Systems Engineering in Large-scale Agile Software Development

Will Menner

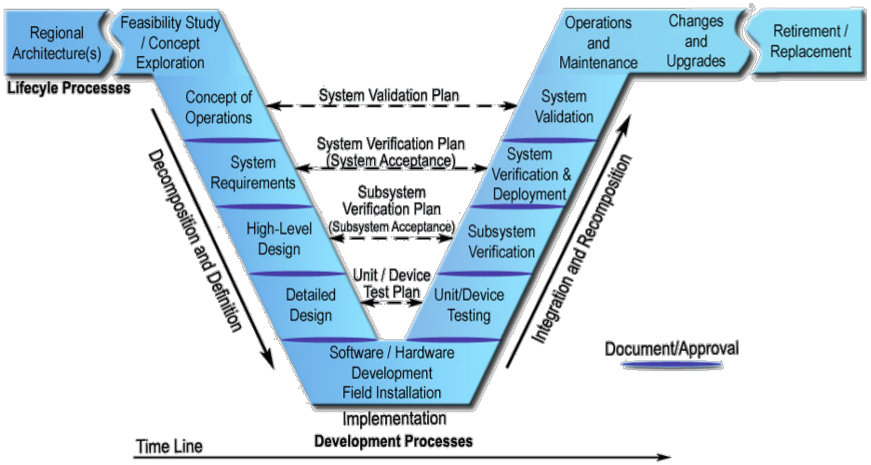
Johns Hopkins University

Will.Menner@jhuapl.edu

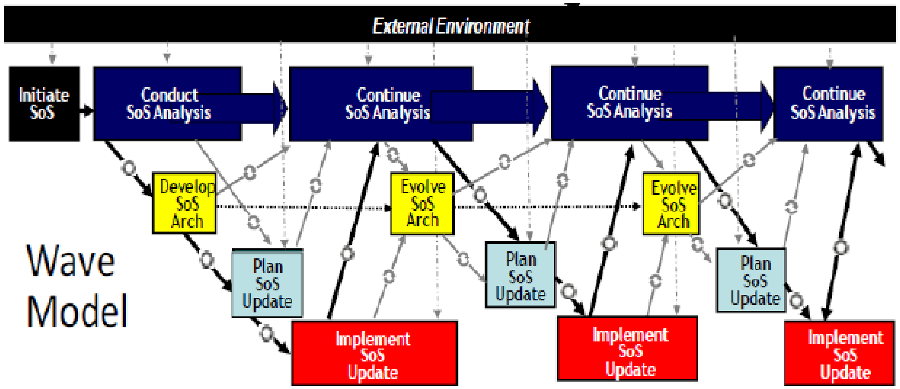
240.228.5354

BLUF

Bottom Line Up Front



ops.fhwa.gov



sercuarc.org

My experience: attempts to find a home for agility within systems engineering constructs have largely come up empty.

> *Instead of starting with a systems engineering model and attempting to find room for agility, start with an agile model and figure out where systems engineering can be of greatest utility.*

Motivation

- 2014: NDIA Acquisition Reform Letter - reduced regulatory burden, better-educated acquisition workforce, improved oversight
- 2014: USD(AT&L), Performance of Defense Acquisition System - cycle time is increasing
- 2013: NDIA/ADAPT report: Better Buying Power in Software Acquisition
- 2012: GAO report: Effective Practices for Federal Challenges
- 2011: UK Institute for Gov't report on "*fixing the flaws*" in gov't IT
- 2010: HASC report: "*the acquisition system is particularly poorly designed for the acquisition of information technology*"
- 2009: DSB report: AoA-to-IOC averaging 91 months
- 2008: GAO reports: DoD acquisition programs averaging 21-month delay

The underlying cause of acquisition problems may be largely due to our attempts to control uncertainty. Acquisition reform may need to embrace uncertainty and explore how we can develop methods and systems to better manage it.

