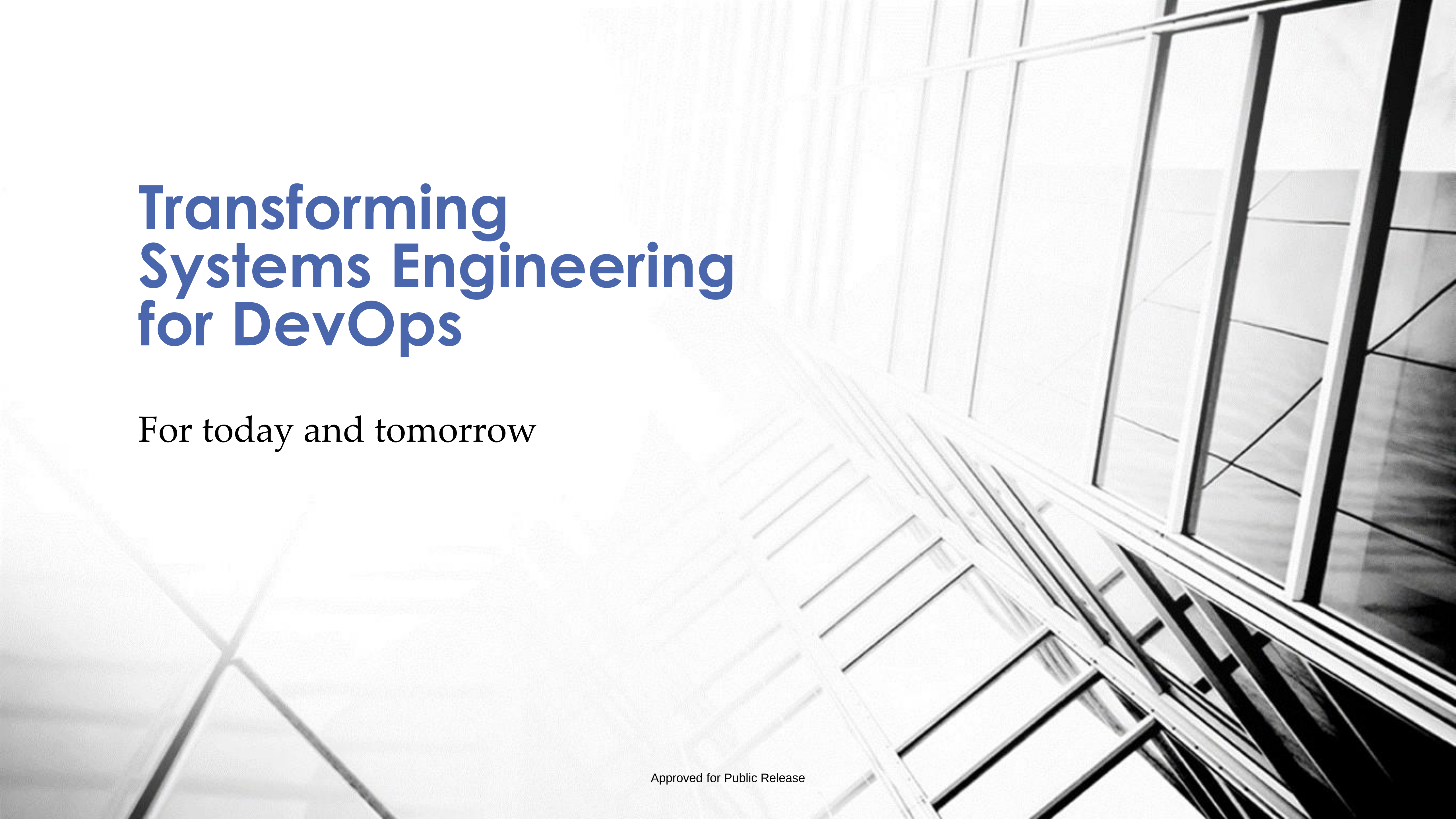


Transforming Systems Engineering for DevOps

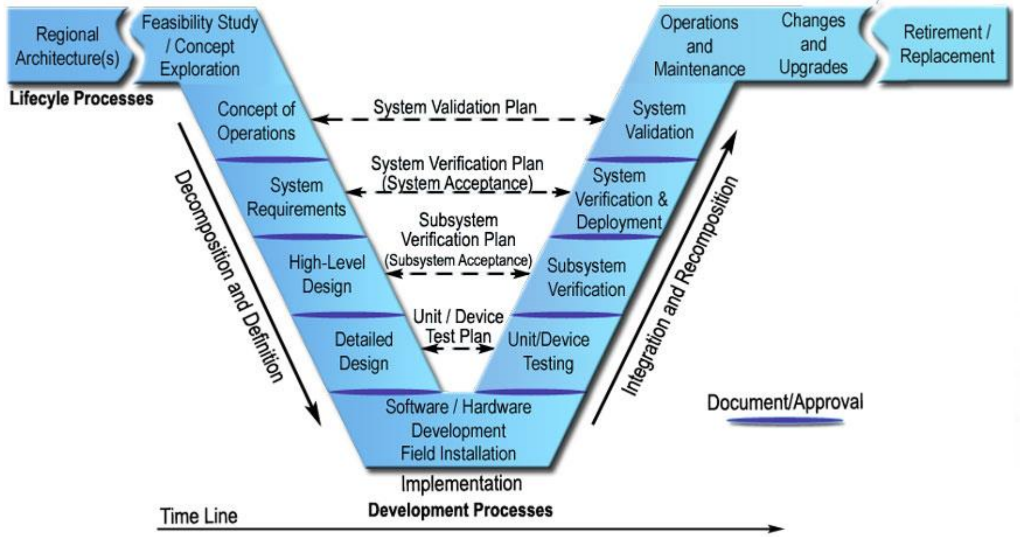
For today and tomorrow

Vision

- The customer community is increasingly dissatisfied with the traditional acquisition methods for systems and services that require BDUF (Big Design Up Front) and years in development and test. They are looking at methodologies like DevOps for a more tactical, nimble acquisition of products and services with shorter cycle times, faster integration and delivery, frequently delivered enhancements, and dynamic product maintenance to combat the ever-changing cyber security threat.

Transformation of Systems Engineering

- Traditional Systems Engineering must adapt to the emerging dynamics of DevOps
  - BDUF of the Systems Engineering "V" model needs to be re-evaluated as it is not designed to support the DevOps methodology

