

DAU

Software Relevancy

Garth.McMurray@DAU.edu



“Have you ever walked into a defense contractor and watched us build software? Well, it’s just a nightmare. And if you look at every one of our programs, it’s a nightmare across the board. Why, because the United States doesn’t know how to build software?

We’re the leading software nation in the world. We just haven’t translated that into the Department of Defense.”

- Gen Hyten, Vice Chairman Joint Chiefs of Staff
2019-2021

Operational Relevancy

The Department of Defense prepares for war – for the sake of peace.

Acquisitions must be optimized for war-time, not peace-time.

Capabilities must be:

1. Delivered quickly (OODA Loop)
2. Effective, Suitable and Survivable



“We have a single purpose, really, which is to -- either to prepare for war or to fight a war, and we are laser-focused on that.”

SECDEF Lloyd Austin

“Speed of Relevance”



“The acquisition system & culture must adapt to the reality that hardware & software systems must **change** on a frequent basis to meet warfighter needs, adapting to the **speed of relevance.**”

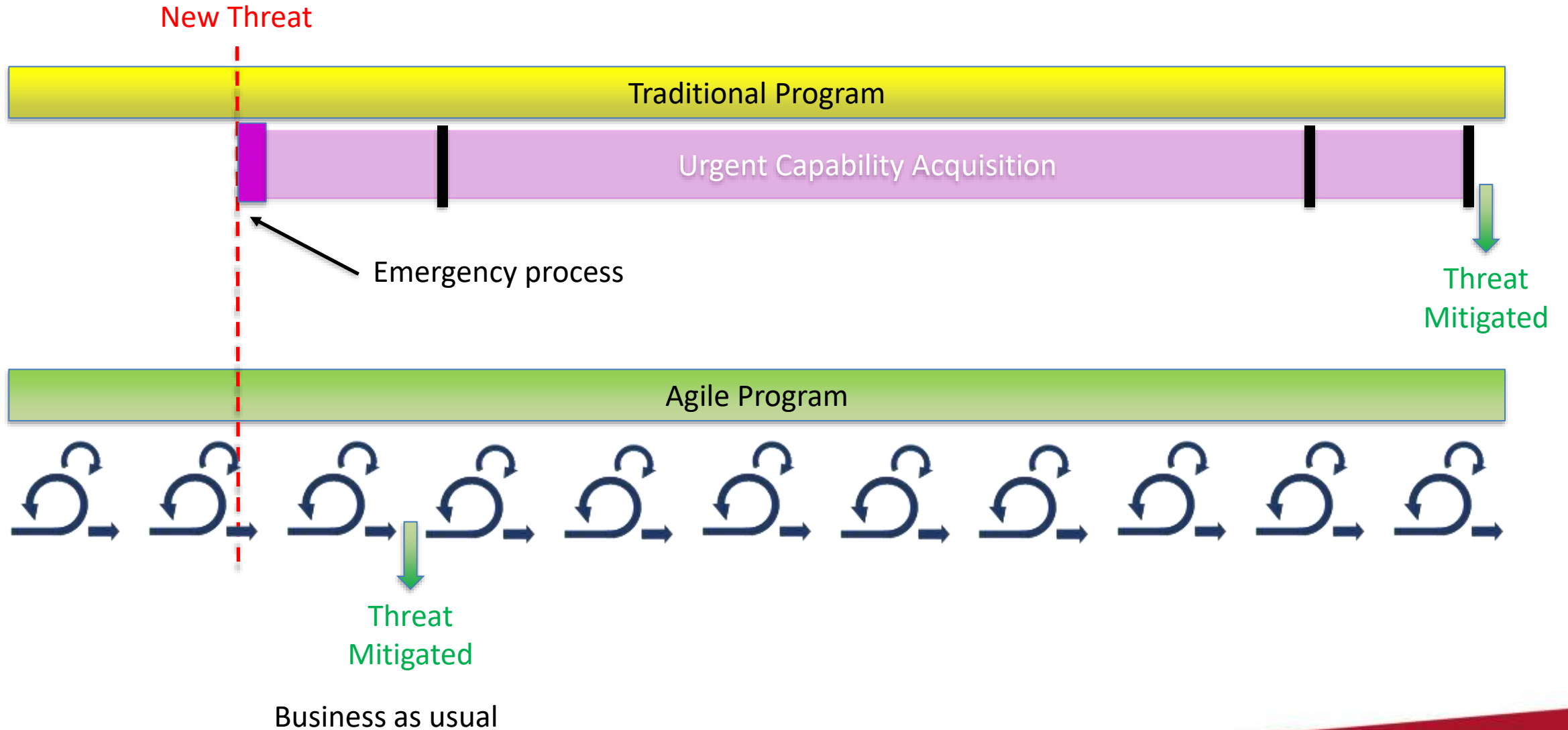
– Former United States Secretary of Defense James Mattis

