21st Annual National Defense Industrial Association

Systems and Mission Engineering Conference

Software Acquisition: Facing the Challenge, Valuing Velocity!

James Thompson

Office of the Under Secretary of Defense for Research and Engineering

October 24, 2018

Classical Engineering/Project Management

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.

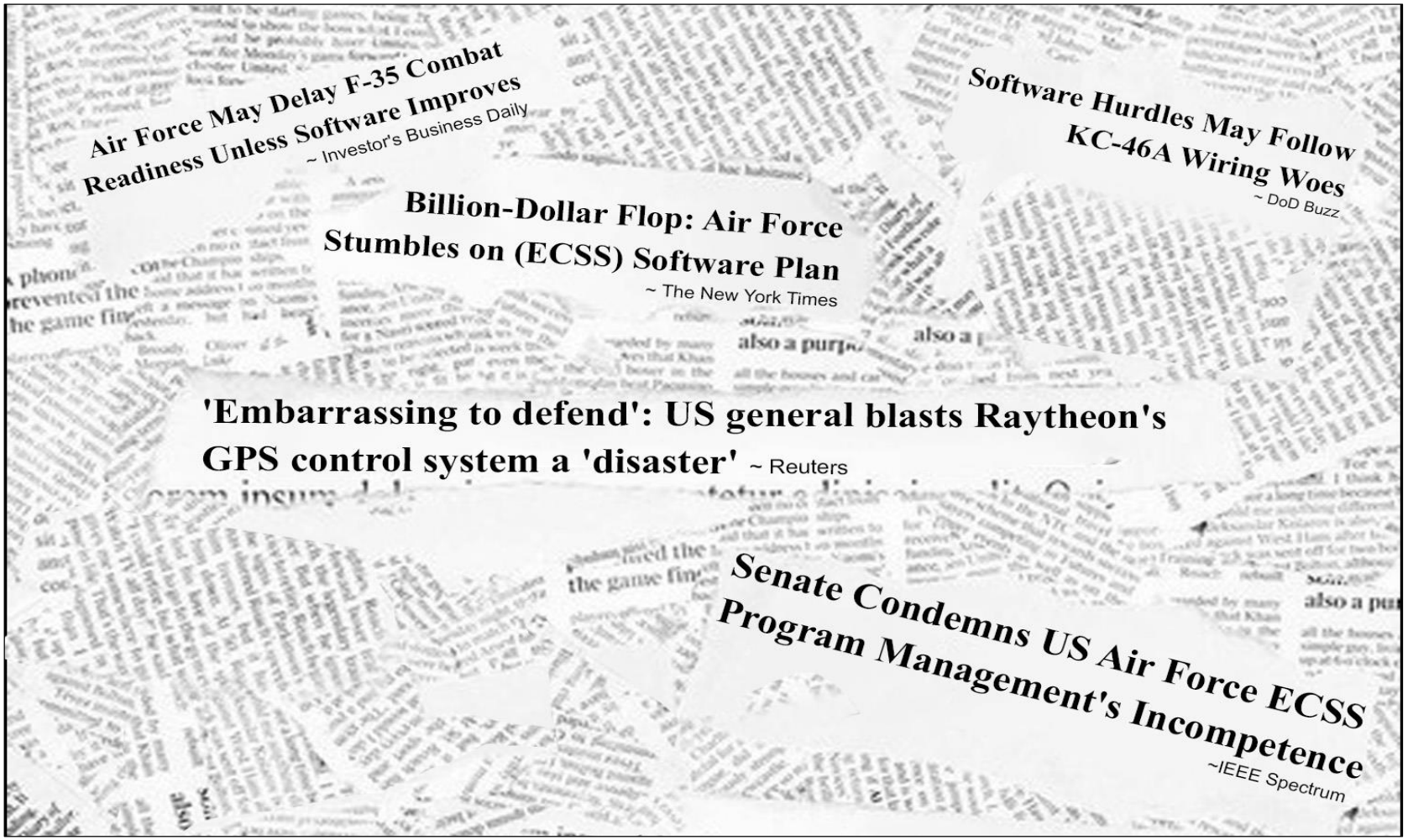
Code of Hammurabi, 229, http://avalon.law.yale.edu/ancient/hamframe.asp Accessed Sept 17 2017

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 [...]

Marcus Vitruvius Pollio. "De Architectura", Liber X, Introduction, paragraphs 1-2. Translation by Bill Thayer. ~15 BC . http://penelope.uchicago.edu/Thayer/E/Roman/Texts/Vitruvius/10*.html. Accessed 17 Sept 2017

DoD Software Failures in the Press



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

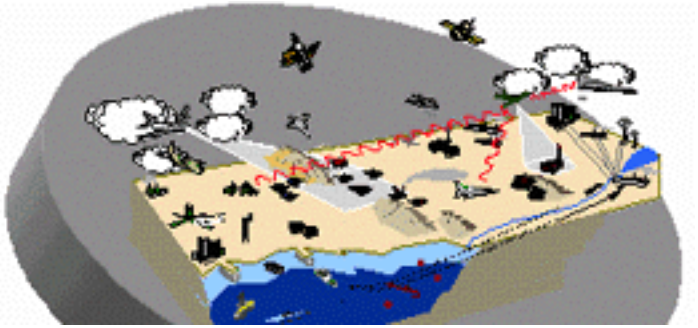


image credit SPAWAR.navy.mil

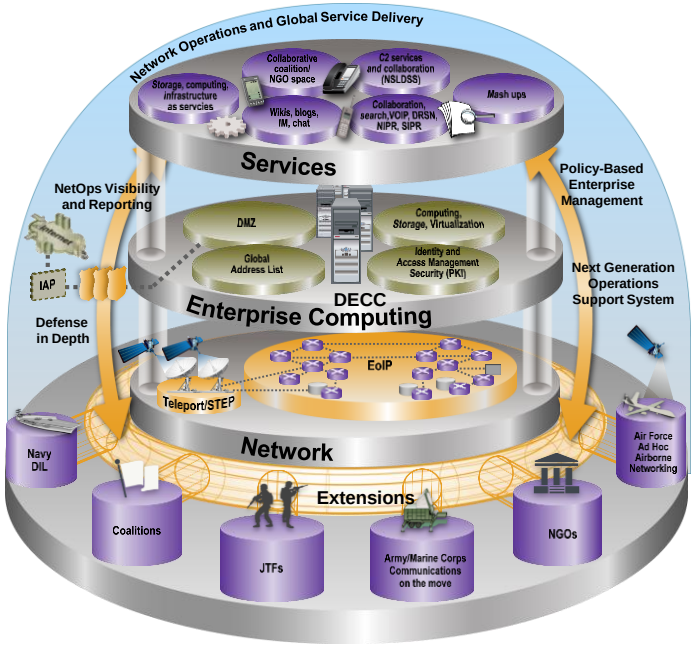
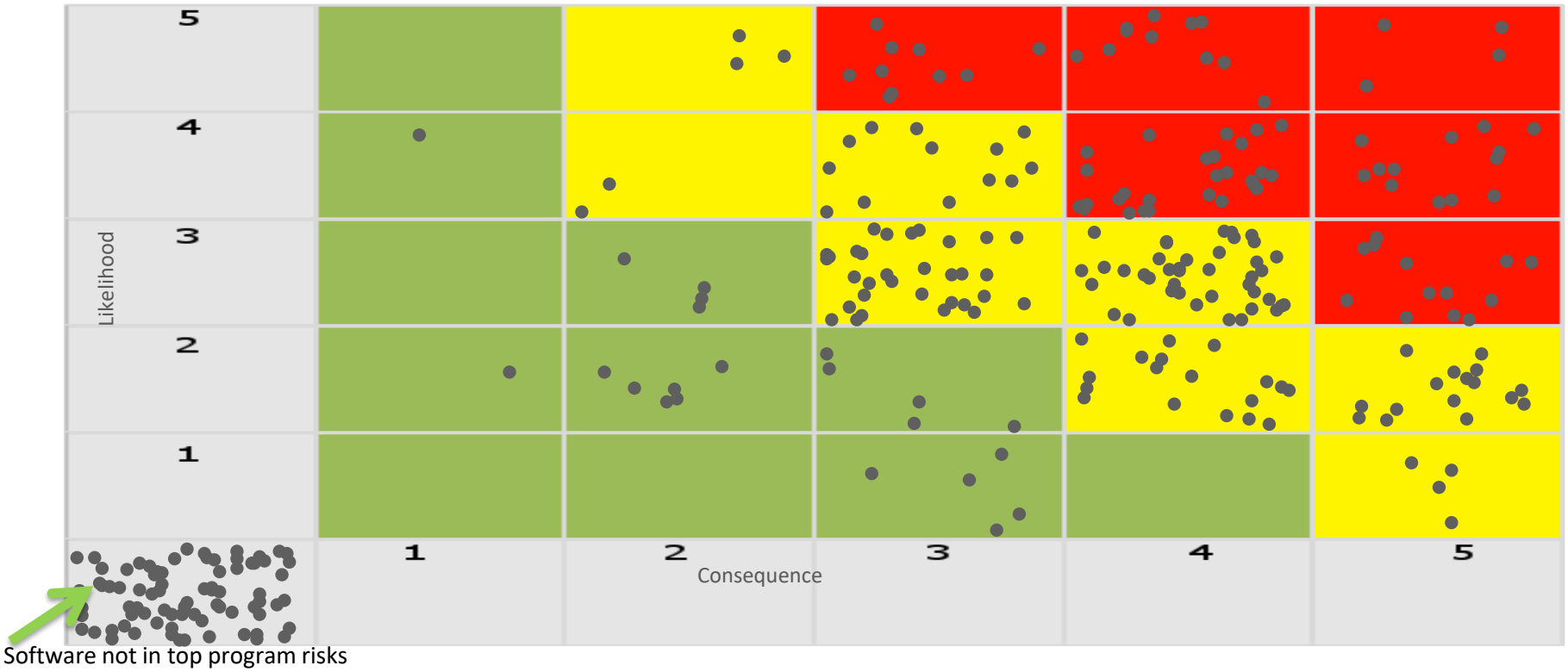


