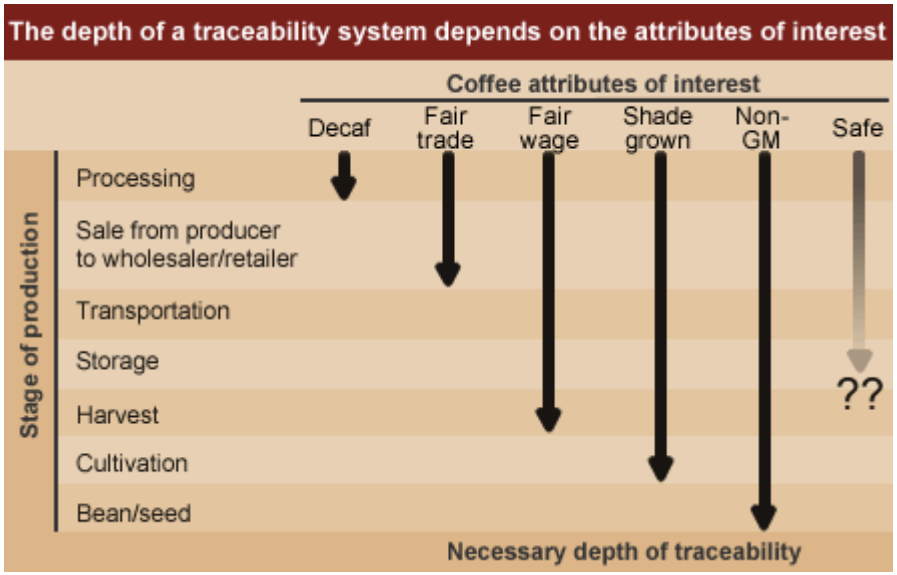


Traceability in the US food supply, from http://www.ers.usda.gov/Briefing/Traceability/overview.htm.

Traceability

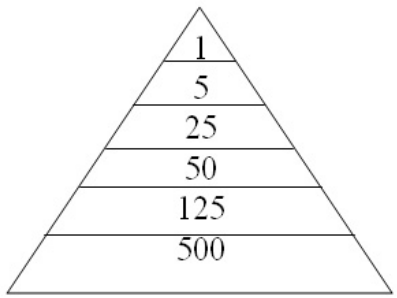
CSSE 371, Software Requirements and Specification
Steve Chenoweth, Rose-Hulman Institute
October 22, 2004

In the book – This is Ch 27 - 28

What's traceability all about?

- **Ch 27:** Tracing Requirements
  - Why is tracing important?
  - What tools can you use?
- **Ch 28:** Managing Change
  - How do you capture change requests?
  - How do you respond to these (individually & overall)?
  - How does this tie-in with Ch 27?

Cost to Fix Errors



Why we care – Remember Dr. Ardis's Sep 4 Process Intro Lecture?

The problem…

- How do you know, if you're at one of these later stages, that you have a requirements fault?

