

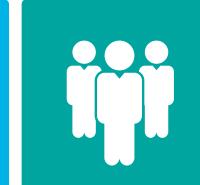


# The Wheels of Change

Department of Defense is implementing change in a big way.



Lockheed Martin. Your Mission is Ours.



# Lockheed Martin



- Headquartered in Bethesda Maryland
- ~100,000 Employees
- Defense Contractor
- 5 Lines of Business

# Journey



- Began Agile Journey in 2002
- Began DevOps Journey in 2012
- Forth year presenting at DOES
- Lockheed Martin This Year
  - Lockheed Martin Software Dojo 2018
  - Government changes

# Wheels of Change

## National Defense Strategy



<https://www.defense.gov/Portals/1/Documents/pubs/2018-National-Defense-Strategy-Summary.pdf>

## Software Productivity Trends and Issues (IDA)

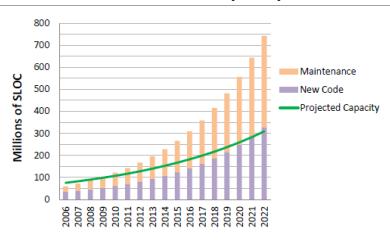


Figure 4. Forecast Supply vs. Unconstrained Demand

[https://www.ida.org/idamedia/Corporate/Files/Publications/IDA\\_Documents/CARD/2017/D-8367.pdf](https://www.ida.org/idamedia/Corporate/Files/Publications/IDA_Documents/CARD/2017/D-8367.pdf)



## Contracting Strategy for F-22 Modernization

<http://www.dodig.mil/reports.html/Article/1475333/contracting-strategy-for-f-22-modernization/>

## Defense Innovation Board



DEFENSE INNOVATION BOARD

<https://innovation.defense.gov/>

### Recommendations

Defense Innovation Board  
Ten Commandments of Software

## Defense Digital Service (DDS)



<https://www.dds.mil/>

## Defense Innovation Unit Experimental (DIUx)



<https://www.diuix.mil/>

## Defense Science Board (DSB) Task Force on the Design and Acquisition of Software for Defense Systems



[https://www.acq.osd.mil/dsb/reports/2010s/DSB\\_SWA\\_Report\\_FINAL\\_delivered2-21-2018.pdf](https://www.acq.osd.mil/dsb/reports/2010s/DSB_SWA_Report_FINAL_delivered2-21-2018.pdf)



James N. Mattis  
Secretary of Defense



Patrick M. Shanahan  
Deputy Secretary of Defense



Ellen M. Lord  
USD (A&S)



Dr. Michael D. Griffin  
USD (R&E), CTO

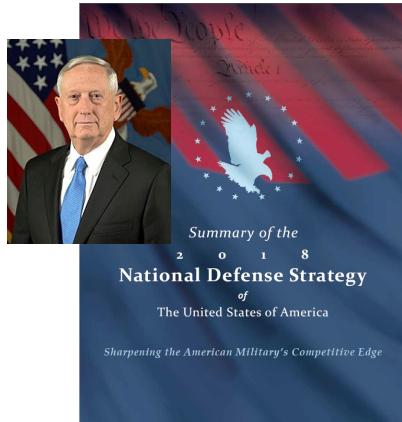


Jeff Boleng  
SEI, CTO  
Special Ass't for  
Software Acquisition



Kristen Baldwin  
Mission Engineering  
and Integration

# Wheels of Change National Defense Strategy



<https://www.defense.gov/Portals/1/Documents/pubs/2018-National-Defense-Strategy-Summary.pdf>

- Build a More Lethal Force
    - Modernize key capabilities  
..., space, cyberspace, C4ISR, ...
- *Advanced autonomous systems.* The Department will invest broadly in military application of autonomy, artificial intelligence, and machine learning, including rapid application of commercial breakthroughs, to gain competitive military advantages.

