The GRC/DevSecOps Nexus
Continuous Governance challenges,
as prerequisite to automated governance

Objective [**2 minutes**]:
- necessity of operational visibility and logistics-schema
  - as the foundation to horz scaling [individual to multi]
  - as prerequisite to vert scaling [to multi-division and to multi-organization]

Overview [**4 minutes**]:
- 1] history and evolution of development systems and process models
- 1] every continuous model is basically a Vee-model in time
- 1] perspective, plan, schema, specification, horizontal/vertical scaling, RACI, RBAC/visibility, merit
- 1] clockwise vs counter-clockwise, GAMP-5 Vee-model

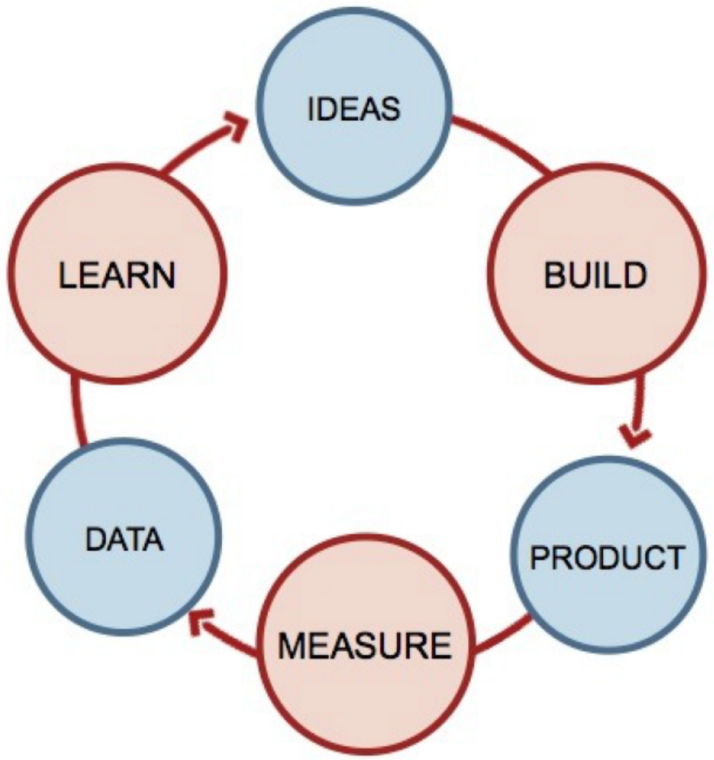
Vee-models and SAFe [**1 minute**]:
- 2] System Arch Vee [What, Why, and Who], Entity Vee [How], and Dual-Vee [V&V] [User:Greghc; Wikipedia]
- 3] Design-to and Build-to realization iterations, Multi-Vee; [User:Greghc; Wikipedia]
- 4] Silos in SAFe still have orthogonal opaque inputs:
  - Portfolio Flow > Solution Train Backlog
  - Value Stream Management > [Solution Demo, Continuous Delivery Pipeline, DevSecOps]
  - Lean UX > Team Backlogs

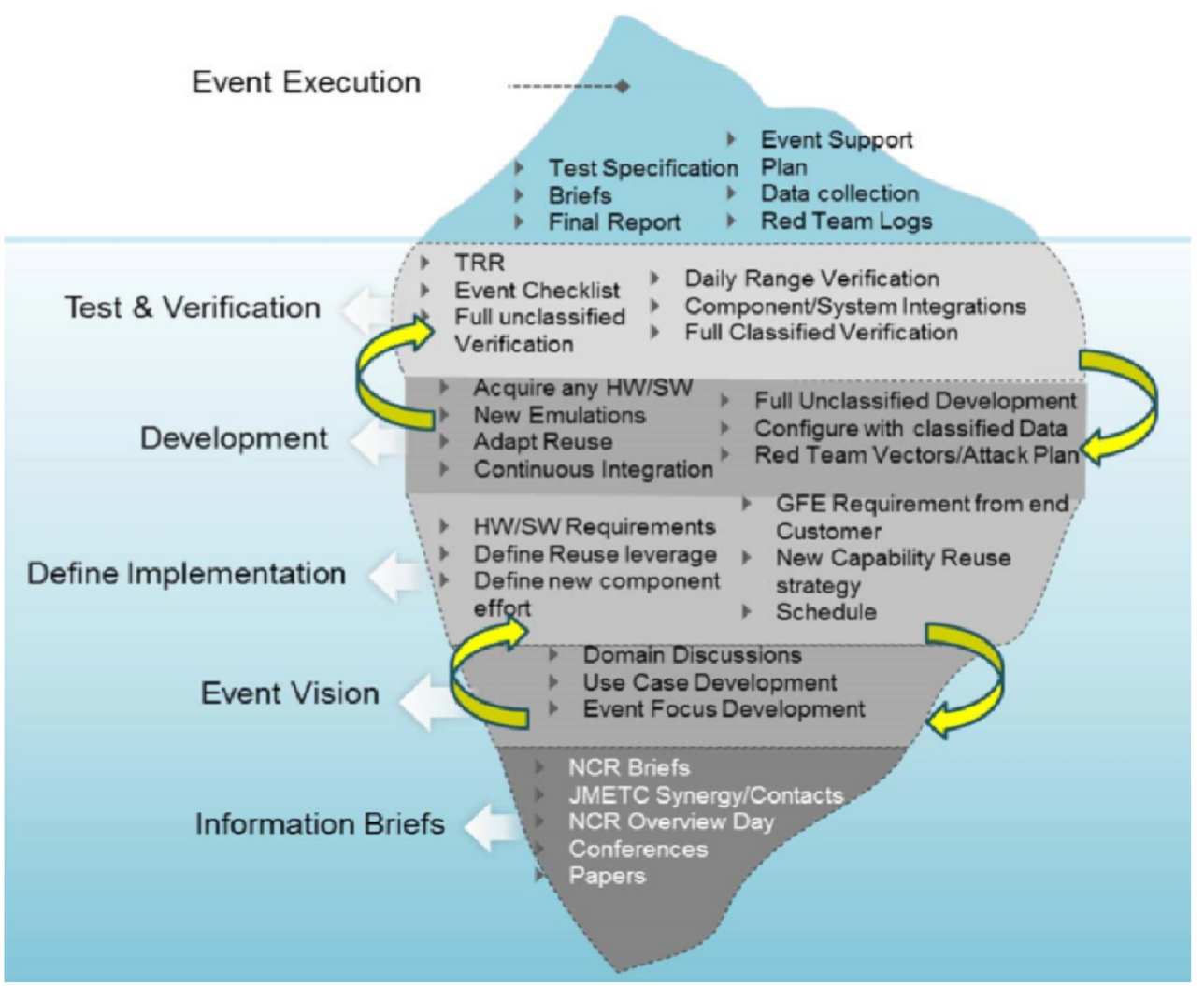
Continuous Governance [**2 minutes**]:
- 5] Planning Must-Not-Exclude, lexicon of GRC from perspectives of Regulatory, Representation, Implementation, and Quality
- 5] From stack of turtles [beginning] to automation celebration [end]
- 5] a manual process improved through iterations, implies automation as a consequence of development
- 5] automation needs programming, needs API, needs logistics schema,
  needs digitization, needs systems architecture, needs function/data flow
- 5] Bidirectional arrows represent development; direction changes with transition from manual push to automated pull
- 5] Manual one-way logistics pull @ authorization, transitions to authorization of automation, and logistics push @ quality gate
- 5] Questions: [**8 minutes**]:

