Lecture 12
Communicating the Requirements: Rationale

mountvernon.org

Com S/SE 409/509

Robyn Lutz

rlutz@iastate.edu

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 posted; due 10/15

Rationale answers: “Why this requirement??”

- Def’n: The rationale is the **reason**, or justification, for a requirement.” (Chap. 12)
- Helps developers find the fit criteria
- Helps developers identify overly strict requirements
- Helps developers decide among trade-offs
- Helps testers understand priority of requirement
- Helps maintainers understand change impact

NASA Study on Flight Software Complexity
Dan Dvorak, Editor, 2009

“2.2 **Emphasize Requirements Rationale**

2.2.1 **Finding**

Requirements that are unnecessary or that specify unnecessarily stringent performance targets cause extra work and add complexity, whether in analysis, design, software, testing, operations, or some combination thereof. **The standard defense against unnecessary requirements is a statement of rationale that substantiates why a particular requirement is necessary.** However, the study found that rationale statements are often missing or superficial, or even misused in the sense of providing more detail about a requirement (rather than substantiating it).

In one mission a scientist levied a requirement for “99% data completeness,” implying that science results would be greatly diminished by any interruptions in science observations. The flight software team took the requirement seriously and designed and developed a system with redundant elements and fast onboard fault detection and response, making for a more complex system. Later in the project, somebody questioned the value of 99%, and the scientist—realizing that it was **overly stringent**—quickly relaxed the requirement. Unfortunately, the damage was already done; an unsubstantiated requirement had spawned an unnecessary cascade of time-consuming analysis, design, development, and testing, not to mention the dismay of the software team who had worked so hard."

NASA Study on Flight Software Complexity

“2.2.2 Recommendation 1

Project management should emphasize the importance of requirements rationale to the people who write requirements and ensure that they know how to write a proper rationale. “

A more recent study confirmed that missing, inadequately documented and incorrect requirements rationales contributed to defects during integration & testing of spacecraft software [Lutz et al., 2013]

Reducing the Risk of Overly Strict Requirements

- Consider easing it
- Consider removing it
- Consider variations and alternatives
  - Some delivery drones need less precise location info than others
- Consider relaxing it at runtime

