4.4 Characteristics of Requirements

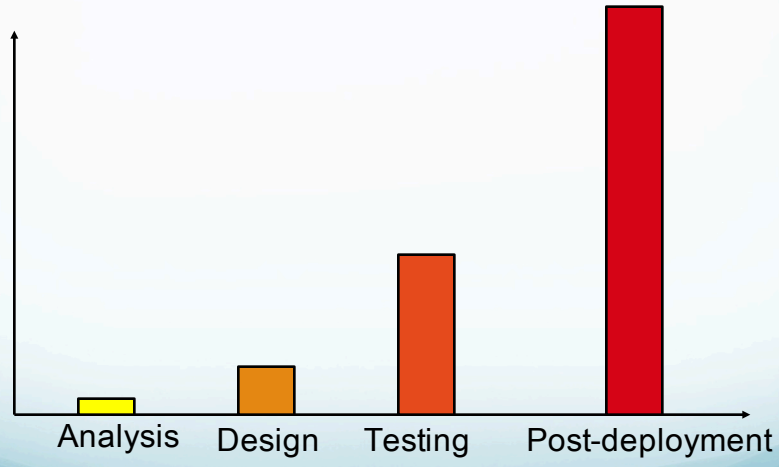
- Correct
- Consistent
- Unambiguous
- Complete
- Feasible
- Relevant
- Testable
- Traceable

Requirements elicitation/analysis

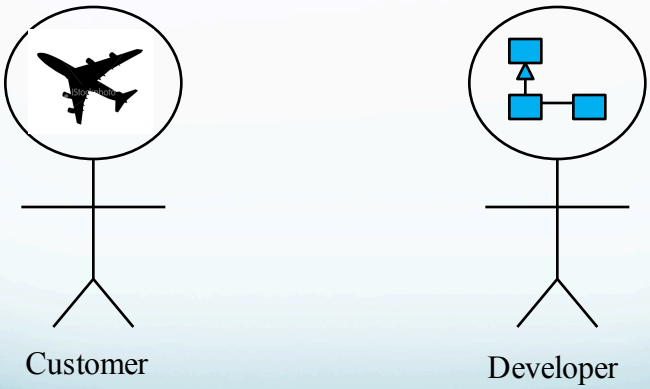
Topics:

- Problem statements, requirements, and elicitation

Cost of requirements errors by phase

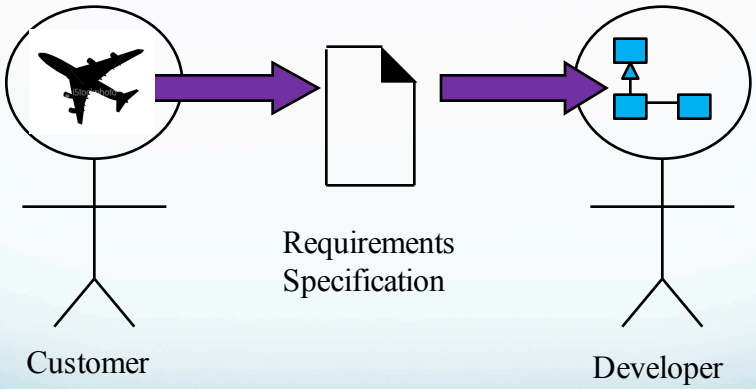


"Gulf" between client and developer perspectives on software requirements



http://www1.istockphoto.com/file_thumbview_approve/325621/2/istockphoto_325621_airplane_silhouette.jpg

"Bridging" the gulf



The requirements specification

Critical artifact, for many reasons:

- freezes "what" is to be developed
  - should be no new requirements added once development begins
  - equivalent to a "project handout" in a programming course
- part of the "contract" between client and developer

Two schools of thought on notion of "freezing" reqts

- It is a myth to think we can freeze requirements; therefore, we must develop software assuming new reqts will arrive after development begins
- It is vain to think we can develop a quality system if new reqts are added after development begins; therefore, the reqts spec should be thorough and subjected to quality assurance

Waterfall processes subscribe to the second view

Issues

Given the critical nature of the requirements specification, important to get it right!

Meta-requirements (i.e., reqts of a reqts spec):

- consistent
- complete
- understandable by both client and developer

Question: Why might these meta-requirements be difficult to satisfy?