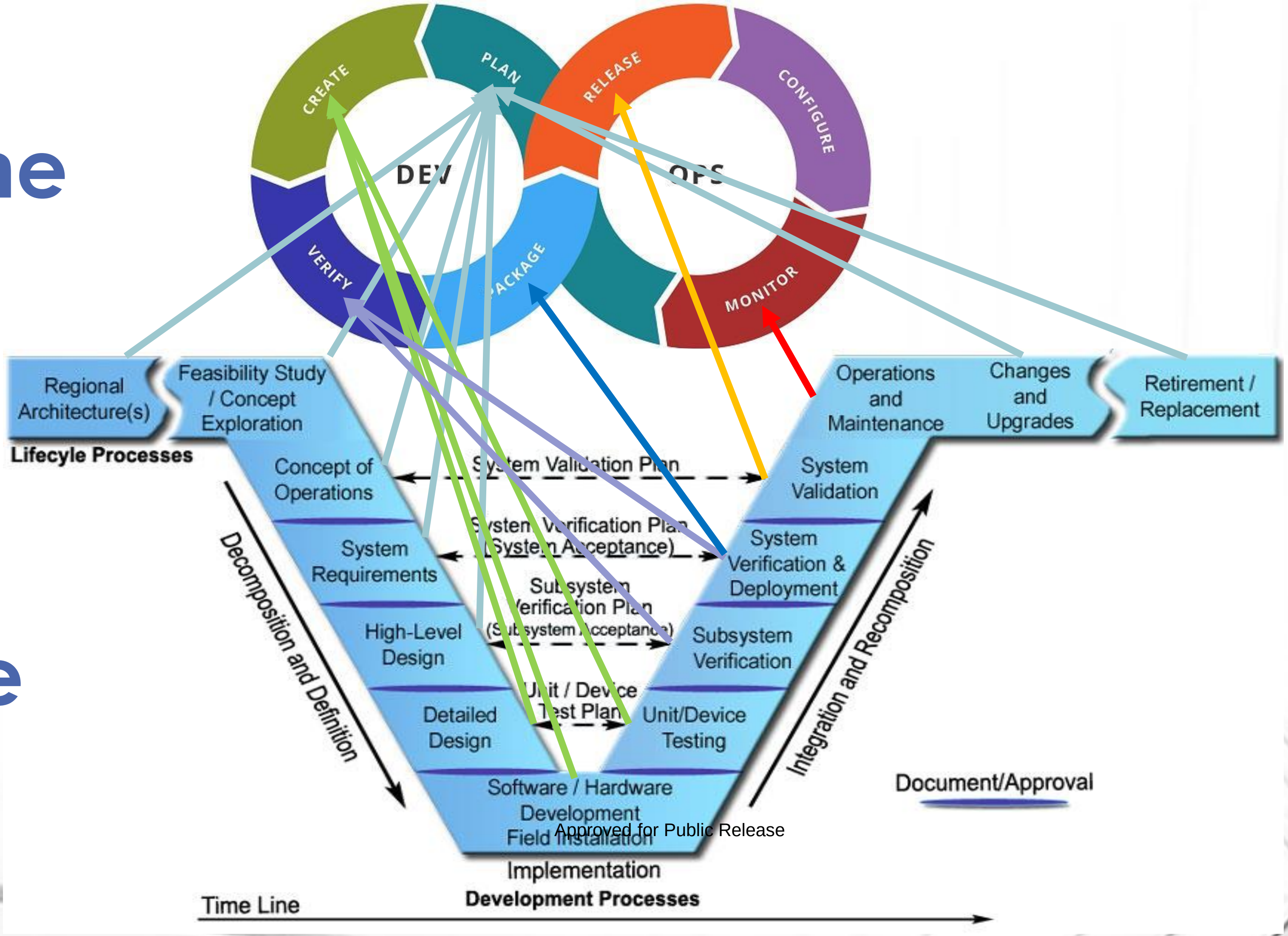


- New approaches to Systems Engineering must be identified that can support both the strategic view of the system and the tactical development of products and services

Systems Engineering "V" to DevOps

Every Time

One Time



How Did We Get To Today?

- We have always done engineering this way
  - Complex Systems & System of Systems
  - Complex Missions & Warfighter needs
  - Hardware centric
- Customer standards and compliance drive us to do engineering this way
  - MIL-STD-1521B Military Standard: Technical Reviews and Audits for Systems, Equipments, and Computer Software
  - MIL-STD-973 Military Standard: Configuration Management
    - Section 5.6.3 – PCA shall include an audit of the released engineering documentation and quality control records to make sure the as-built or as-coded configuration is reflected by this documentation. Including product specification, Interface Design Document, and VDD.

How Do We Get To Tomorrow?

- Bound & Scope the Upfront Design
- Embed Systems Engineers & Architects into DevOps
- Define the Minimum Viable Product
- Change the Rules on Standards & Compliance

