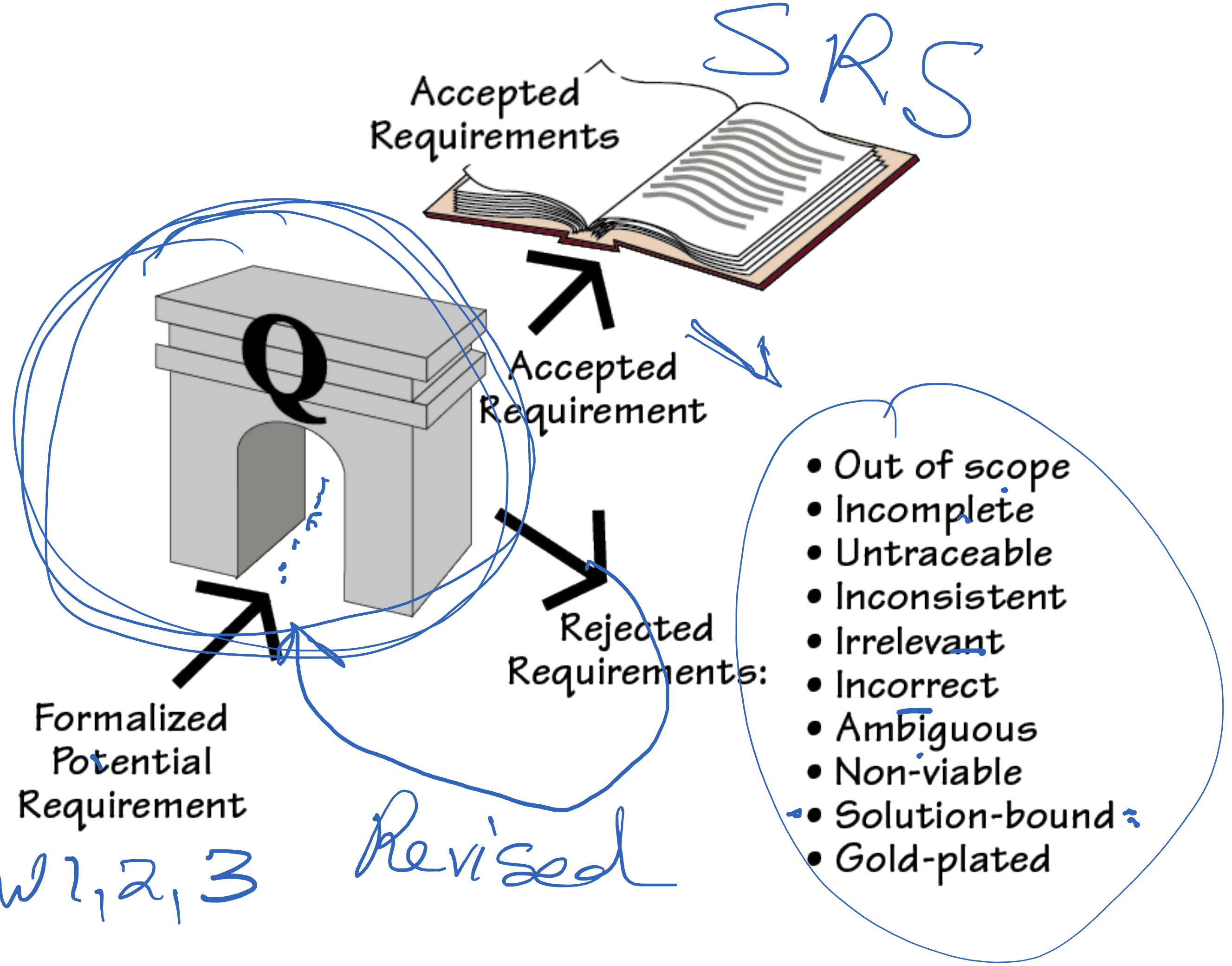
(An extract from the Volere requirements process.) The Quality Gateway is the activity where each requirement is tested to ensure its suitability. Suitability in this sense means that the requirement provides downstream activities with a clear, complete, unambiguous description of what to build. To ensure a suitable requirements specification, all requirements must be validated by the Quality Gateway.



From *Mastering the Requirements Process, Third Edition,* by Suzanne Robertson and James Robertson (ISBN-13: 978-0-321-81574-3)

Figure 13.2

The Quality Gateway tests each requirement for correctness and suitability. Accepted requirements are added to the specification; rejected requirements are returned to their originator.



SRS

Accepted Requirements

Accepted Requirement

Rejected Requirements:

- Out of scope
- Incomplete
- Untraceable
- Inconsistent
- Irrelevant
- Incorrect
- Ambiguous
- Non-viable
- Solution-bound
- Gold-plated

Formalized Potential Requirement

HW1,2,3

Revised

From *Mastering the Requirements Process, Third Edition,* by Suzanne Robertson and James Robertson (ISBN-13: 978-0-321-81574-3)