- Strengthen Alliances and Attract New Partners
- Reform the Department for Greater Performance and Affordability

"The current bureaucratic approach, centered on exacting thoroughness and minimizing risk above all else, is proving to be increasingly unresponsive."

"Deliver performance at the speed of relevance"

"Our response will be to prioritize speed of delivery, continuous adaptation, and frequent modular upgrades."

"Streamline rapid, iterative approaches from development to fielding."

"A rapid, iterative approach to capability development will reduce costs, technological obsolescence, and acquisition risk."

# Wheels of Change USD (A&S)

**Software is the “thread that runs through all our programs. It’s the functional area that I have focused on.”**

**“both the department and industry are behind the curve in terms of modernization of software practices.”**

**“I believe we are at an inflection point in terms of doing things differently. We are pivoting from the traditional waterfall software development methodology to agile and DevOps. So we are coding every day, testing every night.”**



**Hon. Ellen M. Lord**  
Under Secretary of Defense for  
Acquisition, Technology, and Logistics



**Jeff Boleng**  
SEI, CTO  
Special Assistant, Software  
Acquisition

# Wheels of Change DSB Software Task Force

Defense Science Board (DSB) Task Force on  
the Design and Acquisition of Software for  
Defense Systems

(Feb 2018):

"The task force concluded that the Department needs to change its internal practices to encourage and incentivize new practices in its contractor base. The assessment of the Task Force is that the Department can leverage best practices of iterative development even in its mission critical software systems."

[https://www.acq.osd.mil/dsb/reports/2010s/  
DSB\\_SWA\\_Report\\_FINALdelivered2-21-2018.pdf](https://www.acq.osd.mil/dsb/reports/2010s/DSB_SWA_Report_FINALdelivered2-21-2018.pdf)



OFFICE OF THE SECRETARY OF DEFENSE  
3140 DEFENSE PENTAGON  
WASHINGTON DC 20301-3140

MEMORANDUM FOR UNDER SECRETARY OF DEFENSE FOR RESEARCH AND ENGINEERING

SUBJECT: Final Report of the Defense Science Board (DSB) Task Force on the Design and Acquisition of Software for Defense Systems

I am pleased to forward the final report of the DSB Task Force on the Design and Acquisition of Software for Defense Systems, chaired by Dr. William LaPlante and Dr. Robert Wisniewski.

The Task Force has made seven recommendations on how to improve software acquisition in defense systems. A base recommendation underlying all others is to emphasize the importance of the software factory and to incorporate the software factory as a key evaluation criterion in the source selection process. Next, the Department of Defense (DoD) and its defense industrial base partners need to adopt continuous iterative development best practices. The study recommends DoD adopt best practices on risk reduction and metrics in formal program acquisition strategies. Software strategies must be better incorporated in current and legacy programs from development, production, and sustainment. The Task Force recommends ways to improve the software and acquisition workforce, in both software development expertise and the broader functional acquisition work force. Next, software is immortal and contracts must be framed to allow for software sustainment. Finally, the Task Force recommends further research into machine learning and the implementation of an independent verification and validation process for machine learning and autonomy in software systems.

Software is a crucial and growing part of weapons systems and the Department needs to be able to sustain immortal software indefinitely. The Task Force concluded that the Department of Defense would benefit from the implementation of continuous iterative development best practices as software becomes an increasingly important part of defense systems.

I concur with the Task Force's conclusions and recommend you forward the report to the Secretary of Defense.

A handwritten signature in black ink, appearing to read "Craig Fields".