Agile SD Methods, ©Will Menner

Domain of Interest

Vision

System Capabilities

Champion (Product Owner)

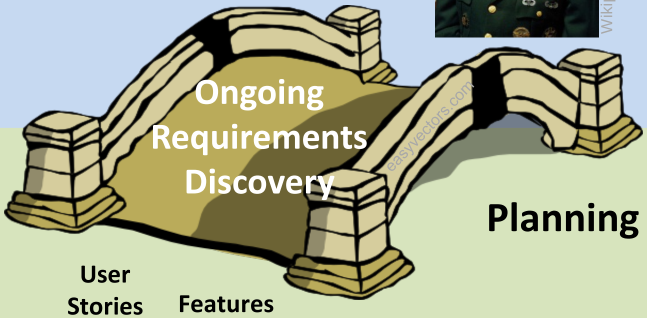
Users

Analysts
OPS Mngmt
Warfighters

Deployment & Support

Clarification Collaboration

Alignment



Planning

User Stories **Features**

Iteration Backlog

Product Backlog

Greater Fidelity **High**

Estimates

Priorities

Lesser Fidelity **Low**

Dev Team, 7 ± 2
Cross Functional
Self-Organized

Stories → Tasks

1st **Write Test** ↔ **Write Code** **Refactor**

Review Code
Integrate Continuously
Automate Tests
Standardize Coding
Manage Configurations

Product Owner

Leverage:
App Store
Service Catalog
Architectural Models

Reuse

Team 2

Team 3

⋮

Team T

Foster Collaboration
Use Scrum of Scrums

Developing

Integration
& System Test
Acceptance
Definition of Done
Learning
Metrics (Quality)
Functional
Structural
Process

Sprint Reviews

Sprint Retrospectives

Integrated Product Increment

Coordinate
Synchronization
Points

Release Plan

