

Aligning the Language of Systems Engineering and Agile

Kelly Weyrauch

Kelly@AgileQualitySystems.com

Copyright © 2018 by Agile Quality Systems. Permission granted to INCOSE to publish and use.

NASA SE Handbook

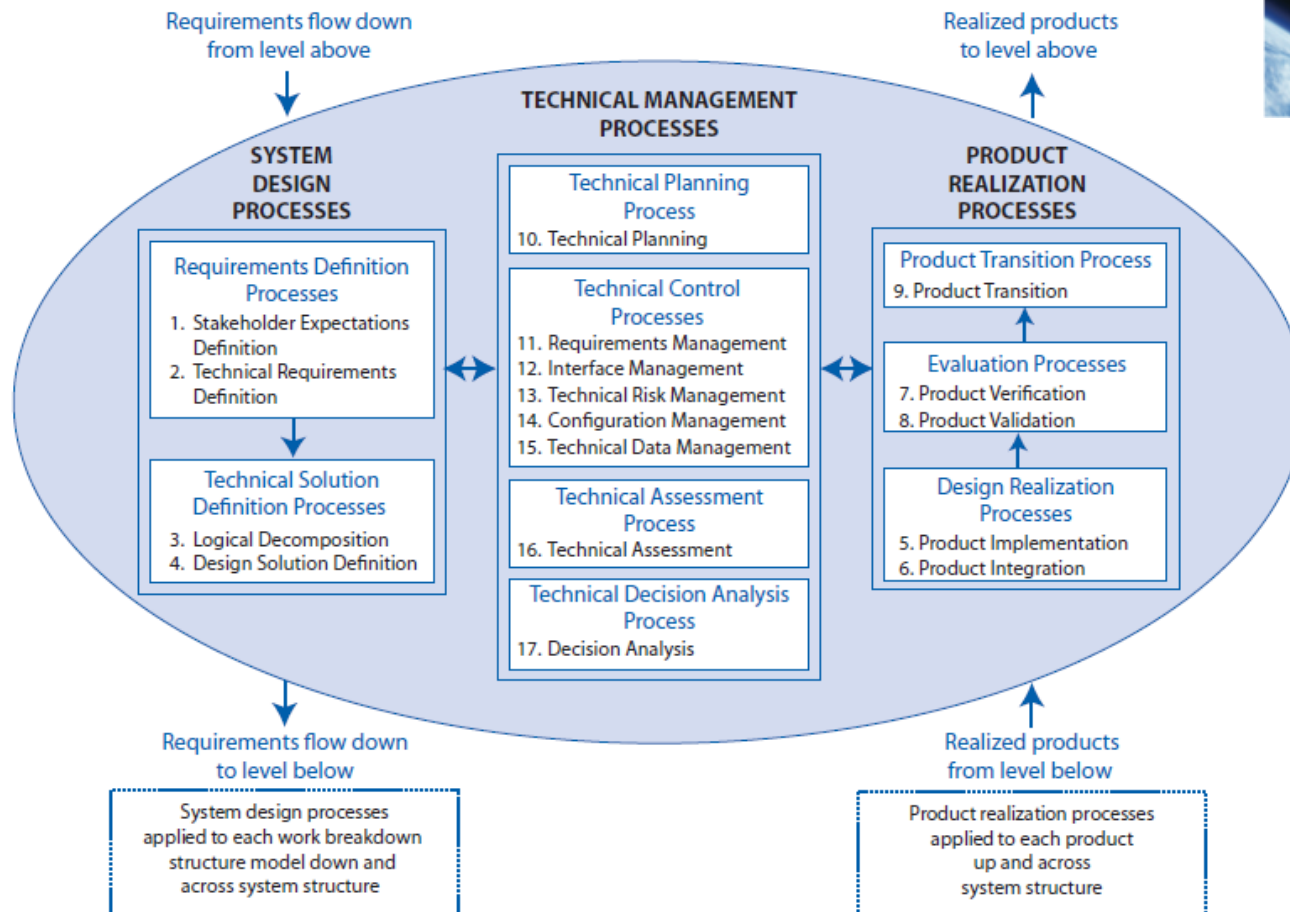
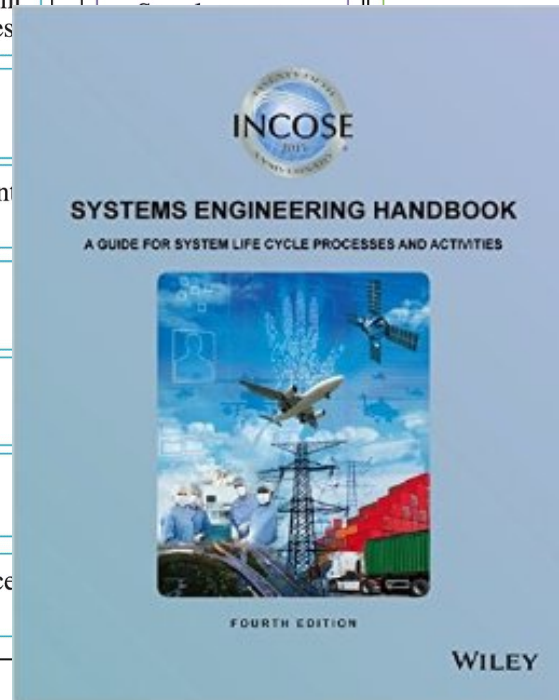
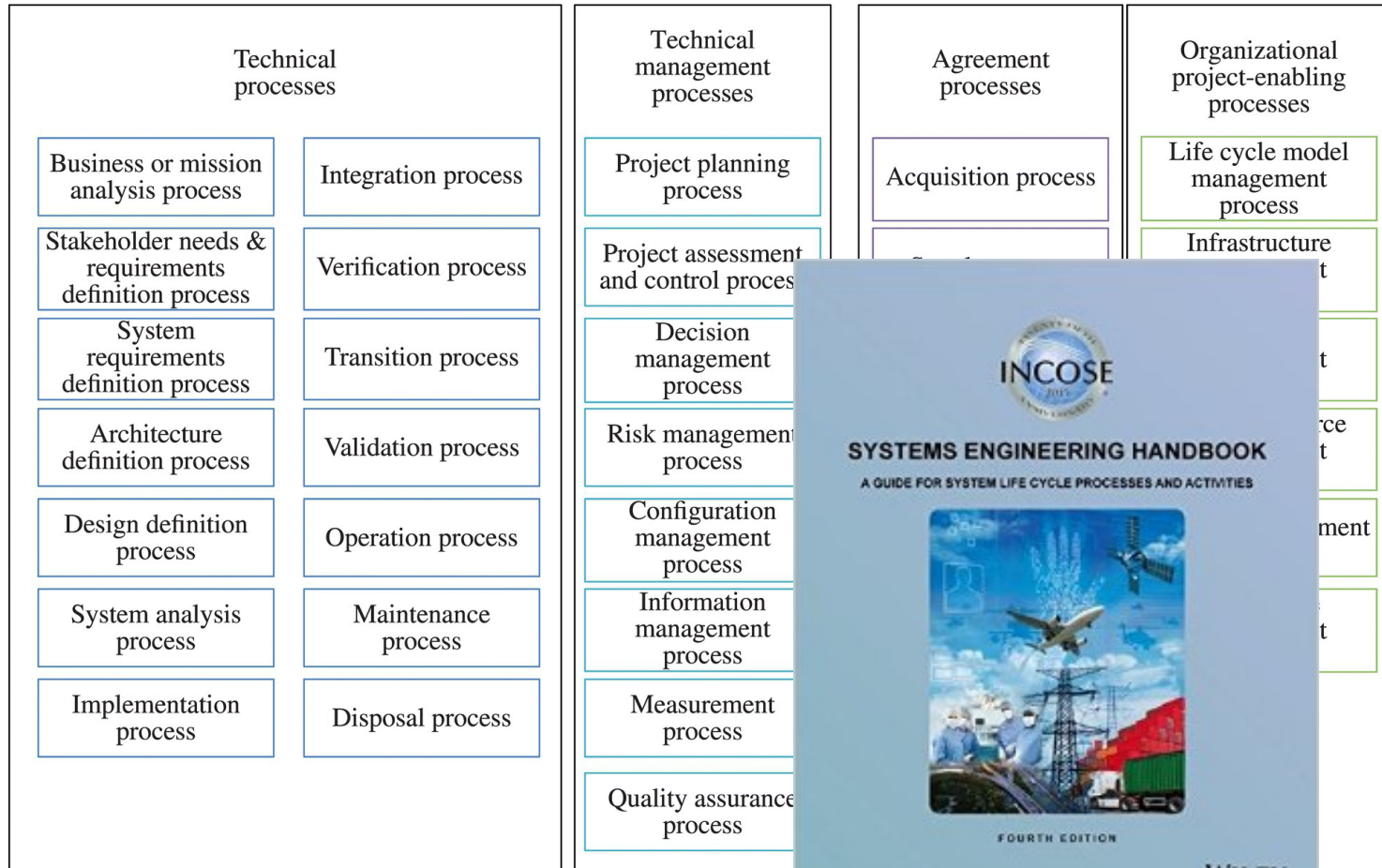