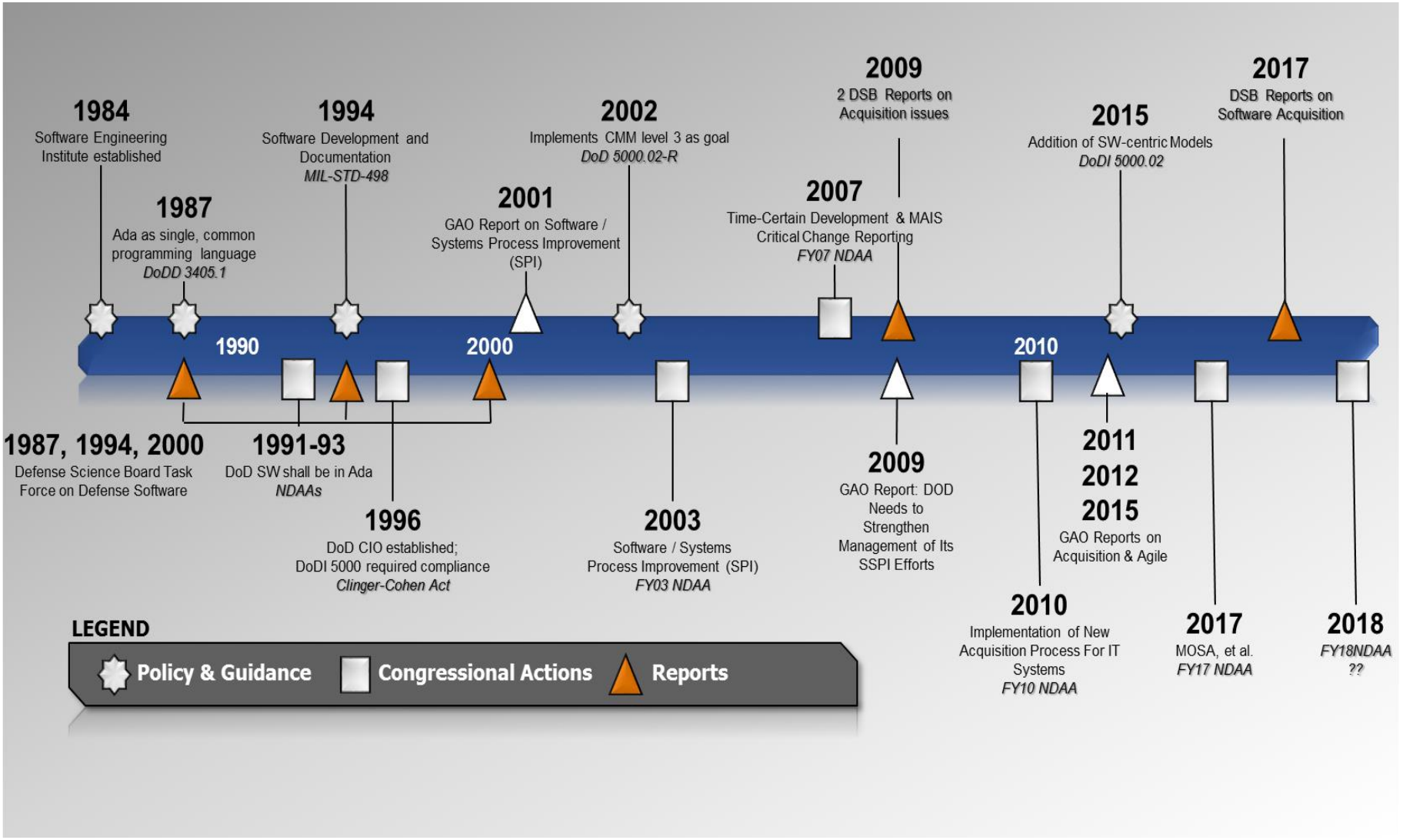
image credit DISA.mil

Software Risk Assessed by DoD Program Offices

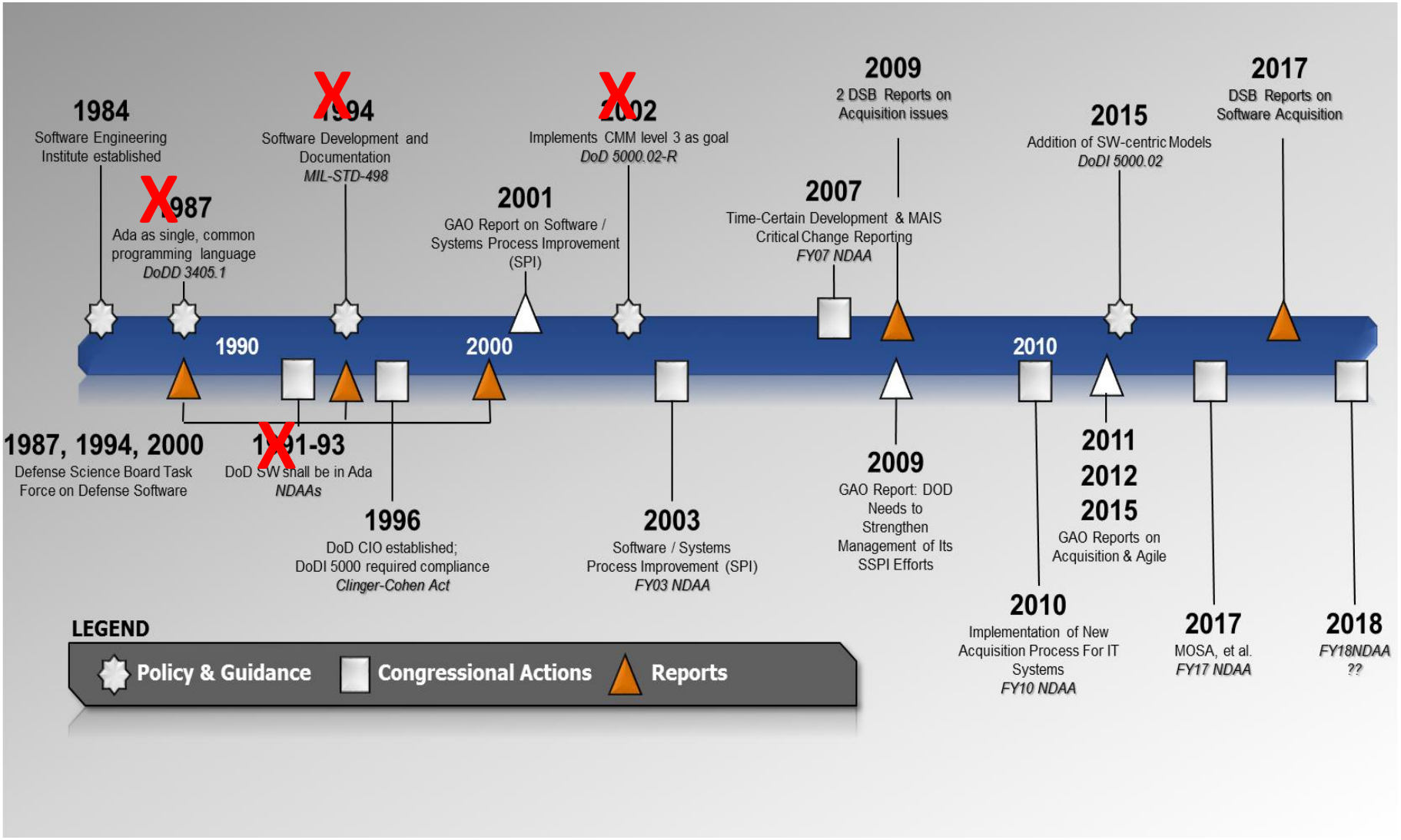


Software among most frequent and most critical challenges, driving program risk on ~ 60% of acquisition programs