Low-Hanging Fruit Challenge [**4 minutes**]:
- 6] Enhancing NIST Document Metadata Access
- 6] Questions

Deck:
- notes: https://github.com/georgalis/pub/tree/main/know/control/GRC_DevOps_Nexus.yml
- slides: https://github.com/georgalis/pub/tree/main/know/control/GRC_DevOps_Nexus.pdf

DevOps





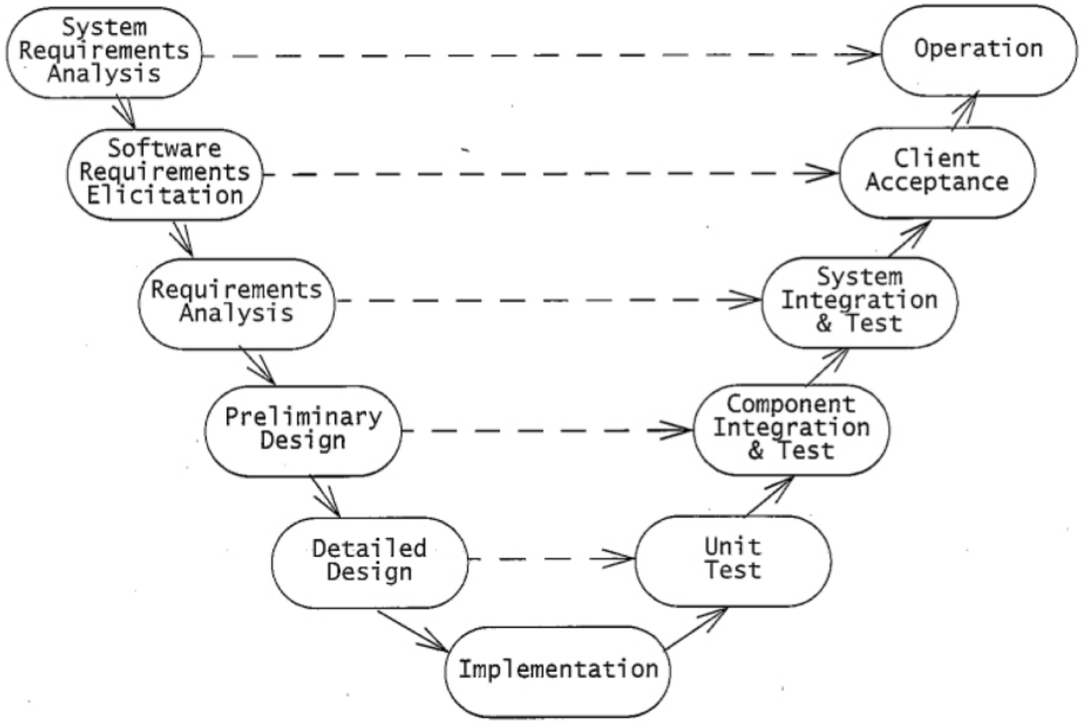
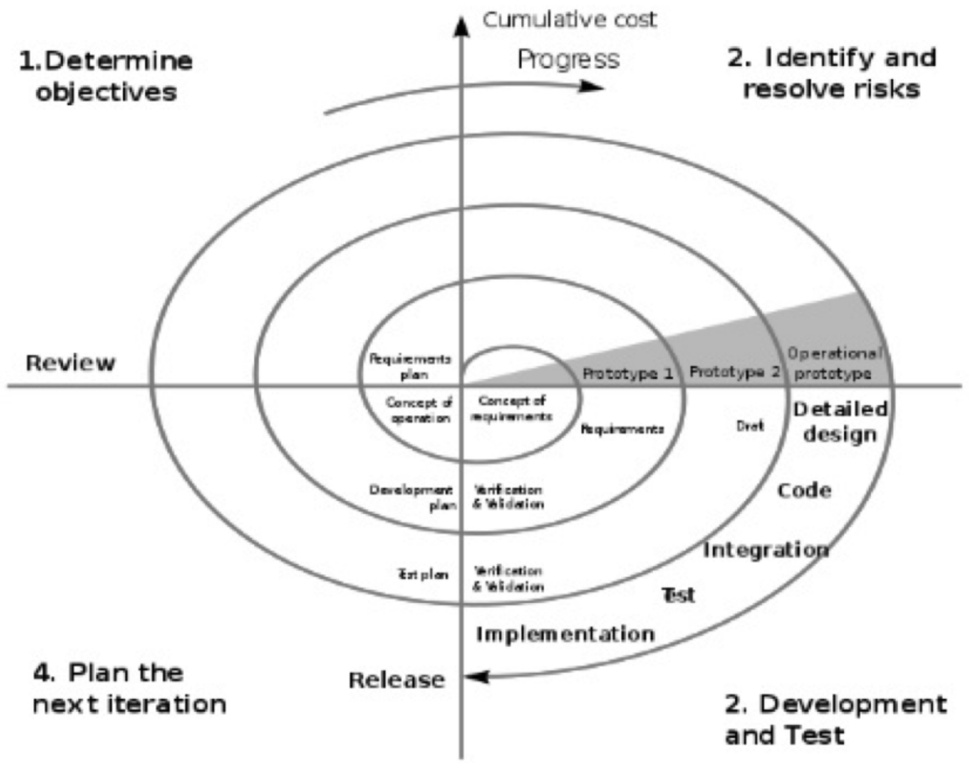


Figure 15-9 V-model of software development (UML activity diagram; adapted from [Jensen & Tonies, 1979]). The horizontal object flow denotes the information flow between activities at the same abstraction level. The V-shape layout of the activities was conserved to reflect the original drawing. However, the layout of the activities has no semantics in UML.

