National Defense Industrial Association Systems and Mission Engineering Conference

November 2022

Systems Engineering Transformation in a Digital World

Mr. Allan Dianic

Director Software Engineering

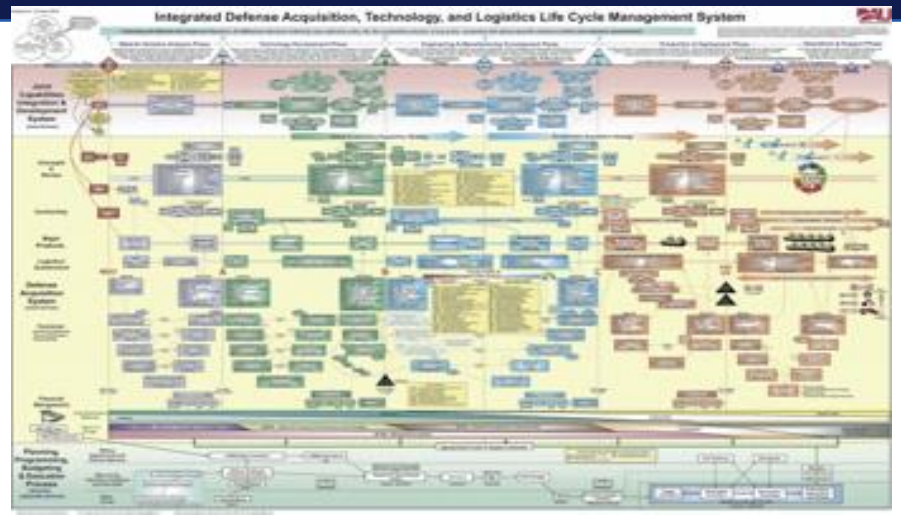
Office of the Executive Director, Systems Engineering and Architecture

Office of the Under Secretary of Defense for Research and Engineering

As we strive to deliver capability faster …

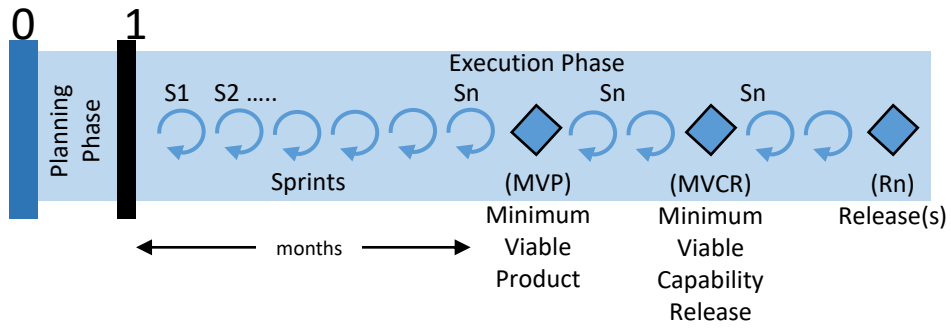
Software Engineering Is Changing

DoDI 5000.02 milestones, models, and documentation did not provide the proper structure for managing software development.



"Going forward, DoD should shift away from its traditional linear system development process and adopt a nimbler approach to iteratively design, test, and fix systems. ,"
-The Honorable Heidi Shyu, USD, R&E

The new way facilitates rapid and iterative development of software – systems engineering is still essential but must be more agile