Historical Approaches to Address DoD's Software Challenges



Historical Approaches to Address DoD's Software Challenges



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

https://www.acq.osd.mil/dsb/reports/2010s/DSB_SWA_Report_FINALdelivered2-21-2018.pdf

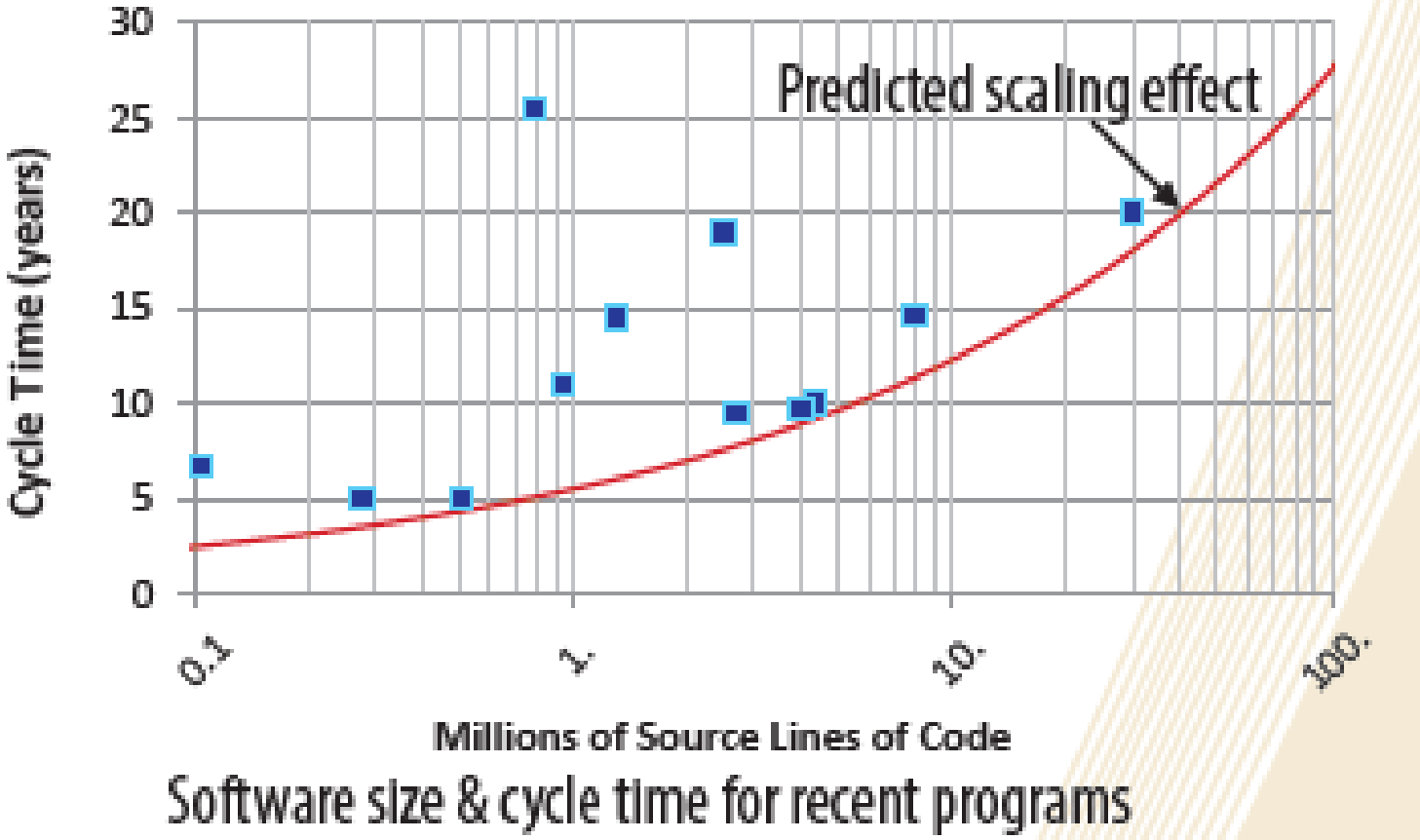
Addressing the Challenge!

Valuing Velocity

Challenge/Opportunity	Approach
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 M**V**P / 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"

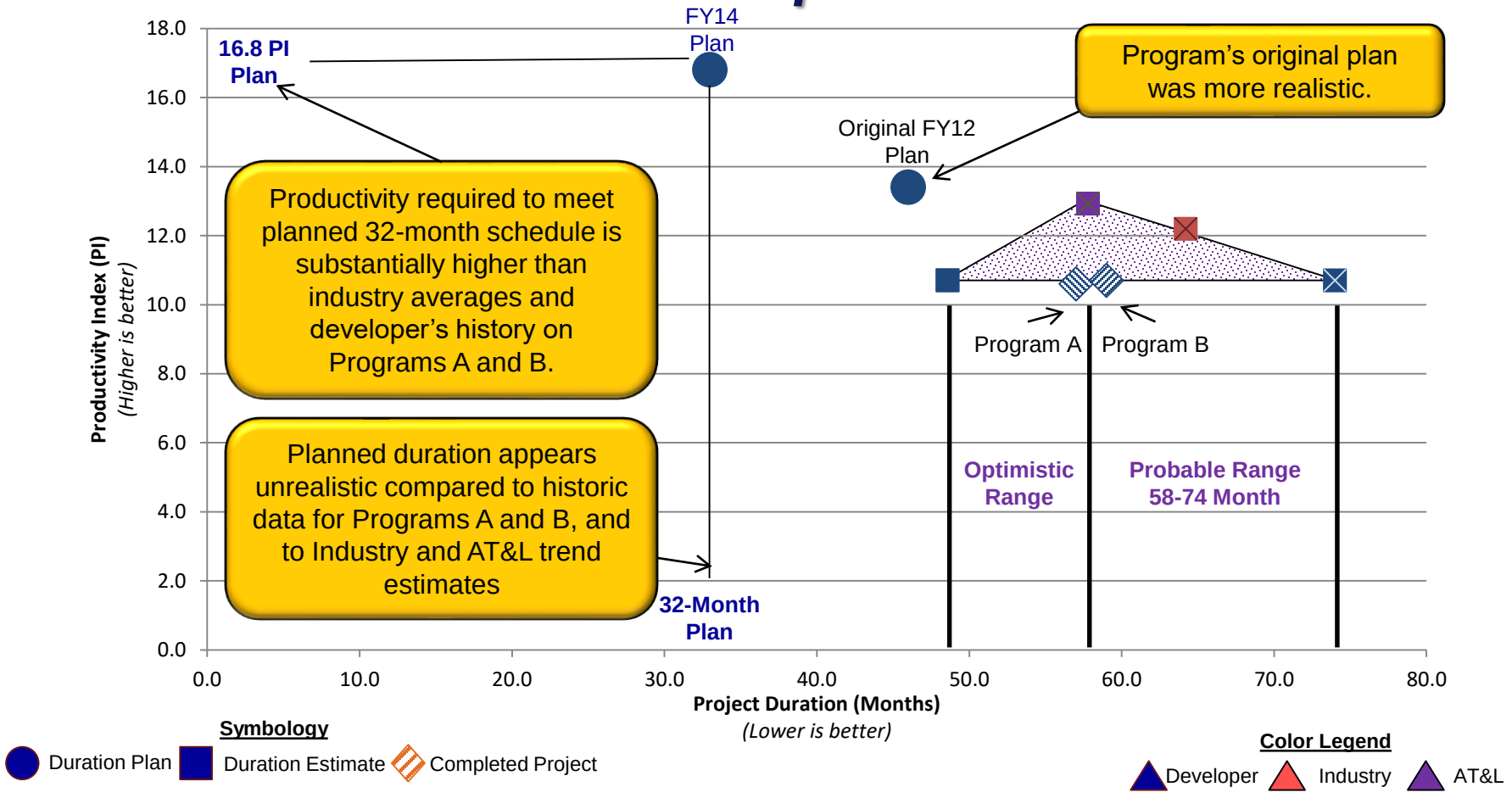
Implement on current and legacy programs