	Iteration N	Iteration N+1	Iteration N+2
Mission Capabilities	■ ■ ■	■ ■ ■ ■	■ ■ ■ ■ ■ ■
Architectural Elements	▲ ▲ ▲ ▲ ▲	▲ ▲ ▲ ▲	▲ ▲ ▲
Research (Technical, Marketing, …)	● ●	● ● ●	●

Short Iterations producing Working, Tested Software
Evolution of Product and its Underlying Architecture

The Product Owner Team

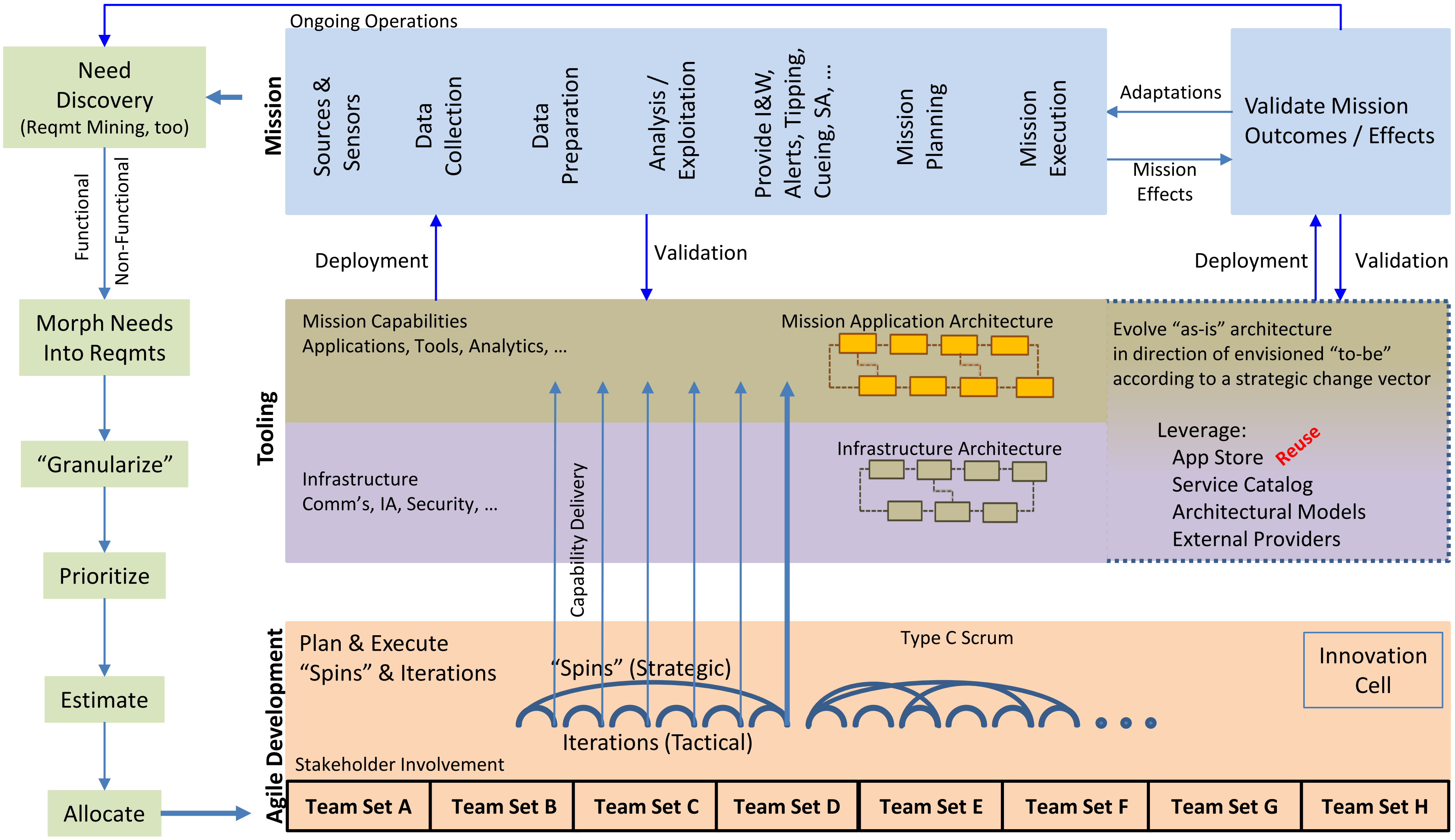
- Coordinate with external projects and systems
- Help write and prioritize user stories
- Manage Product Backlog(s)
- Identify Iteration Backlogs
- Devise alignment mechanisms
- Allocate user stories to teams
- Help define and execute acceptance tests
- Develop and maintain metrics
- Coordinate between implementation, architecture, and research teams
- Ensure teams have the technical resources necessary to be successful
- Conduct or facilitate retrospectives to ensure continual improvement
- Create and conduct education and training sessions for stakeholders
- Facilitate the collection of feedback from end-users
- Synthesize feedback into new user stories

