



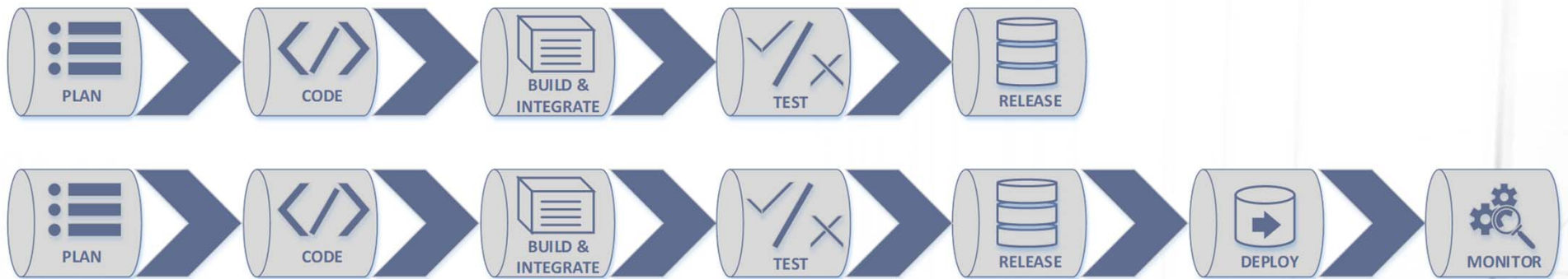
DevOps Development – Dynamics of Engineering Concurrent Multi-Product Efforts

Approved for Public Release

Vision

- Engineering and managing a DevOps product development pipeline is challenging. Engineering and managing multiple concurrent DevOps product development pipelines, some of which are unique or stand alone, some of which are shared, and some of which have cross product dependencies can elevate those challenges to a whole new level.

