Software Acquisition:  
Facing the Challenge,  
Valuing Velocity!  

James Thompson  
Office of the Under Secretary of Defense for  
Research and Engineering  

October 24, 2018
Code of Hammurabi (~1754 BC)

If a builder build a house for some one, and does not construct it properly, and the house which he built fall in and kill its owner, then that builder shall be put to death.


De Architectura on Cost/Schedule Overruns (~15 BC)

• [...] When an architect [in Ephesus] was entrusted with the execution of a public work, an estimate thereof being lodged in the hands of a magistrate, his property was held, as security, until the work was finished. [...] But when more than one-fourth of the estimate was exceeded, he was required to pay the excess out of his own pocket. [...] 

• Would to God that such a law existed among the Roman people, not only in respect of their public, but also of their private buildings, for then the unskillful could not commit their depredations with impunity, and those who were the most skillful in the intricacies of the art would follow the profession. Proprietors would not be led into an extravagant expenditure so as to cause ruin; architects themselves, from the dread of punishment, would be more careful in their calculations, and the proprietor would complete his building for that sum [...] 

DoD Software Failures in the Press

Air Force May Delay F-35 Combat Readiness Unless Software Improves
~ Investor's Business Daily

Billion-Dollar Flop: Air Force Stumbles on (ECSS) Software Plan
~ The New York Times

'Strongly misleading': US general blasts Raytheon's GPS control system a 'disaster'
~ Reuters

Software Hurdles May Follow KC-46A Wiring Woes
~ DoD Buzz

Senate Condemns US Air Force ECSS Program Management's Incompetence
~ IEEE Spectrum

Also, many successful software programs
Software Is Everywhere

- DoD relies on software to provide decisive advantages to our forces
- The complexity required to achieve this advantage demands specific capabilities and tight coupling
- Partial solutions are inadequate
- We can’t omit requirements
  - Because they don’t fit the schedule
  - Because it simplifies refactoring
### Software Risk Assessed by DoD Program Offices

<table>
<thead>
<tr>
<th>Likelihood</th>
<th>Consequence</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

*Software not in top program risks*

Software among most frequent and most critical challenges, driving program risk on ~ 60% of acquisition programs.
Historical Approaches to Address DoD’s Software Challenges

- 1984: Software Engineering Institute established
  - Software Development and Documentation MIL-STD-498
- 1987: Ada as single, common programming language DoDD 3405.1
- 1990: DoD SW shall be in Ada NDAAs
  - DoD CIO established; DoDI 5000 required compliance Clinger-Cohen Act
- 2000: Time-Certain Development & MAIS Critical Change Reporting FY07 NDAA
- 2002: Implements CMM level 3 as goal DoD 5000.02-R
- 2003: Software/Systems Process Improvement (SPI) FY03 NDAA
- 2008: 2 DSB Reports on Acquisition issues
- 2009: Addition of SW-centric Models DoD 5000.02
- 2010: GAO Reports on Acquisition & Agile
- 2015: Implementation of New Acquisition Process for IT Systems FY10 NDAA
- 2017: MOSA, et al FY17 NDAA
- 2018: FY18NDAA ??

Legend:
- Policy & Guidance
- Congressional Actions
- Reports

Distribution Statement A: Approved for public release. Distribution is unlimited. DOPSR Case #18-S-1517
Historical Approaches to Address DoD’s Software Challenges

1984
Software Engineering Institute established

1987
Ada as single, common programming language DoDD 3409.1

1990
DoD SW shall be in Ada NDAA

1994
Software Development and Documentation MIL-STD-498

1996
DoD CIO established; DoD 5000 required compliance Clinger-Cohen Act

2000
Implements CMM level 3 as goal DoD 5000.02-R

2001
GAO Report on Software / Systems Process Improvement (SPI)

2002
Time-Certain Development & MAIS Critical Change Reporting FY07 NDAA

2003
Software / Systems Process Improvement (SPI) FY03 NDAA

2004
2 DSB Reports on Acquisition issues

2007
Addition of SW-centric Models DoD 5000.02

2009
GAO Report: DOD Needs to Strengthen Management of Its SSP Efforts

2010
Implementation of New Acquisition Process For IT Systems FY10 NDAA

2011
MOAA, et al. FY11 NDAA

2015
DSB Reports on Software Acquisition

2017

LEGEND

 Policy & Guidance  Congressional Actions  Reports
Defense Science Board Report on Software Acquisition

