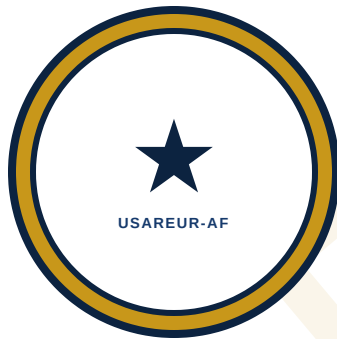


DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

CONCEPTS GUIDE

SL 4N



CONCEPTS GUIDE — SL 4N COMPANION — UI/UX DESIGNER · MAVEN SMART SYSTEM (MSS)

Specialist Course Manual

HEADQUARTERS
UNITED STATES ARMY EUROPE AND AFRICA
(USAREUR-AF)
Wiesbaden, Germany

DRAFT — NOT FOR OFFICIAL USE. FOR TRAINING PLANNING PURPOSES ONLY.

26 MARCH 2026

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

CONCEPTS GUIDE — SL 4N COMPANION — UI/UX DESIGNER · MAVEN SMART SYSTEM (MSS)

Forward: The Designer is the voice of the user on the product team. Every interface a Soldier touches was shaped — well or poorly — by design decisions. SL 4N ensures those decisions are informed, validated, and implementable. **Purpose:** Develops mental models required to practice Soldier Centered Design, build effective information architectures, and deliver accessible, operational application designs on MSS. Read before beginning SL 4N task instruction. **Prereqs:** SL 3 REQUIRED (and SL 1 + SL 2 implied). Design tool familiarity recommended. *HQ USAREUR-AF · v1.0 · 2026 · DISTRIB: USG only*

SECTION 1 — THE UI/UX DESIGNER'S ROLE ON MSS

BLUF: The Designer is the voice of the user on the product team. Every interface a Soldier touches was shaped — well or poorly — by design decisions.

MSS workforce tiers (design perspective):

| Tier | Designation | Relationship to Designer |
|-------|-----------------------|---|
| SL 1 | Maven User | Primary end user — designs must serve this population |
| SL 2 | Builder | Builds Workshop apps from Designer specifications |
| SL 3 | Advanced Builder | Builds complex Workshop/Pipeline apps from Designer specifications |
| SL 4J | Program Manager | Balanced team partner — owns backlog, validates priorities |
| SL 4L | Software Engineer | Balanced team partner — implements custom code from Designer specs |
| SL 4N | UI/UX Designer | Conducts research, defines design, validates usability |
| SL 4O | Platform Engineer | Infrastructure partner — defines deployment/performance constraints |

The Designer's distinguishing criterion: you are the team member whose primary job is understanding the user. PMs understand stakeholder priorities. SWEs understand technical feasibility. The Designer understands what the person actually sitting at the keyboard needs to see, do, and decide — and translates that understanding into implementable design artifacts.

Designer boundaries (owned by others): - Backlog prioritization decisions — SL 4J - Code implementation — SL 4L, SL 3 - Infrastructure and deployment — SL 4O - Data modeling and pipeline design — SL 3, SL 4H

The boundary is not about capability — it is about primary responsibility and design ownership.

SECTION 2 — SOLDIER CENTERED DESIGN (SCD) MENTAL MODEL

BLUF: SCD is not commercial UX with a military skin. The operational context fundamentally changes what "good design" means.

Commercial UX vs. SCD:

| Dimension | Commercial UX | Soldier Centered Design |
|----------------|---|---|
| User access | Unlimited; schedule around user availability | Limited windows; operational tempo dictates access |
| Environment | Controlled office/home | TOC, field, vehicle, SCIF — variable lighting, noise, stress |
| Classification | Public/private data; no handling restrictions | Data classified at multiple levels; UI must enforce marking |
| Stakes | Revenue, satisfaction | Operational decisions; potential life-safety impact |
| Feedback loop | Continuous analytics, A/B testing | Intermittent access; no commercial analytics in classified environments |
| User diversity | Segmented by persona | Cross-rank, cross-MOS, cross-nationality (coalition) |
| Network | Always-on, high bandwidth | DDIL — may be degraded, intermittent, or unavailable |

SCD implications for design practice: 1. Research must be planned around operational tempo — you cannot interrupt a battle drill for a usability test 2. Designs must function across classification levels — the same application may display different data at different markings 3. Error tolerance is lower — a commercial app crash costs revenue; an MSS app failure during operations costs decision advantage 4. Designs must degrade gracefully under DDIL — show what you have, indicate what is stale, do not show a blank screen

SECTION 3 — INFORMATION ARCHITECTURE PRINCIPLES

BLUF: Military users are information-dense decision-makers. Do not simplify — structure.

The Decision-First Hierarchy:

```

What decision does the user need to make?
|
What information supports that decision?
|
What is the priority order of that information?
|
How should it be spatially organized?
|
What interactions let the user drill deeper?

