U.S. GEOLOGICAL SURVEY

ShakeAlert® Earthquake Early Warning System

Production System Code Review – Statement of Work (9/1/2026)

Document	ShakeAlert Production System Code Review
Period of performance	12 months from contract award
Version / Date	Version 2.0 – September 2026
Prepared by	U.S. Geological Survey, Earthquake Science Center

1. Introduction

The U.S. Geological Survey (USGS) seeks a qualified firm to conduct a focused, independent code review of the ShakeAlert® Earthquake Early Warning (EEW) Production System. ShakeAlert operates continuously to detect earthquakes and deliver life-safety warnings to more than 50 million people along the U.S. West Coast. Its software is a collection of real-time, safety-critical modules that have evolved over many years and are maintained by a small team under significant operational constraints.

The objective of this contract is to improve the reliability, maintainability, and long-term sustainability of the ShakeAlert production code base through two complementary activities:

- A rigorous, module-by-module review of source code quality, correctness, and maintainability.
- An assessment of current software development and deployment practices, with concrete recommendations for modernizing the CI/CD pipeline and GitLab workflow.

Work will be performed in phases over a period not to exceed 12 months. All work shall be conducted with regular coordination with USGS ShakeAlert staff. All work can be performed remotely.

2. Background

Earthquakes pose a significant national risk. FEMA estimates annualized losses to the U.S. building stock at $14.7 billion per year, with California alone accounting for 65% ($9.6 billion). The West Coast states together account for 78% of total national average annual losses.

To reduce these losses, USGS operates ShakeAlert, which became publicly available in California in October 2019, Oregon in March 2021, and Washington in May 2021. Expansion to Nevada and Alaska is under active planning.

The ShakeAlert Production System ingests real-time waveform data from approximately 1,700 seismic sensor locations across four regional seismic networks, as well as GNSS data from approximately 1,100 sites. Data flows through eight parallel processing pipelines distributed across four geographically separated data centers. The software stack includes modules for waveform processing, multi-algorithm earthquake source characterization (EPIC, FinDer, GFAST), ground-motion estimation (eqinfo2GM), solution aggregation and decision making (SA/DM), alert publishing, and system health monitoring.

Because this is a real-time, safety-critical system with very low latency tolerance, code quality directly affects alert accuracy, reliability, and speed. The codebase has grown organically over many years, and the ShakeAlert team lacks the bandwidth to conduct systematic internal code reviews. An independent, expert review is needed to identify latent defects, reduce unnecessary complexity, and establish sustainable practices for ongoing code quality assurance.

2.1 Context for This Work

Future Systems Engineering Analysis contracts may assess the broader ShakeAlert architecture, end-to-end performance, and alerting decision logic. This Code Review contract is complementary and narrower in scope: it focuses specifically on source code quality, software engineering practices, the development pipeline, and the software deployment pipeline. Findings from this contract will inform future systems engineering work.

The contractor is not expected to measure or evaluate the correctness or appropriateness of scientific methods that are embodied in the code.

Portions of this Statement of Work were drafted with assistance from an artificial intelligence (AI) tool and were fully reviewed, edited, and approved by USGS/DOI personnel to ensure accuracy, integrity, and compliance with applicable policies.

3. Scope of Work

The contractor shall conduct an independent review of the ShakeAlert production code base and software development practices. The work is organized into two major tasks.

3.1 Task 1 – Production Code Review

The contractor shall review the source code of all production modules listed in Section 4, plus any modified third-party libraries that the ShakeAlert team maintains. The review shall at a minimum cover the areas described below.

3.1.1 Quantitative Code Metrics

For each module, the contractor shall compute and report at minimum:

- Total executable lines of code (LOC), distinguishing production code from test code and comments.
- Cyclomatic complexity: per-function and per-module distributions, with identification of functions exceeding recommended thresholds.
- Halstead metrics (volume, difficulty, effort) as a supplementary complexity indicator.
- Unit test code coverage (line, branch, and function coverage), measured using the project's existing test framework where available.
- Documentation coverage: proportion of public functions/APIs with adequate documentation (docstrings, header comments, or equivalent).
- Dependency graph: identification of circular dependencies, overly broad coupling, and modules with high fan-in or fan-out.

The contractor may propose other metrics to supplement or replace those listed that, in their professional opinion, would best characterize the system, uncover deficiencies, and lead to improvements. Metrics shall be compiled into a baseline report (Deliverable 2) and used to prioritize areas for deeper review.

