New Trends in Preventing System Modernization Project Failure

For Agency Leaders

Mark S.R. Williams
Fast Enterprises, LLC
Trend #1: Major software projects usually fail...

- Only 29 percent of software projects succeed. *This percentage has been constant for a number of years.*

- Software projects larger than $15 million deliver on average 56 percent less functionality than the project was designed to deliver.

- 31.1 percent of software projects are canceled for failure to achieve project management targets.

- For every 100 software projects that start, there are 94 project restarts.

- 70 percent of organizations have gone through at least one major software project failure.

There have been successes, but motor vehicle system modernization projects are still very high risk....

- “Half-finished $208 million DMV technology overhaul cancelled because little progress was being made...” 2013

- “DMV terminates contract with developer after $17 million in expenses...” 2015

- “DMV computer project cost more than $46 million so far, remains unfinished after almost a decade of work...” 2016

- “DMV sues developer over notorious DMV computer project...” 2016

- “DMV pulls the plug on its computer project after spending $5 million and firing the contractor...” 2012

Translating the trend...

The single greatest responsibility for agency leaders doing systems modernization projects is to reduce the risk of project failure.
Trend #2: Increasing project risk by seeking to reduce it...

- “The most common response to a history of failed projects is to increase project oversight, with a particular emphasis on reporting. However, more is often mistaken for better when it comes to governance. The key to changing project outcomes is to focus on effective governance, not increased reporting.”

- “Successful projects are characterized by less bureaucracy in governance and by greater focus on outcomes.”

- “Project management should simplify everything about the project.”

Translating the trend...

A project’s intellectual capital, which is a limited resource, needs to be deployed in optimum balance between development and oversight to reduce project risk. There is a cost to over-investment in oversight.
Trend #3: Agile is replacing Waterfall, but which Agile?...
Waterfall vs Agile in terms of project risk...

<table>
<thead>
<tr>
<th>SIZE</th>
<th>METHOD</th>
<th>SUCCESSFUL</th>
<th>CHALLENGED</th>
<th>FAILED</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Size Projects</td>
<td>Agile</td>
<td>39%</td>
<td>52%</td>
<td>9%</td>
</tr>
<tr>
<td></td>
<td>Waterfall</td>
<td>11%</td>
<td>60%</td>
<td>29%</td>
</tr>
<tr>
<td>Large Size</td>
<td>Agile</td>
<td>18%</td>
<td>59%</td>
<td>23%</td>
</tr>
<tr>
<td>Projects</td>
<td>Waterfall</td>
<td>3%</td>
<td>55%</td>
<td>42%</td>
</tr>
<tr>
<td>Medium Size</td>
<td>Agile</td>
<td>27%</td>
<td>62%</td>
<td>11%</td>
</tr>
<tr>
<td>Projects</td>
<td>Waterfall</td>
<td>7%</td>
<td>68%</td>
<td>25%</td>
</tr>
<tr>
<td>Small Size</td>
<td>Agile</td>
<td>58%</td>
<td>38%</td>
<td>4%</td>
</tr>
<tr>
<td>Projects</td>
<td>Waterfall</td>
<td>44%</td>
<td>45%</td>
<td>11%</td>
</tr>
</tbody>
</table>

Source: 2015 Standish Chaos Report, 10,000 projects in baseline
The twelve simple principles of Agile...

<table>
<thead>
<tr>
<th>Principle</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.</td>
<td>Working software is the primary measure of progress.</td>
</tr>
<tr>
<td>Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.</td>
<td>Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.</td>
</tr>
<tr>
<td>Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.</td>
<td>Continuous attention to technical excellence and good design enhances agility.</td>
</tr>
<tr>
<td>Business people and developers must work together daily throughout the project.</td>
<td>Simplicity--the art of maximizing the amount of work not done--is essential.</td>
</tr>
<tr>
<td>Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.</td>
<td>The best architectures, requirements, and designs emerge from self-organizing teams.</td>
</tr>
<tr>
<td>The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.</td>
<td>At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.</td>
</tr>
</tbody>
</table>
Have morphed into this...

Agile Tribes – Subway Map

Safe Drivers · Safe Vehicles · Secure Identities · Saving Lives
From one of the 17 founders of Agile...

- “The word ‘agile’ has been subverted to the point where it is effectively meaningless”;

- “The word ‘agile’ became a magnet for anyone with points to espouse, hours to bill, or products to sell. It became a marketing term”;

- “Once the word ‘agile’ became meaningless, developers could no longer use it as a guide to what is useful in their practice”;

- “Instead, let’s substitute ‘agility’ for ‘agile.’ You aren’t an agile programmer, you program with agility; you aren’t on an agile team, your team exhibits agility; you don’t use agile tools, you use tools than enhance your agility.”

SOURCE: Dave Thomas, “Agile is Dead,” March 4, 2014
Translating the trend...

Flexibility and simplicity are what Agile methodology is about, not a branded set of tools. Using the methodology for its “agility” will increase the likelihood that your modernization project will be successful.
Trend #4: COTS and Agile can be disruptive to public sector systems development processes and procedures...

- State IT project oversight and IV&V may have been developed for projects in the Waterfall era—i.e., phase gate approvals that are sequential and not ideal for evaluating iterative projects.

- It now can be faster and less expensive to rebuild a piece of software than to develop a folder that documents what needs to change.

- COTS systems reduce or eliminate the need to process map the agency’s current state and expected future state in project pre-planning.

- If project deliverables are demonstrations of ongoing system capabilities, the need for review binders is reduced or eliminated.

- Agency SOPs may be hard-wired into screen flows and documented in context-sensitive help screens, reducing or eliminating the need for stand alone procedures manuals.
Translating the trend...

Agencies may need to evolve processes to exploit new opportunities in software development that have proven to reduce risk.
For More Information

Contact:
Mark S.R. Williams
Fast Enterprises, LLC

mrwilliams@fastenterprises.com
303-905-9126