“The department will be innovative at a **speed** and scale that matches a **dynamic threat** landscape”

– Current United States Secretary of Defense Lloyd Austin



How Speed Works



“Speed of Relevance” Example

Ukrainian forces used SpaceX’s Starlink for high-speed remote internet access



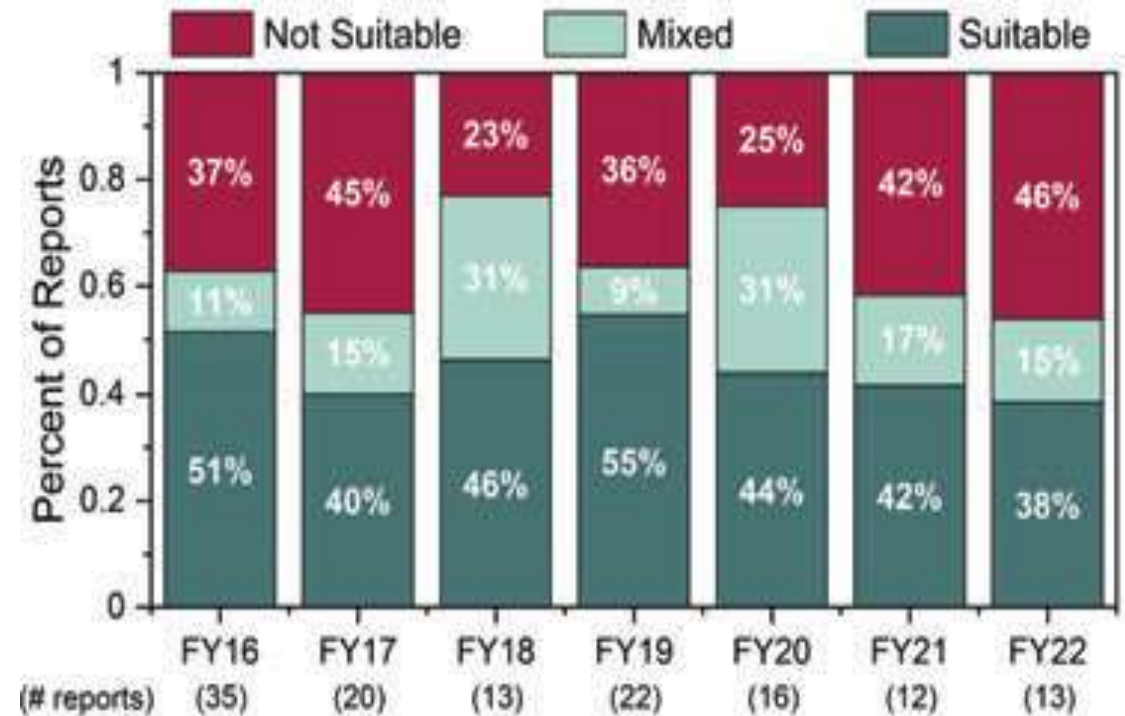
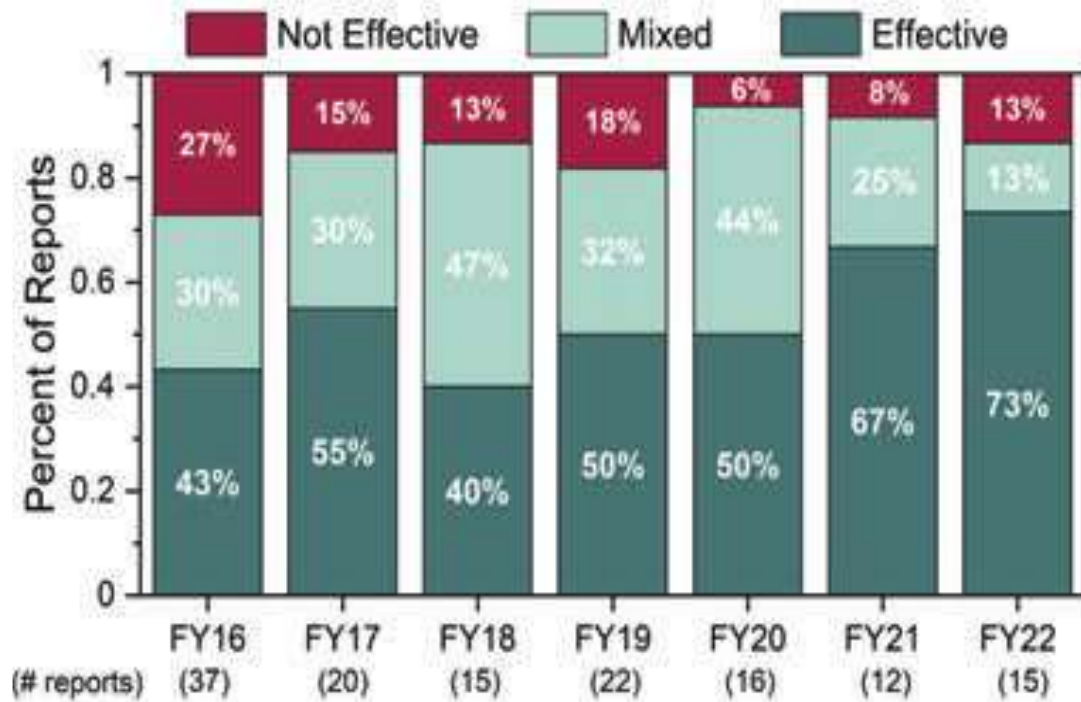
Russian Forces used Electronic Warfare assets to jam Ukrainian Starlink capabilities