3.1.2 Qualitative Code Review

The contractor shall perform a deep qualitative review of each module, examining:

- Correctness: logic errors, off-by-one errors, incorrect assumptions about data types or input ranges, race conditions, memory leaks, and any code paths that could produce silent errors or incorrect alert parameters.
- Reliability and fault handling: adequacy of error detection and recovery; handling of unexpected inputs (missing data, out-of-order messages, sensor dropouts, partial failures); behavior under degraded conditions.
- Multithreading issues: thread contention, circular dependencies, deadlocks, concurrency bugs, cache thrashing, and thread leaks.
- Real-time safety: potential blocking calls, unbounded loops, unsafe memory operations, or operations inconsistent with real-time processing requirements.
- Inter-process interactions: correctness of message-passing interfaces (currently ActiveMQ, transitioning to NATS); adherence to defined message schemas; handling of message ordering and timing dependencies.
- Tunable parameters (“free variables”): identification of all runtime-configurable parameters in each module; documentation of their expected ranges, sensitivity, and default values; assessment of whether parameter validation is adequate.
- Uncertainty propagation: whether each module correctly propagates, quantifies, or flags uncertainty in its outputs.
- Known issues: review of specific issues identified in prior work (e.g., differences in solutions across server instances due to variations in data arrival time or order, network path latency effects, operating system resource contention, and differing computational loads), and assessment of whether any code-level changes could reduce them.
- Code clarity and maintainability: adherence to consistent naming conventions, absence of dead code and commented-out code blocks, clarity of algorithmic intent, adequacy of inline comments.
- Coding standard compliance: assessment against applicable coding standards (see Section 5.2).

3.1.3 Cross-Module Integration Analysis

In addition to module-level review, the contractor shall assess:

- Data flow and interface consistency: whether module inputs and outputs are consistent in format, units, coordinate systems, and time references.
- Failure propagation: how exceptions, errors or anomalies in one module affect downstream modules; whether error states are adequately isolated.
- Logging: that exceptions, errors, and intermediate results are properly logged.
- Distributed network: timeouts, retry limits, backoff strategies
- Solution aggregation logic: specific attention to the Solution Aggregator (SA) logic that merges EPIC, FinDer, and GFAST solutions, including edge cases (e.g., split events, foreshock-mainshock sequences, magnitude saturation).
- Alert decision logic: review of the Decision Module (DM) logic used to determine if ShakeAlert Message should be published to public Alert Servers.
- Alert pause and update logic: review of the alert-pause feature and alert-update thresholds for correctness, with attention to the 2022 Ferndale earthquake bug case (logic not triggering updates after pause expiry).

3.2 Task 2 – Software Development Practices and CI/CD Pipeline

3.2.1 GitLab Code Management and Build Workflow

The contractor shall review the current GitLab repository structure, branching strategy, merge request process, and tagging/versioning practices, and shall recommend improvements that are achievable within the program's resources. Areas to examine include:

- Branch protection rules and merge request approval requirements.
- Use of GitLab CI/CD pipeline for automated build and test on every commit.
- Consistency and completeness of commit messages and change documentation.
- Release tagging and version management.

3.2.2 Automated Static Analysis in the CI/CD Pipeline

The contractor shall recommend specific static open-source analysis tools compatible with the ShakeAlert code base (language, build system, licensing constraints) and demonstrate their integration into the GitLab CI/CD pipeline. Recommendations shall address:

- Linting and style enforcement (e.g., clang-tidy, cppcheck, or equivalent depending on language).
- Automated complexity and metrics reporting on every merge request.
- Security-relevant checks (e.g., detection of unsafe string operations, integer overflow risks).
- Test coverage reporting integrated with merge requests, with configurable coverage-decline alerts.

The contractor shall implement at least a baseline CI/CD pipeline configuration in the project's GitLab repository as part of Deliverable 6, not merely recommend it be done.

3.2.3 Coding Standards

The contractor shall review the project's existing coding standards documentation (if any), assess compliance across the code base, and recommend a concise, enforceable coding standard appropriate for a small, resource-constrained team. The standard shall address naming conventions, file structure, error handling patterns, and documentation requirements.

3.2.4 Code Deployment Practices

The contractor shall review the current deployment process (Ansible and related tooling) and assess:

- Reproducibility: whether deployments are deterministic and traceable to a specific code version.
- Rollback capability: whether a failed deployment can be safely and quickly reverted.
- Configuration management: whether runtime configuration files are version-controlled and validated before deployment.
- Deployment testing: whether there is an automated smoke test or health check following each deployment.

4. Modules in Scope

The contractor shall review all of the following production modules, plus any Earthworm modules and third-party libraries that the ShakeAlert team has modified and maintains in the project repository.