Figure 2.1-1 The systems engineering engine

ISO 15288 System Life Cycle Processes

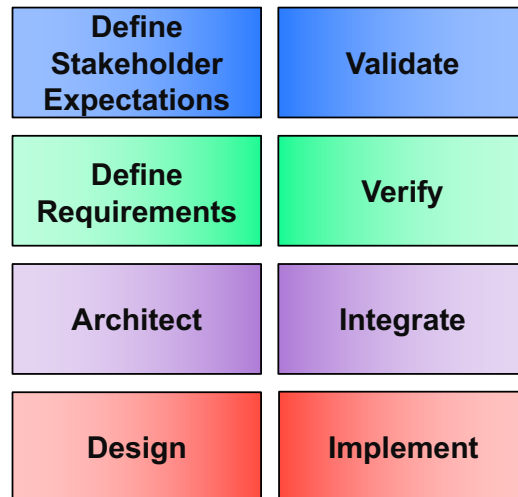


The Systems Engineering Engine



System Development Processes

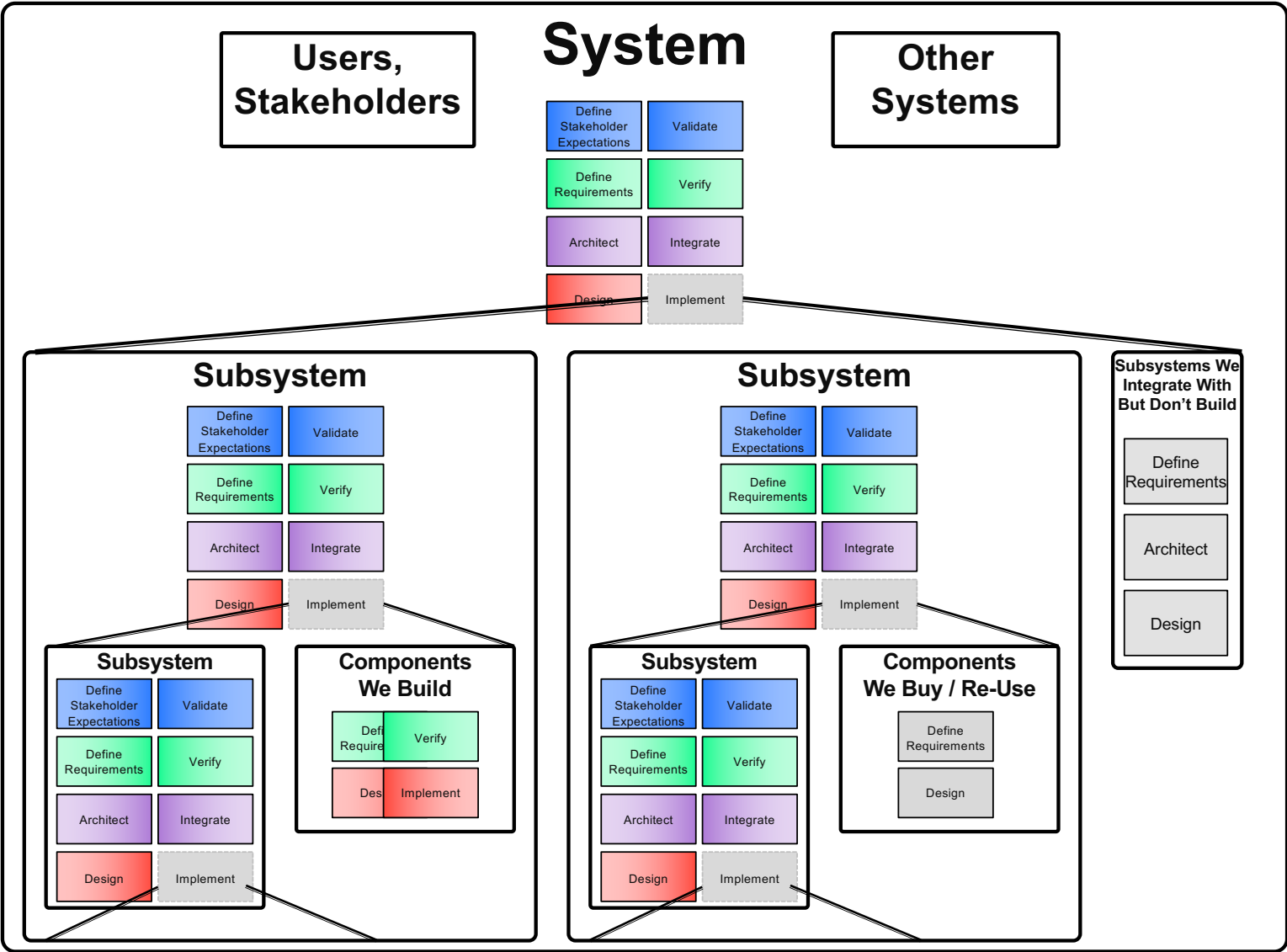
System Design Product Realization



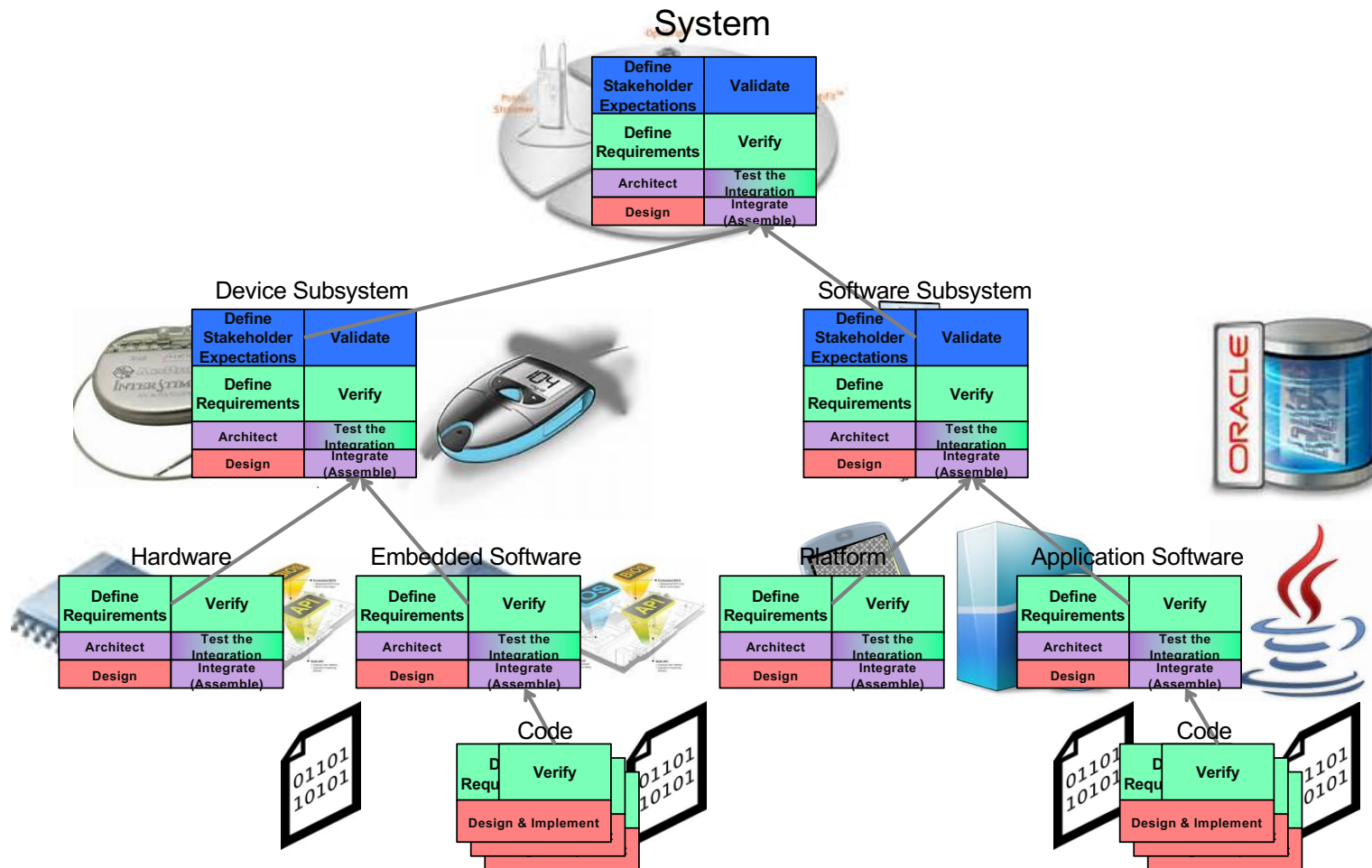