In **ONE DAY** SpaceX created and deployed a software solution which defeated Russian jamming!



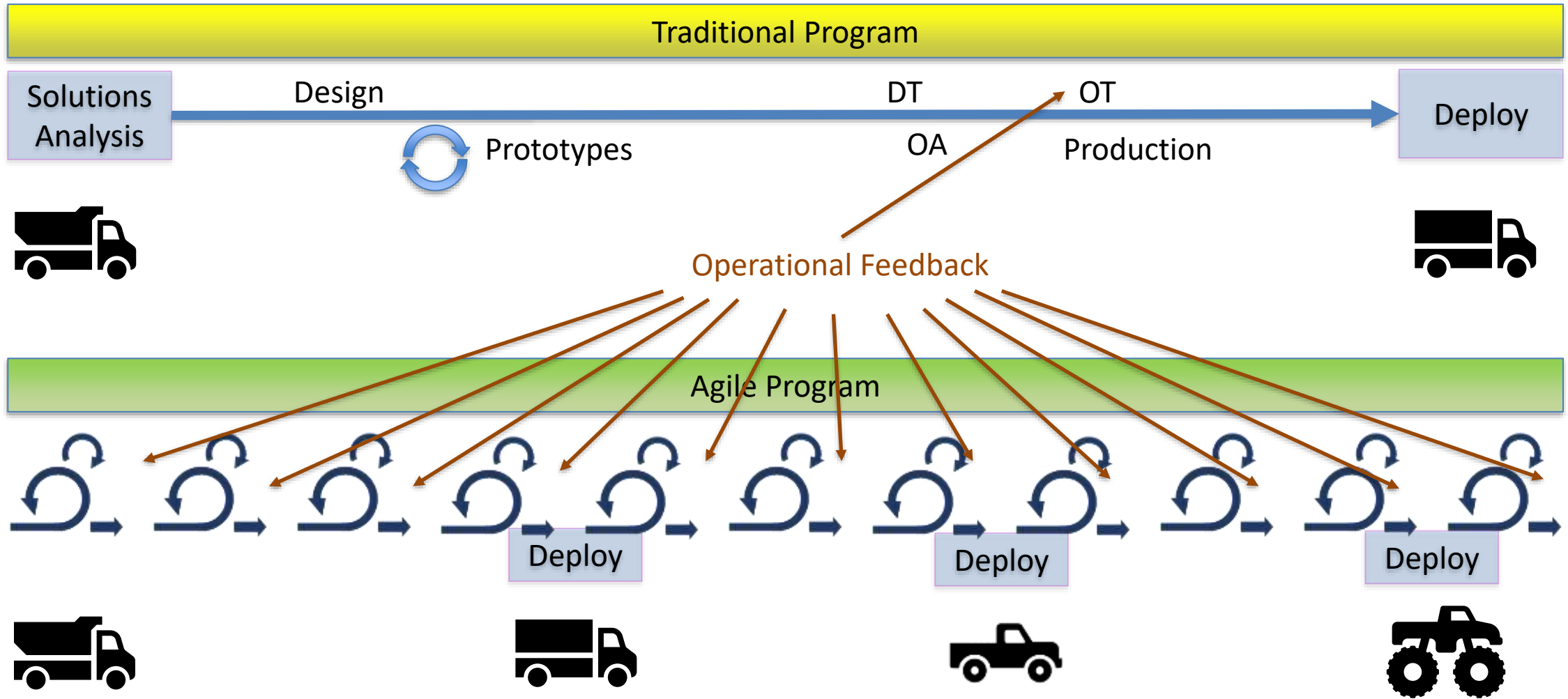
Effective and Suitable

Here's how our systems are doing in OT:



<https://www.dote.osd.mil/Annual-Reports/2022-Annual-Report/>

Relevant Systems



Are teams delivering working software to at least some subset of real users every iteration (including the first) and gathering feedback?
– DIB Guide: Detecting Agile BS

Start Small, Build on Success

- Do **NOT** slowly evolve to Agile!!

Hybrid is High Risk

- Traditional processes and Agile processes are oil and water – they don't mix
- Fundamentally and philosophically, they approach acquisition in different ways:
 - Requirements
 - Contracts
 - Team composition
 - Metrics and measures
 - Management / control
 - Test
 - Documentation
 - Timeline and funding
 - Sustainment
 - Deployment



“Is the full ecosystem of your project agile? (Agile programming teams followed by linear, bureaucratic deployment is a failure.)”