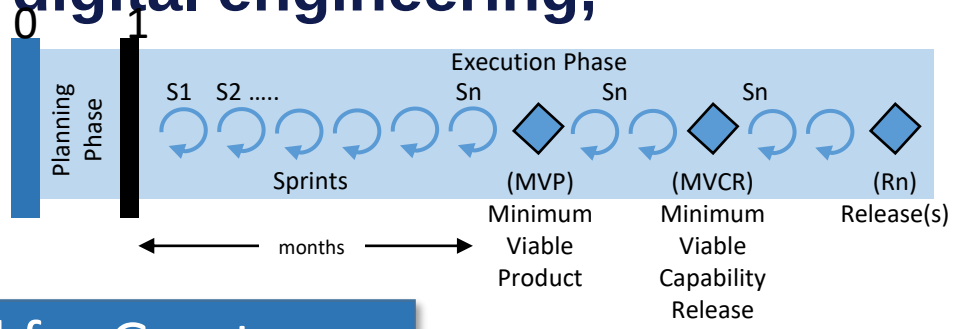


OLD WAY

NEW WAY

Why Rapid Iterative Approaches

- **Promise to field new capability faster**
- **Strategic guidance and directives from leadership**
- **Automated software development pipelines encourage iteration**
- **SWE technology advances** **Containerization, Micro Services, Zero Trust, Cloud, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Continuous Integration Continuous Delivery (CI/CD)**
- **Artificial intelligence/machine learning, digital engineering, models and simulations**
- **Cybersecurity needs rapid adaptation**



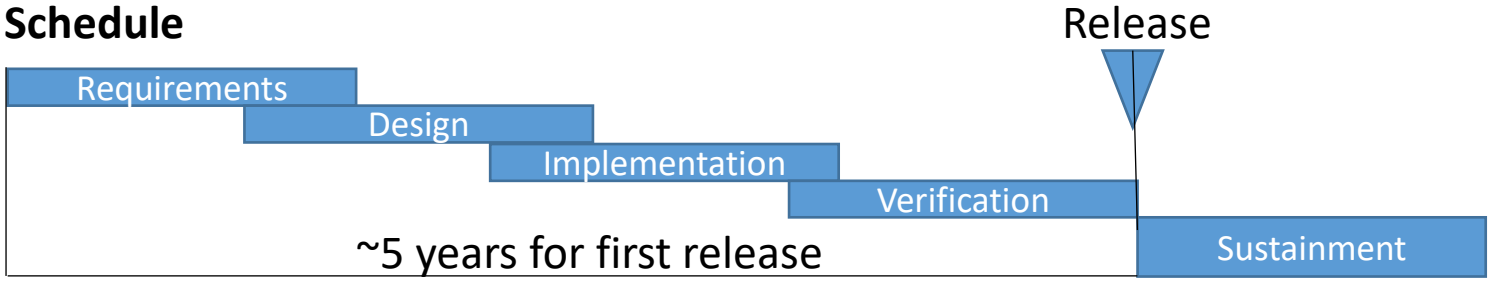
Digital Engineering Needed for Greater Speed and Flexibility

Systems and Software Engineering – Legacy Approach

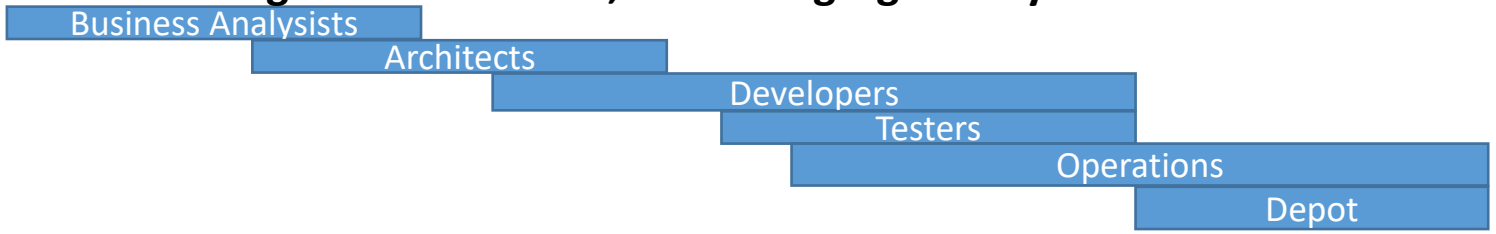
Waterfall

- **Inflexible requirements** –assumes detailed knowledge of needs 5 years in advance.
- Software processes **mimic manufacturing process**
- Schedule or funding short falls result in **technical debt.**
- Work conducted in **silos**
- **Late stage testing/security** scans
- Throw it over the wall mentality
- Delivery often doesn't meet user needs