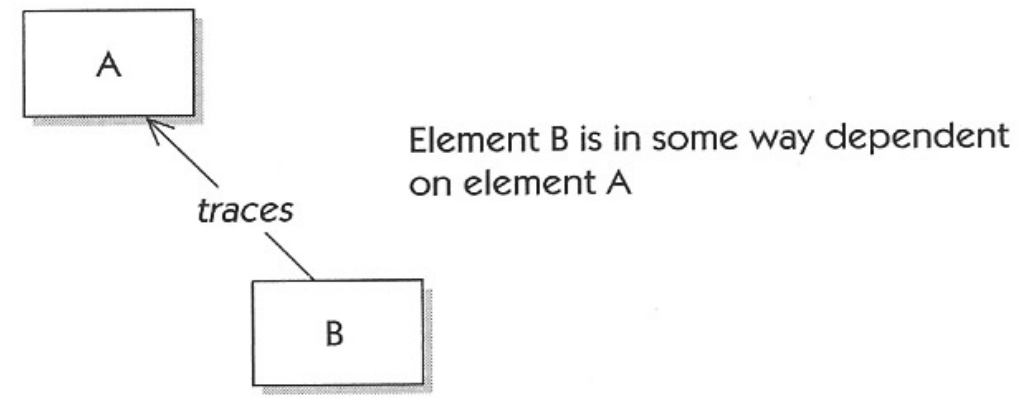


Figure 27–1 A trace dependency relationship

In general, how to trace…

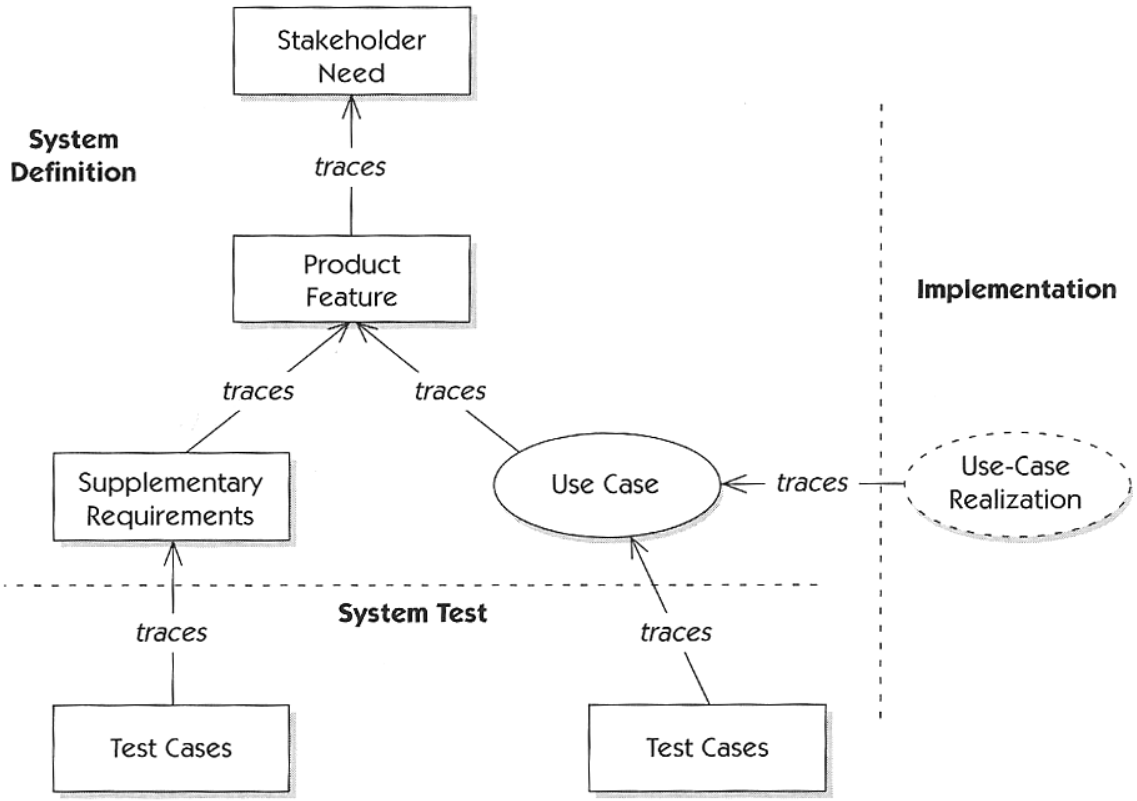


Figure 27–3 Generalized traceability hierarchy

With use cases, for instance…

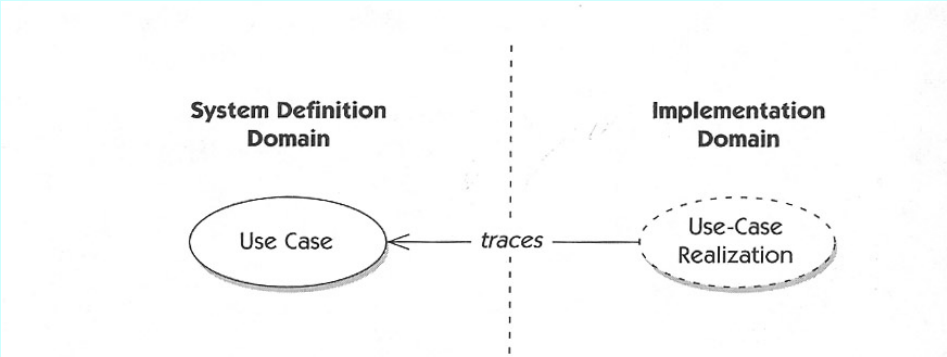


Figure 27–4 Tracing from use case to use-case realization

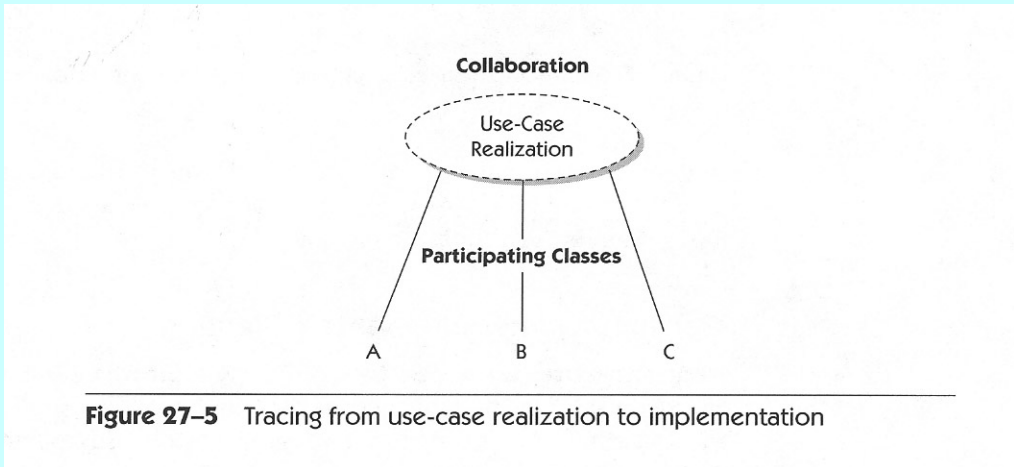


Figure 27–5 Tracing from use-case realization to implementation

Tracing problems from bad tests…