Technical Management Processes

Technical Planning
Requirements Management
Interface Management
Risk Management
Configuration Management
Data Management

Adapted from: NASA Systems Engineering Handbook, NASA/SP-2007-6105

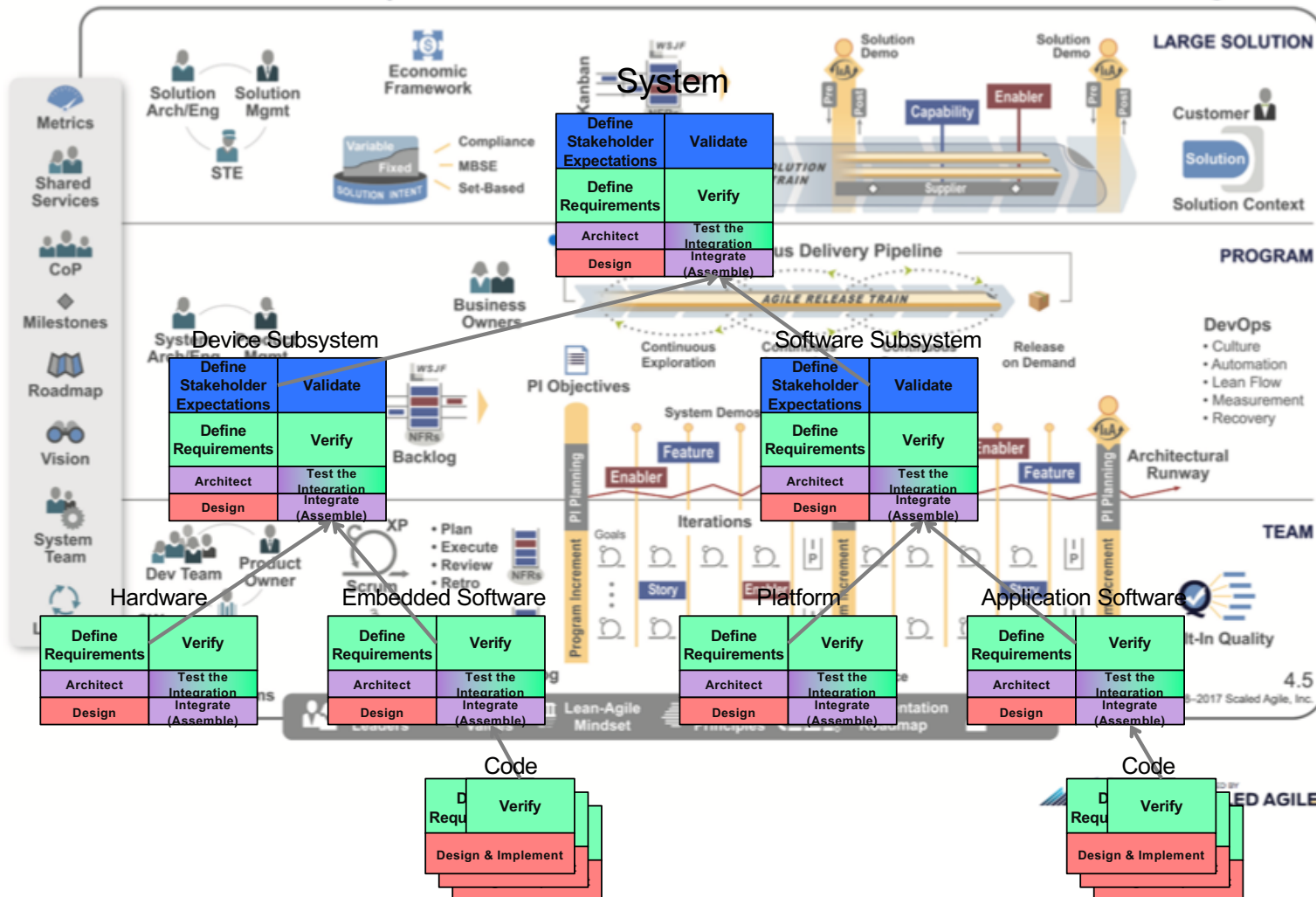


What is the System? And What are the Activities & Deliverables?