Software Cycle Time for Recent Programs



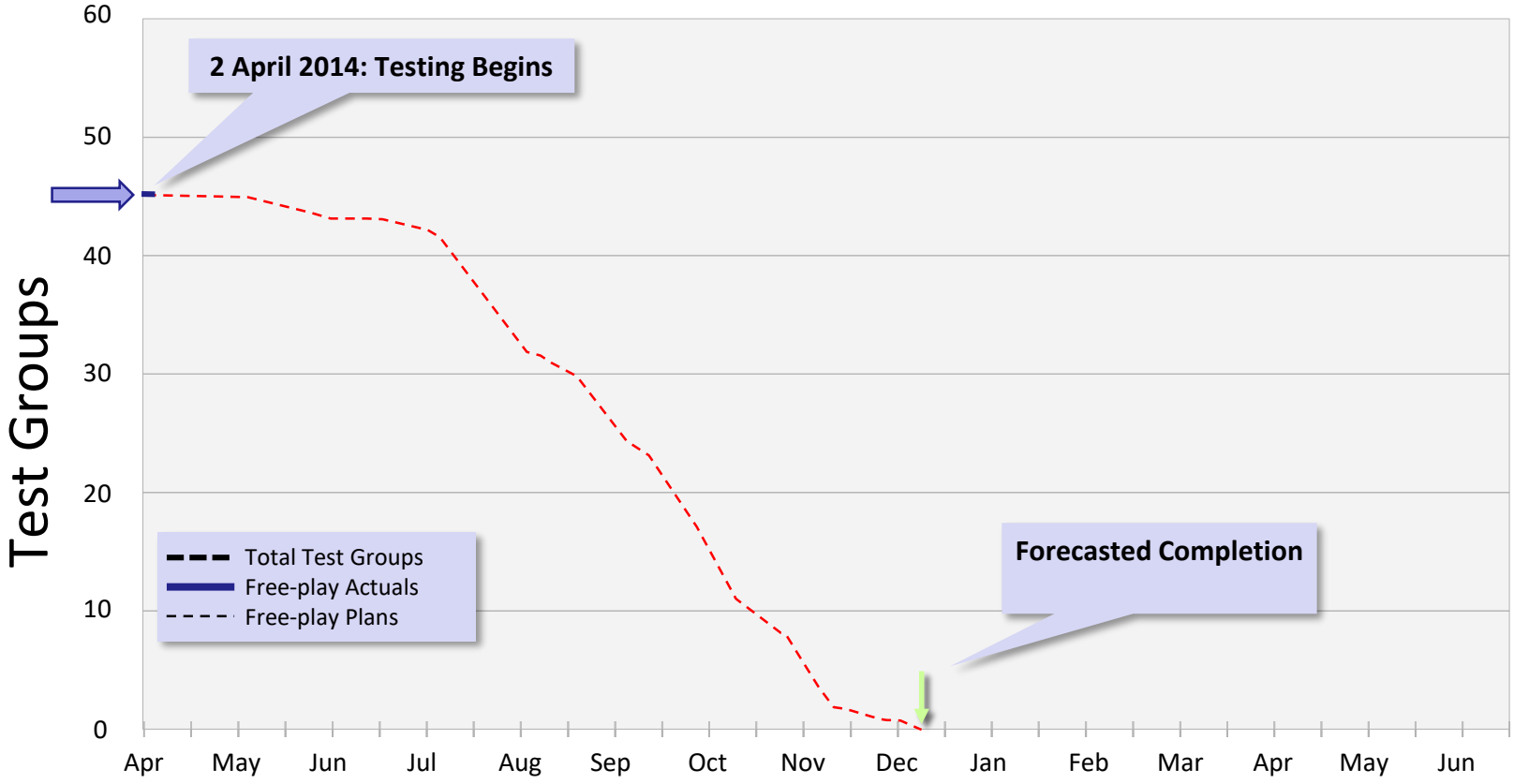
David M. Tate, Software Development May Drive Future Acquisition Cycle Times, (Revised)IDA Document NS D-8053 (Revised) October 2016. Log: H 16-000790 - Approved for public release; distribution is unlimited.

