Making Flexible and Modular Happen—An Architecture-Centric Approach

Getting to Plug and Play

Modular adaptable ships need a design cycle for adaptable hardware elements, and the software needed to make those cyber-physical capabilities run. The objective is to have literal "plug-and-play" to rapidly change the ship's mission performance.

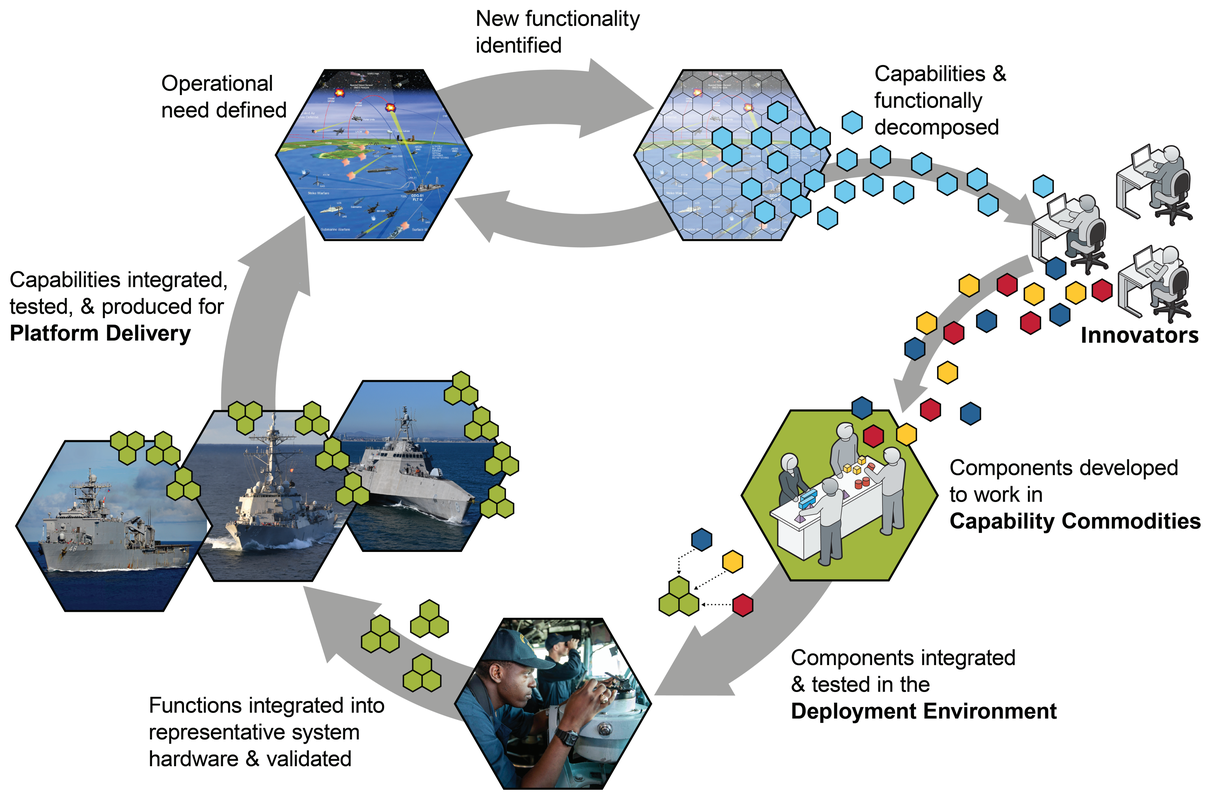
Previously successful practices of the past 20 years are insufficient to achieve the agility needed to deliver new capabilities that meet rapidly evolving needs. To stay ahead of our adversaries we need to increase our pace of change and continuous deliver innovation of fielded capability.

Rethinking Architecture

A new "architecture-centric" strategy establishes the equities of conformant quality attributes and managed variability while sustaining minimally coupled designs and inherently interoperable data. This strategy will requires an enterprise architecture construct in which modules of capability can be evaluated and certified as safe for use on all combatants as soon as they are ready.

- Conventional approaches of acquisition must be rethought at every level; funding, contracting, technical, etc.
- Organize these capabilities into risk-prudently replaceable product commodities
- Use modern and automated methods to assesses quality
- Field reusable capacities that are reliable and rapidly delivered
- Use continuous engineering methods to evolve and modernize products over the life cycle

While components and functions are acquired and delivered as separable mission capability elements, they must be presented to operators in a tightly-integrated manner.



A More Nimble Development Cycle

The cycle would include the following steps:

- **Cyber Secure – all the time as an operational need**
  - Security updates and patches all the time It's needed, we are under attack—all the time
  - Provide fixes to vulnerabilities within hours of being found
- **Identify new functionality**
  - Changes driven by new commercial products or military research results (push)
  - Changes driven by warfighting need (pull)
- **Decompose capabilities and functionally decomposed**
  - Capabilities are functionally decomposed so they can be packaged as deployable modules
  - Using an *"architecture-centric"* strategy to establish rules of construction to ensure quality attributes (security, loose coupling, integratability, testability, performance, etc.) are in the final integrated system.
- **The software framework enables common features to be created once and reused across multiple capabilities**
  - Virtualization and componentization using modern commercial architectures
- **Let innovators attack and solve technical issues**
  - Leverage commercial organizations,
  - DoD contractors, academia, Gov't labs and small business to participate equally (innovate and deliver expertise)
- **Develop components to work as Capability Commodities**
  - Collaboratively derived and government-led architecture enables innovative Capability
  - Commodities to be delivered quickly across all platforms
- **Integrate and test components rapidly**
  - Establish the Deployment Environment to create a high delivery tempo (i.e. DevOps)
- **Government needs to own the baseline**
  - Make sure we have the authority and ability to deploy the capability to warfighters
- **Integrate, test, and produce capabilities for Platform Delivery**
  - Reuse common capability and incorporate unique performance
  - Certifying and accrediting modules individual enable faster certification of the final system

Containerization and micro-services help reduce development risk, and increase overall product robustness. They can be combined with Agile and "DevOps" methods to bring flexibility and speed to delivering capability.

- **Containerization is an operating system (OS) feature that supports isolated user-space instances that enable the execution of distributed applications without launching an entire virtual machine (VM). Containerization can turn multiple components into a robust application.**
- **Micro-services are an evolution of the service-oriented architecture (SOA) architectural style that structures an application as a collection of loosely coupled fine-grained services connected via lightweight protocols. Modular capabilities implemented as micro-services more efficiently use multiple cores, clouds technology, storage evolutions, etc.**

Distribution Statement A: Approved for Public Release;
Distribution is Unlimited

Harry L Levinson | hll@sei.cmu.edu