Dr. Craig Fields  
Chairman, DSB

# Initiatives

# DSB Recommendations

Recommendations	Description
<b>Software Factory:</b>	Integrated Tool Set, Process Blue Prints, and Technical skills to deliver software continuously.
<b>Continuous Iterative Development:</b>	Decomposing work in order to deliver Minimum Viable Product (MVP) and set of incremental Next Viable Products (NVP).
<b>Risk Reduction and Metrics for New Programs:</b>	Leverage metrics that focus on outcomes such as burn downs, velocity, and control charts
<b>Current and Legacy Programs in Development, Production, and Sustainment:</b>	Prime contractors should transition execution within contractual constraints. Incorporate continuous iterative development into a long-term sustainment plan.
<b>Invest in the Work Force:</b>	The U.S. Government does not have modern software development expertise in its program offices or the broader functional acquisition workforce. This requires Congressional engagement and significant investment immediately.
<b>Software is Immortal: Software Sustainment :</b>	All documentation, Test files, software code, design documents, results of fault and performance tests should be delivered to government.
<b>IV&amp;V for Machine Learning:</b>	Establish a machine learning/autonomy data repository, collect and share data.

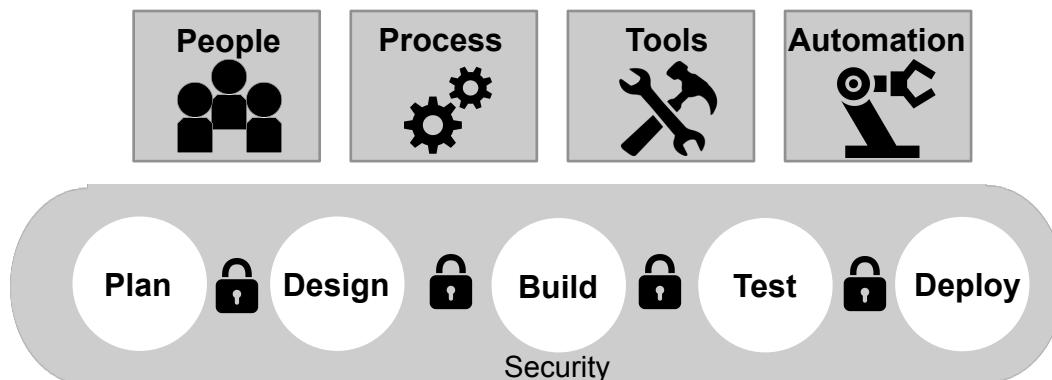


# Facilitating Collaborative Transformation



# Lockheed Martin Approach Software Factory

- People
- Process
- Tools (integrated tool chains)
- Automation

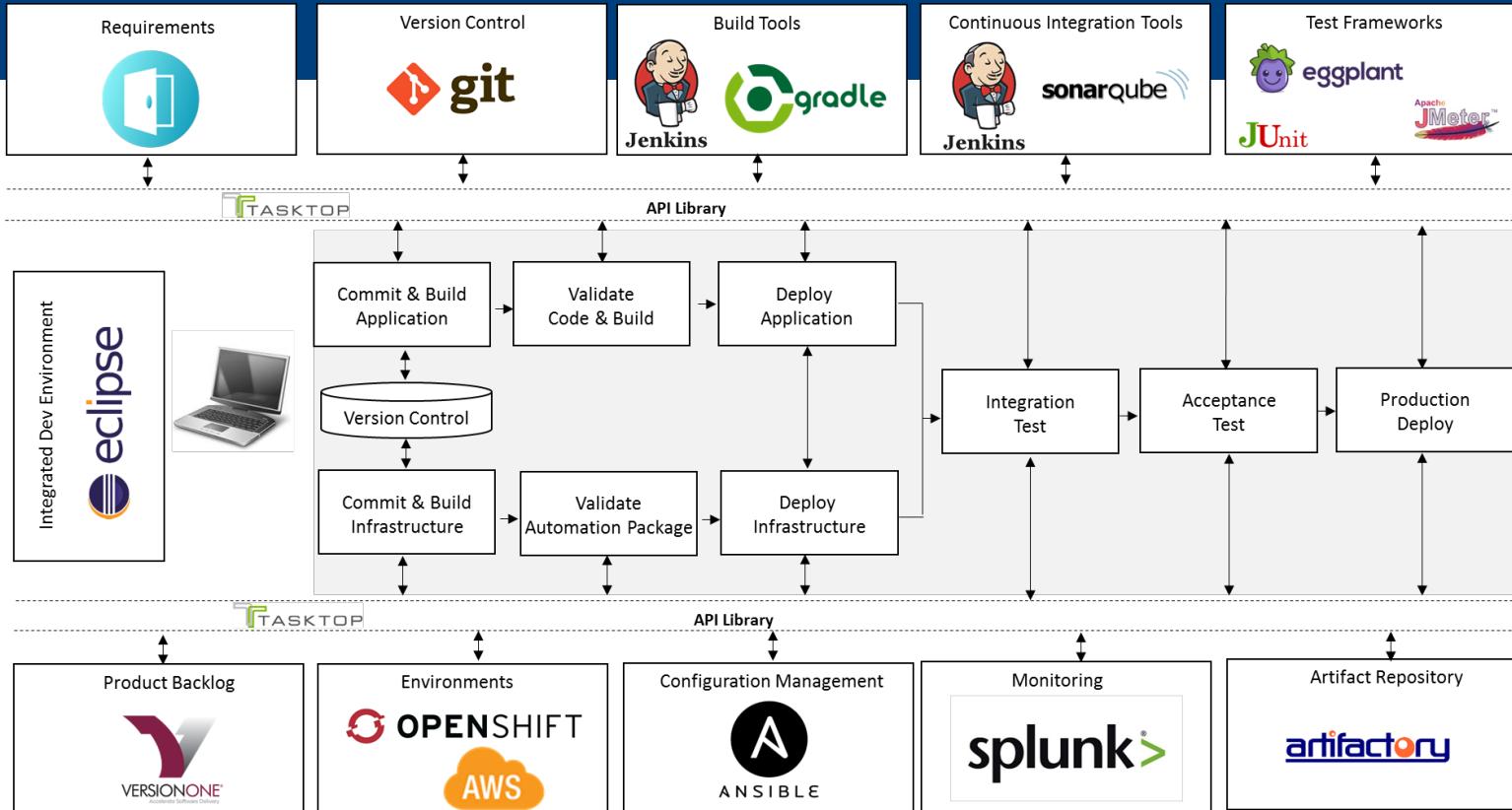


## DSB Recommendation

- ✓ **Software Factory**
- Continuous Iterative Development
- Risk Reduction Metrics
- Transition current and Legacy
- Invest in the workforce
- Software Sustainment
- IV&V for Machine Learning

