

The Fixed Price contract From Hell

by Mary Poppendieck

My story is about the fixed price contract from hell. It was a state government contract, negotiated by a local entrepreneur who was trying to start up a software development company in Malaysia to do a lot of government contract work. He partnered with a company in Indiana which had written similar software for a similar agency in other states. The idea was, two systems developed by the company in Indiana, one that worked in Indiana and one that worked in Montana, would be converted by the developers in Malaysia to satisfy Minnesota requirements.

The contract was fixed price, and the price was based on how much money the state had in the budget, not on how much the system would cost. The deadline was nonnegotiable based on a legislative mandate. If it were a simple matter of converting software which worked in other states to run in Minnesota, the job might have been possible. But Minnesota had been promised that the two systems would be integrated with a financial package that didn't exist yet – it was being developed for Texas.

After a while the integration problem looked so difficult that everyone decided to abandon the original plan of using the Indiana and Montana system and instead, use the system being developed for Texas, because that was going to be integrated with a single financial package. This meant starting over half way through a very tight schedule. But to make matters worse, the Texas system had an imaging system which was deemed to be too expensive, so a new and different imaging system – the heart of the workflow – would be developed to work with the Texas system.

Okay, so you can see we have an impossible situation here, going from bad to worse as time goes on. Let's examine some of the issues here:

1. The vendor, desperate for business and very naive, bid fixed price, fixed schedule with little idea what they were really promising.
2. The customer had already spent millions of dollars and five years trying to replace their system. They were well aware that the promised deliverable would most likely cost far more and take far more time than the contract called for. But they signed the contract hoping that the obligation would force the vendor to dig deep into its own pockets to deliver the software.
3. For a while, the vendor, desperate for follow-on business, did just that. But the customer got sloppy, unrefactored software, and only half of the system was ever delivered. Eventually, the vendor simply went out of business.
4. The software that was delivered was done using a Scrum technique with weekly (rather than monthly) deliveries that went straight into production. This approach violated all of the state guidelines – the contract required a classic waterfall process – but it worked.
5. The governing contract was, in my opinion, responsible for a lot of the problems.

In retrospect: The state was playing a very risky game by insisting on a waterfall process which allowed projects to burn money for months and years without showing any results. The state should also be blamed for negotiating a contract which it knew from the start was in all probability a win-loose contract (state wins, vendor loses). In the end, it turned into a loose-loose contract. The naive vendor evaporated, leaving the state holding the bag. Neither the waterfall process nor an air-tight contract protected the state.

I believe it is incumbent upon those negotiating from a position of strength to realize that they are not protecting their own best interests when they select the cheapest and least experienced vendor and negotiate contracts which takes advantage of these vendors. In addition, when you add a sequential process on top of a poor contracting practice, you get the potential for enormous losses that go undetected until the end of a contract.