Software Procurement Has Failed Us Completely, But No More

Waldo Jaquith
Story Time
Healthcare.gov
Welcome to the Marketplace

Enroll now in a plan that covers essential benefits, pre-existing conditions, and more. Open enrollment continues until March 31, 2014.

Choose your state and we'll tell you your next steps

If you live in Ohio, you'll use this website, HealthCare.gov, to apply for coverage, compare plans, and enroll. Here's what you need to know before you apply. You can also see if you qualify for lower costs and preview plans and prices.

How the Marketplace works
Rhode Island Unified Health Infrastructure Project
64 software bugs, complex union rules and a $15.8 million mistake: Why S.F. can’t pay its teachers on time

Software flaw hid signs of $576 million state unemployment fraud

Audit: Troubled Rhode Island system has cost $400M so far

Software Failure, Supply chain Blamed for School’s Inability to Turn Off Lights for Over a Year

States $170 million project with IBM over failure to deliver computer system it promised
Software Procurement Has Failed Us Completely
Of all government software development contracts over $6M, only 13% are successful (cost, schedule, performance).

Source
46% of systems developed across $37 billion dollars worth of DoD spending failed to meet real needs even though they met written, contractual specifications.
A study of 400 projects found that only 10% of waterfall-developed code was ever actually deployed. **Only 2% was ever used.**
The World Bank’s Government Procurement Benchmark for the average US Government procurement action is 260 days just to award contracts that will fail.
I am going to create an amazing list of requirements that is so comprehensive.
"Deloitte presented much too rosy of a picture to us," Gov. Raimond said. "I sat in meetings with Deloitte and questioned them and they gave us dashboards that showed us everything was green and ready to go, and the fact of the matter was it wasn't."
But No More
1. User-centered design
2. Agile software development
3. Product ownership
4. DevOps
5. Building out of loosely coupled parts
6. Modular contracting
User-centered design
Iterative Design

- Concept
- Try it out ("prototype" or "draft")
- Incorporate feedback
- Gather feedback (testing with users and stakeholders)
Agile software development
The premise of Agile

Design, Build, Test

Inspect
Agile development model

Feature 1:
- Define
- Design
- Code
- Unit tests
- Acceptance tests
- Integration tests
- Document

Feature 3:
- Define
- Design
- Code
- Unit tests
- Acceptance tests
- Integration tests
- Document

Feature 5:
- Define
- Design
- Code
- Unit tests
- Acceptance tests
- Integration tests
- Document

Feature 2:
- Define
- Design
- Code
- Unit tests
- Acceptance tests
- Integration tests
- Document

Time
Constantly delivering value

Not like this....

Like this!
## Waterfall control model

<table>
<thead>
<tr>
<th>Task</th>
<th>Vendor</th>
<th>Government</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identifying tasks</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>User research</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Writing code</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Testing</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>QA</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Deployment</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>System integration</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
## Agile control model

<table>
<thead>
<tr>
<th>Task</th>
<th>Vendor</th>
<th>Government</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identifying tasks</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>User research</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Writing code</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Testing</td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>QA</td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Deployment</td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>System integration</td>
<td></td>
<td>✔️</td>
</tr>
</tbody>
</table>
Product ownership
DevOps
Development

+ Operations
+ Security
+ Automation

People do the things they’re good at; computers do the things they’re good at.
Building out of loosely coupled parts
Interoperability standardization yields system-level thinking.

Bespoke monoliths are terrifying.

The industrial revolution has come to software.
Modular contracting
Traditional contracting
Modular contracting

contract module

contract module

contract module
1. User-centered design
2. Agile software development
3. Product ownership
4. DevOps
5. Building out of loosely coupled parts
6. Modular contracting
Bringing It All Together
Q&A