QA Consultants

The HealthCare.gov fiasco was avoidable

The HealthCare.gov fiasco was avoidable

The dust has not completely settled but what we now know is not a pretty IT project outcome. HealthCare.gov is a high profile $400M web portal that did not launch on time, does not work well, and has produced significant political and vendor embarrassment. How did this disaster happen and what can all public and private organizations do to avoid these problems?
It has been publicly acknowledged that a big part of the HealthCare.gov disaster trace to inadequate software testing, much of which traces to upstream problems in the development process. Poor outcomes are not uncommon with large IT projects. Fortunately, they are not as publicly embarrassing as this one.

Based on congressional testimony by the four vendors, we know that the failure had many dimensions:

Insufficient Testing

This complex portal was tested for only two weeks, clearly an insufficient amount of time given the problems and challenges such as delays in gathering requirements, working with multiple government departments, and databases, a large number of users, many stakeholders, heterogeneous infrastructures and politically-inspired deadlines. The number and severity of these issues virtually guaranteed there would be quality and delivery problems with the code, necessitating much more than two weeks of testing.

The Fix:
Testing must be elevated to a strategic activity within the project plan. Sufficient amounts of fixed time as well as qualified resources must be allotted to conducting proper testing activities throughout the project life cycle. For a portal like HealthCare.gov, expect to perform at least three months of functional, scalability, security and performance testing once the code is released. When only two weeks of testing is possible, CIOs and their business sponsors need to carefully consider the risks of releasing poor quality code.

Limited end-to-end view of quality assurance

Building HealthCare.gov utilized at least four different vendors (and now Google, Oracle and Red Hat). They were building an extremely complicated system with a wide variety of internal and external stakeholders, data formats, and IT infrastructures. Yet, no one had a single, end-to-end view and accountability for the quality assurance process. This limits the amount of integration and 'end-to-end' testing activities that can occur, increasing the probability of system failure and the inevitable blame game that follows. Furthermore, overall management responsibility of HealthCare.gov was given to the Medicaid/Medicare agency, which lacked the technical expertise and resources to independently monitor the overall project, and credibly advise the elected officials and address problems when they arise.

The Fix:
Our Managed Services practice addresses this challenge by looking at testing as part of an entire quality assurance continuum. This approach considers process, talent and cultural issues. For HealthCare.gov, we would have emphasized three best practice strategies: 1) follow defined multi-vendor system integration processes that encourage regular collaboration; 2) conduct realistic scalability, performance and user experience planning to meet expected and possible peak demands and; 3) add independent validation and verification (IV&V) to provide overall governance, standards compliance and end user advocacy.
Every large public organization would benefit from having a centralized IT think tank that can provide expert, timely and independent IV&V on large, risky and high profile projects. This group should be arms length from vendors. They would be accountable to the highest elected officials and senior bureaucrats, ensuring that there is proper vendor oversight, sufficient funding and realistic timelines.

Bad Practices

Large organizations, but especially the government, have a tough time delivering large IT projects on-time, on-budget and on-spec. Computer World, a leading IT magazine, recently published a study that analyzed the last 10 years of US government IT projects. The report found that 96 percent of them were deemed a failure. While some of the blame must fall on the vendors (see above), other problems are more systematic such as how the public sector writes its requirements, defines a development strategy & launch, and ensures proper governance. Adding more cooks to the HealthCare.gov kitchen (i.e.

Google, Oracle, and Red Hat) will likely not help. It will add more system integration issues, greater process complexity and more finger pointing.

The HealthCare.gov project features many examples of bad practices that ended up compromising development and testing: the development cycle of a very complicated portal was compressed to “several” months despite plenty of lead time; government contractors reportedly did not receive their project specifications until a few months before launch, guaranteeing poor code quality. These specifications were not detailed enough and ended up changing many times. Finally, the launch date was driven more by political considerations than what was technically realistic.

The Fix:
Project leaders need to go back to the basics like insisting on proper requirements with extensive stakeholder input, a realistic release schedule with proper milestones, and the use of approved service level agreements not to mention having sufficient time and talent available for testing. Utilizing an external IV&V group would also ensure someone monitors and governs the end-to-end development process and reports directly to the Government, keeping all the vendors honest.

For further information please contact:

Harry France
Sr. VP Sales
(519) 575-0314
hfrance@qaconsultants.ca