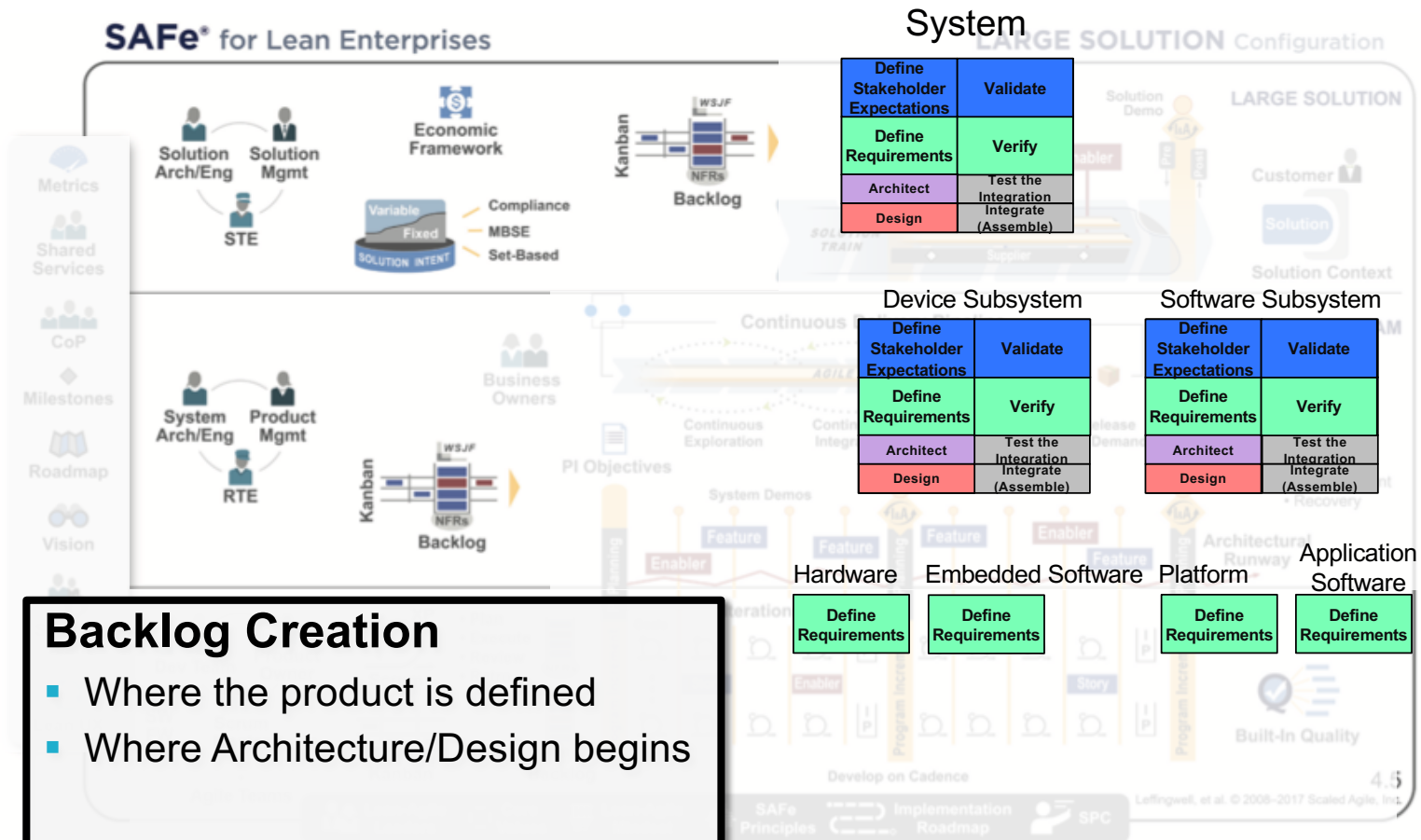


SAFe® for Lean Enterprises

LARGE SOLUTION Configuration



SAFe® for Lean Enterprises



| | |
|---------------------------------|---|
| Define Stakeholder Expectations | Validate |
| Define Requirements | Verify |
| Architect | Test the Integration Integrate (Assemble) |
| Design | |

Device Subsystem

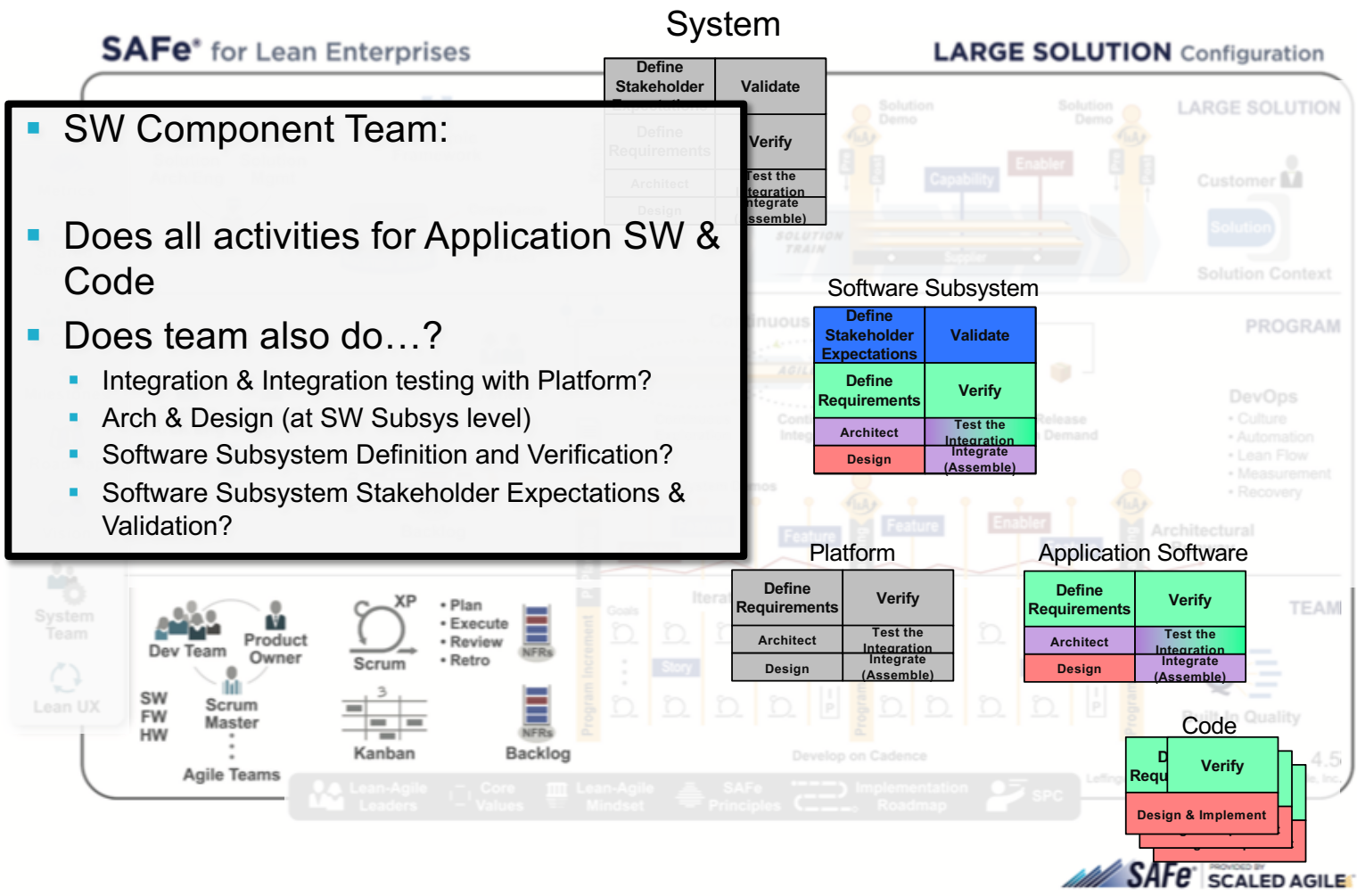
| | |
|---------------------------------|---|
| Define Stakeholder Expectations | Validate |
| Define Requirements | Verify |
| Architect | Test the Integration Integrate (Assemble) |
| Design | |