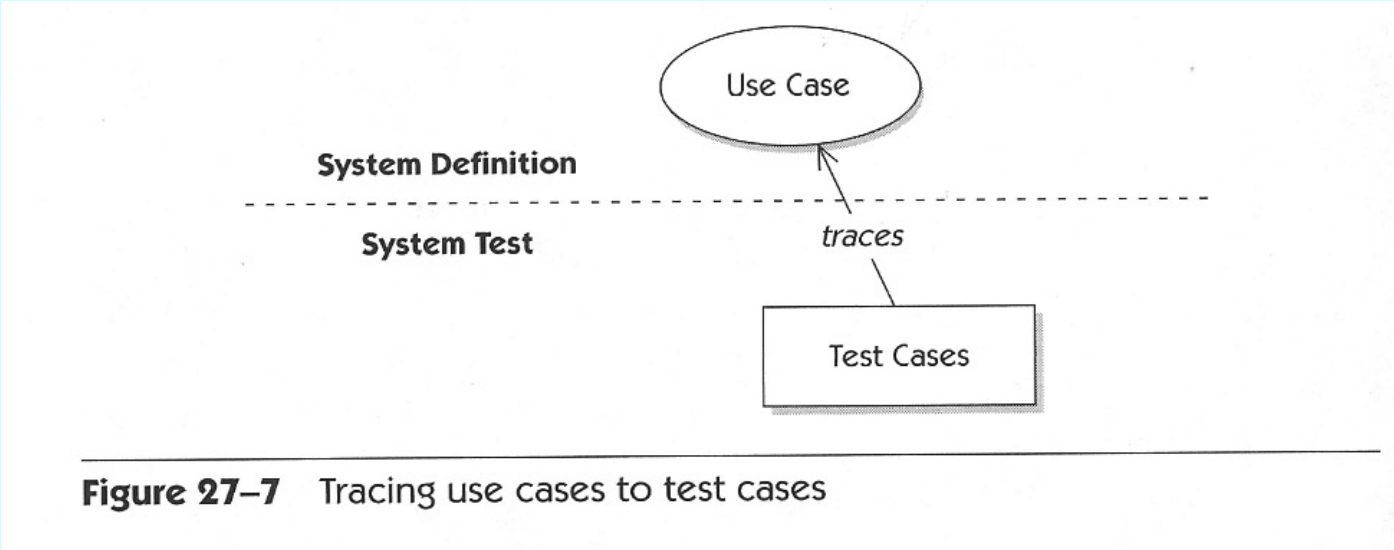


Figure 27–7 Tracing use cases to test cases

Today's team exercise – 3 minutes to brainstorm:
Help me solve yesterday's dilemma! Namely, …

When the designers begin to detail your use cases, they will need their own "version." But they will not just *add* things, they will *change and delete* them. So, how do you keep them in synch with the "external requirements" you've collected, while keeping track of who said what?

