Profiles in Failure
Learning Lessons from Others’ Pain
Introduction

“Perhaps the history of the errors of mankind, all things considered, is more valuable and interesting than that of their discoveries.” Ben Franklin

I learn a lot from projects that failed
I’ll study three different projects here
Common themes in what could have saved them
- Earlier testing
- Better quality throughout the project
- Tighter collaboration between testers and other stakeholders
- Better vendor quality management

Let’s see how….
Failed Project A

- A banking project to customize a browser-based COTS home loan package
- Existing application in Java
- Plan for six weeks of acceptance testing
- After twelve weeks of acceptance testing, project was stopped
- Testing service provider was fired
- RBCS engaged to identify the problems
Case Study Home Equity Loan Application Bugs
Product Quality Problems Analysis

Date Opened/Closed

www.rbcs-us.com
Profiles in Failure
Copyright (c) RBCS 2013
Case Study Home Equity Loan Application Bugs
Daily and Rolling Closure Period

Test release and bug confirmation test processes introduce average one-day delay
Case Study Home Equity Loan Application Bugs
Subsystem Breakdown

Subsystem Most Affected by Bug
www.rbcs-us.com
Case Study Home Equity Loan Application Bugs
Opened Count Breakdown

Number of Bugs Opened

Opened Count

www.rbcs-us.com
Copyright (c) RBCS 2013
Project A Conclusions

Failure could have been avoided through:

- Better quality throughout the project: needed better bug management, more realistic bug fix process
- Tighter collaboration between testers and other stakeholders: needed defined requirements, rather than letting testers make requirements up
- Better vendor quality management: interaction between bank, testing service provider, and COTS software provide was dysfunctional; bank merger exacerbated situation

Result: wasted millions of dollars, about a dozen person-years of effort, and considerable ill-will generated
Failed Project B

- Insurance project to merge a group of legacy insurance product lines to a single, modern, maintainable system
- Huge project, planned for five years, organized as a sequence of major phases
- Things broke down during phase 3B, about three years into the project
- RBCS engaged at the end of phase 3B to assess and give recommendations on improving testing and quality for phases 4 and beyond
Main Findings of the Assessment

- ST and UAT were organized, effective, and efficient
  - DDE 95-98%
  - Design spec and use case coverage 100% (see below)
  - ROI 1300% (effort saved)

- SDLC described as “Agile” but was not (esp. as regards unit testing)
- When trouble started, SDLC became purely reactive
- When test team tried to improve broken unit test and design processes, they were resisted, characterized as counter-productive
- Code reviews were desultory and ineffective
- Test team forced to accept an unrealistic test execution period, even though it had estimated accurately
- Bug reports used as a retroactive specification mechanism due to poorly documented use cases and designs
- There was another, larger problem with bugs, though…
An Ominous Trend

- A typical number of bugs were found, based on project size
  - However, four subsystems (out of 100) contained 50% of the bugs (next slide)
  - As Phase 3B proceeded, bug repair slowed (slide after next)
- Intractable design bugs were revealed
  - Performance
  - Data quality
  - Code maintainability
- These indicated fatal flaws
- Especially critical was bad database design done by incompetent staff at the beginning of the project
Longer and Longer

Closure Period (in Calendar Days)
Red line = rolling avg; dots = daily avg
Bug Clusters from Hell

Bug Category

*About 100 categories with less than 70 each*
**Project B Conclusions**

- Phase 4 never happened — the project was cancelled
- Failure could have been avoided through:
  - Earlier testing: thorough unit testing and better testing in earlier phases would have revealed the bugs
  - Better quality throughout the project: reviews of the database design and realistic early performance testing would also have found and removed the most fatal bugs
  - Tighter collaboration between testers and other stakeholders: awarding losing development bidder the testing work created disincentive to collaborate for quality
  - Better vendor quality management: the project managers could have insisted on better SDLC processes, especially quality-related one
- Result: waste of 150 person-years of effort and millions of dollars, and lost opportunities
Failed Project C

- Entertainment company creating a new system-of-systems including a Unix-based, international network of IVR servers and sophisticated call centers
- Large project, planned for 18 months, following a sequential lifecycle, developing and testing seven major systems and the system-of-systems
- Subcontractor and vendor development, responsible for code reviews, unit test, unit integration test, and component test
- RBCS was involved as the independent test team, responsible for system pre-integration test, system test, and system integration test
## System Pre-Integration Test Plan

