EECS 394
SOFTWARE DEVELOPMENT

Chris Riesbeck

Improving
Improving

- Retrospectives
- The 5 Whys
- Process Changes

Monday, November 19, 2012
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.
<table>
<thead>
<tr>
<th>Problem</th>
<th>Mitigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Went over schedule</td>
<td></td>
</tr>
<tr>
<td>Software did not work</td>
<td></td>
</tr>
<tr>
<td>Problem</td>
<td>Mitigation</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>Went over schedule</td>
<td>Increase project slack time</td>
</tr>
<tr>
<td></td>
<td>Schedule additional milestones</td>
</tr>
<tr>
<td>Software did not work</td>
<td>Test more</td>
</tr>
<tr>
<td></td>
<td>Schedule structured code review sessions</td>
</tr>
<tr>
<td>Problem</td>
<td>Mitigation</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>Went over schedule</td>
<td>Increase project slack time</td>
</tr>
<tr>
<td></td>
<td>Schedule additional milestones</td>
</tr>
<tr>
<td>Software did not work</td>
<td>Test more</td>
</tr>
<tr>
<td></td>
<td>Schedule structured code review sessions</td>
</tr>
</tbody>
</table>

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

Software did not work
**Problem**

- Went over schedule

**Cause**

- multi-year timeline encouraged delay
- third-party encryption package was delayed

**Problem**

- Software did not work

**Cause**

- platform issues at deployment sites
CAUSE-BASED CHANGE

Problem

Went over schedule

Cause

multi-year timeline encouraged delay

third-party encryption package was delayed

Problem

Software did not work

Cause

platform issues at deployment sites

Mitigation

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

Monday, November 19, 2012
"Success has many parents. Failure is an orphan."
"Success has many parents. Failure is an orphan."

But in reality failure never has just one source.
"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

blog:  
5 WHYs EXAMPLE

Analysis

Repair

Monday, November 19, 2012
When an administrator clicked on edit project, nothing seemed to happen.
When an administrator clicked on edit project, nothing seemed to happen

... because the server redirected the user back to the same page
5 WHYS EXAMPLE

Analysis

• 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

Repair

Aaron Steinfeld EECS 394 Spring 2011

Monday, November 19, 2012
5 WHYS EXAMPLE

Analysis

- 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

Repair
5 Whys Example

Analysis

- 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

Repair

Aaron Steinfeld EECS 394 Spring 2011
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.
**5 WHYS EXAMPLE**

**Analysis**

- 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

**Repair**

- 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

Aaron Steinfeld EECS 394 Spring 2011
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".

Analysis

Repair
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

Aaron Steinfeld EECS 394 Spring 2011
**5 Whys Example**

**Analysis**
- 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

**Repair**
- 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.

Aaron Steinfeld EECS 394 Spring 2011

Monday, November 19, 2012
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.
IT'S ALSO REALLY HARD TO DO
PITFALLS

- Multiple 1-step causals instead of a 5-step causal chain
- Causals involving 3rd party actions ("because client didn't send reminder")
- Shallow explanations ("we forgot" "we didn't think it was important")
Focus on what you can change

- The goal is improvement, not punishment
- Given a failure, what kinds of changes in process are relevant? In order of preference:
  - **Detour**: take an equally cheap but alternate path that avoids the cause
  - **Detect**: set up something to detect if it's about to happen again
  - **Defend**: set up barriers to prevent or reduce negative effects if it happens again
  - **Duplicate**: set up redundant processes to keep going if it happens again
- Find causals that tie directly to one of the above