This is Ch 28: Managing Change

In general, how to manage change…

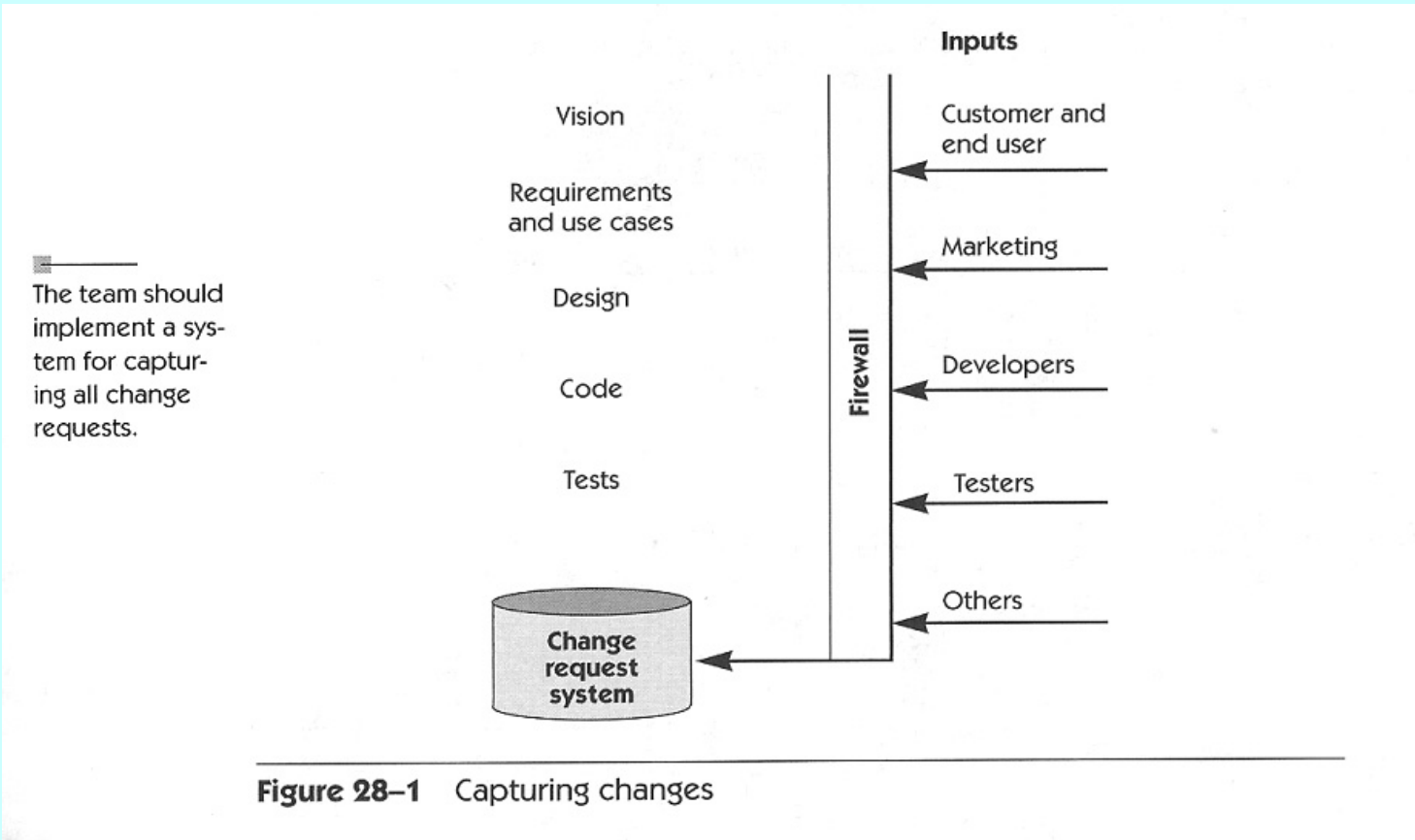


Figure 28–1 Capturing changes

It requires processes…

(These steps are out of Ch 28)