Estimated Schedule Durations for a Software Development Effort

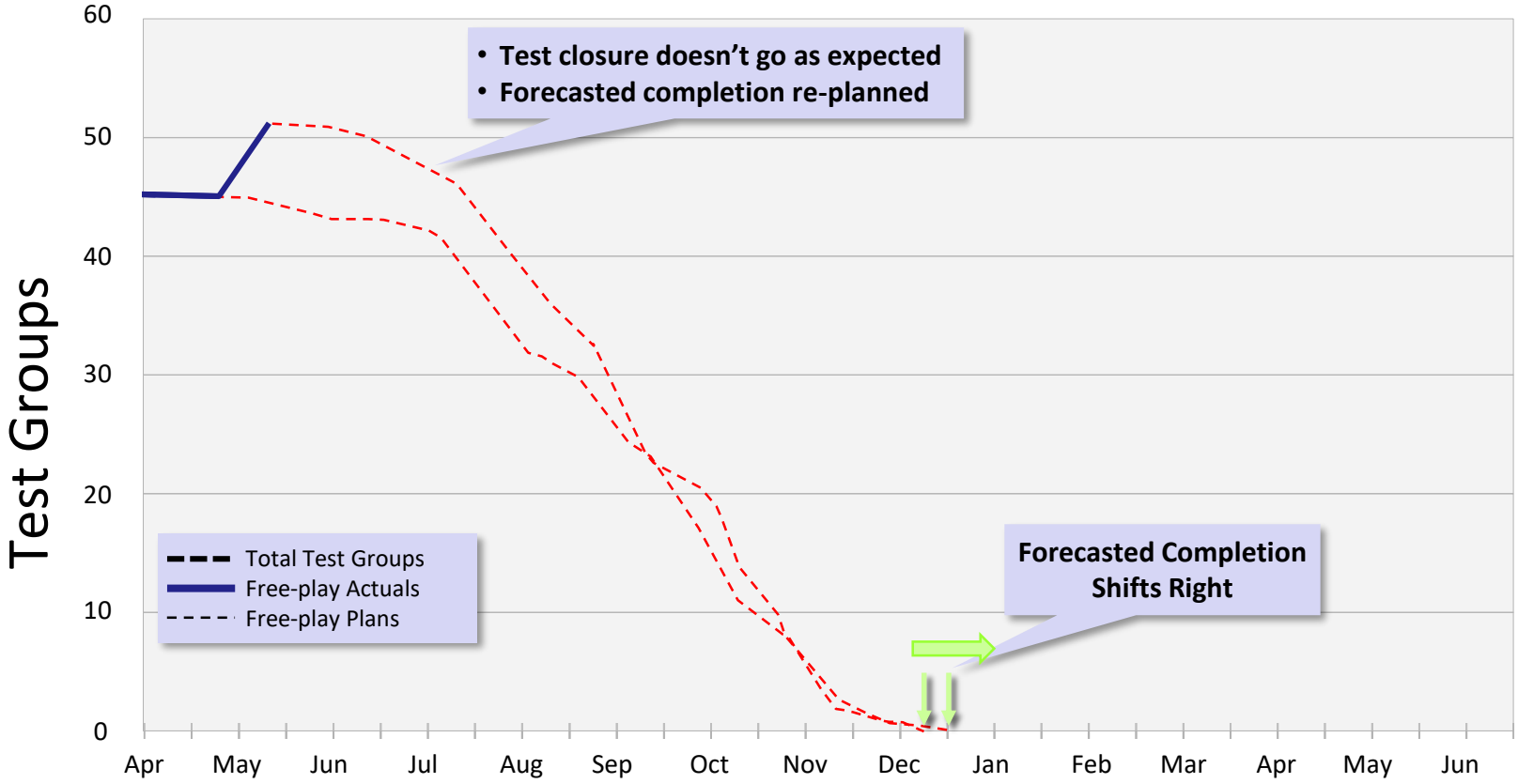


Software benchmarks promote credibility in scoping, enable data-driven decisions

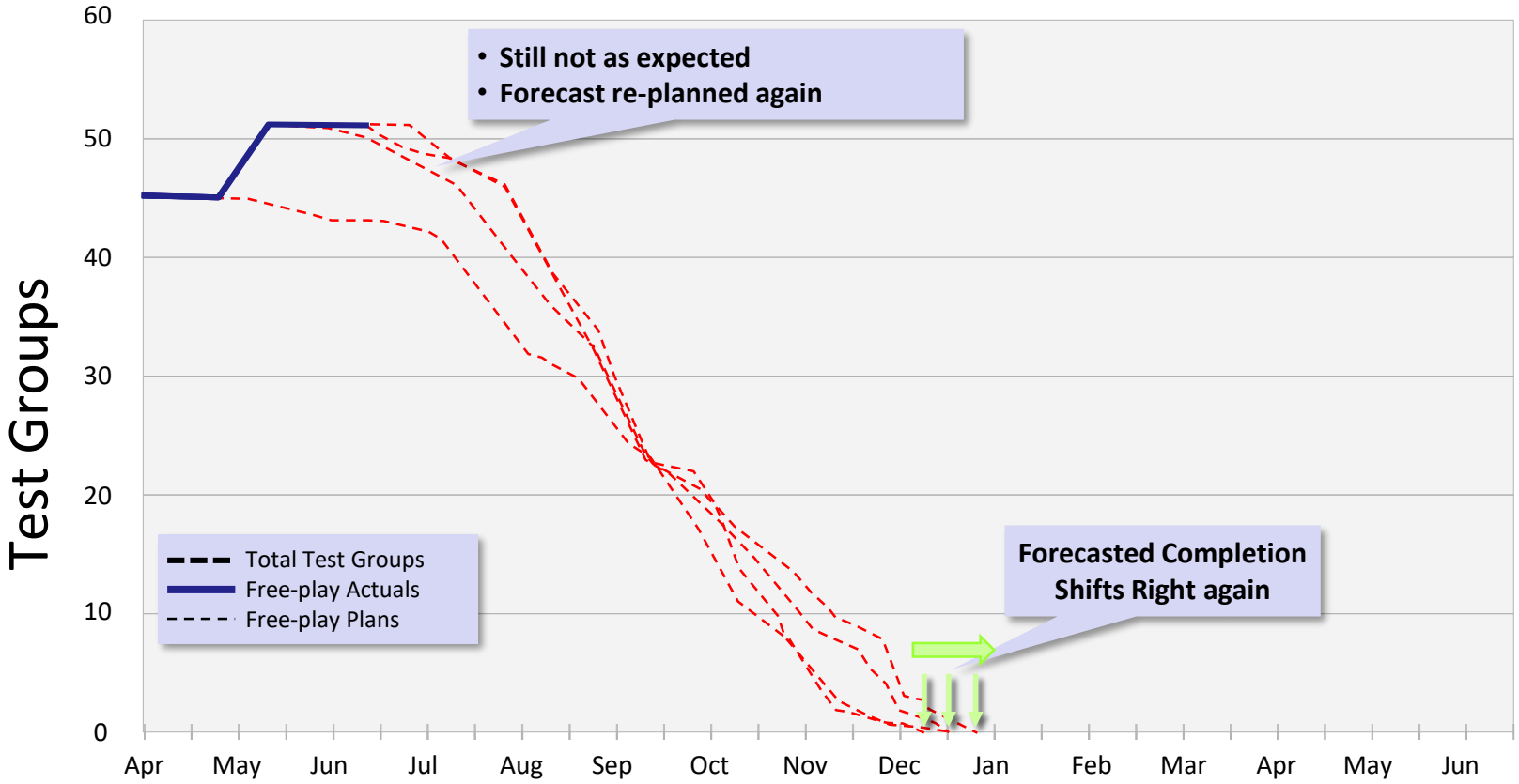
Sample Metrics Testing Optimism



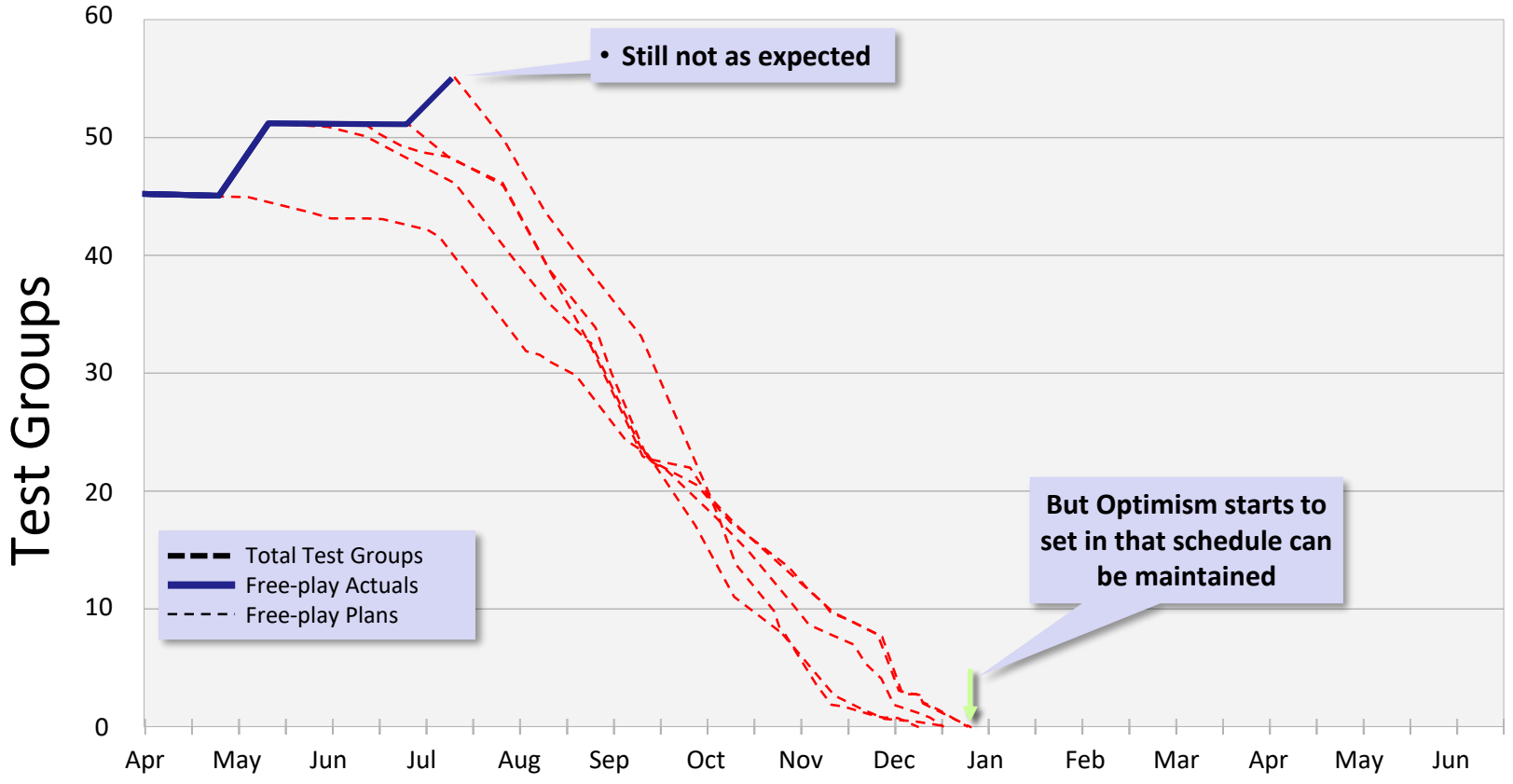
Sample Metrics Testing Optimism



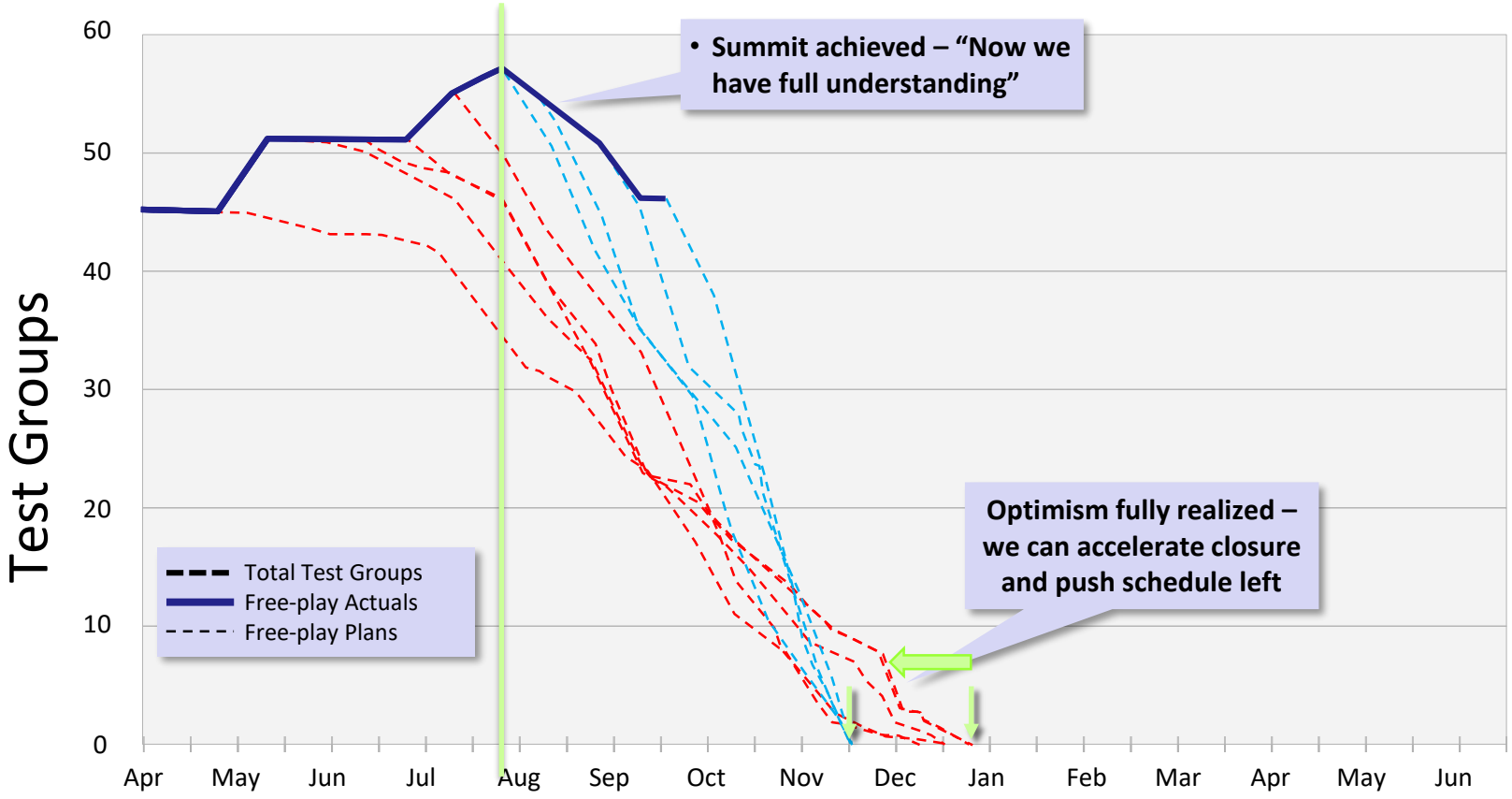
Sample Metrics Testing Optimism



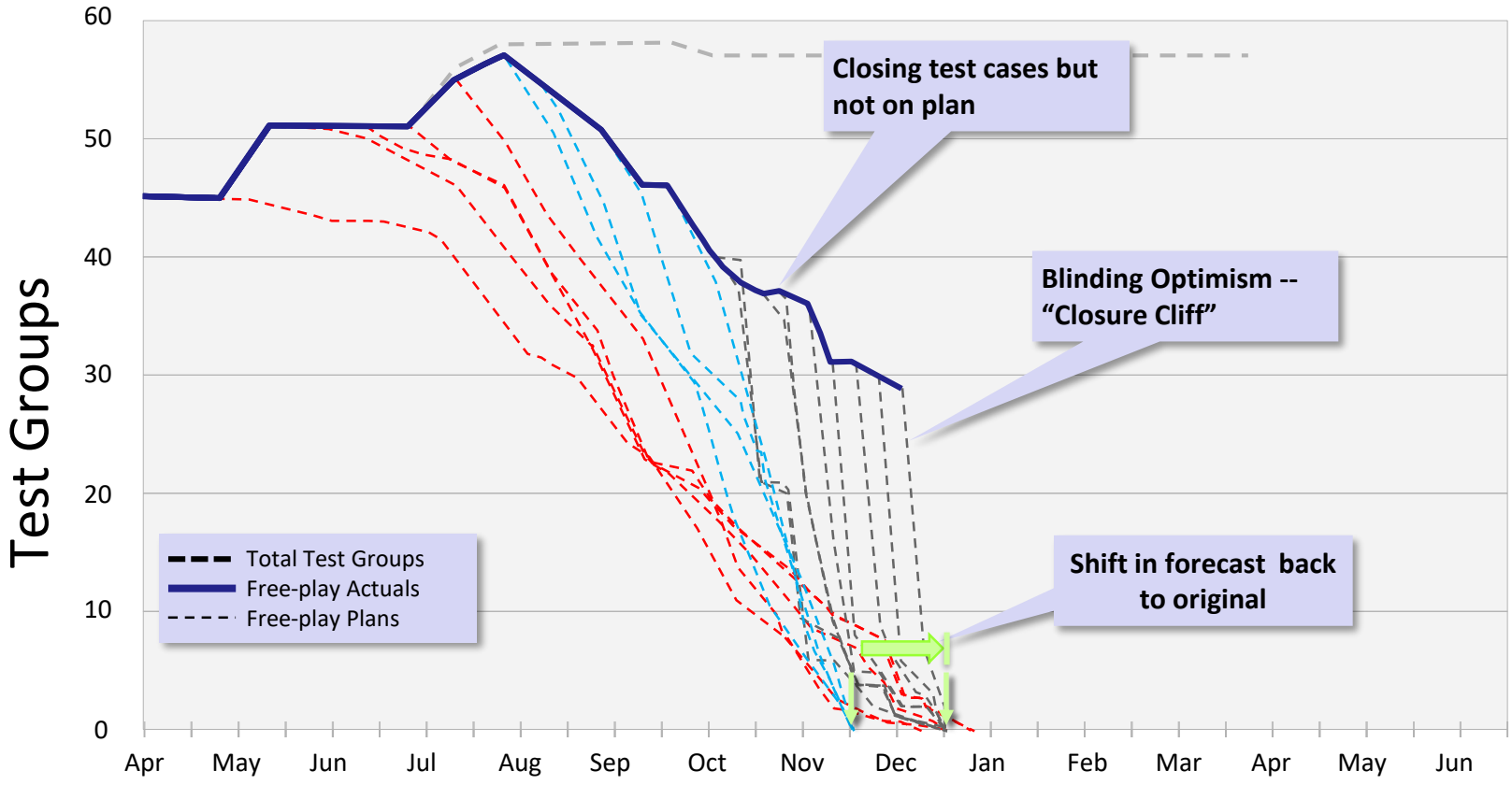
Sample Metrics Testing Optimism



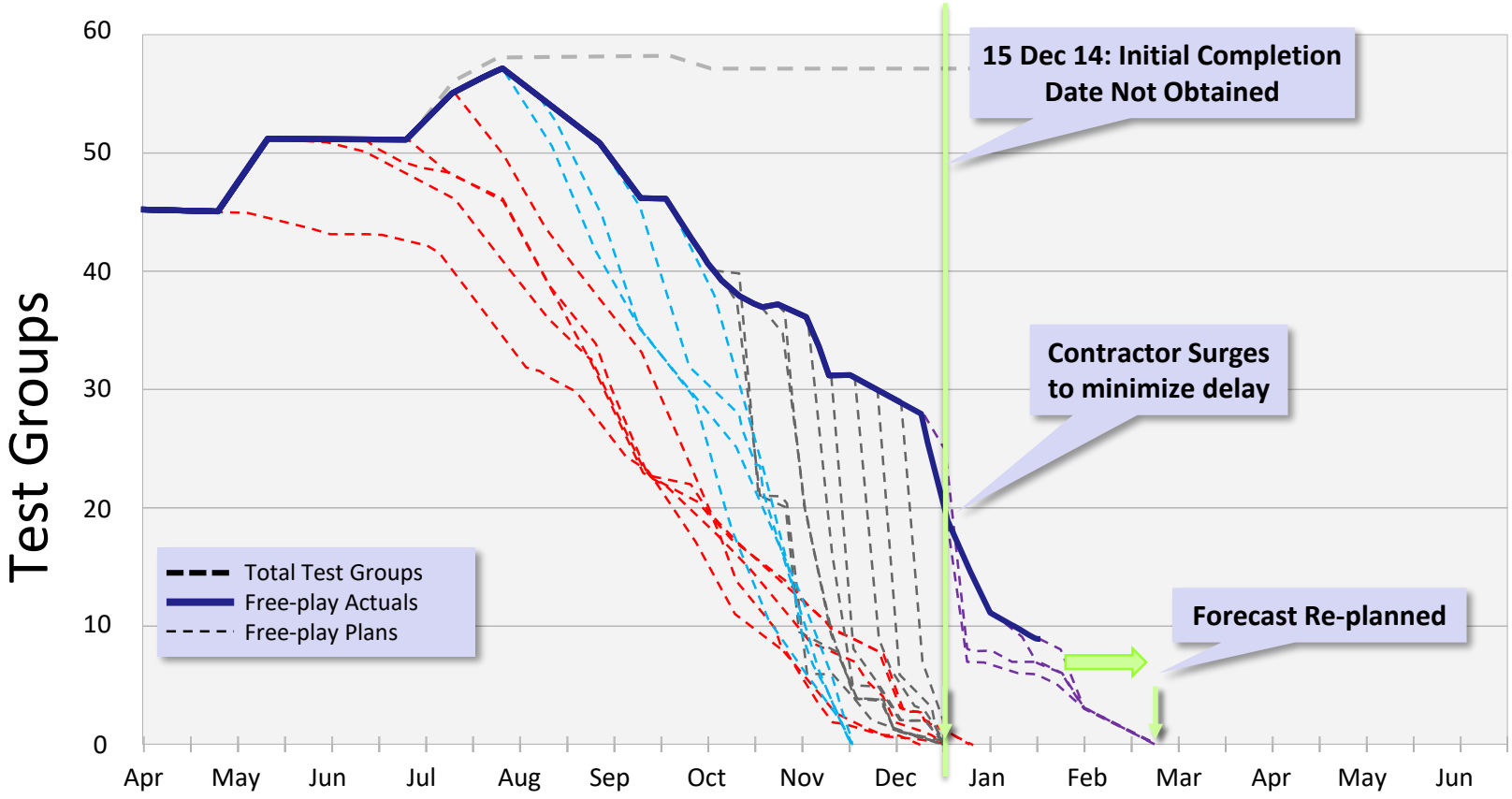
Sample Metrics Testing Optimism