Module	Role in production system
EWP2	EPIC waveform processor
EPIC (E2)	Point-source earthquake detection and magnitude estimation
ffd2	FinDer waveform processor

FinDer	Finite-fault rupture detector; handles M≥6 characterization
GFAST	Geodetic (GNSS-based) magnitude estimation
eqinfo2GM	Converts source parameters to ground-motion estimates and produces Contour and Grid Products
HeartbeatAggregator	System health monitoring and heartbeat aggregation
Solution Aggregator / Decision Module (SA/DM)	Combines algorithm solutions; applies alerting logic and thresholds
SA2IPAWS	Creates and transmits CAP format alerts to FEMA IPAWS / WEA

USGS will provide the contractor with repository access, build instructions, and relevant documentation for each module at contract start. The contractor shall notify USGS of any module for which the provided documentation is insufficient to perform the review, and USGS will provide supplemental technical briefings as needed.

5. Technical Requirements

5.1 General

- All review work shall be based on the production branch of the code at the time of repository access. USGS will notify the contractor of any significant code changes deployed during the contract period.
- The contractor shall request and document, for each module, the design intent, known limitations, and any open issues tracked in the project's issue tracker. This contextual information shall inform the review and shall not substitute for independent code-level analysis.
- The contractor shall communicate findings to USGS ShakeAlert staff progressively (not only in final reports), to allow USGS to prioritize fixes and to avoid misunderstanding and duplication of effort.
- Primary recommendations shall be technically sound, clearly linked to identified deficiencies, and actionable within the constraints of the ShakeAlert program (limited staff, operational system, a federally regulated IT environment). Additional recommendations that require more extensive effort, specialized expertise, or expanded staffing may also be included. These may represent longer-term opportunities, options for future phases, or work that could be outsourced to external partners when appropriate. Recommendations that require a complete rewrite of major modules are unlikely to be actionable and should be avoided unless there is compelling justification.
- The contractor shall acknowledge, where relevant, that the message-passing infrastructure is transitioning from ActiveMQ to NATS. Recommendations related to inter-process communication shall be written to remain valid under the anticipated NATS architecture or shall explicitly address both current and future states.

5.2 Applicable Standards

The contractor shall apply, at minimum, the following standards and best practices as reference frameworks for the code review:

- MISRA C/C++ guidelines (or equivalent; to be agreed at kickoff based on language profile of code base).
- CERT C/C++ Secure Coding Standard for security-relevant findings.

- SEI CERT guidelines for concurrency and real-time considerations.
- Federal government IT security and software assurance guidance relevant to USGS systems.

At kickoff, the contractor and USGS shall agree on the specific standards and tools to be applied, taking into account the languages and build environment of the ShakeAlert code base.

5.3 Access and Coordination

- USGS will provide the contractor read-only access to the ShakeAlert GitLab repository (code.usgs.gov/EEW/ShakeAlert) and relevant wikis, issue trackers, log files, and CI/CD configuration.
- Write access to the GitLab repository CI/CD configuration will be provided for Deliverable 6 (pipeline implementation), under USGS review and approval of all changes before merging.
- The contractor shall acknowledge they understand the Intellectual Property rules which govern the ShakeAlert production code and protect the code accordingly.
- The contractor shall participate in a kickoff meeting (in person or virtual) within 30 days of contract award to review scope, clarify access, agree on standards and tools, and establish a communication cadence.
- The contractor shall hold at least monthly progress meetings with USGS ShakeAlert technical staff throughout the contract period.
- USGS ShakeAlert staff will be available for technical questions, clarification of design intent, and architectural briefings. Contractor questions shall be submitted in writing (e.g. email or issue tracker) or captured in meeting minutes to create a record.

6. Deliverables

The following deliverables are required. All written reports shall be submitted in both PDF and editable Word (.docx) format. All deliverables shall be reviewed by USGS within 30 days of receipt; the contractor shall address USGS comments within 30 days thereafter.

#	Deliverable	Due (months after award)	Format
1	Kickoff meeting and work plan	Month 1	Meeting + written plan
2	Code metrics baseline report (all modules)	Month 3	Written report + data files
3	Detailed code review findings (all modules)	Month 6	Written report with prioritized findings
4	CI/CD and development practices assessment	Month 8	Written report with implementation guide
5	Final report and recommendations	Month 11	Comprehensive report; briefing to ShakeAlert staff
6	Demonstration of CI/CD improvements in repository	Month 12	Working pipeline changes in project GitLab

6.1 Deliverable Descriptions

Deliverable 1 – Kickoff Meeting and Work Plan

Within 30 days of contract award, the contractor shall conduct a kickoff meeting with USGS ShakeAlert technical staff and deliver a written work plan that specifies the review methodology, tools and standards to be applied, task schedule with milestones, points of contact, and communication plan.

Deliverable 2 – Code Metrics Baseline Report

A quantitative characterization of the code base covering all modules in Section 4. The report shall include per-module and aggregate metrics (LOC, cyclomatic complexity, Halstead metrics, test coverage, documentation coverage), a dependency graph, and a prioritized list of modules and subsystems flagged for deeper review based on the metrics. Raw metric data shall be provided as machine-readable files (e.g., CSV or JSON) in addition to the narrative report.