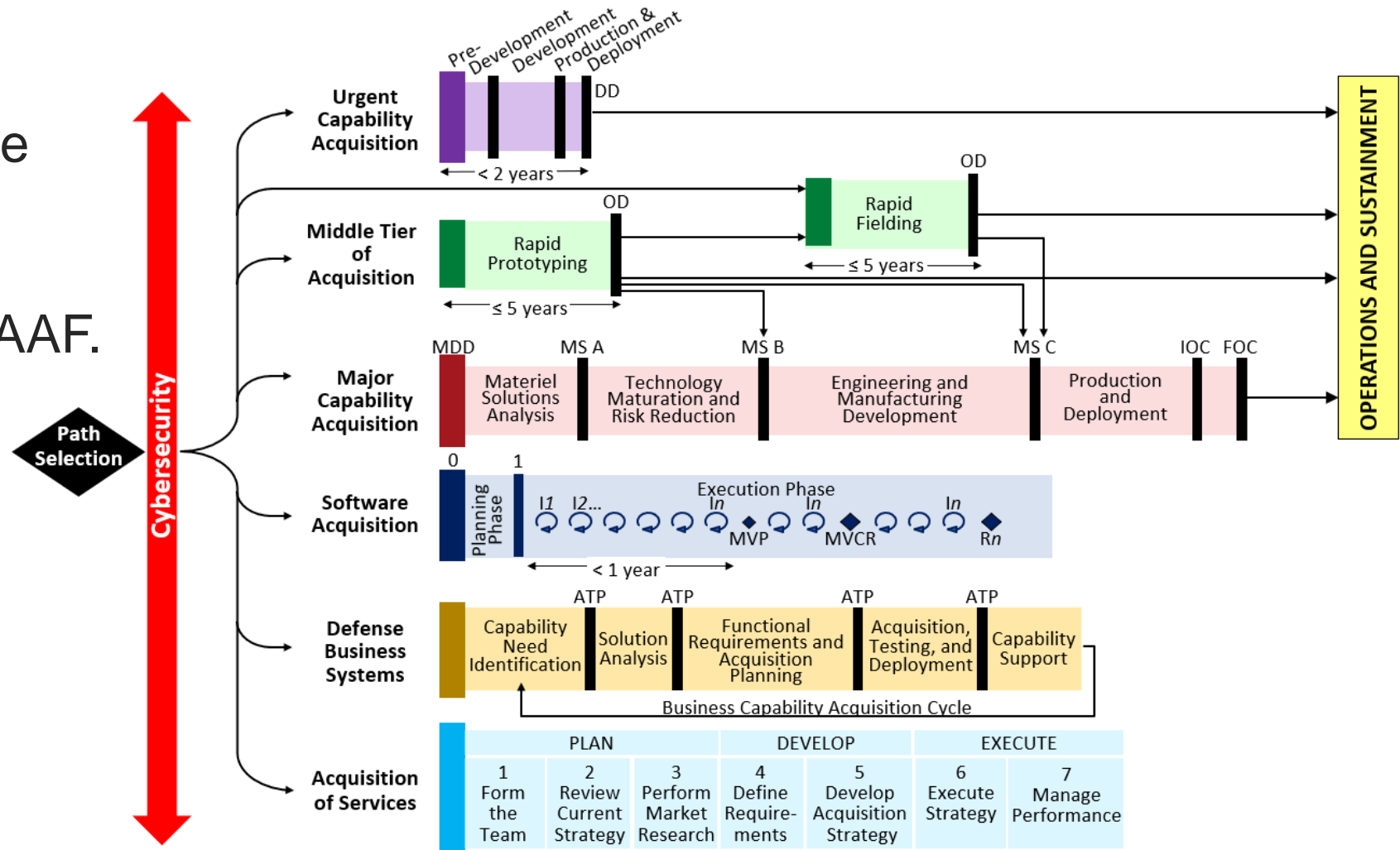
Start Small, Build on Success

- Do **NOT** slowly evolve to Agile!!
- Modularize: build *all* Agile processes for a small, less important sub-system first
- Learn from small Agile implementation
- Apply working processes to another sub-system
- Expand to more as processes mature

This approach is often called the Strangler Pattern

Pathways

During transition, the program will have multiple sets of processes, like the AAF.



Questions?

“In the decade since the waterfall model was developed, our discipline has come to recognize that setting the requirements is the most difficult and crucial part of the software building process, and one that **requires iteration** between the designers and users. In **best modern practice**, the early specification is embodied in a prototype, which the intended users **can themselves drive** in order to see the consequences of their imaginings. Then, as the design effort begins to yield data on the cost and schedule consequences of particular specifications, the designers and the users revise the specifications.”

- **1987** Defense Science Board report to Congress on software

<https://apps.dtic.mil/sti/pdfs/ADA188561.pdf>