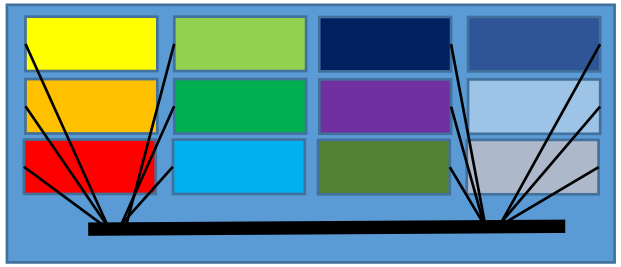
Schedule



Workforce Organization – Silos, Teams Segregated by Phase



Architectural Approach – Monolithic, Modular, n-Tier



- Highly complex
- Brittle
- Difficult to maintain

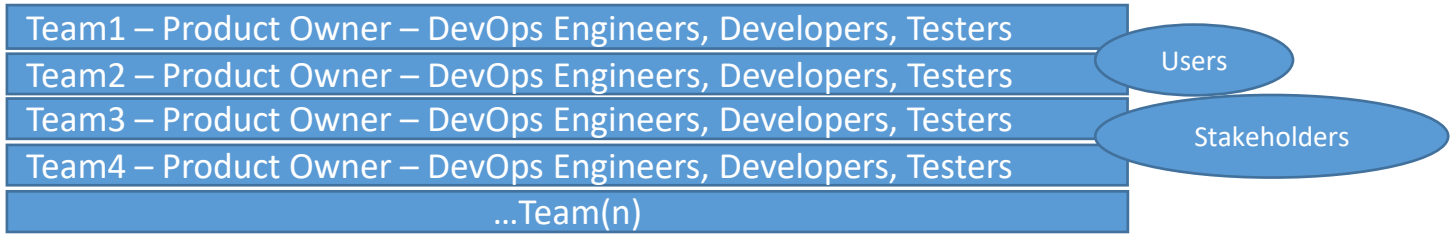
Systems and Software Engineering – New Way

Digital Approach Supporting Agile Software

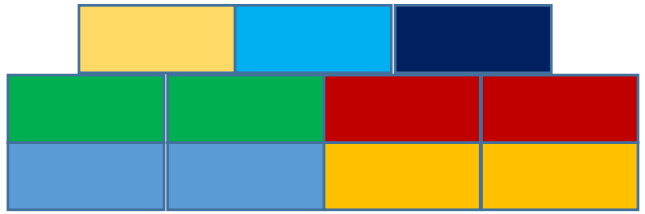
- Software is never done
- **MVP** within **6-12 months**
- **MVCR/CR** every **3-6 months**
- Cross Functional Teams
- Loose Coupling
- **Zero Trust**
- **Continuous ATO**
- **Continuous Sustainment**
- **Cloud-Native** Microservices
  - Scalability
  - Elasticity
  - Reliability
- **Evolutionary Architecture**

Schedule

Workforce Organization – Cross Functional Teams



Architectural Approach - Microservices



MVP = Minimum Viable Product
MVCR = Minimum Viable Capability Release
CR = Capability Release