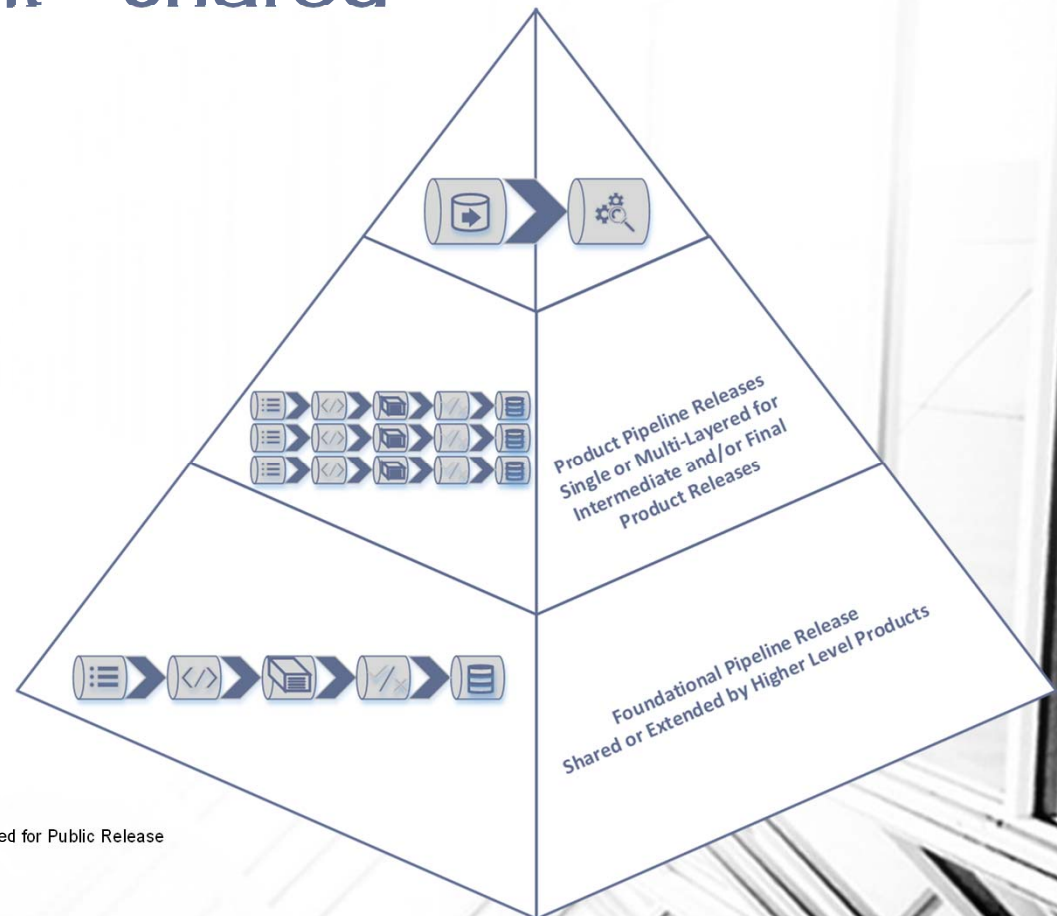
DevOps Development - Unique



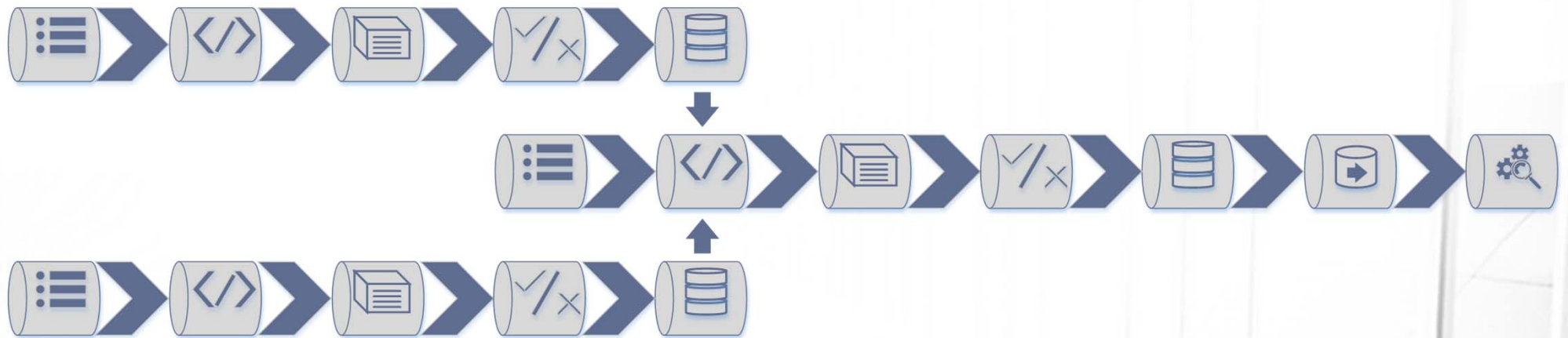
- Unique & Stand Alone Product Pipelines
 - Low to Moderate Barrier to Entry
 - Self Contained – All tooling, code, planning, and resources dedicated to the product
 - Generally Low Complexity – Limited Interfaces, Few Dependencies

DevOps Development - Shared

- Shared Product Pipelines
 - Moderate to High Barrier to Entry
 - Multi-tiered Products often including Infrastructure, Middleware, and Mission Components
 - Tooling & Resources usually shared
 - Planning & Integration more complex
 - Medium to High Complexity – Compartmentalized Components, Moderate to High Volume of Interfaces, Component Level Dependencies within Product



DevOps Development – Dependent



- Dependent Product Pipelines

- High Barrier to Entry
- Often Multi-tiered & Enterprise Products
- Tooling & Resources usually shared, non-linear release processes are typical
- Planning, Integration, & Deployment can be significantly more complex
- High Complexity – Large Volume of Interfaces, Code level Dependencies

DevOps Development: Elevating Challenges

- Development Contract versus Sustainment Contract
- Complexities of release content planning: Unique → Shared → Dependent
- Managing release compatibilities & dependencies

Development vs. Sustainment

Development

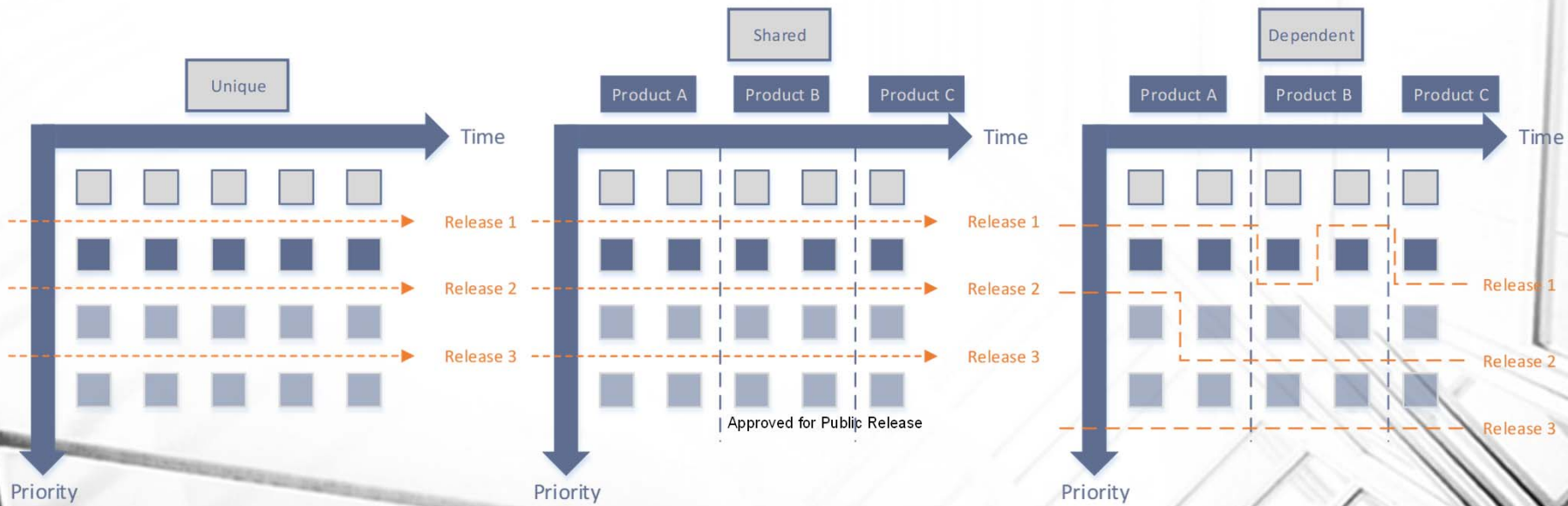
- Evolving DevOps maturity
- Emerging Tech Baseline
- Developing for Standards & Compliance
- Building for Automation
- Development Milestones
- DT&E
- Designing for O&S