Spiral Model (Boehm, 1988)



Architecture Development Vee

System Development

Integration, Verification, & Validation Planning

Solution/System Realization

Subsystem Development

Integration, Verification, & Validation Planning

Subsystem Realization

I, V, and V Planning

LCI Lowest Configuration Item Development

LCI Lowest Configuration Item Realization

Architecture Decomposition & Definition

Architecture Integration & Verification

Entity Vee

Stakeholder Requirements and Implementation Context

Customer Confirmation

Validation

Define Entity Requirements (Behavior and Performance)

Verification and Validation Planning

Validation Preparation

Verification

PDR

Customer Confirmation

Customer Confirmation

Concept & Architecture Selection and Design-to Specification

Verification- *Inspection, Test, Demonstration, Analysis*

CDR

Definition Sequence and Elaboration of Detail

Build-to and Code-to Artifacts

Verification Planning

Verification- *Inspection, Test, Demonstration, Analysis*

Assembly and Performance Assurance Sequence

Opportunity and Risk Investigation

Buy, Build, Code

Anomaly Investigation

Solution Realization

Off-core user discussions and approvals (In-process validation)

"Are the proposed baselines acceptable?"

Approved Baseline

Baseline Being Considered

Baseline Verification

"How to prove it's built right"

Planned Verification

Baselines to be Considered

Baselines to be Verified

"Time Now" (Vertical Line) With upward and downward iterations as required

Core of the Vee

Plans, specifications, and products are under progressive configuration management

Time and Baseline Maturity

Off-Core opportunity & risk management investigations and actions

"How are the opportunities and risks of the proposed baselines being resolved?"

Off-Core User Approval of Baseline and Baseline Modification

"Is the verified performance acceptable?"

Baselines to be Verified

Defined System Performance

Baseline Validation

"Are we building the right thing?"

Defined Entity Performance

Baseline Being Verified

Baseline Verification

"Are we building it right?"

Core of the Vee

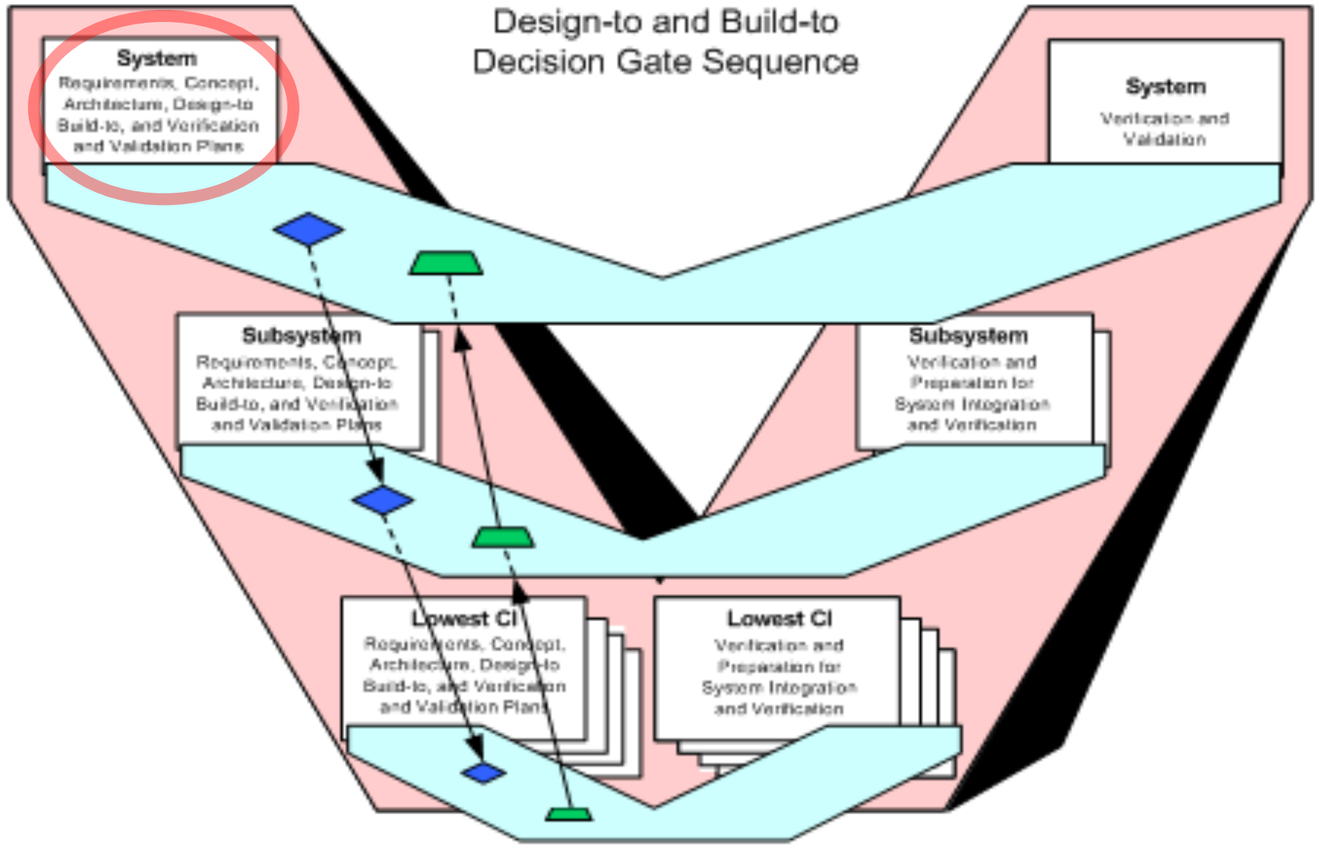
Plans, specifications, and products are under progressive configuration management