Activities for Systems Engineers on Large-scale Agile Teams

Activity-based Decision Framework

An Example based on Data Analysis Workflow for a Generic “Mission”

Agile SD Methods, ©Will Menner



Need Discovery (Reqmt Mining, too)

Functional

Non-Functional

Morph Needs Into Reqmts

“Granularize”

Prioritize

Estimate

Allocate

Ongoing Operations

Mission

Sources & Sensors

Data Collection

Data Preparation

Analysis / Exploitation

Provide I&W, Alerts, Tipping, Cueing, SA, …

Mission Planning

Mission Execution

Adaptations

Mission Effects

Validate Mission Outcomes / Effects

Deployment

Validation

Deployment

Validation

Tooling

Mission Capabilities
Applications, Tools, Analytics, …

Mission Application Architecture

Infrastructure
Comm’s, IA, Security, …

Infrastructure Architecture

Capability Delivery

Evolve “as-is” architecture in direction of envisioned “to-be” according to a strategic change vector

Leverage:
- App Store
- Service Catalog
- Architectural Models
- External Providers

Reuse

Agile Development

Plan & Execute “Spins” & Iterations

“Spins” (Strategic)

Type C Scrum

Innovation Cell

Iterations (Tactical)

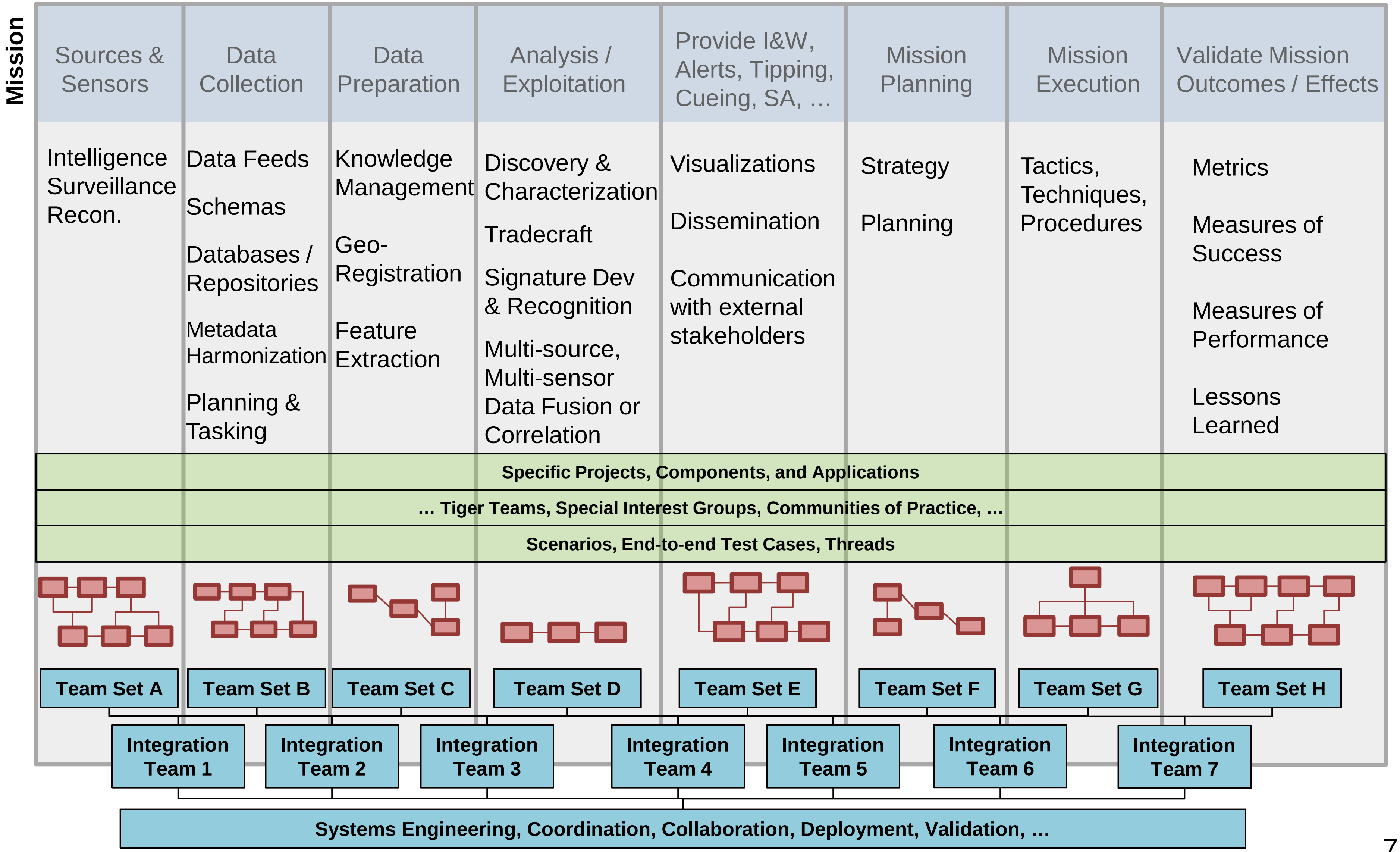
Stakeholder Involvement

Team Set A	Team Set B	Team Set C	Team Set D	Team Set E	Team Set F	Team Set G	Team Set H

Functional Description of “Mission” used to Organize and Allocate Requirements to Teams

Agile SD Methods, ©Will Menner

An Example based on Data Analysis Workflow



Mission

Sources & Sensors	Data Collection	Data Preparation	Analysis / Exploitation	Provide I&W, Alerts, Tipping, Cueing, SA, …	Mission Planning	Mission Execution	Validate Mission Outcomes / Effects
Intelligence Surveillance Recon.	Data Feeds Schemas Databases / Repositories Metadata Harmonization Planning & Tasking	Knowledge Management Geo-Registration Feature Extraction	Discovery & Characterization Tradecraft Signature Dev & Recognition Multi-source, Multi-sensor Data Fusion or Correlation	Visualizations Dissemination Communication with external stakeholders	Strategy Planning	Tactics, Techniques, Procedures	Metrics Measures of Success Measures of Performance Lessons Learned
Team Set A	Team Set B	Team Set C	Team Set D	Team Set E	Team Set F	Team Set G	Team Set H

Specific Projects, Components, and Applications

… Tiger Teams, Special Interest Groups, Communities of Practice, …

Scenarios, End-to-end Test Cases, Threads

Integration Team 1 | **Integration Team 2** | **Integration Team 3** | **Integration Team 4** | **Integration Team 5** | **Integration Team 6** | **Integration Team 7**