Completeness/consistency problems

Consistency problems:

- Multiple interpretations of similar terms
  - developer's vocabulary vs. client's
- Concepts "built upon" undefined concepts/terms
  - E.g., scheduling system based on notion of "constraint"
  - But constraint never really defined
    - ★ E.g., is a "recurring commitment" one or multiple constraints?

Completeness problems:

- Customer may have only a fuzzy understanding of what he or she wants
- Developer lacks "implicit knowledge" of client domain

Elicitation techniques

Client View of Domain

Clients cannot be expected to have rigorous or formal view of domain

Hence, cannot be expected to completely be aware of domain-problem relationship

Some knowledge is explicit

- Easier to get at…

Some knowledge is implicit

- Many constraints are implicit
- Hard to get at…

Technique: Initial client interview

Goal: Discover as many requirements as you can in a limited amount of time

Implications:

- Essentially an information-extraction process
- Ask open-ended questions
  - ask them in more than one way
- Your analysis should be very limited
  - OK to ask follow up questions, but don't get bogged down analyzing one requirement, or you will run out of time
  - Never (during this interview):
    - ★ suggest a "better way to think about it"
    - ★ express opinions on answers

Question Structure is Critical

What is the client's problem?

- what, precisely, is the problem to be solved?

When does the problem occur?

- what generates the problem?
- situations, are they new or old? Transient

Where does the problem occur?

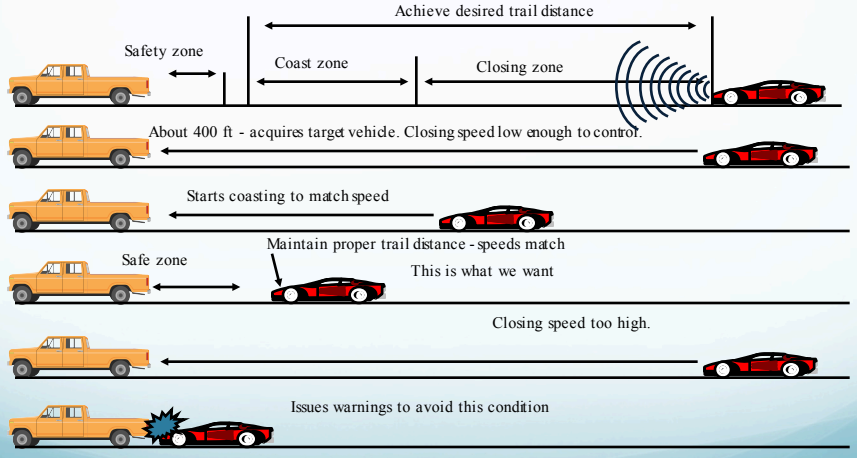
- what are the problem domain boundaries?

How is the problem handled now?

Why does the problem exist?

Remember, this is a diagnosis / information extraction process

Sample System: Smart Cruise Requirements



Closed-ended questions

Q: When a vehicle cuts in front of the car, you have to slow down quickly and not hit it, right?

A: Yes

You learned absolutely nothing.

Open-ended questions

New reqt!

Q: What happens when a car cuts in front of you?

A: Well, if the lead car is too close, the driver has to intervene or else a crash results. I guess we need a warning light in this case. If the car is moving faster, you don't have to do anything. He's pulling away. I guess the only time brakes are used is when the closing speed is too high for the distance and yet within the capabilities of the system to slow down. But I guess if a collision is imminent, we should max out the braking.

Clarification

Now, we learned something...

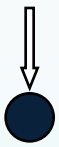
Dialogue with different responses

Q: Tell me what should happen if a car cuts in front of our car too close to avoid a collision?

A: I guess since there is nothing the system can do, turn off the controller and hope the driver brakes in time.

Q: What? Are you nuts? We should at least try to stop. Shouldn't we?

A: Perhaps...



You are done at at this point, and still unresolved.

Q: We have quite a bit of braking power in the system. What would happen if we used it here?

A: Well, I guess it could avoid a collision and at least get the car slowed down but the attorneys tell me we don't want the system active when a collision occurs.

Ah ha! Non-technical constraint

From elicitation to analysis...