- “Software Factory”
- Continuous Iterative Development
  - Requirements/MVP
- Risk Reduction & Metrics
  - Competition
  - Cost/schedule scoping estimation techniques
  - Execution metrics framework
- Transition to Current and Legacy Programs
- Workforce improvement
- Software is Immortal - Sustainment
- Machine Learning – IV&V, Cyber

## Addressing the Challenge!

### Valuing Velocity

<table>
<thead>
<tr>
<th>Challenge/Opportunity</th>
<th>Approach</th>
</tr>
</thead>
</table>
| Software/data are critical assets, software/related fields are critical skills | • Cultivate workforce, centers of excellence, capabilities, base  
• Software Factory, repository  
• Outreach to industry, academia |
| Software has become rate controlling step in acquisition | • Credibility in planning & execution – improve scoping & metrics framework  
• Establish/demonstrate Software Factory early (TMRR), use in evaluations  
• Common production/mission representative development and test environments  
• Harness technology (e.g., automation (build, test), machine learning, etc.)) |
| Grand scales and unified architectures are challenging | • Modify requirements process - “Big R” - enable MVP / P3I  
• Modify acquisition processes - right-size programs and approaches  
• Value modularity, consider decoupling/loosening integration  
• Reduce customization (enterprise optimization)  
• Promote multi-functionality |
| Software enables opportunity for continuous capability delivery | • Software is immortal  
• Iterative development – initial development and sustainment |
| Hardware limitations are barriers | • Hardware abundance (cpu, bandwidth, etc.) supporting iterative acquisition  
• Promote hardware/software independence |
| Cyber/program protection / resilience add layers to challenge | • Build in capability  
• Refresh opportunity  
• Manage risk  
• “Software by the pound” |

**Implement on current and legacy programs**
Software Cycle Time for Recent Programs

David M. Tate, Software Development May Drive Future Acquisition Cycle Times, (Revised) IDA Document NS D-8053 (Revised)
October 2016. Log: H 16-000790 - Approved for public release; distribution is unlimited.
Estimated Schedule Durations for a Software Development Effort

Productivity required to meet planned 32-month schedule is substantially higher than industry averages and developer's history on Programs A and B.

Planned duration appears unrealistic compared to historic data for Programs A and B, and to Industry and AT&L trend estimates.

Program’s original plan was more realistic.

Software benchmarks promote credibility in scoping, enable data-driven decisions.
Sample Metrics Testing Optimism

2 April 2014: Testing Begins

Forecasted Completion

Total Test Groups
Free-play Actuals
Free-play Plans

Test Groups

Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun
Sample Metrics Testing Optimism

- Test closure doesn’t go as expected
- Forecasted completion re-planned

Test Groups

- Total Test Groups
- Free-play Actuals
- Free-play Plans

Forecasted Completion Shifts Right
Sample Metrics Testing Optimism

- Still not as expected
- Forecast re-planned again

Forecasted Completion Shifts Right again
Sample Metrics Testing Optimism

- Still not as expected

But Optimism starts to set in that schedule can be maintained
Sample Metrics Testing Optimism

- Summit achieved – “Now we have full understanding”

Optimism fully realized – we can accelerate closure and push schedule left

Test Groups

- Total Test Groups
- Free-play Actuals
- Free-play Plans

Distribution Statement A: Approved for public release. Distribution is unlimited. DOPS Case #18-S-1517
Sample Metrics Testing Optimism

- Closing test cases but not on plan
- Blinding Optimism -- "Closure Cliff"
- Shift in forecast back to original

Graph showing test groups over time with various test case closures and forecast adjustments.
Sample Metrics Testing Optimism