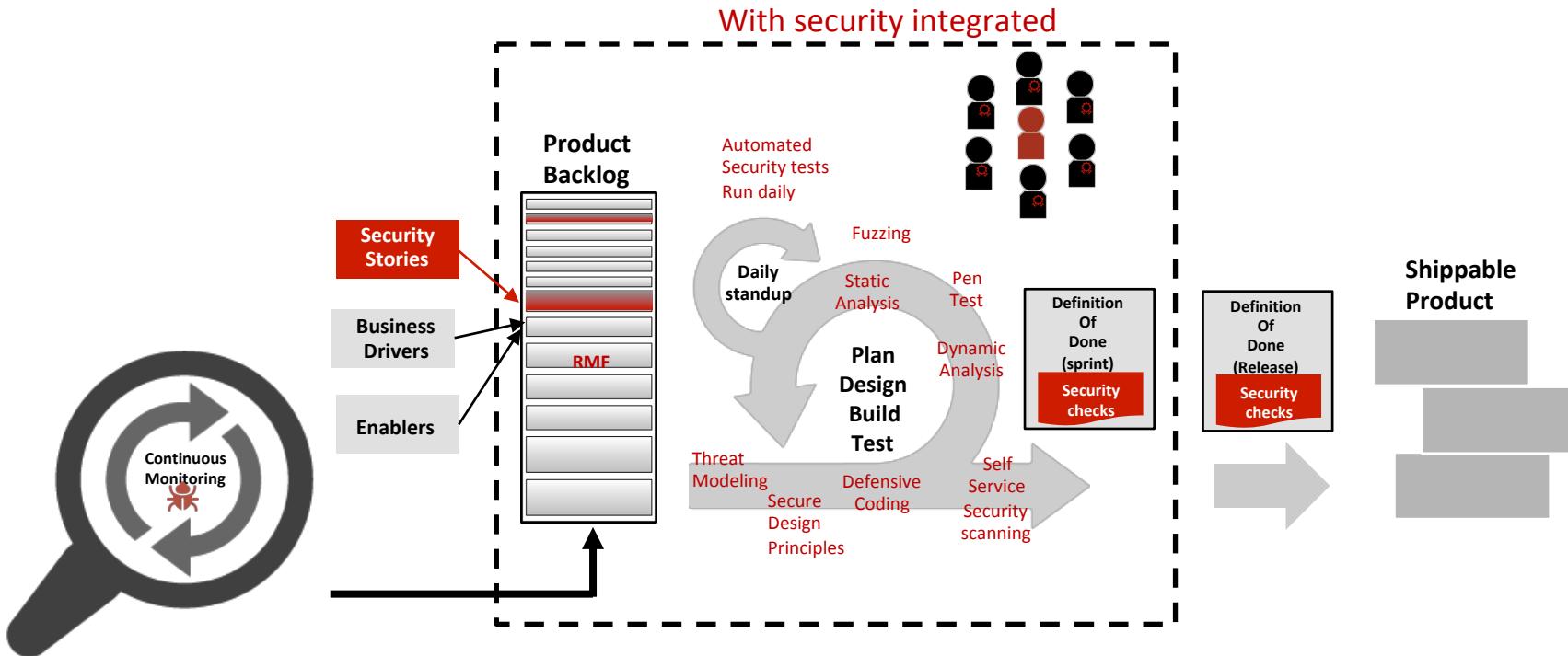
# Initiatives

# Example Software Factory



# Initiatives

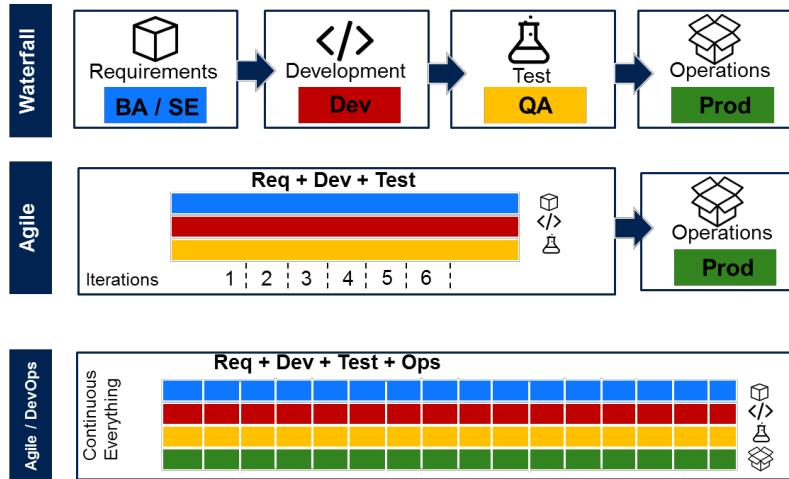
# Secure Software Factory



# Lockheed Martin Approach

## Continuous Iterative Development

- Migrating Legacy programs a step at a time
- Software Dojo and starter kits



## DSB Recommendation

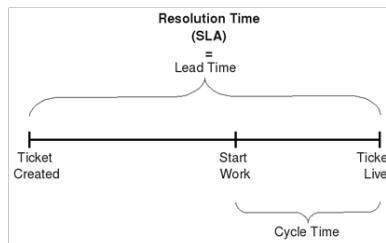
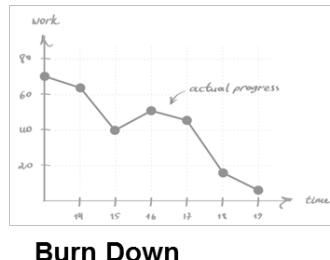
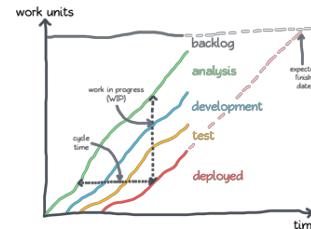
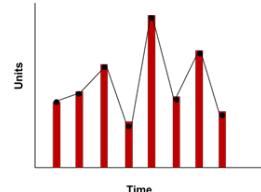
- ✓ **Software Factory**
- ✓ **Continuous Iterative Development**
- Risk Reduction Metrics
- Transition current and Legacy
- Invest in the workforce
- Software Sustainment
- IV&V for Machine Learning

# Lockheed Martin Approach Metrics

- Looking at different Metrics
- Activity based Accounting



vs



## DSB Recommendation

- ✓ Software Factory
- ✓ Continuous Iterative Development
- ✓ Risk Reduction Metrics
- Transition current and Legacy
- Invest in the workforce
- Software Sustainment
- IV&V for Machine Learning

# Lockheed Martin Approach **Incrementally Transition programs**

Many have already begun the journey, others  
are just starting

- F-35
- F-16
- F-22
- Aegis
- Orion

## DSB Recommendation

- ✓ Software Factory
- ✓ Continuous Iterative Development
- ✓ Risk Reduction Metrics
- ✓ Transition current and Legacy
  - Invest in the workforce
  - Software Sustainment
  - IV&V for Machine Learning

# Experience F-16

## Lessons Learned

- Improved product definition through increased collaboration
- Improved project maturity through incremental releases
- Improved capability through early feedback from stakeholders

## Single Greatest Benefit

- Greater collaboration between agile and program teams by scaling agile with SAFe

