OR-001 — Operation by Identified User Categories

Go to Operational Requirements

Original Requirement

> *The system SHALL support operation by:*
>
> > *Developers*
> > *DevSecOps Engineers*
> > *Platform Engineers*
> > *System Administrators*
> > *Security Engineers*
> > *Compliance Officers*[C1]

Assessment of Original Requirement

The Original Requirement preserves the approved intent that the identified categories of users participate in Crucible operation, but it does not express independently testable normative statements.

The following Specification Discipline and Authoring findings apply:

- **The system** does not use the defined system name Crucible
- **Support** is a weak verb that does not identify an observable Crucible behavior
- **Operation** is ambiguous because it does not identify the actions performed by the listed user categories
- **Operation by** does not distinguish whether a user category requests, initiates, configures, administers, executes, approves, reviews, monitors, observes, or responds to a Crucible activity
- The Original Requirement does not identify which Crucible operations apply to each listed user category
- The Original Requirement does not identify whether every listed user category participates in the same operations
- The Original Requirement does not identify the interfaces through which each listed user category interacts with Crucible
- The Original Requirement does not identify the inputs provided by each listed user category
- The Original Requirement does not identify the outputs received or reviewed by each listed user category
- The Original Requirement does not identify an acceptance condition for operation by each listed user category
- The Original Requirement combines six independently testable user categories within one normative statement
- Crucible can support operation by one listed user category while failing to support operation by another
- The Original Requirement does not provide separate requirement identifiers for verification and

traceability of each listed user category

The word **operation** is grammatically valid as a mass noun describing the general act of operating Crucible. For requirements purposes, however, the term does not identify an observable action.

Changing **operation** to **operations** would indicate that multiple actions exist, but pluralization alone would not identify those actions.

The Original Requirement identifies the following categories of users:

- Developers
- DevSecOps Engineers
- Platform Engineers
- System Administrators
- Security Engineers
- Compliance Officers

The Original Requirement does not classify these user categories as Operational Roles and does not require:

- A particular role model
- A role-assignment mechanism
- A role-based authorization model
- A particular access-control model
- Allocation of permissions through roles
- Multiple-role assignment
- Role-assignment removal
- Recording of an authorizing role

These concepts should not be introduced as derived requirements unless the Concept of Operations, architecture, security model, or another controlling source establishes them.

The Original Requirement therefore requires:

- Clarification of the intended meaning of **operation**
- Identification of the Crucible operations applicable to each listed user category
- Separation of the six independently testable user categories
- Reuse of existing requirements wherever those requirements already define the applicable operations

The decomposition preserves `OR-001` as the stable parent requirement identifier. Each replacement requirement receives a lettered identifier and a separate requirement page.

ConOps Relationship

The Crucible Concept of Operations [C2] provides operational context for interpreting the ambiguous word **operation** in the Original Requirement.

The Concept of Operations identifies the following high-level Crucible operations under **Build, Capture, Deploy**:

- **Build** — Crucible builds Machine Images while applying the applicable image-hardening configuration
- **Capture** — Crucible captures dependencies and compliance findings into managed storage
- **Deploy** — Crucible provisions infrastructure and executes the applicable pre-deployment and post-deployment steps

The Concept of Operations also addresses:

- Phase 1 command-line operation
- CI/CD pipeline integration
- Image-layer baseline composition
- Infrastructure baseline composition
- Machine Image construction
- Dependency capture
- Compliance-finding capture
- Infrastructure deployment
- Connected-to-disconnected transfer
- Offline package-repository population
- Compliance Evidence generation
- Phase 2 web management
- Strategic native execution

The Concept of Operations therefore supports interpreting **operation** as participation in one or more identifiable Crucible operations rather than as an undefined general capability.

The cited ConOps description assigns execution of Build, Capture, and Deploy to Crucible. It does not identify how each listed user category participates in those operations.

For example, **operation by** a listed user category could mean that Crucible:

- Accepts a request to perform an operation from the user
- Accepts Controlled Inputs from the user
- Initiates an operation in response to a user request
- Presents operation status to the user
- Presents the result of a completed operation to the user
- Presents captured dependencies to the user
- Presents compliance findings to the user
- Presents generated Artifacts or Evidence to the user
- Accepts a decision, approval, disposition, or other response from the user

These examples are derived interpretations. Neither the Original Requirement[C1] nor the cited Concept of Operations[C2] expressly assigns these interactions to a particular listed user category.

The requirement owner should clarify:

- Whether **operation** means participation in Build, Capture, and Deploy
- Whether additional Crucible operations identified elsewhere in the Concept of Operations apply
- Which Crucible operations apply to each listed user category
- Whether each user category requests, initiates, configures, administers, executes, approves, reviews, monitors, observes, or responds to each applicable operation
- Which interfaces support each user interaction
- Which inputs each user category provides

- Which outputs each user category receives or reviews
- Which existing requirements already define each interaction
- Whether any uncovered interaction requires a new normative requirement

Until this clarification occurs, the child requirements derived from OR-001 assume that **Crucible operations** include Build, Capture, and Deploy as described by the Concept of Operations.

This assumption:

- Does not establish that every operation applies to every listed user category
- Does not establish how a listed user category participates in an operation
- Does not allocate permissions or authorizations
- Does not replace the need to identify the applicable existing requirements
- Remains subject to confirmation by the requirement owner

The Concept of Operations contextualizes the System Requirements Specification but does not override an approved requirement. Where the Concept of Operations and the System Requirements Specification conflict, the System Requirements Specification controls.

Proposed Statements

The Original Requirement requires separate treatment of the following independently testable user categories:

- OR-001a — Developer Operations
- OR-001b — DevSecOps Engineer Operations
- OR-001c — Platform Engineer Operations
- OR-001d — System Administrator Operations
- OR-001e — Security Engineer Operations
- OR-001f — Compliance Officer Operations

Each child requirement:

- Assumes that Crucible operations include Build, Capture, and Deploy as described by [C2]
- Identifies that the assumption remains subject to confirmation
- Does not assume that every operation applies to the applicable user category
- Identifies candidate interactions requiring clarification
- Reuses existing requirements wherever those requirements already define the applicable operation or interaction
- Creates additional atomic requirements only when the reuse assessment identifies a coverage gap

- OR-001a — Developer Operations
- OR-001b — DevSecOps Engineer Operations
- OR-001c — Platform Engineer Operations
- OR-001d — System Administrator Operations
- OR-001e — Security Engineer Operations
- OR-001f — Compliance Officer Operations