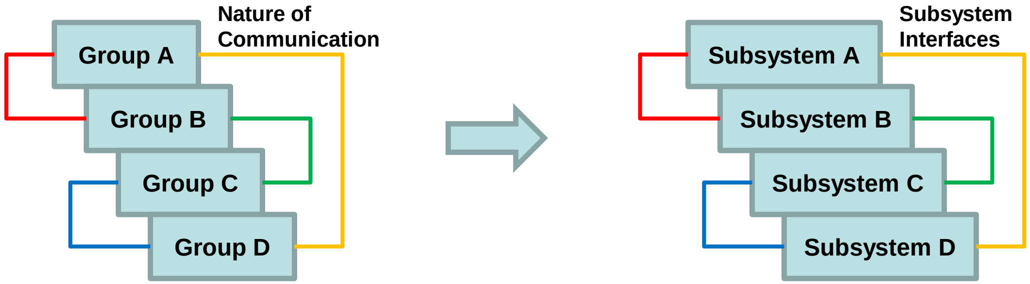
Systems Engineering, Coordination, Collaboration, Deployment, Validation, …

Conway’s Law

“Any organization that designs a system (defined more broadly here than just information systems) will inevitably produce a design whose structure is a copy of the organization's communication structure.” **- Melvin E. Conway, 1968, Datamation**

buzzfeed.com

“art imitates life”



- Why let your existing organization structure drive the way you tackle problems? (Especially if it’s not a good match.)
- Understand the problem space; then reshape your organization to efficiently attack it.

Summary

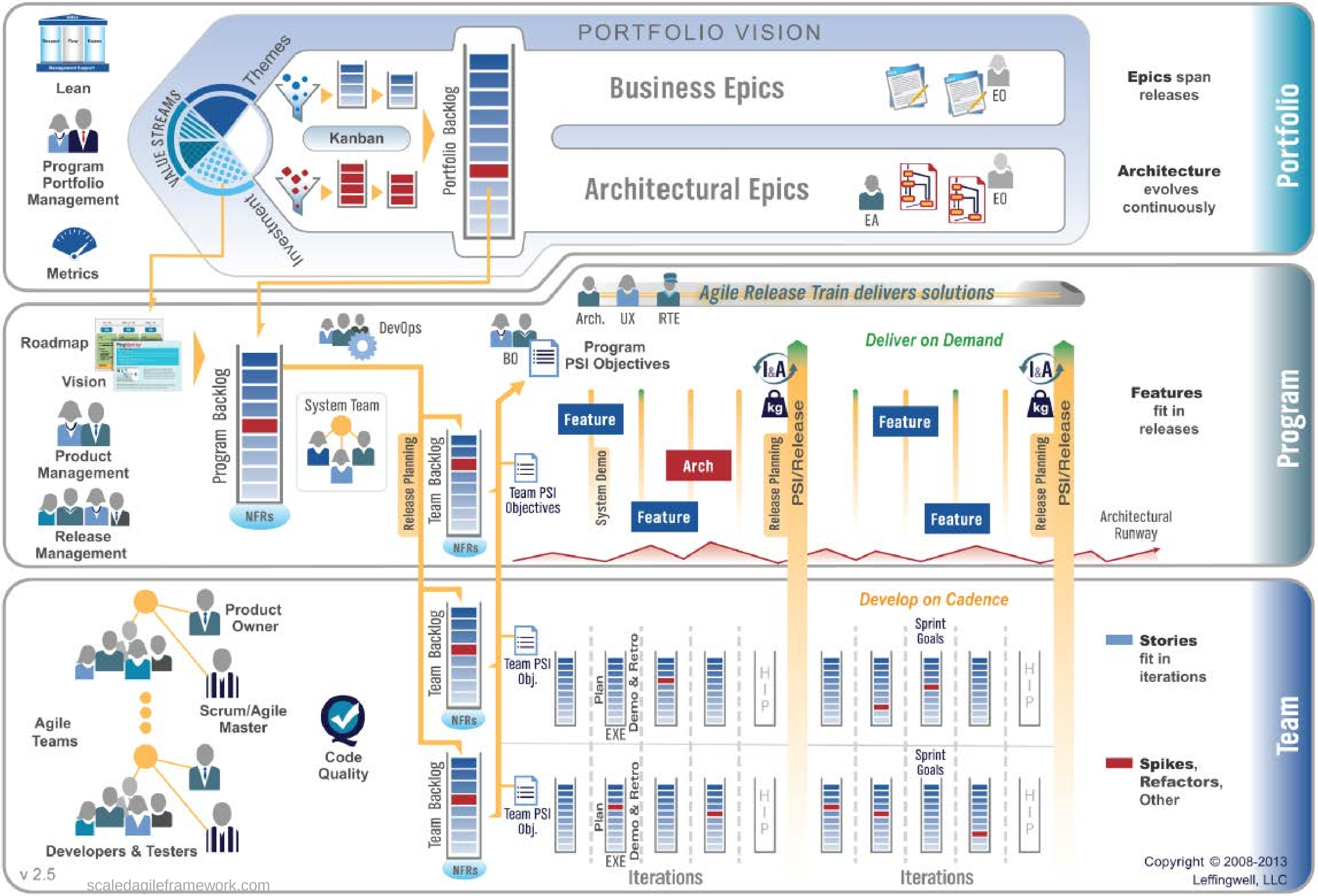
- Agile principles and practices extend well to large-scale development
- Product Backlogs become multi-dimensional
  - Architectural issues get extra attention
  - Research and technology development get extra attention
- Product Owners have a team
  - Systems engineers may help in many ways, including …
  - … spreading the vision, validating the product, and promoting reuse strategies
- Organizations need to flex their structures to match what is best for operations (not the other way around)
  - Allows better alignment and allocation of work to teams
  - Allows flexibility through the use of Type C Scrum