Software Subsystem

| | |
|---------------------------------|---|
| Define Stakeholder Expectations | Validate |
| Define Requirements | Verify |
| Architect | Test the Integration Integrate (Assemble) |
| Design | |

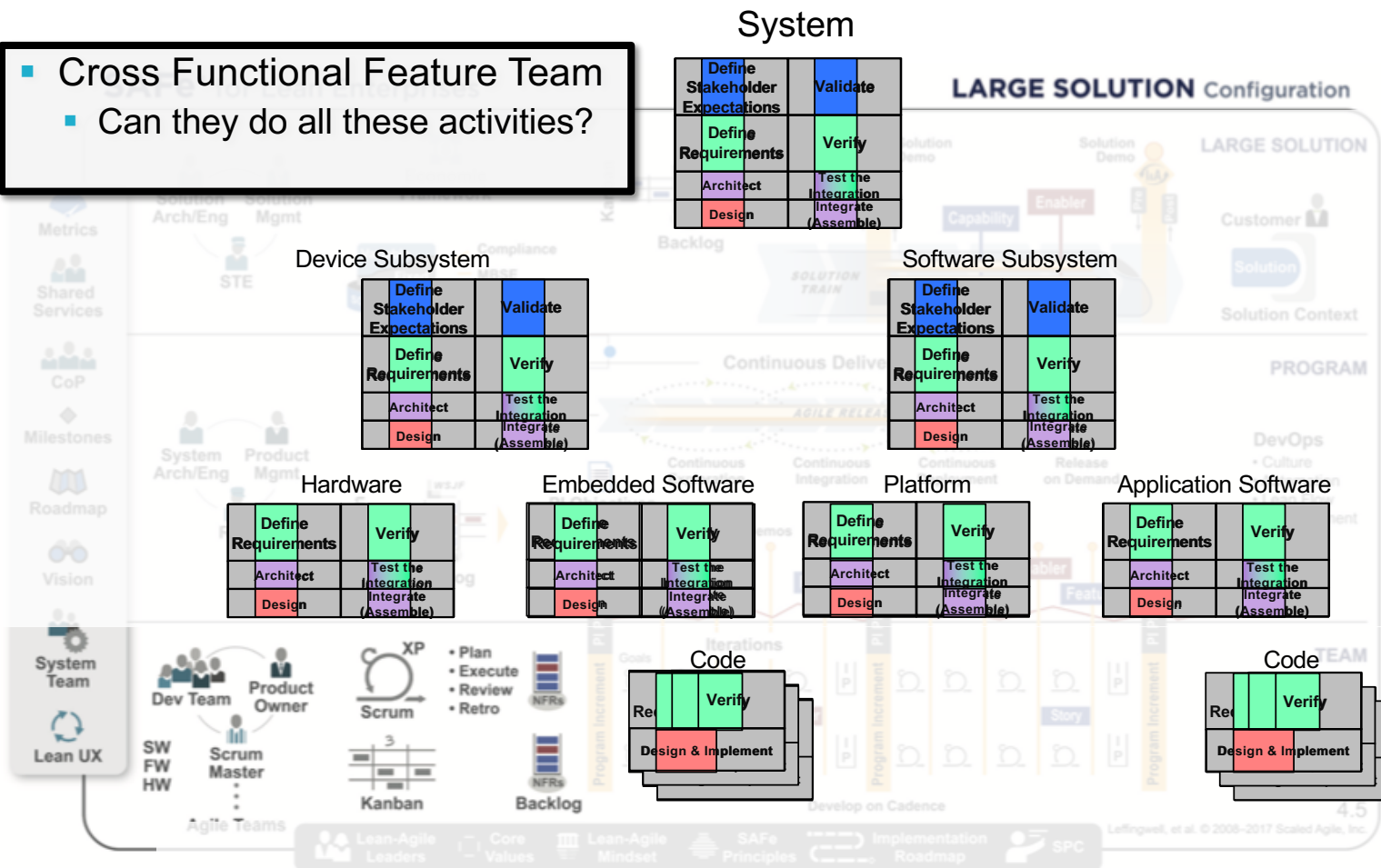
| | | | |
|---------------------|---------------------|---------------------|----------------------|
| Hardware | Embedded Software | Platform | Application Software |
| Define Requirements | Define Requirements | Define Requirements | Define Requirements |

- ### Backlog Creation
- Where the product is defined
 - Where Architecture/Design begins
 - But how far?
 - Variable? Fixed?