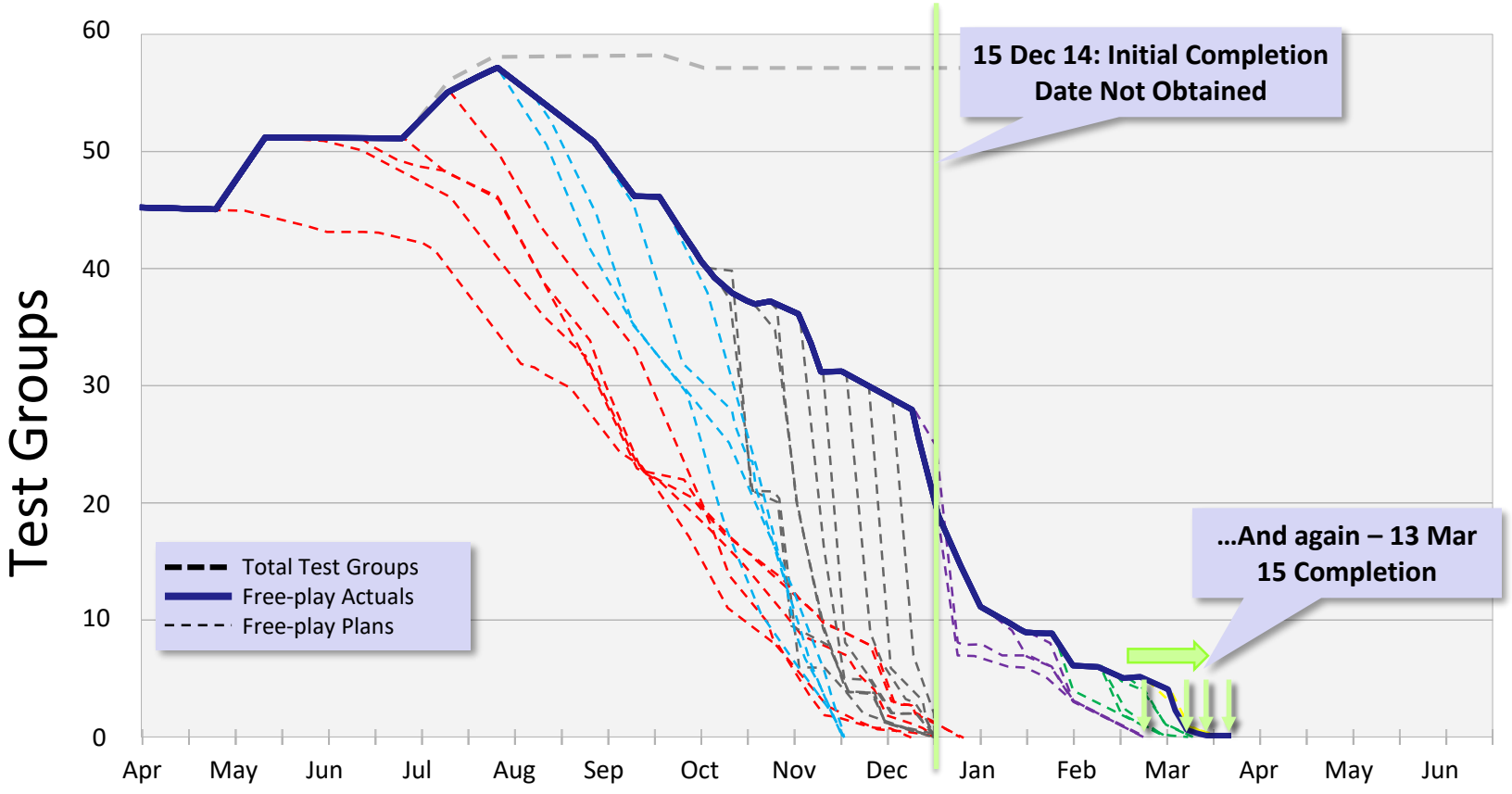
Sample Metrics Testing Optimism



Sample Metrics Testing Optimism

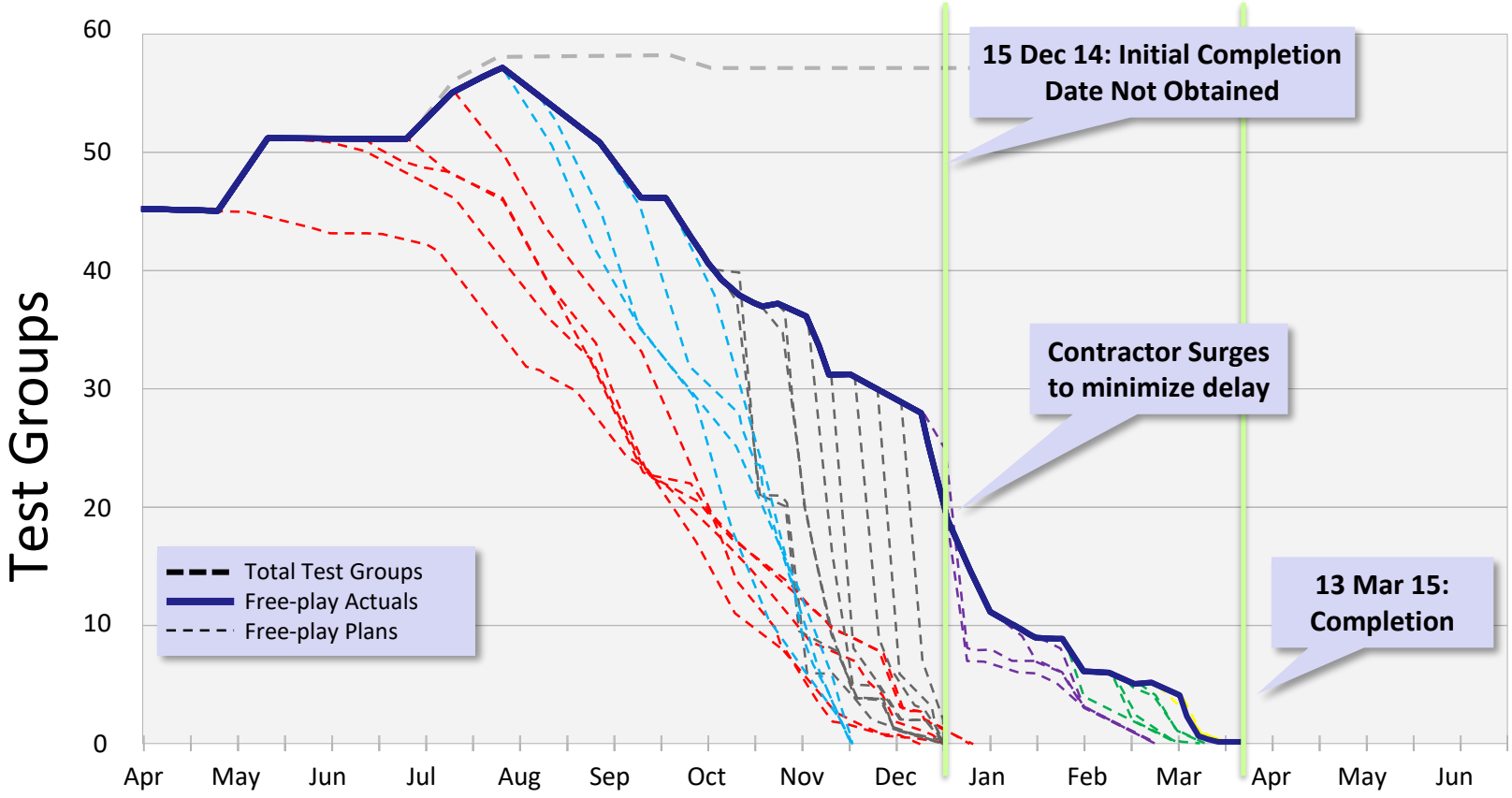


Sample Metrics Testing Optimism



Sample Metrics Testing Optimism

Summary



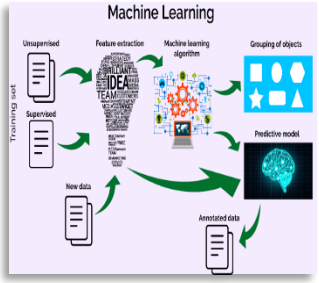
Opportunities

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

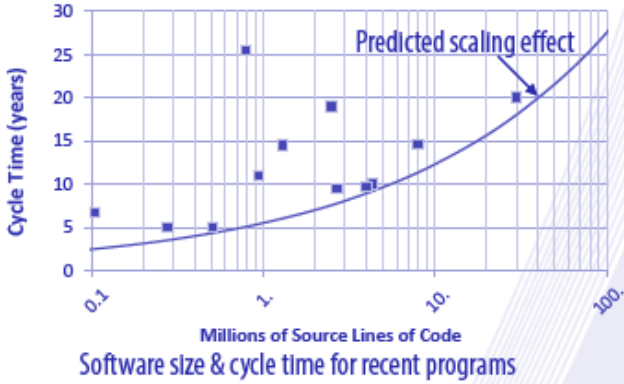
Challenges

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



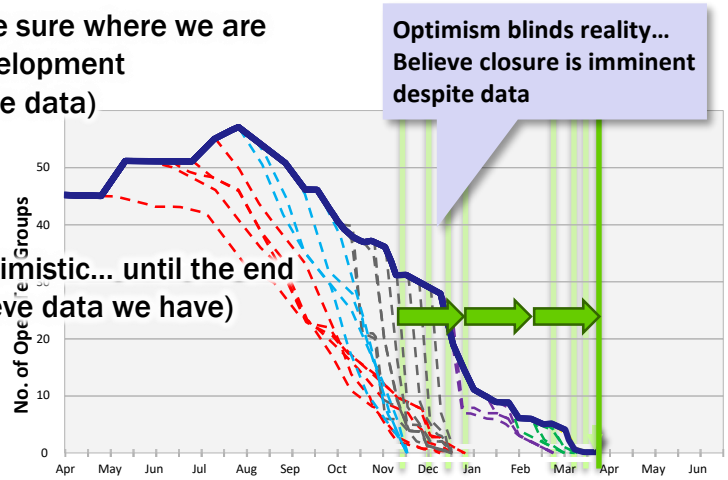
Software size & cycle time for recent programs

