

ERP Failure #7: How Does A Well-Intentioned Project Go Off the Rails?

Transcript © 2015-2017 by Warren S. Reid All Rights Reserved

Hi, I'm Warren Reid, and today, we're going to explore how otherwise successful, well-run, and well-intentioned Fortune 500 companies and Tier 1 integrators end up in software failure litigation.

In this actual case, a large State Child Support & Enforcement System mandated by Feds needed to be implemented, as the State was fined approximately \$15mm per year it didn't have the system. It was already several years late. The State engaged a huge Tier 1 software developer/systems integrator to build and maintain the new system.

Project was already 3 years past the original 2 year estimate when it was terminated. It was stuck in systems testing – never ready for implementation. Here's how it went down:

1. Good Faith: Customer negotiated in good faith, and expected the Integrator to do the same. The purpose of the Contract's was to flush out & allocate risks before the project began & assign responsibility for managing and mitigating those risks. But the Integrator made several material false promises to secure job -- which the Customer reasonably relied on to award the job.

2. False Estimate: Specifically, the Integrator's estimates were NOT based on relevant historical metrics, comparable systems development projects/efforts, or industry standard parametric models. Integrator hid calendar concerns, hoping to save time by moving to agile methods– a false assumption.

3. Poor Staffing: The Integrator's promised "A Team" never showed. Replacement staff was unqualified and couldn't perform the Integrator's "best practices". The Integrator's best staff arranged reassignment off this "dead project". Integrator threw incomplete, poor quality work over fence for the Customer to finish/approve, which the latter did not have the skills to perform, and was the reason why the Customer hired the Integrator in the first place.

4. "Rosier than actual" Project Status: Reports lacked actionable metrics to keep the project on track. Overestimated earned value and unmanaged critical path meant that the risk of not meeting Go-Live date with a suitable, high-quality system was extremely high. The project schedule was extended 4 times; cost increased more than doubled. The Integrator kept repeating the same mistakes.

5. Unwise SDLC shortcuts: The shortcuts caused big foreseeable problems later (e.g., JAD sessions without prototyping; requirements were contradictory/misunderstood; a bad design followed the bad requirements; the RTM was abandoned; the code quality was not reviewed).

6. Testing Shortcuts: Testing was cut short at every stage level – but code was still promoted. The absolute number of test data/scenarios was insufficient. Defect symptoms were addressed – not root causes. An Automated Regression Testing Suite not created or used appropriately. GUI was inconsistent throughout the system.

7. Poor Testing Plan/Execution: Testing was poorly planned and executed. Certified test pros were not used. The estimated 5 month Systems Test expanded to 18 months. Each fix cycle

uncovered more errors! The system test pattern of errors proved they were injected during earlier project phases. The cost to fix defects rose exponentially. Status reports hid the gravity of the error pattern and that the project was in crisis. When approached, the Integrator was either unwilling or unable to fix the weaknesses.

8. No Estimate To Complete: Neither party knew how much work, re-work and effort remained to deliver the contracted-for system; the Integrator could not reliably estimate a Go-Late date. There was no mutual plan to re-source, re-work, re-test, train & proceed. Finger pointing was rampant.

9. Analysis & Opinions: I analyzed (and testified), among other opinions, that:

- a. Continuing with current project management, project staff, “seat of the pants” SDLC, error patterns and lack of useful QA/IV&V, the project would take more than 10 yrs to deliver -- i.e., it could not be accomplished.
- b. My overview of case artifacts led me to opine that there was little/no re-use value. No Integrator would want to be responsible to fix, complete, and maintain this low-quality system that the original Integrator could not itself finish testing.

Termination & lawsuits resulted.

Everyone Loses! It's the same old thing again.