The End

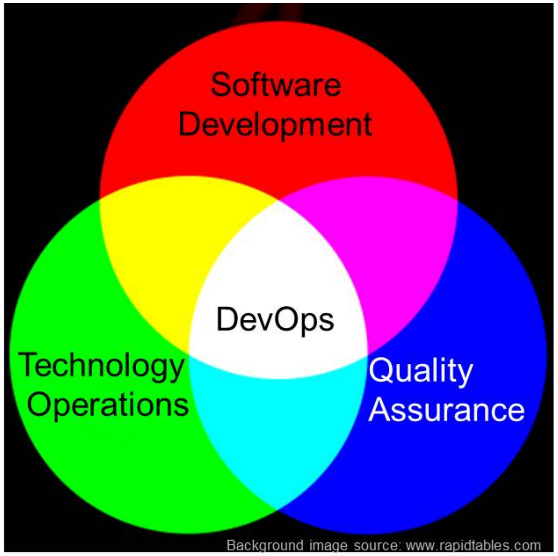
JOHNS HOPKINS
UNIVERSITY
THE JOHNS HOPKINS UNIVERSITY BALTIMORE 1876

Scaled Agile Framework (SAFe)

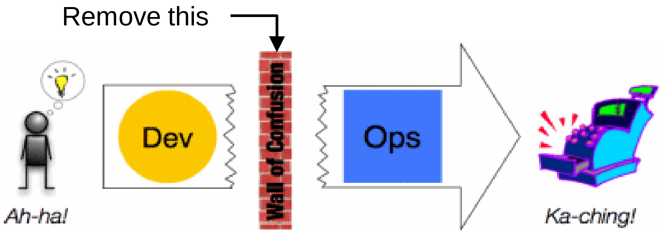
Backup
-
er



DevOps



Background image source: www.rapidtables.com



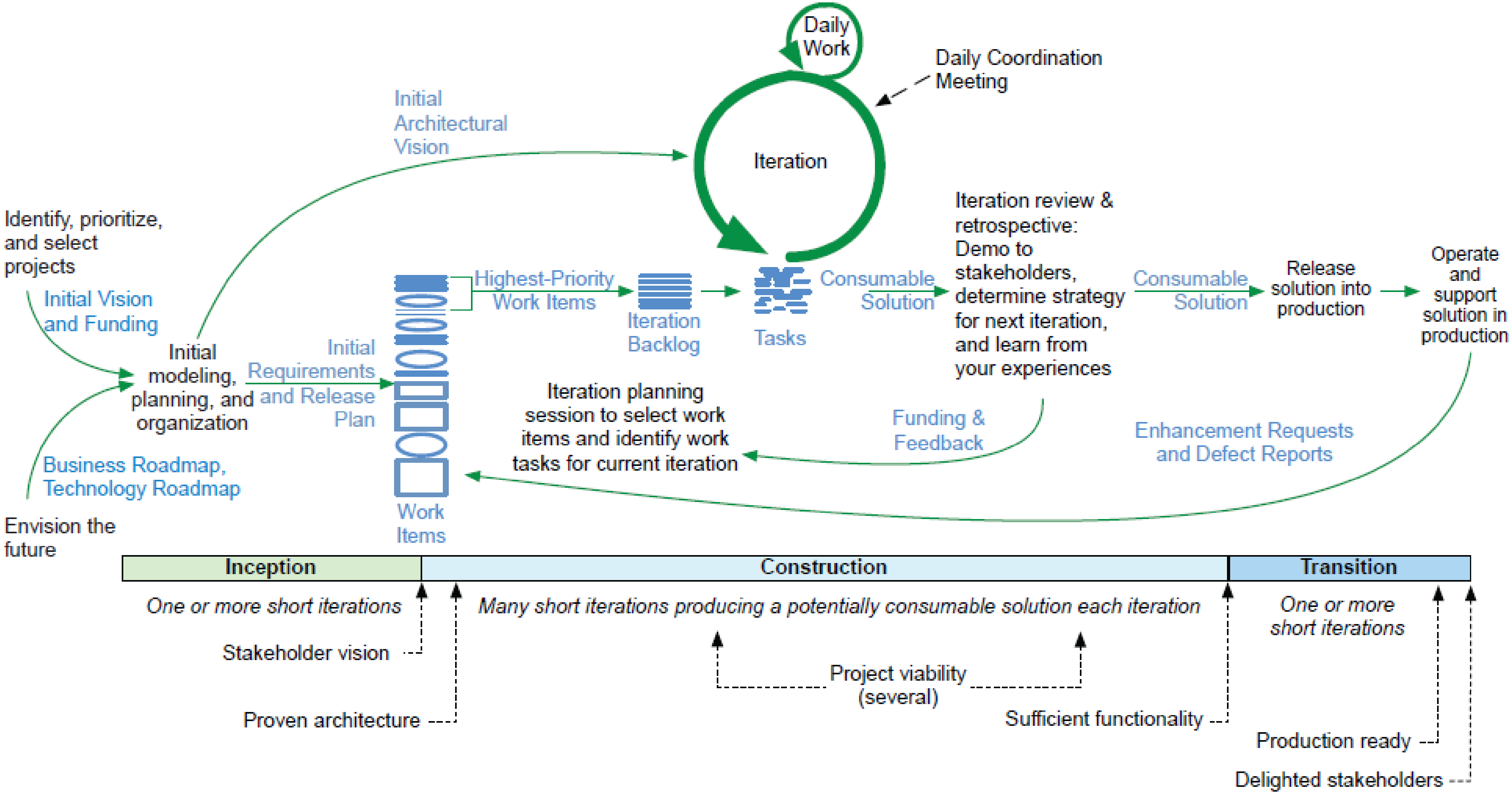
http://dev2ops.org/2010/02/what-is-devops

- DevOps involves the efficient integration of organizational functions to streamline the path between concept and cash
- Sustained innovation is one of the few ways left to gain competitive business advantage
- Organizations must quickly transform ideas into marketable and revenue-producing products
- Each functional unit must strive to no longer be a rate-limiting impediment on the product’s path from development to delivery

Agility needs to permeate the entire organization!

Disciplined Agile Delivery

Backup


Daily Work
Daily Coordination Meeting
Initial Architectural Vision
Iteration
Identify, prioritize, and select projects
Initial Vision and Funding
Initial modeling, planning, and organization
Initial Requirements and Release Plan
Business Roadmap, Technology Roadmap
Envision the future
Work Items
Highest-Priority Work Items
Iteration Backlog
Tasks
Consumable Solution
Iteration review & retrospective: Demo to stakeholders, determine strategy for next iteration, and learn from your experiences
Consumable Solution
Release solution into production
Operate and support solution in production
Iteration planning session to select work items and identify work tasks for current iteration
Funding & Feedback
Enhancement Requests and Defect Reports
Inception
Construction
Transition
One or more short iterations
Many short iterations producing a potentially consumable solution each iteration
One or more short iterations
Stakeholder vision
Proven architecture
Project viability (several)
Sufficient functionality
Production ready
Delighted stakeholders