<table>
<thead>
<tr>
<th>System</th>
<th>BB0</th>
<th>BB1</th>
<th>BB2</th>
<th>BB3</th>
<th>BB4</th>
<th>BB5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. IVR HW</td>
<td>Alpha</td>
<td>Beta</td>
<td>Quad</td>
<td></td>
<td></td>
<td>Prod</td>
</tr>
<tr>
<td>B. Network</td>
<td>LAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WAN</td>
</tr>
<tr>
<td>C. IVR App</td>
<td>C1-E1</td>
<td>C2-D1</td>
<td>C3-D2</td>
<td>C4-D3</td>
<td>C7-F2</td>
<td>C9-D8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C3-G3</td>
<td>C5-F1</td>
<td>C8-D6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C6-E4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D. Cust Svc App</td>
<td>D1-C2</td>
<td>D2-C3</td>
<td>D3-C4</td>
<td>D6-C8</td>
<td>D8-C9</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D4-G4</td>
<td>D7-F3</td>
<td>D9-F5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D5-E5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E. Call Center</td>
<td>E1-C1</td>
<td>E2-G2</td>
<td>E3-G4</td>
<td>E4-C6</td>
<td>F2-C7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>E1-G1</td>
<td></td>
<td></td>
<td>E5-D5</td>
<td>F3-D7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F4-G5</td>
<td></td>
</tr>
<tr>
<td>F. Data Warehouse</td>
<td>G1-E1</td>
<td>G2-E2</td>
<td>G3-C3</td>
<td>F1-C5</td>
<td>F5-D9</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>G4-E3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G. Content Mgmt</td>
<td></td>
<td></td>
<td></td>
<td>G4-D4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G5-F4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What Ultimately Happened

Profiles in Failure

Pre-integration test starts, but few touchpoints actually tested

System test starts, but most systems incomplete

System integration test starts, two months late, with systems still incomplete

www.rbcs-us.com
Copyright (c) RBCS 2013
Who Did the Most Damage

NOP Bugs
Subsystem Affected Breakdown

Bugs by Subsystem

IVR App 764
CSA App 476
CM App 48%
IVR Telephony 123
IVR Platform 101
GTC App 68
RDT 63
900 App 4
Web Platform 3
Web App 0
WAN 0
PSTN 0
LAN 0

Profiles in Failure

www.rbcs-us.com
Copyright (c) RBCS 2013
Project C Conclusions

Failure could have been avoided through

- Earlier testing: proper vendor testing could have revealed the bugs; actually doing the planned pre-integration testing would have, too, and it was the last chance
- Better quality throughout the project: design and code reviews, programming best practices would have prevented many of bugs to begin with
- Tighter collaboration between testers and other stakeholders: the individual system teams were too siloed, and the last possible collaboration forcing mechanism (pre-integration testing) was mostly skipped
- Better vendor quality management: most subcontractors, especially the rogue IVR software vendor, ran roughshod over the project

Result: waste of 100 person-years and millions of dollars, near-collapse of the business
Conclusions

- Proper testing and quality assurance (including SDLC) are central to project success
  - Earlier testing
  - Better quality throughout the project
  - Tighter collaboration between testers and other stakeholders
  - Better vendor quality management
- We’ve seen three cases where application of these known best-practices would have prevented failures
- In addition to costs, there are career implications and individual fulfillment issues associated with failed projects
- Be smart: learn from other people’s failures!
To Contact RBCS

For almost 20 years, RBCS has delivered consulting, outsourcing and training services to clients, helping them with software and hardware testing. Employing the industry’s most experienced and recognized consultants, RBCS advises its clients, trains their employees, conducts product testing, builds and improves testing groups, and hires testing staff for hundreds of clients worldwide. Ranging from Fortune 20 companies to start-ups, RBCS clients save time and money through improved product development, decreased tech support calls, improved corporate reputation and more. To learn more about RBCS, visit www.rbcs-us.com.

Address: RBCS, Inc.
31520 Beck Road
Bulverde, TX 78163-3911
USA

Phone: +1 (830) 438-4830
Fax: +1 (830) 438-4831
E-mail: info@rbcs-us.com
Web: www.rbcs-us.com