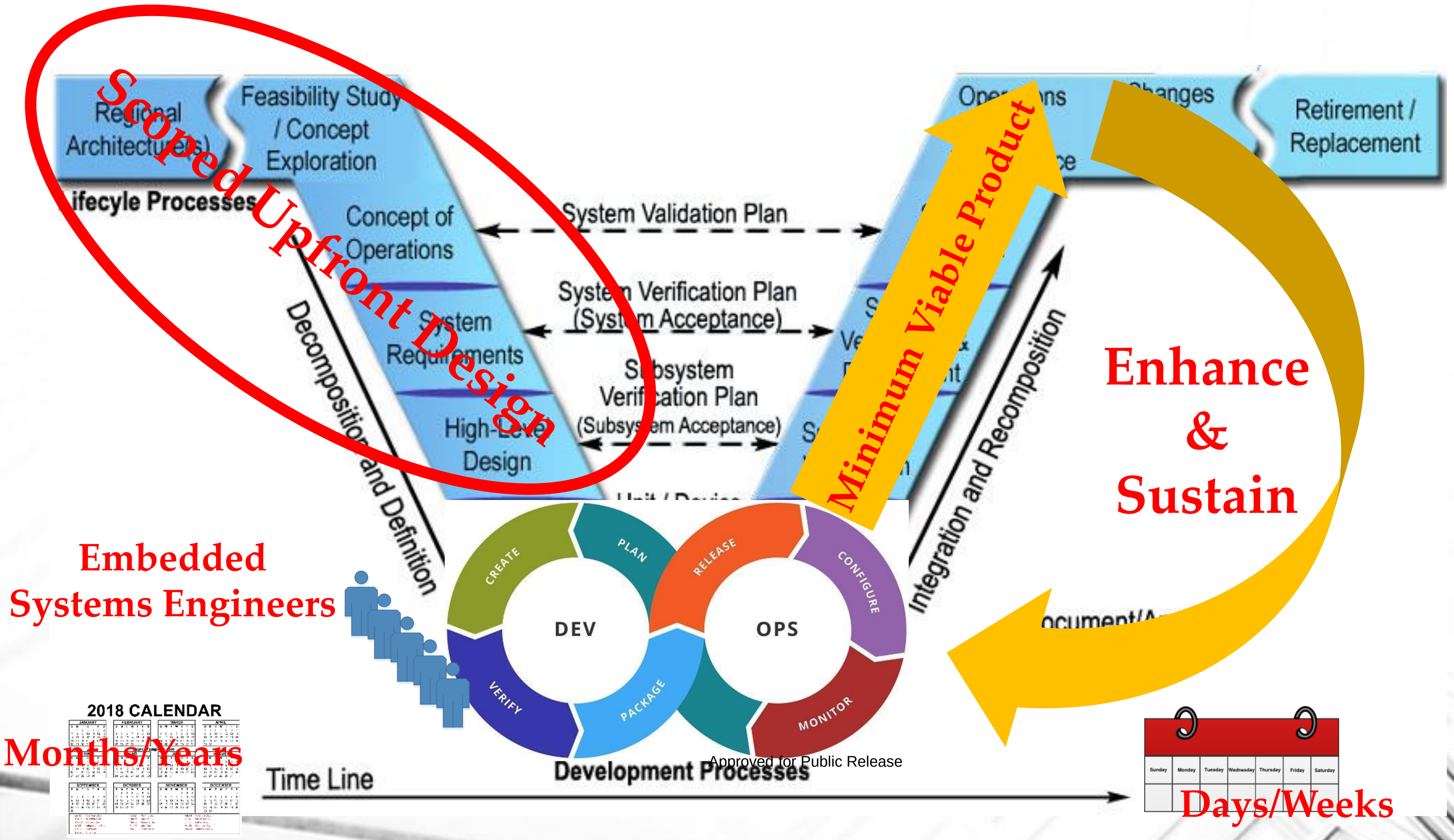
Recommendations

- Bound & Scope the Upfront Design
  - Organize the SE Technical Baseline to support DevOps by focusing on:
    - Unique features that can be developed and encapsulated into a product
    - Products that can be bundled and delivered to support a mission thread (or use case)
    - Mission threads that combine to build the system
  - Tailor the upfront design to “cap” at the entry point into DevOps
- Embed Systems Engineers & Architects into DevOps
  - Because we have capped the upfront design at the DevOps entry point:
    - Systems Engineers need to be embedded into the DevOps teams to decompose the higher level design into the DevOps process and to set a plan with achievable discrete objectives and measurable progress metrics
    - Systems Architects integrated across all the DevOps work providing governance, direction, and ensuring that the features/capabilities being developed align with the higher level products, mission threads, and system

Recommendations

- Define the Minimum Viable Product
  - In order to achieve the tactical/nimble delivery of products and services with shorter cycle times Systems Engineering must be able to define the smallest set of features and capabilities that when brought together constitute a deliverable product consisting of the minimum capabilities necessary to perform the core mission of the system.
  - All subsequent deliveries augment, enhance, build upon, and sustain the initial Minimum Viable Product
- Change the Rules on Standards & Compliance
  - DevOps core fundamentals do not align well standards and compliance directives like MIL-STD-1521B and MIL-STD-973
    - Tailor program standards and compliance baseline to better fit scoped upfront design and DevOps work products
    - Make the DevOps processes, automation, and delivered products the standards and compliance evidence → "We are compliant because we delivered the system and it works"

Recommendation Realized



Approved for Public Release

Questions?