Sustainment

- Stable DevOps Pipeline(s)
- Established Tech Baseline
- Maintaining Standards & Compliance
- Leveraging & Augmenting Automation
- Sustainment Lifecycles
- OT&E
- Supporting the Mission & Growth

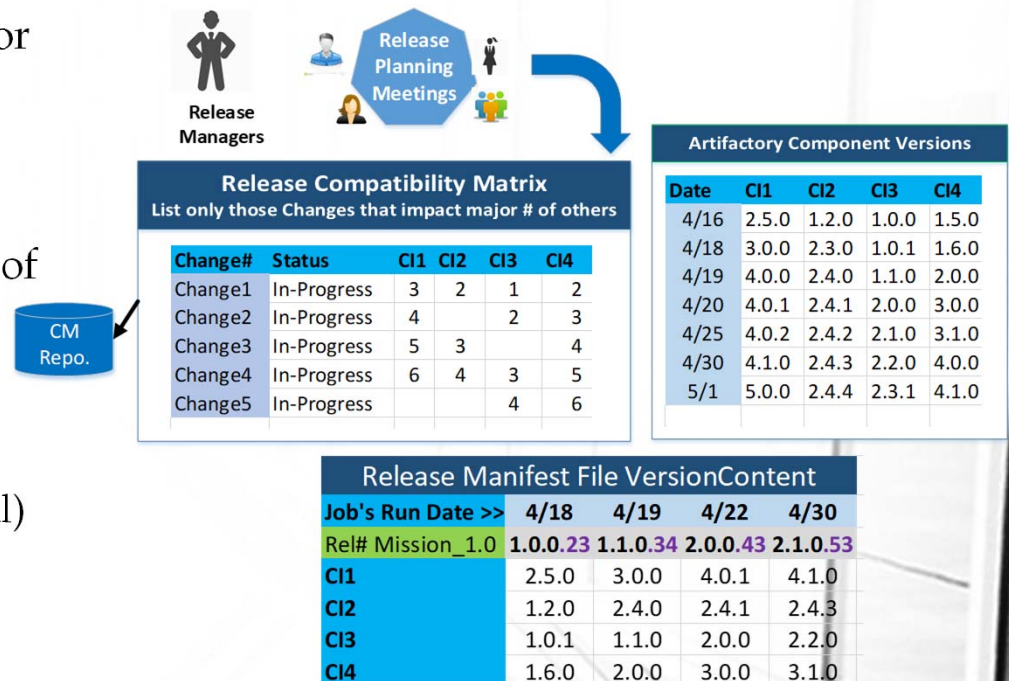
Release Content Planning

- The simpler the product, the lower the volume of interfaces, the fewer the dependencies, the easier it is to do release content planning
- As product volume, shared design, and inter-product dependencies grow so does the complexity and need for release content planning
- Whether back of the napkin, spreadsheet, tool driven, or formal project schedules release content planning is critical and a key indicator of program health



Compatibilities & Dependencies

- Volume & Complexity of shared, tiered, or dependent products drive compatibility and dependency issues
- Semantic Versioning to establish compatibility matrices and management of change including:
 - Mission (custom) Software
 - COTS, FOSS, & GOTS
 - Environments (development & operational)
- Architecture & Software dependency matrices to analyze change & inform release planning



Summary

- DevOps Product Development Pipelines can vary greatly in volume and complexity
- Stage of the program, Development versus Sustainment, is a major contextual differentiator of the DevOps engineering and management needs
- As the complexity of your DevOps Product Development Pipelines increases so does the criticality of release content planning
- The greater the complexity of the system the more critical the active management of compatibilities and dependencies becomes
- Keep It Simple → Design for Unique, Build Shared when required, but Avoid Dependent!

Questions?



Approved for Public Release