

The CONOPS CONOPS: Using Concept of Operations as a Design Tool

Morgan LaFavers | Monterey Technologies, Inc. | mlafavers@mti-inc.com



System development is most successful when all stakeholders are aligned to the vision of the system. The human systems integrator takes a leading role: decomposing, building upon, and tracing to the CONOPS, guiding discussions about functionality to ensure the system is mission-effective in its intended context.

Definitions and Relationships

"A ConOps is a user-oriented document that describes system characteristics for a proposed system from the users' viewpoint" (IEEE 1362-1998); includes

- Mission context, phases, and modes
- High-level depiction of the goals / tasks to accomplish within means
- The user's role in accomplishing the goal (both on- and off-system)
- Environmental conditions and other context
- Concepts for deployment, maintenance, sustainment, and retirement

Other key terms:

- **OPSCON**: Operations Concept; document that describes how a system will be implemented from engineering and operations perspectives
- **USE CASE**: Interaction between an actor and a system to accomplish a goal
- **SCENARIO**: Scene-based, in-context example of a specific potential actor interaction with the system

Application and Use

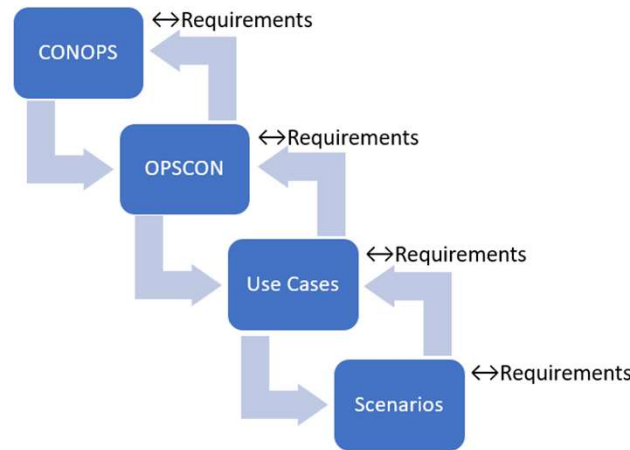
SYSTEM ENGINEERS:

- CONOPS is a holistic, unified framework for decomposing stakeholder needs to requirements, with traceability across the effort
- When updating legacy systems, is also a source of truth and rationale
- Informs plans for ensuring requirements and standards compliance
- Drives verification and validation plans ("test like you fly")
- Ensures functionality decomposition is complete / minimizes late discovery and associated development risk
- Drives traceability to user needs in MBSE environments

HUMAN FACTORS / USER EXPERIENCE:

- Provides a foundation for task decomposition and analyses
- Supplies context for user research and engagement
- Supports a comprehensive HSI Program Plan driven by mission value
- Provides vision for how individual features / components contribute to the success of the whole system

Fig 1. Iterative Development Process



Starting with the CONOPS is essential, as it provides the needed context and source of truth for requirements mapping, developing the operational concept, delineating use cases, and how those use cases will be satisfied by the system in common (and critical) scenarios.

Fig 2. Rabbit or Duck?



Failure to understand the CONOPS prior to system development can lead to building a system for the wrong context. System developers can think the system features, when connected, build a "rabbit" but instead could be a "duck" (or vice versa).

Resources

- NASA SE and HSI Handbooks
- INCOSE SE Handbook
- IEEE 29148:2011 Requirements Engineering
- DAU Acquippedia and CONOPS Template

Approach to Development

Usually get mission needs from the user

- May not address or only imply aspects like context, maintainability, supportability, cybersecurity, etc.

Address how the system as whole (including users, support personnel, and other stakeholders) supports accomplishing the mission

- Ensures complete coverage of explicit and implied needs
- Facilitates mapping requirements to delivered functions
- AND identifying weaknesses / gaps in workflows for improvement

High-level CONOPS may be written from a system or mission perspective, it is essential to decompose to the user perspective

- Also include information applicable to other stakeholders and interface (e.g., upstream and downstream stakeholders, commanders, etc.)

As a Design Tool

- How new features fit into the whole system to accomplish the mission
- Adding features without context can add unnecessary complexity to the system and create illogical / overcomplicated workflows
- Source of truth for system user goals and mission
- Facilitates finding opportunities for improvement, whether new features, improved workflows, or updates to UI design to fit common system user behavior / patterns
- Ask me about examples from the MTI portfolio!

Common Mistakes

PROBLEM: Lack of common understanding / terminology
IMPACT: Delays development, risks re-work to correct missteps
SOLUTION: Define project scope early and use / maintain it

PROBLEM: CONOPS development late in the product life cycle
IMPACT: Delivery of out-of-context features, overcomplicated workflows
SOLUTION: Start early and maintain first principles

PROBLEM: No plan for system deployment / maintenance / support / retirement
IMPACT: System down-time, costly repairs, poor mission efficiency
SOLUTION: Incorporate in CONOPS development, identify plan and potential issues early

And others...