- "Step 2: Baseline the Requirements"
  - This means they are signed-off on, and
  - From then on, they fall under change control – see *below*

- "Step 3: Establish a Single Channel to Control Change"
  - No *ad hoc* additions
  - No *ad hoc* fixes, either

Picture: Baselining at the University of Minnesota. From http://www-users.cs.umn.edu/~dutta/usta/Baseline_brochure.htm.

which don't break under pressure…

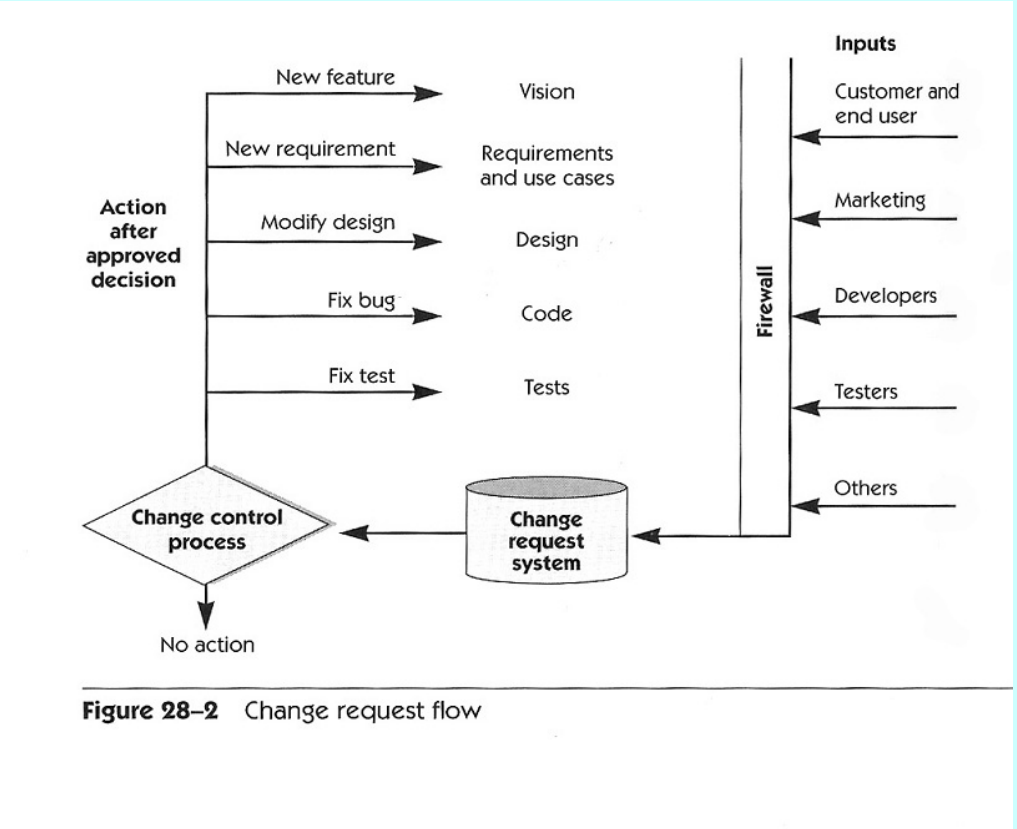
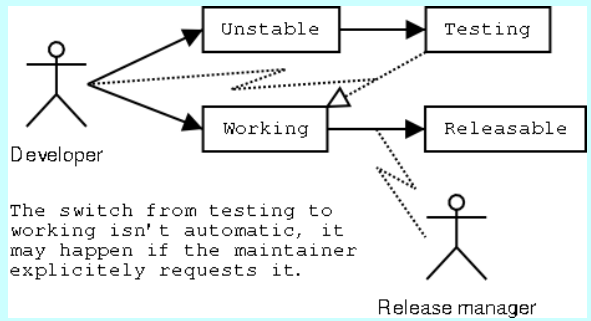


Figure 28–2 Change request flow

In ← this big picture, you especially need to know what "release management" ↓ is!



Picture at *right* courtesy of www.debian.org/vote/ 2002/platforms/raphael.