

Software Procurement in the Public Sector

Ken Stevens, Ph.D.

I recently had the opportunity to participate as a member of a panel at a ITAC Health workshop. The goal of the workshop was to bring public sector buyers and sellers together to fix a broken procurement system. I participated as a member of the vendor panel, representing mid-sized Canadian software companies. The buyer side was well represented, including an Auditor General of Ontario and the Assistant Deputy Minister for Ontario Shared Services. The following paper outlines the key points from my talk as well as some suggestions other vendors made for improving the public sector procurement process.

Building a Software System is not like Building a Subway System

As a Software Engineer, the main barrier I see to successful software delivery in the public sector is a misunderstanding about the nature of software. Software projects in the public sector tend to be managed like major construction projects: like building a new subway system. How do you manage risk when building a new subway system? You do years of up-front planning, specifying all the details of the entire project long before the first shovel breaks earth. Why do you do this? Because re-routing a subway tunnel is very, very expensive. You have to get it right the first time.

The mistake we make in the public sector, is that we treat software systems the same way. Software is different. Unlike a major construction project, it is in fact relatively inexpensive to alter a software system after it has been built. Building a software system is more like building a successful political campaign platform. Successful political campaigns commit very little up front: they throw out teasers and then poll intensively to suss out the public mood – which parts of the new platform does the electorate hate, which parts get the public excited – then based on this feedback, the direction of the campaign is altered. It is not a subway tunnel, planned years in advance. It is built incrementally, guided by constant feedback from the electorate.

To give a specific example, consider Ontario's Medication Management Request for Proposal (RFP). Ontario started writing this RFP six years ago. The RFP was finally issued three years ago. Today it is still not awarded. We've been planning this project for six years now, and all we have to show for it is a stack of paper. If a middle-aged person arrives at an emergency room today unconscious, we have no way of knowing what medications they are on. Why are we at this point? It's because the scope of the Ontario Medication Management system is huge. It's being procured like a new subway system: with massive scope and comprehensive specifications. The scope includes real-time transactional integrations into pharmacy systems, real-time e-prescribing integration into physician systems, real-time patient lookup into patient registries, etc... These are ambitious plans.

Imagine instead, if six years ago we decided to build the software incrementally. We started with a nightly batch upload of all prescriptions from all pharmacies to a central database and then gave EMRs access to this database. Pharmacies already batch upload their prescriptions to other partners, so adding a provincial drug database to the feed would be a simple project for them. Had Ontario started with this scope, I believe we would have had a comprehensive province-wide prescription

database within a year. Sure the prescription data is a day old, but day old data is a heck of a lot better than no data. More significant than that, having a real system out there, in the hands of users, allows you to start polling your users. Which parts of the new platform do the users hate? Which parts get the users excited? Based on this feedback, you can steer the evolution of the system, often in ways you could never have anticipated at the start.

Six years is an eternity in the world of technology. Six years ago, there were no iPhones. Three years ago, when the Medication Management RFP hit the streets, there were no iPads. Subway trains don't change that much. But software changes a lot. When you're planning a software project, particularly something as important as a public health system, if you plan too big for too long, your system will be obsolete by the time it's launched. Start small, get feedback, and incrementally improve.

Fix the Q&A Process

I also wanted to touch on an aspect of that process that I think is particularly broken, and that is the RFP Q&A process. The intent of the RFP Q&A rules are to level the playing field – to ensure all vendors have a fair chance at winning. The effect of the RFP Q&A rules is exactly the opposite. Vendors on the inside already understand what the customer needs and those on the outside have no way of finding out.

The rules require that any question a vendor has about an RFP must be submitted on paper on the public record, with the answers to the questions being shared with all bidders. While this process looks good on paper, this is what actually happens:

- Meaningful clarification questions are rarely asked for fear of giving away your requirements analysis advantage to your competitors.
- On the rare occasion when meaningful clarification questions are asked, nine times out of ten the answers don't help.

Software requirements cannot be understood through a public Q&A process. Delivering software is fundamentally different from delivering milk. What's required is a conversation, with only the vendor and the customer in the room, where the vendor can ask insightful questions and the customer can answer honestly. It is very, very hard to successfully bid on a project when you don't understand what the customer actually needs and a public Q&A process has proven that it can't get us there.

To make this work, there would need to be some sort of short-list qualifying process so there are a manageable number of meetings. But often times there are only two or three bidders. Even a confidential Q&A would be fairer than what we currently have.

Contract Templates

We heard a lot at the workshop about how attaching a unique contract in each RFP delays the bidding process, and how many clauses in these contracts are showstoppers for many vendors, especially clauses regarding unlimited liability, unlimited indemnification and IP ownership.

Why not publish, say, five contract templates that are to be used for all public sector procurements, and then have the RFP simply refer to the contract template by name. This would allow vendors to pre-qualify which of those templates they can bid on and which they can't. Such pre-qualification could even be announced by vendors if they so chose so that when selecting a contract template, governments would know in advance which vendors they are excluding.

RFP Star Rating

Those of us who regularly read RFPs know that the quality can vary greatly. Vendors prefer RFPs that describe the business problem and leave the implementation details to the vendor. Vendors prefer RFPs with clear scope and deliverables.

One of the things Marion Macdonald (ADM Ontario Shared Services) mentioned at the meeting is that Ontario plans to change the system it uses to manage the RFP process. I would suggest to Marion that she consider allowing vendors to anonymously assign a star rating to posted RFPs. I imagine a future where budding RFP authors download all the five star RFPs as guides for how to write a really good RFP.

References vs. Innovation

My co-panelist Michael Martineau raised an important point, which is that we make a lot of noise about fostering innovation, but then when it comes time to RFP, the RFP inevitably requires "three sites in Canada where the software has been installed for over a year" or some such reference requirement. You're not going to introduce innovative solutions into the marketplace, particularly from other countries, so long as RFPs contain language like that.

About the Author

Ken Stevens – Director, Solution Delivery

Project Manager and Technical Analyst with over 15 years experience defining, developing, integrating and managing complex business applications and building roadmaps including focused technology assessments and large strategic direction and planning initiatives. Ken holds a Joint Honours in Mathematics and Philosophy from the University of Waterloo and a Ph.D. in Mathematics from the University of Toronto. For the last five years, Ken has led e-Health development and integration initiatives collaborating with government ministries, health agencies, healthcare providers and development teams.

About Intelliware Development Inc.

Intelliware is a custom software, mobile solutions and product development company headquartered in Toronto, Canada. Intelliware is a leader in Agile software development practices which ensure the delivery of timely, high quality solutions for clients. Intelliware is engaged as a technical partner by a wide range of national and global organizations in sectors that span Financial Services, Healthcare, ICT, Retail, Manufacturing, and Government.

Intelliware placed among the Top 5 Mobile Technologies Companies in the 2012 Branham300 report, the definitive listing of Canada's Information and Communication Technology (ICT) industry leaders, as ranked by revenues.

1709 Bloor Street West
Suite 200
Toronto, Ontario
M6P 4E5, Canada

200 Adelaide Street West
Suite 100
Toronto, Ontario
M5H 1W7, Canada

416.762.0032

416.916.3457

www.intelliware.com