## Program Info

Teams	15 - 18
Practices	Scrum, Kanban, XP Paired Programming
Scaled Approach	SAFe
Tools	VersionOne, Jenkins, Sonar Cube
Agile Architecture	No
Test Automation	No
Metrics	Velocity, Capacity, Flow, WiP, Burn Down

# Experience F-22

## Lessons Learned

- Product-centric cross-functional teams decreased dependencies and enabled collaboration
- Pair-programming work cells for instantaneous product review
- Modern tool stack and pipelines maximized feedback
- DDD and TDD for bounded context between teams

## Single Greatest Benefit

- Increased team and team-of-teams collaboration through user-centric outcomes

## Program Info

Team	~30
Practices	Scrum, Kanban, XP, Pair Programming, DDD, TDD
Scaled Approach	SAFe
Tools	Openshift, PCF, VersionOne, Jira
Agile Architecture	No
Test Automation	No
Metrics	Velocity Delivery Tracking Capability Maturity

# Experience F-35

## Lessons Learned

- Necessity of cross-functional teams
- Increased communication in collaborative sessions
- Value of VersionOne training
- Backlog creation/grooming habits
- Work awareness and prioritization

## Single Greatest Benefit

- Increased planning, communication, and stakeholder involvement through PI planning, work decomposition, and V1 Tracking

Program Info	
Teams	15
Practices	Scrum, Scumban
Scaled Approach	SAFe
Tools	Jira
Agile Architecture	No
Test Automation	Yes
Metrics	Cumulative Flow Earned Value SLOC, Defects

# Experience Aegis

## Lessons Learned

- Automate manual events
- Rapid defect identification through comprehensive test cycles
- Embedding customer in regular reviews
- Continuous Integration and Continuous Testing

## Single Greatest Benefit

- Improve agile planning and execution by moving to a flow based system to manage CSL scaled