Your interview should result in a large volume of facts that must be analyzed to derive requirements

- Here "analysis" involves both analysis and synthesis
- Synthesis: attempt to compose a coherent "model" of the problem requirements

A model can be analyzed to:

- identify potentially inconsistent facts, and
- infer facts that should be true

Both of these issues must be clarified, often via a second client interview

Putative questions

Asks about a situation in a way that tests your model of the domain

SWE:
"If a lead vehicle turns, or otherwise is not in front of the car anymore, the car can resume the previous speed, correct?"

CLIENT:
"Yes, exactly."

Very specific question that tests the idea of cruise plus collision avoidance

Sample Interview I

SWE:
Could you tell me about the cruise control system?

CLIENT
Yes, normal cruise control holds a fixed speed. What we want is to make the car "smart" so that it slows down when there is a vehicle in front of it.

Establish facts, but open ended. It's seemingly obvious what happens "now"

SWE:
What does a driver currently do in this situation?

CLIENT
Currently, the driver can step on the brakes to disengage the cruise, or turn the cruise off completely. Or, not use the cruise.

SWE:
Why is turning off the cruise this way a problem?

i.e., Why do you need "smart" cruise?

Try to get at the motivation for the problem

Sample Interview II

CLIENT
In an urban environment, say I-75 in Detroit, using the cruise becomes irritating, but really we are more interested in avoiding collisions.

The system is mis-named. This info is good useful

SWE:
Tell me more about the collision avoidance aspect, please.

open, info gathering

CLIENT
If we limit how close a lead vehicle can get, and control the speed while the car is in trail, the chances of a collision can be greatly reduced.

SWE:
How would a system avoid a collision in a typical scenario?

Looking for process/behavior information

CLIENT
Suppose the driver is following a truck, but at a higher speed than the truck. As the car closes, the system could alter the speed to match the speed of the truck.

Specific request for facts

SWE:
What does the slowdown profile look like?

Sample Interview III

Great insight

CLIENT
Well, we have discovered that slowing down linearly over a long distance can lead to other cars cutting in front of you. This is also not what a human driver does. Instead, we continue at out current speed and start a coast when we compute that we will get too close.

Very specific, to resolve ambiguity in domain terms

SWE:
What is "too close"

Ok, we can infer what this statement means

CLIENT
Oh, within 2 seconds of trail distance

Time for a putative theorem to verify current model that resolves ambiguity

SWE:
Does that mean at 60 mph, 88 ft/sec, too close is 176 ft?

CLIENT
Yes, closer than 176 ft is too close.

verification

Sample Interview IV

SWE:
What if a car cuts in front of you within the "safe" 2 second distance?

filling in model

CLIENT
I guess since there is nothing the system can do. Turn off the controller and hope the driver brakes in time.

SWE:
The specs indicate we have a fair amount of braking power available. What would be the problem with using it here?

vs. “can’t we use…”

CLIENT
The system does have access to the brakes, which are anti-lock. Technically, we could apply the brakes, but at the moment, our attorneys tell us we'd rather not have the system active if a collision is imminent.

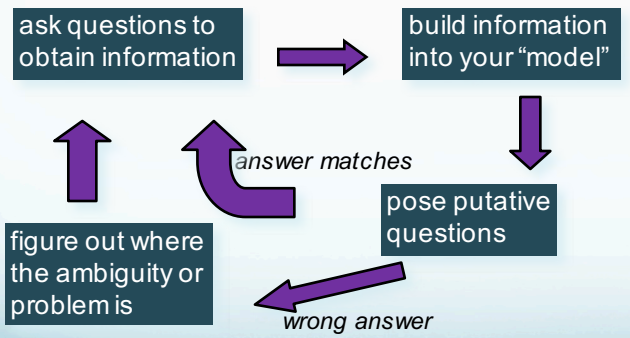
A non-technical issue arises

Types of Questions as Tools

Why...	→	Usually leads to deeper motivations, information on structure.
What...	→	Usually leads to facts
How...	→	Usually leads to a discussion of process, not structure
Could...	→	Maximally open, might lead to no data "could you explain why the safety zone is 2 seconds?"

Elicitation/analysis structure

Elicitation/analysis may require multiple interviews



Summary

Elicitation is critical to:

- address the requirements-completeness problem
- support analysis, which aims to address the requirements-consistency problem

Client interviews are a useful tool, but:

- Must be carefully planned and orchestrated
  - Meetings should focus on a primary goal (e.g., information extraction vs. clarification)
- Big mistake to fail to plan for some iteration here