- SW Component Team:
- Does all activities for Application SW & Code
- Does team also do...?
 - Integration & Integration testing with Platform?
 - Arch & Design (at SW Subsys level)
 - Software Subsystem Definition and Verification?
 - Software Subsystem Stakeholder Expectations & Validation?

- Cross Functional Feature Team
 - Can they do all these activities?



Layers / Kinds of Backlog Items

(as defined by the Scaled Agile Framework, SAFe®)

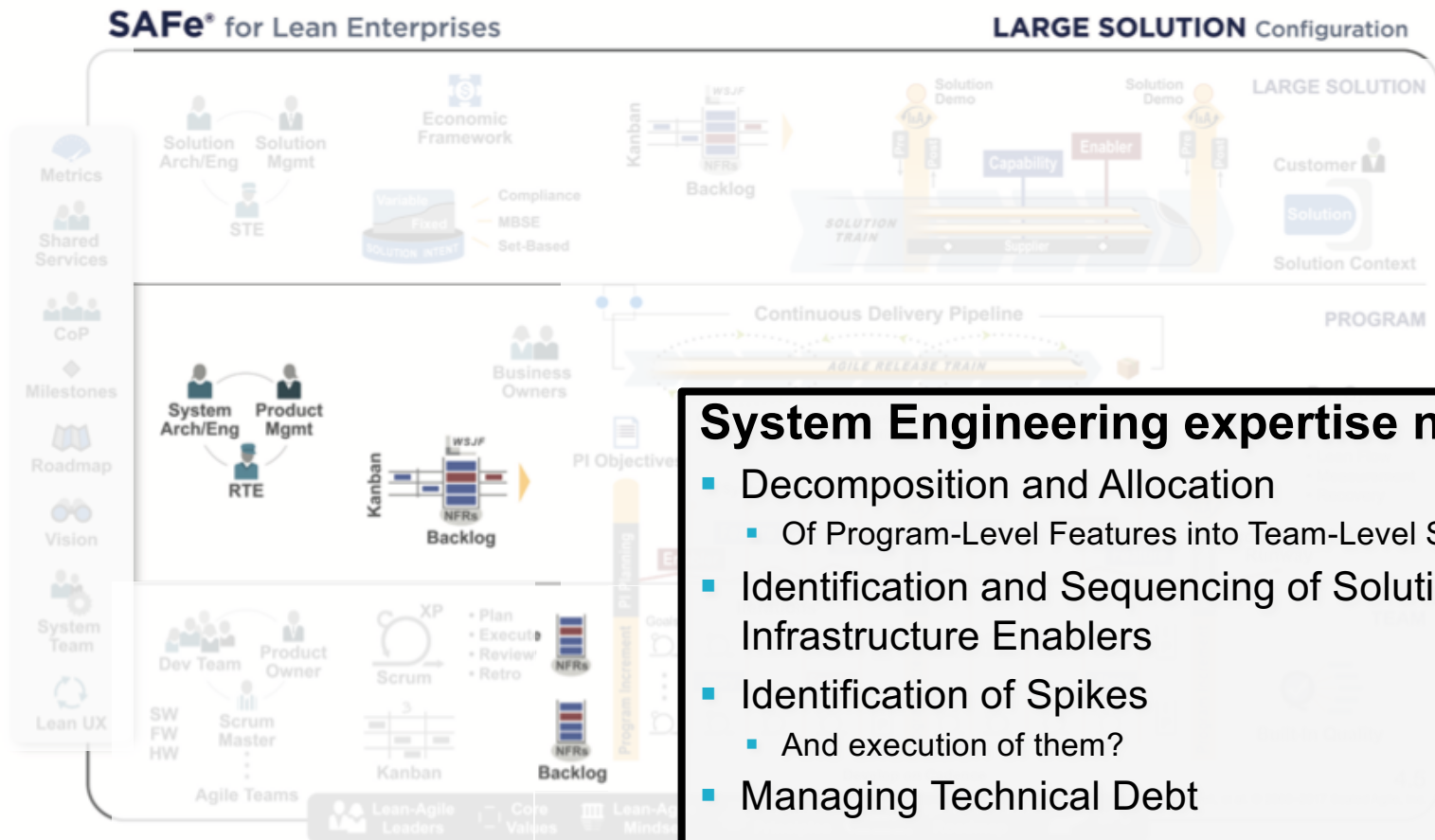
- Four Levels
 - Portfolio-Level Epics
 - Large-Solution-Level Capabilities
 - Program-Level Features
 - Team-Level Stories