Baselines Verified

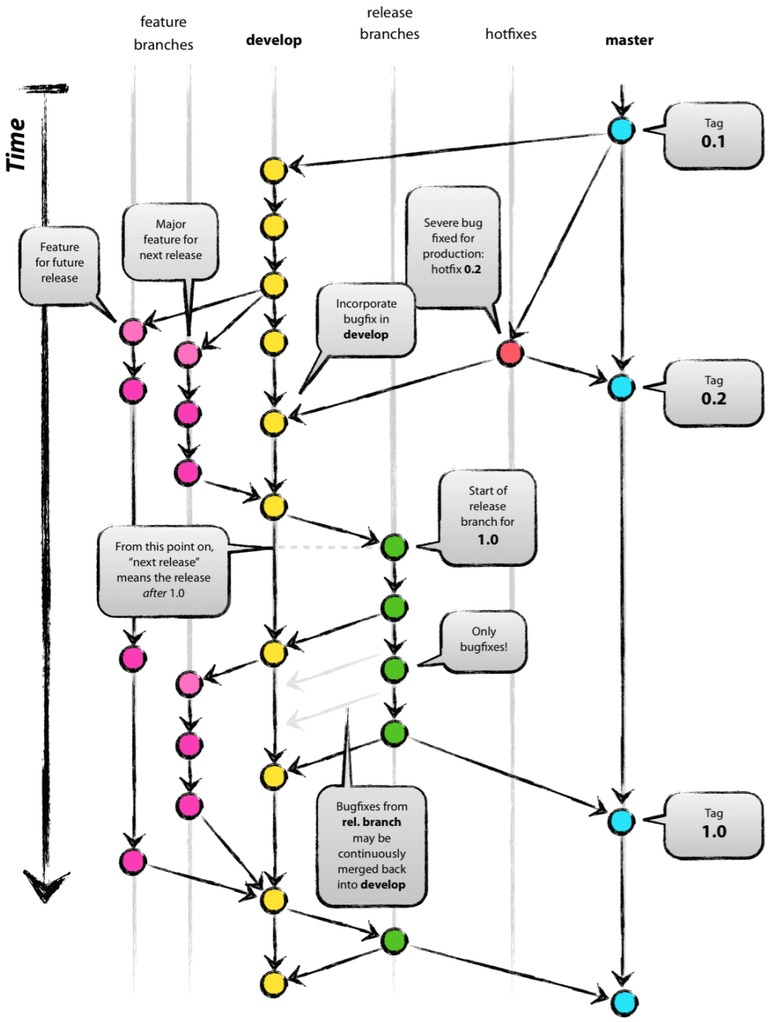
Time and Baseline Maturity

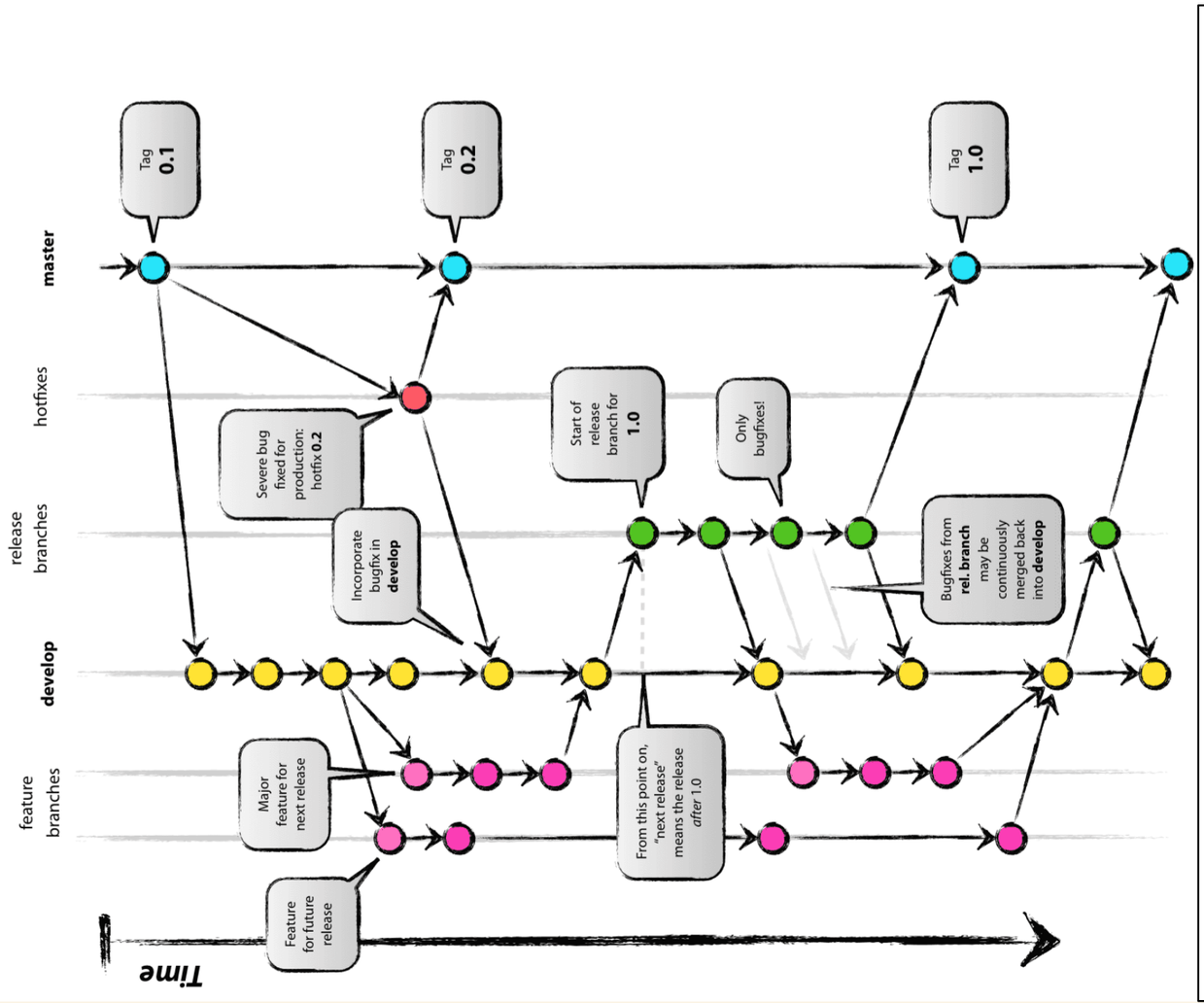
"Off-Core" Verification Problem Investigation and Resolution