---

**Test Groups**

- **Total Test Groups**
- **Free-play Actuals**
- **Free-play Plans**

---

**Graph Details:**
- **15 Dec 14: Initial Completion Date Not Obtained**
- **Contractor Surges to minimize delay**
- **Forecast Re-planned**

---

Distribution Statement A: Approved for public release. Distribution is unlimited. DOPSR Case #18-S-1517
Sample Metrics Testing Optimism

15 Dec 14: Initial Completion
Date Not Obtained

...And again – 13 Mar 15 Completion

Test Groups

Total Test Groups
Free-play Actuals
Free-play Plans
Sample Metrics Testing Optimism

Summary

15 Dec 14: Initial Completion Date Not Obtained
Contractor Surges to minimize delay
13 Mar 15: Completion

Test Groups

Total Test Groups
Free-play Actuals
Free-play Plans
Opportunities

AUTONOMY
- DSB: Autonomous solutions mitigate mission challenges
  - Enable rapid decision making
  - Manage a high volume of data
  - Coordinate complex actions
  - Ensure persistence and endurance

ARTIFICIAL INTELLIGENCE
- Improve on-board sensing
- Exploit time-critical intelligence from seized media
- Manage a dynamic spectrum for protection missions

MACHINE LEARNING
- Adapt acquisition and sustainment to rapid deployment
- Attain predictive logistics and adaptive planning

CONTINUOUS CAPABILITY DELIVERY
- Software “immortality”
- Iterative development and sustainment
  - Agile
Opportunities

MODELING & SIMULATION

- Enable warfighting capability and acquisition
  - Reduce risk
  - Accelerate programs
  - Save lives and $ 

MODEL-BASED SYSTEMS ENGINEERING AND TEST

- System requirements, design, analysis, verification and validation across the life cycle
  - Improve communication
  - Manage increased complexity
  - Improve quality 

EXPANDING SOURCES

- National software resources and capabilities
  - Embrace U.S. leadership in SW
  - Utilize Silicon Valley
Challenges

**Cyber-attack**
- Increased SW provides a vulnerability path to a master OFF switch
- Need to protect key mission components from malicious activity
- As SW increase the potential for vulnerabilities increases exponentially
- Ensure key data is protected from adversary collection
- Strengthen Supply Chain Activities

**SW is rate controlling step in acquisition**

DoD acquisition program cycle time is dictated by amount of SW

**Inadequate insight – performance to plan**

Never quite sure where we are during development (inadequate data)

Always optimistic... until the end (can’t believe data we have)
Challenges

Defects / unintended consequences

Impossible to exhaustively test systems

How do we establish and maintain trust in SW systems?

Grand architectures

Grand scales and unified architectures hamper timely solutions

Need to modify acquisition processes for “Minimal Viable Product”

SW – HW coupling

Current: HW is static in a system; SW evolution is limited by installed HW

Next Gen:
  • Value modularity, consider decoupling/loosening integration
  • HW abundance (cpu, bandwidth, etc.) needs to be factored into iterative acquisition

Software / data / skills are critical resources

DoD workforce is dated

DoD workforce not sufficiently knowledgeable to SW challenges and solutions

Mean age: ~ 46
What About Agile?

- Can Agile address the complexity of DoD systems?
  - Can we decompose tightly-coupled technical requirements into Agile user stories and controlled interfaces?
  - Can we identify authoritative customers - among many diverse stakeholders, including the Adversary - for feedback and iteration?
  - Can we learn from small, agile teams and scale to complex projects?
  - Can we support formal, independent testing over long test cycles?
  - Can we deliver capabilities?

- Can Agile address regulatory challenges?
  - Can we provide enough “up-front” cost, schedule, and risk analysis to satisfy DoD regulatory and statutory requirements?
  - Can we support the persistent oversight and management requirements of DoD acquisitions?
  - Can we mix contractual negotiation with customer collaboration?