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.

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

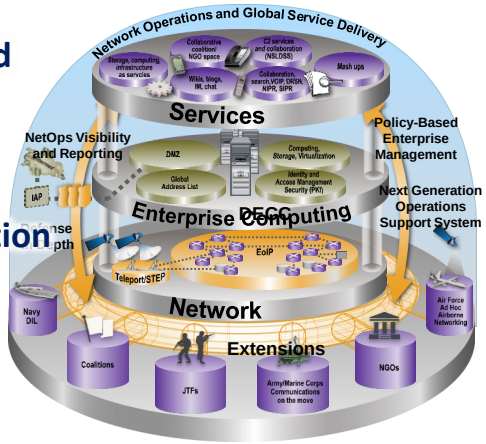
Impossible to exhaustively test systems

How do we establish and maintain trust in SW systems?

Grand architectures

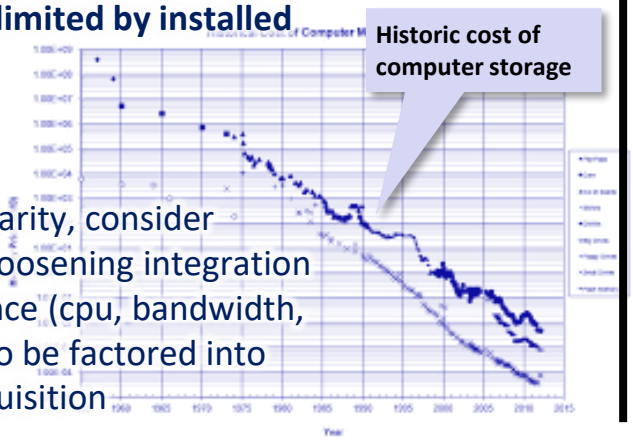
Grand scales and unified architectures hamper timely solutions

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



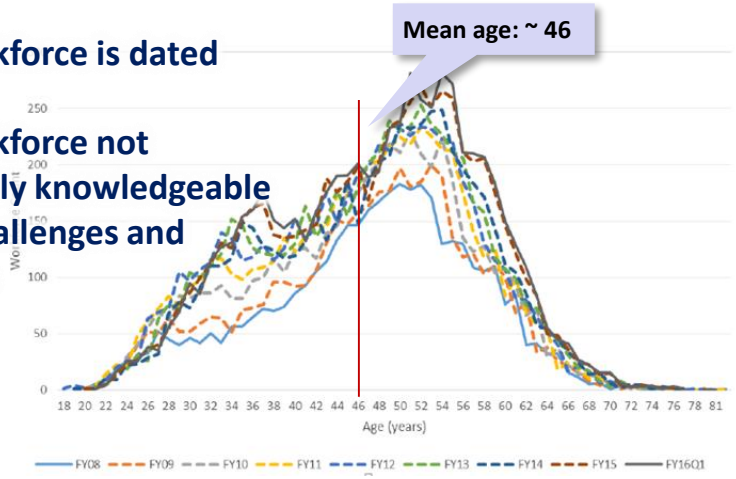
Next Gen:

- 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



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

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

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

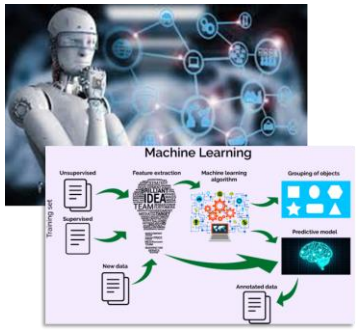
Addressing the Challenge!

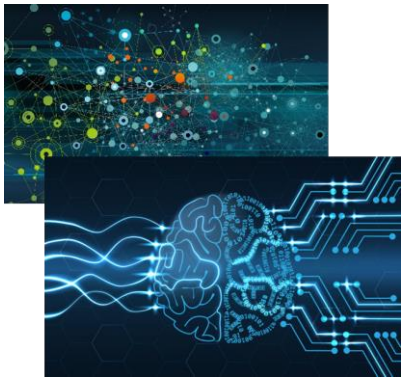
Valuing Velocity

Challenge/Opportunity	Approach
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

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 <u>threat-based acquisition</u> 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/chieftechnologist/

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