- Two Perspectives
 - Customer-Facing, Business Value
 - Solution-Facing, Enablers

| | Epics (Portfolio) | Capabilities (Large Solution) | Features (Program) | Stories (Team) |
|--------------------------|---|--|--|--|
| Responsible | Portfolio Management Enterprise Architect (+PM, +SE) | Solution Management, Solution Architect (+PM, +SE) | Product Management (+PO, +SE) | Product Owner (+Team) |
| Provides Value to | Customers, Business | “Who” of the Capability, Customers, Enterprise Architect, The Epic | “Who” of the Feature, Business Owners, Solution Architect, The Capability/Epic | “Who” of the Story, System Architect, The Feature |
| Delivered by | N/A, delivered through Capabilities & Features | Agile Teams, System Team, Specialty Team? | Agile Teams, System Team | Agile Teams |
| Delivered when | Depends – When all Capabilities/Features complete? Only some? | When all Features complete (1 or more Increments) | Each Program Increment | Each Sprint |
| Demoed | N/A, Demo done with lower layers | System Demo, Validation Studies | System Demo | Team’s Sprint Demo |
| Content | Lightweight Business Case | Description & Benefit, Acceptance Criteria, Definition Of Done | Description & Benefit, Acceptance Criteria, Definition Of Done | Story Pattern, Acceptance Criteria, Definition Of Done |

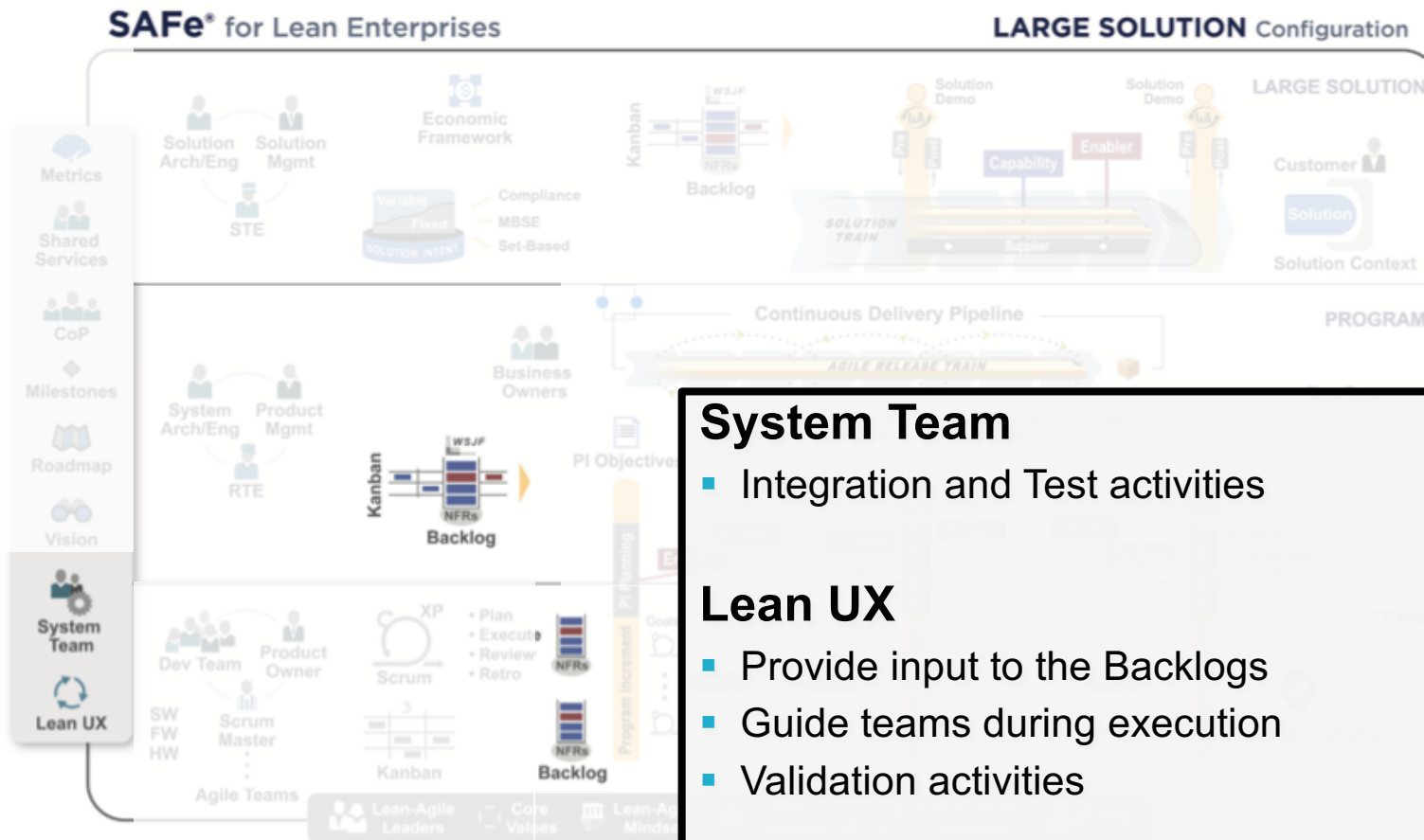
Solution-Facing Backlog Items

- SAFe® term: “Enablers”
 - Enabling the “Architectural Runway” upon which customer-facing value can be delivered.
- Infrastructure (Development, Product)
- Debt
- Spikes (Definition, Technical, Decision)
- System Integration
- Quality System Satisfaction
- ...



System Engineering expertise needed for

- Decomposition and Allocation
 - Of Program-Level Features into Team-Level Stories
- Identification and Sequencing of Solution-Facing Infrastructure Enablers
- Identification of Spikes
 - And execution of them?
- Managing Technical Debt



System Team

- Integration and Test activities

Lean UX

- Provide input to the Backlogs
- Guide teams during execution
- Validation activities

Agile & Systems Engineering

- Agile Development depends on
 - Solid Product Architecture (Physical Architecture)
 - Solid Backlog Architecture (Functional Architecture)
 - Solid Development Team Architecture
 - Systems Engineers speak the language of Architecture and System Development Activities
 - SAFe speaks the language of Lean/Agile
 - Align them to realize the benefits of both
- 