Backup

New Realities and Digital Technology Advances Driving Change

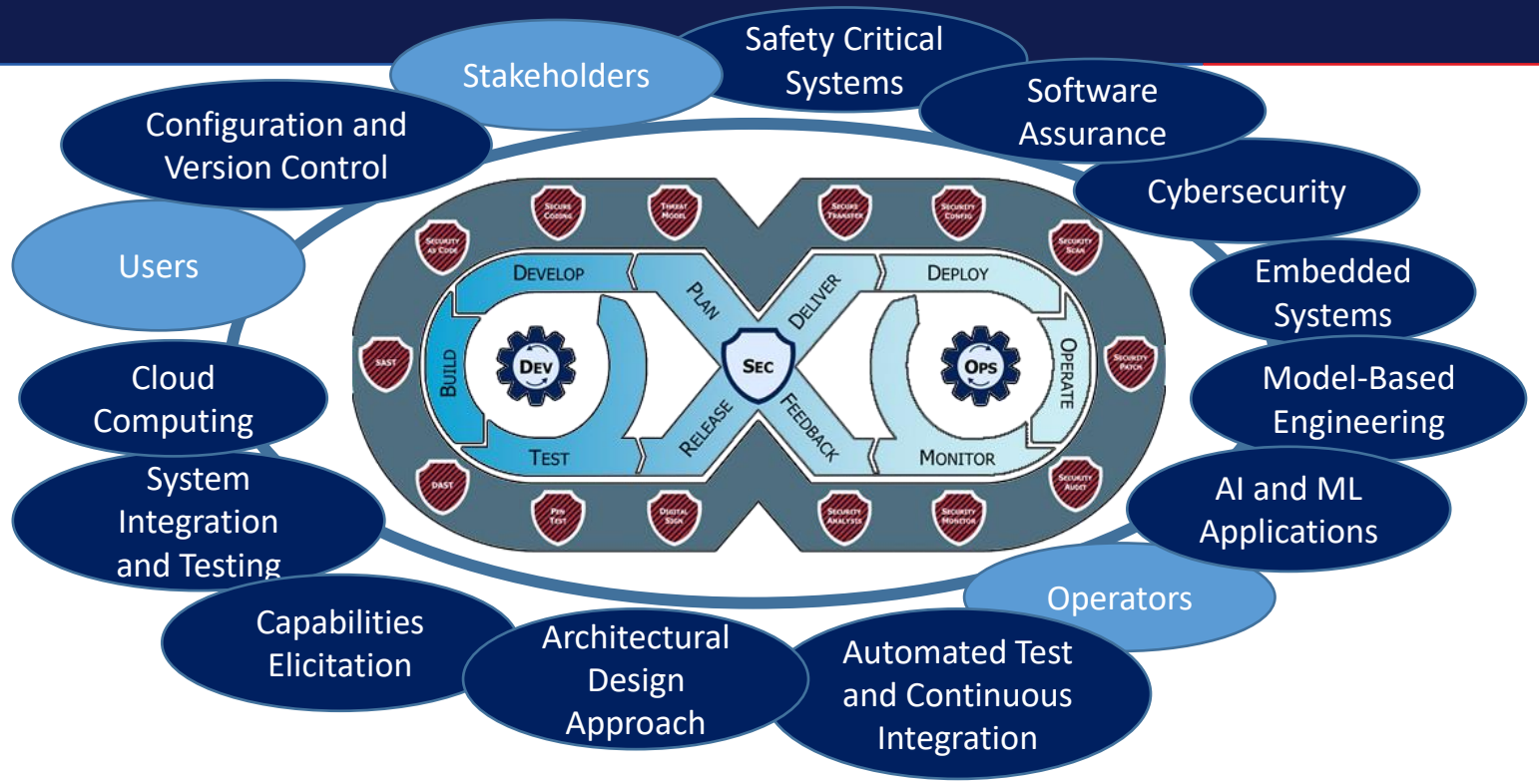
	Old Way	Disadvantages	New Way	Advantages
Policy	Title 10 DoDI 5000.02	Regulatory Requirements Manufacturing Orientation	Adaptive Acq Framework DoDI 5000.87 SW Acq Pathway	Less Bureaucracy Capability Driven Commercial Best Practices Adopt Modern SWE Practices
Method	Document Centric Waterfall	Fixed Requirements Milestone Driven Months/Years Delivery Cycles	Curated Digital Models Continuous Delivery DevSecOps	Days/Weeks Delivery Cycles Rapid Feedback Collaborative Culture Responsive to Needs
Technology	Local Physical Servers	One offs (staff/sched/perf) Dedicated HW infrastructure Slow Tech Insertion Hard to Scale	Cloud Services Containers Microservices	Shared Infrastructure Simpler to Deploy Rapid Tech Insertion Easy to Scale
Security	Perimeter Security Static Trust	Systemic Fragility Poor support for mobile Privilege bound to location Assume secure perimeter	Zero Trust	Systemically Robust Natural Support for Mobile Privilege bound to identity Assume adversary present

DevSecOps Collaborative Culture and Shift Left Strategy Relies on Digital Engineering

People with specific competencies collaborate on the entirety of the pipeline

No competency is restricted to a single phase

Expertise captured and automated into the pipeline whenever possible



Software Competency as defined by R&E sponsored software initiative

https://www.rand.org/pubs/research_reports/RR3145.html