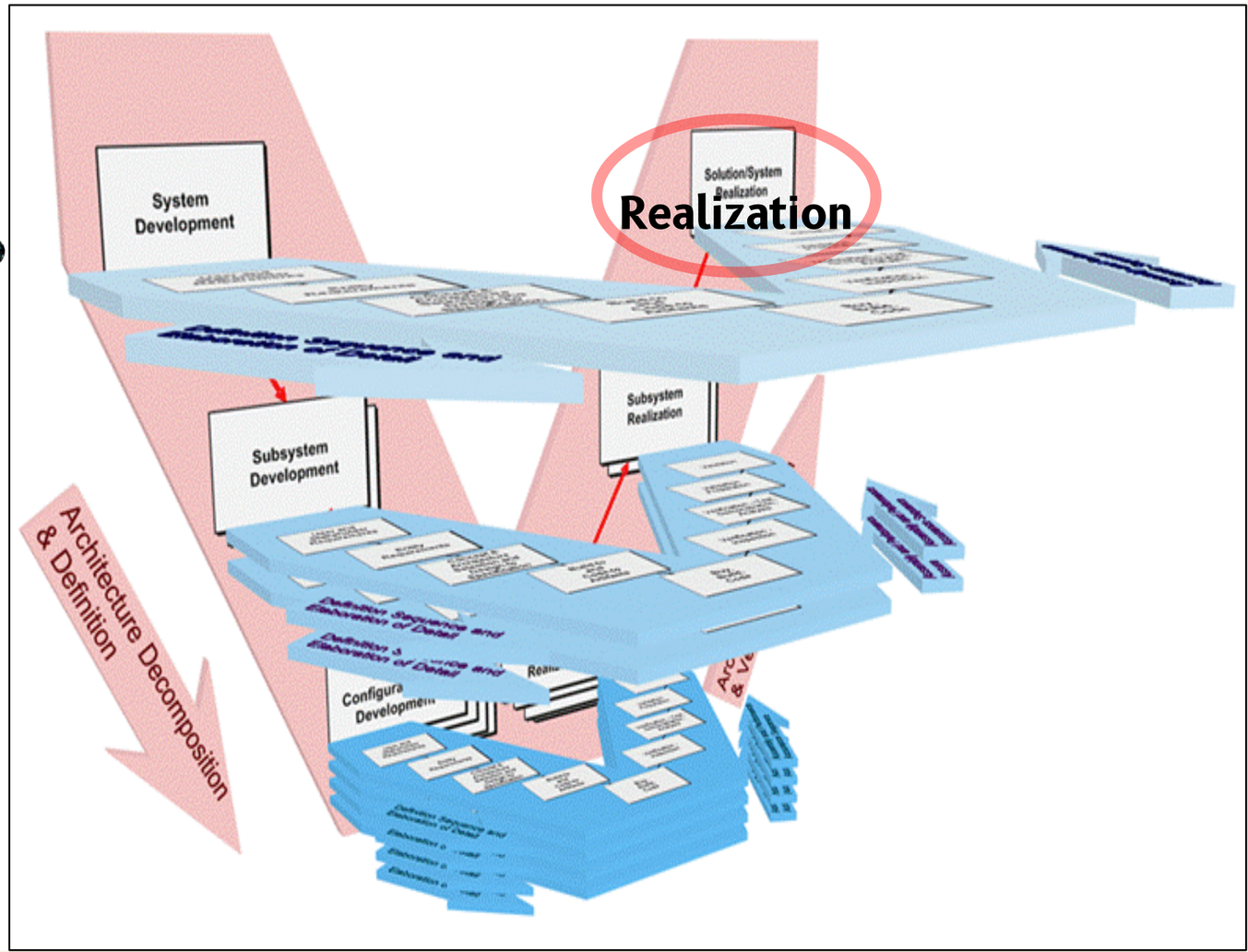
"Is the problem cause understood?"



- Design-to (Preliminary Design Review, PDR)
- Build-to (Critical Design Review, CDR).







SAFe 6.0

Select SAFe configuration

OVERVIEW | ESSENTIAL | LARGE SOLUTION | PORTFOLIO | FULL

Organizational Agility

Enterprise | Government

Operational Value Streams

Lean Portfolio Management

Epic Owners | Enterprise Architect

Enterprise Solution Delivery

Solution Mgmt | Solution Architect | STE

Agile Product Delivery

Business Owners

Product Mgmt | System Architect | RTE

Team and Technical Agility

Agile Teams

Product Owner | Scrum Master / Team Coach

Business & Technology

BUSINESS AGILITY

Portfolio Flow

PORTFOLIO

Strategic Themes | Portfolio Vision | NFRs | Portfolio Backlog | Big Data | PB | Lean Budgets | Guardrails | Value Stream Management | Epic | Enabler | Epic | Coordination | Solutions | KPIs | Development Value Streams

Solution Train Flow

LARGE SOLUTION

WSJF | Variable | Fixed | Compliance | MBSE | Set-Based | NFRs | Solution Train Backlog | Solution Demo | Pre-Plan | Capability | Enabler | Coordinate and Deliver | Supplier | Solution

ART Flow

ESSENTIAL

Customer Centricity | Design Thinking | Lean UX | WSJF | NFRs | ART Backlog | Continuous Delivery Pipeline | AGILE RELEASE TRAIN | Continuous Exploration | Continuous Integration | Continuous Deployment | Release on Demand | Solution | Solution Context

Team Flow

SAFe Scrum | Built-In Quality | SAFe Team Kanban | NFRs | Team Backlogs | System Demos | Enabler | Feature | Story | Enabler | Iteration | PI Planning | CD | CI | Iterations | PI | PI Objectives | Cloud | Sec | DevOps | Architectural Runway

Leffingwell, et al. © Scaled Agile, Inc.

Vision | OKRs | Roadmap | AI | Shared Services | CoP | System Team | Measure & Grow

Lean-Agile Leadership

Lean-Agile Mindset | Core Values | SAFe Principles | Implementation Roadmap | SPC | Continuous Learning Culture

Operational Security and Scale Planning: Must-Not-Exclude

Automation Pipeline

Current systems use **Continuous Deployment** strategies, version control, platform templating, automation frameworks, and dynamic configuration generation. However, scaling presents challenges due to complex requirements, orthogonal cross-functional mandates, and evolution of short-term tactical solutions to problems.