## Program Info

Developers	1,275
Teams	157
Practices	Scrum, Kanban
Scaled Approach	SAFe, SAM
Tools	Jira
Agile Architecture	Yes
Test Automation	Yes
Metrics	Velocity, Burndown, Cycle Time

# Experience Orion

## Lessons Learned

- Setup tools ahead of time
- Offset testing and verification teams
- Initial training & Ongoing Training
- Experienced Agile Coaches

## Single Greatest Benefit

- Communication – Release planning events revealed dependencies and resolved issues ahead of production

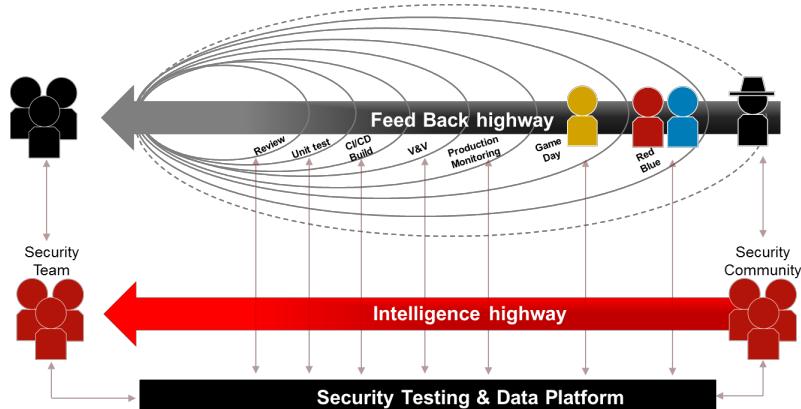
## Program Info

Teams	35
Practices	Scrum, Kanban
Scaled Approach	SAFe
Tools	Jira w/Plugins
Agile Architecture	Yes
Test Automation	Yes
Metrics	Velocity Burn Up/Down, Defect Density, SLOC, Work Added

# Lockheed Martin Approach

## Rethink Software Sustainment

- Platforms
- Design Patterns
- Telemetry
- Factory



## DSB Recommendation

- ✓ Software Factory
- ✓ Continuous Iterative Development
- ✓ Risk Reduction Metrics
- ✓ Transition current and Legacy
- ✓ Invest in the workforce
- ✓ Software Sustainment
- IV&V for Machine Learning

# Lockheed Martin Approach

## Invest in Workforce

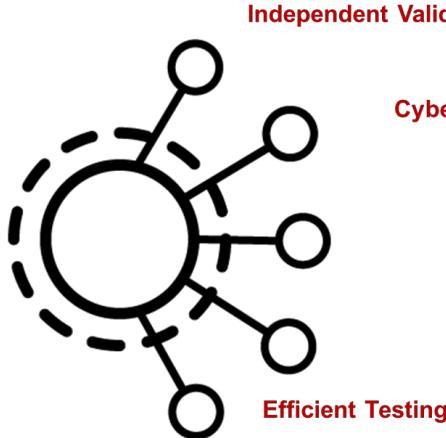
- Communities of Excellence
- Team Safaris
- Mentors
- Pairing
- Improved Tuition Strategy
- Videos
- Hackathons
- Shark Tanks

## DSB Recommendation

- ✓ Software Factory
- ✓ Continuous Iterative Development
- ✓ Risk Reduction Metrics
- ✓ Transition current and Legacy
- ✓ Invest in the workforce
- Software Sustainment
- IV&V for Machine Learning

# Lockheed Martin Approach Machine Learning

- Research
- Hackathons



## DSB Recommendation

- ✓ Software Factory
- ✓ Continuous Iterative Development
- ✓ Risk Reduction Metrics
- ✓ Transition current and Legacy
- ✓ Invest in the workforce
- ✓ Software Sustainment
- ✓ IV&V for Machine Learning

**LOCKHEED MARTIN**