```

Every MSS application design should start with this chain, not with a widget palette.

Information density spectrum:

| Too Sparse | Right Density | Too Dense |
|--|--|--|
| One metric per screen; lots of whitespace; requires 5 clicks to see related data | All critical metrics visible simultaneously; detail available on demand; spatial grouping matches mental model | Wall of numbers with no hierarchy; everything competes for attention; no entry point for the eye |

Pattern: the "glance, scan, commit" test: - **Glance (2 sec):** User can identify the overall status (green/amber/red) without reading - **Scan (10 sec):** User can identify which specific areas need attention - **Commit (30 sec):** User can drill into a specific area and understand the details needed for a decision

If a design fails any of these three levels, it needs iteration.

SECTION 4 — VISUAL DESIGN MENTAL MODEL

BLUF: Color, type, and layout are not aesthetic choices on MSS — they are functional systems with operational consequences.

Color is a safety system: Classification banners, status indicators, and threat levels all use color as a primary encoding. The Designer's job is to ensure the overall color system is coherent — that status green does not clash with classification green, that the palette works for colorblind users, and that every color-encoded meaning has a redundant non-color indicator.

Typography is a readability system: Font choices, sizes, and spacing determine whether a Soldier can read the data they need, in the environment they are in, in the time they have. This is not a style preference — it is a functional requirement.

Layout is a priority system: What appears at the top-left gets looked at first (in LTR languages). What is large gets more attention than what is small. What is grouped together is perceived as related. Every layout decision is a priority decision.

SECTION 5 — ACCESSIBILITY AS OPERATIONAL REQUIREMENT

BLUF: Accessibility is not a compliance checkbox — it is an operational requirement. If 10% of your users cannot use the application, you have a 10% capability gap.

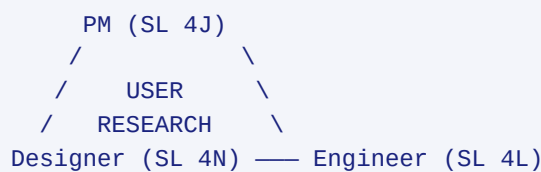
Why accessibility matters beyond compliance: - Colorblind users represent ~8% of male population — in a unit of 200, that is 16 soldiers who may misread color-only status indicators - Field conditions (glare, low light, dust on screens) create temporary visual impairments for everyone - Gloved operation creates temporary motor impairments - Noisy environments (generators, vehicles) create temporary hearing impairments for any audio-based alerts - Stressed, fatigued users under operational tempo have reduced cognitive capacity — clear, simple interfaces are accessibility features

The accessibility-operations alignment: Designing for accessibility and designing for operational conditions produce the same design decisions: high contrast, large touch targets, redundant encoding, clear hierarchy, keyboard navigability, graceful degradation. Accessibility is not extra work — it is the same work as designing for field use.

SECTION 6 — THE BALANCED TEAM AND DESIGN HANDOFF

BLUF: A design that cannot be implemented is a drawing, not a design. The handoff to engineering is where design becomes product.

The balanced team triad:



The three roles form a triangle around the user. The Designer conducts the research. The PM translates findings into priorities. The Engineer translates priorities into working software. The Designer validates the working software against the original user needs. The cycle repeats.

Handoff quality checklist: A design handoff is ready for engineering when a SWE (SL 4L) can: - Build the interface without asking "what should happen when...?" for any state - Identify every data source (Ontology property, computed value, static text) - Implement every interaction (click, filter, sort, submit, navigate) - Handle every edge case (0 items, max items, null values, long strings, slow load) - Verify accessibility without the Designer present

If any of these require a follow-up conversation, the handoff is incomplete.

SECTION 7 — WORKSHOP AND FOUNDRY DESIGN CONSTRAINTS

BLUF: Workshop is a constrained design environment. Know the constraints before you design, not after.

Workshop capabilities (what you can design): - Grid-based layouts with configurable widget placement - Pre-built widget types: tables, charts, maps, filters, action buttons, metric cards, text - Data binding to Ontology properties - Filter chaining across widgets - Action buttons that trigger Ontology Actions - Conditional formatting (color rules based on data values)

Workshop constraints (what you cannot do): - No custom CSS within Workshop (use Slate or custom widgets for pixel-perfect design) - Limited responsive behavior — test at all target screen sizes - Widget catalog is fixed — cannot create new widget types without custom code (SL 4L territory) - Animation and transitions are not supported in Workshop - Complex multi-step workflows require careful state management

Design implication: Design within Workshop's vocabulary first. If the requirement exceeds Workshop capabilities, escalate to SL 4L for custom implementation via Slate or OSDK application. Document which design elements require custom code vs. Workshop-native implementation.