BESS Software Development Lifecycle (SDLC) Standard

Document ID	Version	Status	Classification
BESS-STD-002	1.2	Active	Internal
Effective Date 2026-01-12			

Intent: Ship high-quality, secure software quickly using DeepAgent as the primary generation engine, with clear human review gates, evidence, and repeatable release practices.

1. Methodology

BESS uses a **Hybrid Kanban** approach: continuous flow, prioritized backlog, explicit WIP limits (as practical), and defined quality gates.

2. Phase 1 Intake, Requirements & Design

- Capture requirements (functional + non-functional) and acceptance criteria.
- Identify data sensitivity (PII/PHI/financial) and deployment model (SaaS, single-tenant, on-prem).
- Create Minimum Documentation Set (MDS) artifacts per BESS-TPL-006.
- Create at least one ADR for major decisions (architecture, auth, data storage, hosting).
- Perform lightweight threat modeling for systems handling PII/PHI/financial data.

3. Phase 2 Development & AI Integration

3.1 DeepAgent as Primary Implementation Engine

- DeepAgent may generate: architecture scaffolding, business logic, APIs, UI components, tests, migrations, IaC templates, and documentation.
- DeepAgent outputs are treated as **draft implementation** until reviewed and adopted.

3.2 Human Review Gate (HITL)

- No release to Production occurs without human review and approval by the Principal Engineer (or designated reviewer).
- Review focuses on: authZ/authN correctness, input validation, cryptography usage, dependency choices, error handling, logging, privacy, and performance risks.
- For security-critical components (auth, encryption, payment flows, PHI workflows): require an additional checklist-based review and targeted tests.

3.3 Coding & Repo Standards

- Repository must include: lint/format configuration, test runner config, and CI pipeline definitions.
- Use conventional commits (or equivalent) and meaningful PR descriptions (even if solo).
- Document AI-assisted generation notes in PR or change log when material to maintenance.

4. Phase 3 Verification (Quality Gate)

- Automated tests: unit + integration for critical paths.
- SAST and linting run on every CI build.
- Dependency/SCA scanning and license checks run on every CI build (see BESS-STD-003 and BESS-STD-007).
- Manual UAT for user-facing UI and high-risk workflows.

5. Phase 4 Release, Deployment & Obfuscation

5.1 Environment Separation (Single-Workstation Reality)

BESS can maintain strict separation without multiple physical machines by enforcing **logical and deployment separation**:

- **Development:** local workstation using containers and dev-only credentials/config.
- **Staging:** separate cloud (or client) environment, separate credentials, separate data, and production-like configuration.
- **Production:** separate cloud (or client) environment with locked-down access, monitoring, and change control.

Rule: Production workloads and production data must not run on the development machine except by explicit, documented exception.

5.2 Deployment Workflow

- Dev Staging deployment is required for client-facing systems prior to Production.
- Staging Production requires: passing CI gates, human approval, and release notes.

5.3 Obfuscation & IP Protection

- Apply platform-appropriate protections for distributed binaries.
- Prefer keeping sensitive logic server-side when feasible.

5.4 Symbol / Mapping File Management (Critical)

- When obfuscation/minification is used, BESS must securely store the corresponding mapping files/symbols for each release (e.g., ProGuard/R8 mapping, JS source maps policy, .NET obfuscation maps).
- Mapping files are treated as sensitive artifacts and stored with access controls.
- This enables debugging of production incidents without weakening client-delivered binaries.

6. Phase 5 Operations, Support & Maintenance

- On-call support per contract/SLA.
- Patch SLAs: High/Critical vulnerabilities remediated within defined timeframes (see BESS-STD-003).
- Incident response: document timeline, impact, root cause, remediation, and preventive actions.

7. Offboarding & Escrow-Ready Delivery (Enterprise Readiness)

- Projects must remain maintainable by a competent third-party engineer using the MDS artifacts and runbooks.
- Critical credentials and deployment instructions must be documented and transferable (never embedded in code).
- Where contractually required, BESS supports code escrow arrangements and provides an escrow-ready package (documentation, build instructions, dependency lockfiles, release tags).

Note: This standard describes internal controls; client-specific variations must be documented in the project s SOW and/or exception records.