IMPROVING

Retrospectives

The 5 Whys

Process
Changes

Wednesday, May 16, 2012
RETROSPECTIVES

Agile Samurai, Chapter 10, Sections 5 and on

Agile principle

At regular intervals, the team reflects on how to become more effective and then tunes and adjusts its behavior accordingly.
Analyzing the failure is just step 1.

Equally critical is learning the right lesson and making the right changes.
**GENERIC FIXES**

**Problem**

- Went over schedule
- Software did not work

**Mitigation**

- Increase project slack time
- Schedule additional milestones
- Test more
- Schedule structured code review sessions

*These add cost but don’t directly address the underlying causes of the failure.*
Went over schedule

Multi-year timeline encouraged delay

Third-party encryption package was delayed

Software did not work

Platform issues at deployment sites

Schedule more intermediate releases

Schedule developer task to find alternatives to 3rd party s/w

Schedule early deployment technology spike

Hire expert consultant on deployment platform

Wednesday, May 16, 2012
"Success has many parents. Failure is an orphan."

But in reality failure never has just one source.

Eric Ries on the 5 Whys

video:  
http://ecorner.stanford.edu/authorMaterialInfo.html?mid=2296

color:  
When an administrator clicked on edit project, nothing seemed to happen

... because the server redirected the user back to the same page

... because admins didn't have editing privilege for projects

... because only project members had editing privilege

... because developers assumed only project members edited projects

We changed the interface to not show links the user doesn't have access to.

We added a "page generated on ..." to differentiate a redirected page

We added an error page for "action not allowed"

We fixed the edit project function to accept admins

We changed our process to include proposing new user stories for our client to consider, as appropriate.
This is a near-perfect illustration of how to use the 5 Whys analysis to generate multiple levels of defense against the original failure.

- The control that caused the problem won't appear any more unless the user has the required privilege.
- A result page won't look just like the page that led to it.
- Doing something without the required privilege will return an error page.
- Admin users will have the required privilege.
- Developers won't just assume what should happen.