Continuous Integration ties quality checks to design controls and implementation. To address scaling challenges, elevate the automation framework from the deployment pipeline to a cycle that includes:

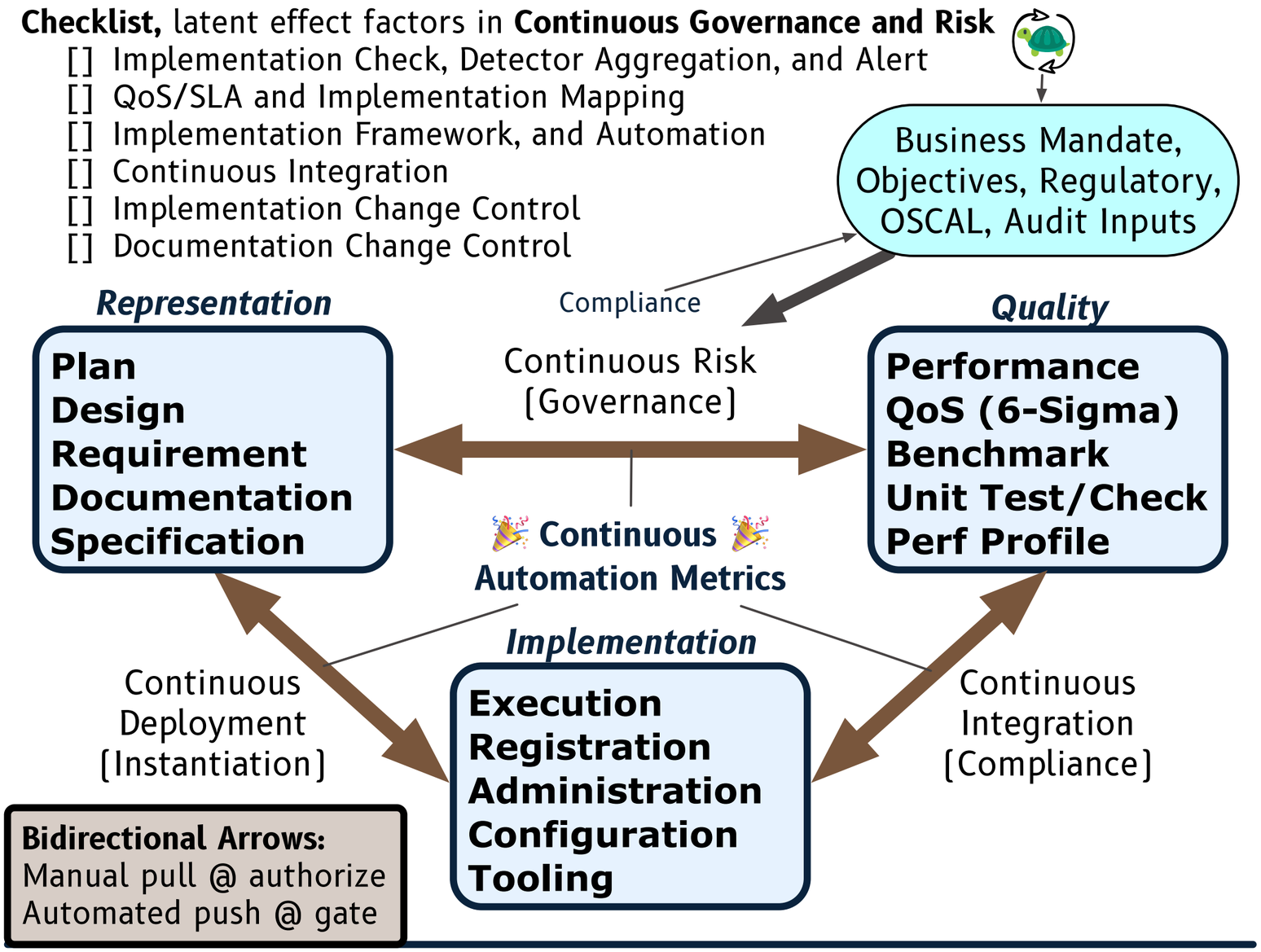
1 Implementation Versioning
2 Selective merge rollback capabilities
3 Version QoS metric comparison

This approach integrates SLA quality metrics with version control, into the automation framework, for change management, not simply as a stop gate. Enabling continuous review and management of complex factors, over time, in deployment at scale.

Change Management / QoS SLA Integration / Metric Mapping

Checklist, latent effect factors in **Continuous Governance and Risk**

- [] Implementation Check, Detector Aggregation, and Alert
- [] QoS/SLA and Implementation Mapping
- [] Implementation Framework, and Automation
- [] Continuous Integration
- [] Implementation Change Control
- [] Documentation Change Control



Bidirectional Arrows:
Manual pull @ authorize
Automated push @ gate

Lean Quality Engineering

The proposed lean approach to operational security and scaling aims to integrate quality metrics, version control, and automation frameworks more tightly into the continuous governance, design, integration, and deployment process. This strategy addresses well-known challenges that arise from division coordination difficulties and incompatibilities, which collectively lead to organizational value degradation. The pattern is designed to be scalable, applicable to both single-person organizations and large enterprise, empowering durable architecture and design through anticipation of future scaling, governance, and regulatory inputs.

Solution Strategy

Representation/Documentation Change Control:
Revision [git controls]
Authorization [nonrepudiation, manual/crypto signature/gate]
Catalog [theme, schema, index, and cross-reference]
Mapping [filename, data, revision and QoS/SLA metric correlation]
Review [business value, risk, feasibility analysis]

Continuous Inputs
Digitization [RBAC visibility and RACI]
Logistics [Specification, Schema, API, Data]

Continuous Management
Automation [Repeatability, detection, versioned quality metrics]

Low-Hanging Fruit : Enhancing NIST Document Metadata Access

- NIST lacks a simple, standardized schema for document metadata access
- No efficient method for continuous local ingestion of NIST catalog data
- The potentially out-of-scope curiosity of https://csrc.nist.gov/Projects/olir/
- Seeking collaborators to formalize recommendation for NIST implementation

Capture highlighted data as JSON supplement to existing pdf publication workflows.