DoD Systems tend to be complex, with independently developed, highly-coupled components
## Agile/Classic Concepts

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Agile</th>
<th>Classic DoD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Involvement</td>
<td>▪ Stakeholder Involvement</td>
<td>▪ Integrated Product Teams</td>
</tr>
<tr>
<td>Approach</td>
<td>▪ Continuous Iterative Development</td>
<td>▪ Build a little, test a little, learn a lot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ Pre-Planned Product Improvement</td>
</tr>
<tr>
<td>Modularity</td>
<td>▪ Incremental, “vertical slice” products</td>
<td>▪ Chunk the problem</td>
</tr>
<tr>
<td></td>
<td>▪ Incremental developments have value, and customers can use them.</td>
<td>▪ Customers cannot start testing parts of the product until everything is</td>
</tr>
<tr>
<td></td>
<td>▪ Work can be broken into parts and conducted in rapid, iterative</td>
<td>complete.</td>
</tr>
<tr>
<td></td>
<td>cycles.</td>
<td>▪ Late changes are expensive or impossible</td>
</tr>
<tr>
<td></td>
<td>▪ Late changes are manageable.</td>
<td>▪ MOSA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ In-phase defect containment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ Total Quality Management</td>
</tr>
</tbody>
</table>

*Incomplete data on agile at scale, with some exaggeration; however... empirical data and strong industry movement to agile development across all domains strongly motivates DoD to move to agile development.*
# Addressing the Challenge!

## Valuing Velocity

<table>
<thead>
<tr>
<th>Challenge/Opportunity</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software/data are critical assets, software/related fields are critical skills</td>
<td>• Cultivate workforce, centers of excellence, capabilities, base&lt;br&gt;• Software Factory, repository&lt;br&gt;• Outreach to industry, academia</td>
</tr>
<tr>
<td>Software has become rate controlling step in acquisition</td>
<td>• Credibility in planning &amp; execution – improve scoping &amp; metrics framework&lt;br&gt;• Establish/demonstrate Software Factory early (TMRR), use in evaluations&lt;br&gt;• Common production/mission representative development and test environments&lt;br&gt;• Harness technology (e.g., automation (build, test), machine learning, etc.)</td>
</tr>
<tr>
<td>Grand scales and unified architectures are challenging</td>
<td>• Modify requirements process - “Big R” - enable MVP / P3I&lt;br&gt;• Modify acquisition processes - right-size programs and approaches&lt;br&gt;• Value modularity, consider decoupling/loosening integration&lt;br&gt;• Reduce customization (enterprise optimization)&lt;br&gt;• Promote multi-functionality</td>
</tr>
<tr>
<td>Software enables opportunity for continuous capability delivery</td>
<td>• Software is immortal&lt;br&gt;• Iterative development – initial development and sustainment</td>
</tr>
<tr>
<td>Hardware limitations are barriers</td>
<td>• Hardware abundance (cpu, bandwidth, etc.) supporting iterative acquisition&lt;br&gt;• Promote hardware/software independence</td>
</tr>
<tr>
<td>Cyber/program protection / resilience add layers to challenge</td>
<td>• Build in capability&lt;br&gt;• Refresh opportunity&lt;br&gt;• Manage risk&lt;br&gt;• “Software by the pound”</td>
</tr>
</tbody>
</table>

### Implement on current and legacy programs
Conclusion

- Unrelenting demand for SW in DoD
- SW is the fuel for innovation and future capabilities
- Need action to meet challenges
  - Diversify the Department’s approach and sources
  - Embrace opportunity in a change-rich environment (threats, technology, process improvements)

Shift to threat-based acquisition demands enhanced velocity
Shift SW acquisition to an enabler of speeding capability
Fearless workers who stood with their back to a TANK roaring toward them to prove its stopping power

Link: http://video.dailymail.co.uk/video/1418450360/2014/01/1418450360_3121005887001_tank-brake-test.mp4
DoD Research and Engineering Enterprise
Solving Problems Today – Designing Solutions for Tomorrow

DoD Research and Engineering Enterprise
https://www.acq.osd.mil/chieftecnologist/

Defense Innovation Marketplace
https://defenseinnovationmarketplace.dtic.mil

Twitter
@DoDInnovation
For Additional Information

Mr. James Thompson
Office of the Under Secretary of Defense
for Research and Engineering
james.j.thompson3.civ@mail.mil