Some requirements are infeasible → Ease it

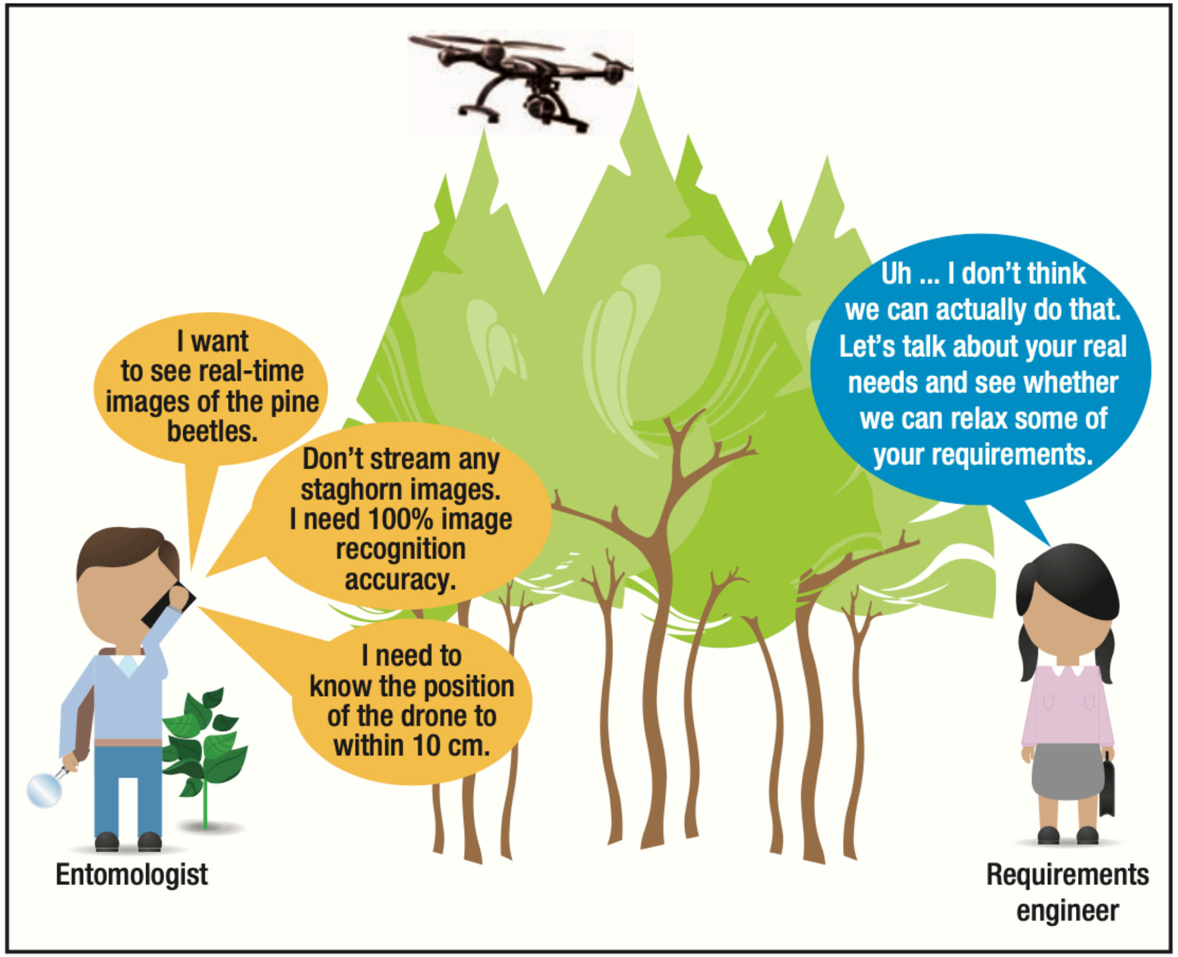
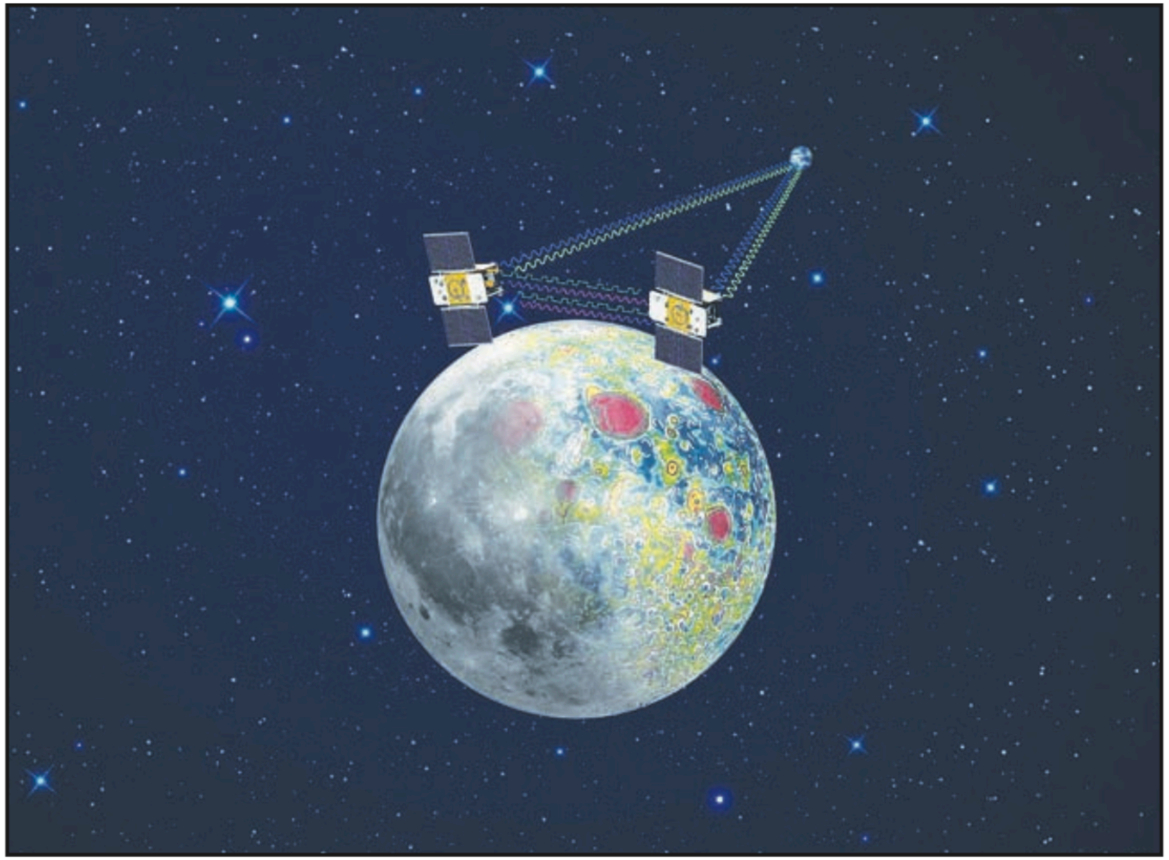

I want to see real-time images of the pine beetles.
Don't stream any staghorn images. I need 100% image recognition accuracy.
I need to know the position of the drone to within 10 cm.
Uh ... I don't think we can actually do that. Let's talk about your real needs and see whether we can relax some of your requirements.
Entomologist
Requirements engineer

FIGURE 1. The essential and sometimes unenviable role of the requirements engineer in reducing the risk of an overly strict requirement.

[Lutz and Cleland-Huang, IEEE Software, 2017]

Some requirements are inconsistent with project’s budget and schedule → remove it



[Lutz and Cleland-Huang, IEEE Software, 2017]

FIGURE 2. The GRAIL (Gravity Recovery and Interior Laboratory) spacecraft, which was launched to do gravity-field mapping of the moon. Such Discovery Program missions have smaller development budgets and shorter development timelines than flagship spacecraft projects. So, some requirements typical of larger spacecraft weren’t justified for inclusion on GRAIL. (Image courtesy of NASA/JPL-Caltech.)