NIST Documentation Catalog Logistics

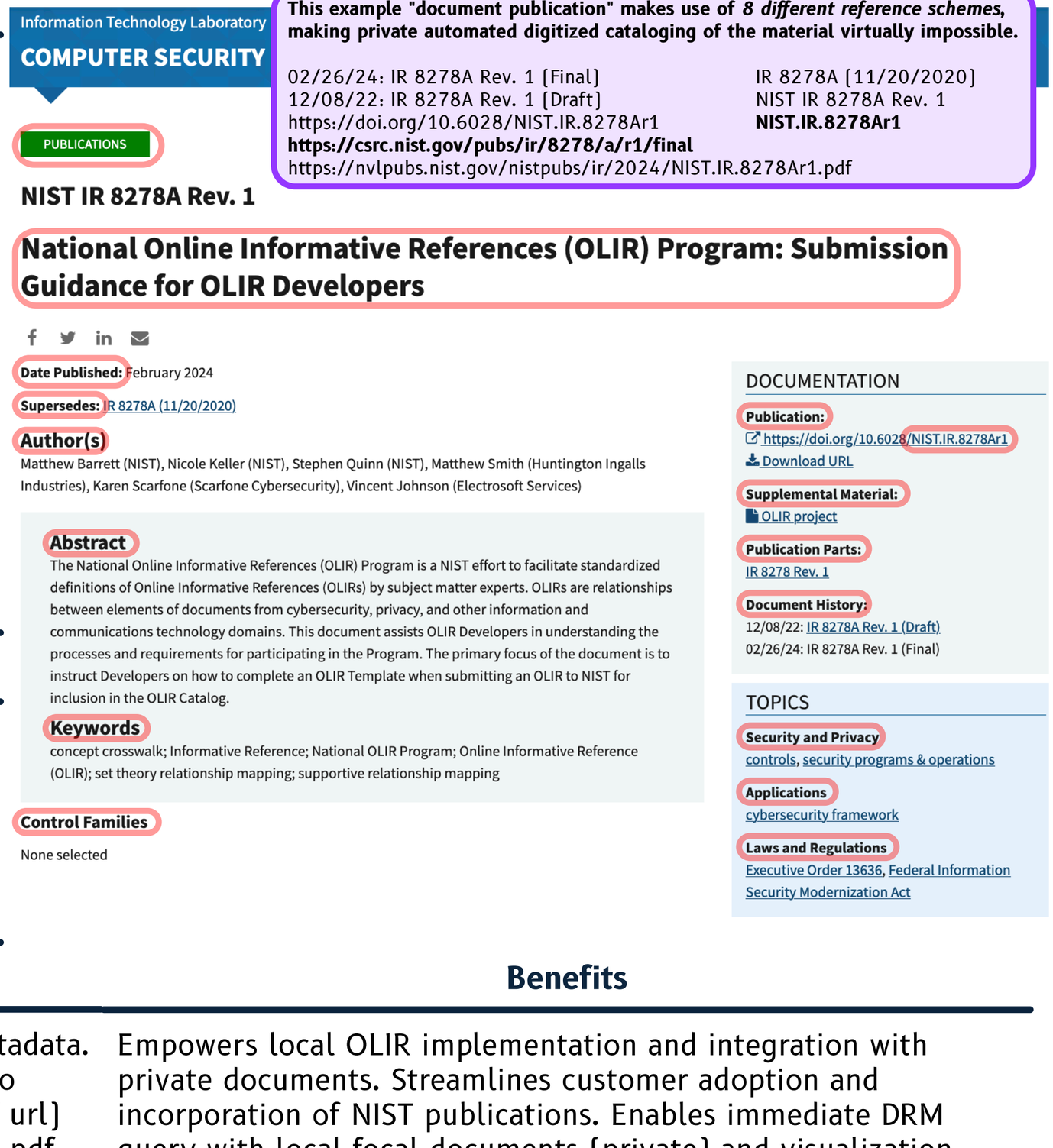
The National Online Informative References [OLIR] Program is a NIST effort to facilitate subject matter experts [SMEs] in defining standardized online informative references [OLIRs] between elements of their documents, products, and services and elements of NIST documents [eg CSF, RMF, OSCAL, etc]

The primary focus of the National Online Informative References [OLIR] Program is:

- **Informative Reference Catalog**: Reference Data Informative References and Derived Relationship Mappings [DRMs]
- **Derived Relationship Mapping** [DRMs]: Analysis Tool provides Users the ability to generate DRMs for Reference Documents with [NIST] Focal Document of the Users' choice.
- **Guidance for OLIR Developers**: how to complete an OLIR Template when submitting an OLIR to NIST for inclusion in the OLIR Catalog.

Key Considerations

NIST's centralized approach impedes focal mapping from private documents to standards. Lack of digitized metadata hinders adoption. Current model limits integration with private systems. Supplement NIST-provided tools with consumer-driven data access, enables efficient mapping and integration of NIST standards in adopters' contexts.



Information Technology Laboratory
COMPUTER SECURITY

This example "document publication" makes use of *8 different reference schemes*, making private automated digitized cataloging of the material virtually impossible.

02/26/24: IR 8278A Rev. 1 [Final]
12/08/22: IR 8278A Rev. 1 [Draft]
https://doi.org/10.6028/NIST.IR.8278Ar1
https://csrc.nist.gov/pubs/ir/8278/a/r1/final
https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8278Ar1.pdf

IR 8278A [11/20/2020]
NIST IR 8278A Rev. 1
NIST.IR.8278Ar1

PUBLICATIONS

NIST IR 8278A Rev. 1

National Online Informative References (OLIR) Program: Submission Guidance for OLIR Developers

Date Published: February 2024

Supersedes: IR 8278A (11/20/2020)

Author(s)
Matthew Barrett (NIST), Nicole Keller (NIST), Stephen Quinn (NIST), Matthew Smith (Huntington Ingalls Industries), Karen Scarfone (Scarfone Cybersecurity), Vincent Johnson (Electrosoft Services)

Abstract
The National Online Informative References (OLIR) Program is a NIST effort to facilitate standardized definitions of Online Informative References (OLIRs) by subject matter experts. OLIRs are relationships between elements of documents from cybersecurity, privacy, and other information and communications technology domains. This document assists OLIR Developers in understanding the processes and requirements for participating in the Program. The primary focus of the document is to instruct Developers on how to complete an OLIR Template when submitting an OLIR to NIST for inclusion in the OLIR Catalog.

Keywords
concept crosswalk; Informative Reference; National OLIR Program; Online Informative Reference (OLIR); set theory relationship mapping; supportive relationship mapping

Control Families
None selected

DOCUMENTATION

Publication:
https://doi.org/10.6028/NIST.IR.8278Ar1
Download URL

Supplemental Material:
OLIR project

Publication Parts:
IR 8278 Rev. 1

Document History:
12/08/22: IR 8278A Rev. 1 (Draft)
02/26/24: IR 8278A Rev. 1 (Final)

TOPICS

Security and Privacy
controls, security programs & operations

Applications
cybersecurity framework

Laws and Regulations
Executive Order 13636, Federal Information Security Modernization Act

Solution

Digitize and publish existing catalog release data as JSON metadata. Implement URL-based access to metadata, map document ID to release pdf and metadata json [eg, .json url derived from .pdf url]
- https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8278Ar1.pdf
- https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8278Ar1.json

Enable programatic access to resources according to digitized ID.

Benefits

Empowers local OLIR implementation and integration with private documents. Streamlines customer adoption and incorporation of NIST publications. Enables immediate DRM query with local focal documents [private] and visualization using common tools [e.g., jq, graphviz]. Requires minimal adjustments to existing NIST publication syndication processes.

references:
- **Dual Vee Model:**
  - **https://en.wikipedia.org/wiki/User:Greghc/Dual_Vee_Model**
  - User:Greghc/Dual Vee Model
  - the power of the Dual Vee Model when applied as a reminder model for development of complex systems
- A successful Git branching model:
  - http://nvie.com/posts/a-successful-git-branching-model/
  - By Vincent Driessen on Tuesday, January 05, 2010
- SAFe Lean-Agile Principles:
  - https://scaledagileframework.com/safe-lean-agile-principles/
  - SAFe is based on ten immutable, underlying Lean-Agile principles
- **GAMP-5 detail and history:**
  - **https://www.slideshare.net/slideshow/gamp5-new/17385991**
  - Gamp5 new; Mar 19, 2013; Kalpeshkumar Vaghela
  - GAMP 5 provides guidance for computerized systems validation. It focuses on risk-based approaches and scalability of efforts based on a system's risk, complexity, and novelty. GAMP 5 also emphasizes leveraging supplier activities and avoiding duplication of efforts. The document provides a framework for life cycle activities from concept to retirement, including planning, specification, development, operation, and retirement of computerized systems.
- **Meta study, organizing framework that clarifies the topology of the literature:**
  - **https://www.researchgate.net/publication/318361926_Process_models_in_design_and_development**
  - Process models in design and development
  - April 2018; Research in Engineering Design 29(2):161-202
  - DOI:10.1007/s00163-017-0262-7; David C. Wynn; P. John Clarkson
- Using V Models for Testing:
  - https://insights.sei.cmu.edu/sei_blog/2013/11/using-v-models-for-testing.html
  - "Using V Models for Testing". insights.sei.cmu.edu. Software Engineering Institute, Carnegie Mellon University. 11 November 2013.
  - The verification and validation of requirements are a critical part of systems and software engineering
- Vee Model of Systems Engineering to System Dynamics Modeling:
  - https://www.researchgate.net/publication/255593138_A_Case_Study_in_Application_of_Vee_Model_of_Systems_Engineering_to_System_Dynamics_Modelling_of_Dryland_Salinity_in_Australia
  - A Case Study in Application of Vee Model of Systems Engineering to System Dynamics Modeling of Dryland Salinity in Australia; January 2008; Naeem U Khan; Alan Mclucas
- **Systems engineering:**
  - **https://en.wikipedia.org/wiki/Systems_engineering**
  - Systems engineering is an interdisciplinary field of engineering and engineering management
- National Online Informative References Program (OLIR):
  - https://csrc.nist.gov/Projects/olir

Ancillary:

LLM alignment of GAMP with DevOps and DevSecOps :

Focus on GAMP-5 and GAMP-6 systems maturity and their alignment with DevOps and DevSecOps practices: |

GAMP-5 and GAMP-6 represent evolving approaches to system maturity in regulated industries, particularly in pharmaceutical and life sciences sectors. Their principles increasingly align with modern DevOps and DevSecOps practices, reflecting the industry's shift towards more agile and integrated methodologies.

GAMP-5, released in 2008, introduced concepts that parallel certain aspects of DevOps:

1. Risk-based approach: Similar to DevOps' continuous risk assessment.
2. Scalable lifecycle model: Aligns with DevOps' iterative development cycles.
3. Supplier assessment: Corresponds to DevOps' focus on integrating third-party tools securely.
4. Emphasis on automated testing: A core principle in DevOps practices.

GAMP-6, released in 2022, further bridges the gap between traditional validation approaches and modern software development practices:

1. Agile and DevOps integration: Explicitly acknowledges and incorporates Agile and DevOps methodologies.
2. Continuous compliance: Aligns with DevOps' continuous integration and delivery (CI/CD) pipelines.
3. Data integrity by design: Parallels DevSecOps' principle of security by design.
4. Critical thinking approach: Resonates with DevOps culture of continuous improvement and learning.

The alignment of GAMP-6 with DevOps and DevSecOps is more pronounced, reflecting the evolving nature of system maturity in regulated environments:

1. Continuous Validation: GAMP-6 supports continuous validation, which aligns closely with DevOps' continuous integration and deployment practices.
2. Automated Testing and Verification: Both GAMP-6 and DevOps emphasize automated testing to ensure quality and compliance throughout the development lifecycle.
3. Risk Management: GAMP-6's risk-based approach complements DevSecOps' focus on continuous risk assessment and mitigation.
4. Collaboration and Communication: GAMP-6 encourages cross-functional collaboration, mirroring the DevOps principle of breaking down silos between development and operations teams.

By adopting GAMP-6 principles alongside DevOps and DevSecOps practices, organizations can achieve:

1. Increased efficiency in validation processes
2. Improved quality and compliance in software development
3. Better alignment between IT and quality assurance teams
4. Faster time-to-market while maintaining regulatory compliance
5. Enhanced ability to adapt to changing regulatory requirements

The evolution from GAMP-5 to GAMP-6 demonstrates a clear trend towards greater alignment with modern software development practices. This progression allows regulated industries to benefit from the agility and efficiency of DevOps and DevSecOps while maintaining the rigorous quality and compliance standards required in their sectors.

Organizations looking to maximize their system maturity should consider integrating GAMP-6 principles with DevOps and DevSecOps methodologies. This integrated approach can lead to more robust, compliant, and efficient software development processes that meet both regulatory requirements and modern development standards.