Deliverable 3 – Detailed Code Review Findings

A module-by-module code review report organized by the review dimensions in Section 3.1.2. For each finding, the report shall identify the affected module and file, describe the issue and its potential impact on alert reliability or correctness, assign a severity rating (Critical / High / Medium / Low), and provide a specific, actionable recommendation. The report shall include a consolidated findings table for easy prioritization. Critical and High severity findings shall include proposed code changes (pseudocode or actual code) where practical.

Deliverable 4 – CI/CD and Development Practices Assessment

A report covering the assessment of GitLab workflow, automated static analysis tooling, coding standards, and deployment practices described in Section 3.2. For each area, the report shall describe current practice, identify gaps relative to applicable standards and industry practice, and provide prioritized, actionable recommendations with estimated implementation effort. The report shall include a step-by-step implementation guide for the recommended CI/CD pipeline improvements.

Deliverable 5 – Final Report and Briefing

A comprehensive final report that synthesizes findings across all tasks, highlights the most impactful improvements, and provides a recommended prioritized action plan for USGS. The report shall include an executive summary suitable for non-technical stakeholders. The contractor shall present the findings to ShakeAlert technical staff and program management in a virtual briefing of approximately two hours.

Deliverable 6 – CI/CD Pipeline Implementation

Working CI/CD configuration changes implemented in the ShakeAlert GitLab repository, reviewed and approved by USGS, that automate at a minimum: build verification, static analysis (linting and complexity checks), unit test execution, and coverage reporting on every merge request. The contractor shall document the pipeline configuration and provide a brief user guide for ShakeAlert staff.

7. Applicable Documents

- ShakeAlert public website: https://earthquake.usgs.gov/data/shakealert/
- ShakeAlert internal code repository: code.usgs.gov/EEW/ShakeAlert
- ShakeAlert GitLab wiki: https://code.usgs.gov/EEW/ShakeAlert/-/wikis/home

- ShakeAlert Alerting Priorities Guidance, May 2023 (internal USGS document; provided to contractor at award).

- Given et al. (2018). Revised Technical Implementation Plan for the ShakeAlert Production System. USGS Open-File Report 2018-1155. https://pubs.er.usgs.gov/publication/ofr20181155
- Lux et al. (2024). Status and performance of the ShakeAlert earthquake early warning system: 2019–2023. Bulletin of the Seismological Society of America, 114(6), 3041–3062. https://doi.org/10.1785/0120230259
- McGuire et al. (2025). ShakeAlert® version 3: Expected performance in large earthquakes. Bulletin of the Seismological Society of America, 115(2), 533–561. https://doi.org/10.1785/0120240189

8. Contract Constraints and Expectations

- Period of performance: 12 months from date of contract award.
- All work may be performed remotely. Travel to USGS facilities is not required but may be proposed by the contractor for the kickoff meeting at contractor expense or by mutual agreement.
- The contractor shall treat all non-public ShakeAlert code, configuration, and documentation as sensitive and shall not disclose it to third parties. A non-disclosure agreement (NDA) or comparable instrument will be executed at contract award.
- All recommendations shall account for the operational nature of the system: ShakeAlert runs continuously and changes to production code must be carefully managed. The contractor is not expected to implement code fixes; that responsibility remains with USGS staff. The exception is Deliverable 6, where the contractor will implement CI/CD pipeline changes in the repository under USGS review.
- All recommendations shall consider the strong preference of USGS to use open-source code, libraries, and tools.
- **Contractor personnel performing the code review shall have demonstrated experience in real-time or safety-critical software systems. Relevant domain experience in seismology or geophysical data processing is desirable but not required.**

9. Glossary

ActiveMQ - Current message-passing middleware used in ShakeAlert.

CAP - Common Alerting Protocol, used for ShakeAlert alerts sent via IPAWS/WEA.

CI/CD - Continuous Integration / Continuous Deployment pipeline.

Cyclomatic Complexity - A measure of branching and logic complexity in code.

Decision Module (DM) - Module that applies alert logic and determines when to publish ShakeAlert Messages. Uses same code as SA with different configuration parameters.

Earthworm - a free open-source earthquake processing software package originally developed by the USGS.

EEW - Earthquake Early Warning.

EPIC - Point-source earthquake detection and magnitude estimation module.

eqinfo2GM - Module that converts earthquake parameters into ground-motion estimates.

FEMA - Federal Emergency Management Agency

FinDer - Finite-fault rupture detection module.

GFAST - Geodetic magnitude estimation module using GNSS data.

GitLab – Source code management and CI/CD platform used for ShakeAlert.

GNSS - Global Navigation Satellite System.

Halstead Metrics - Software complexity measures based on operators and operands.

IPAWS - Integrated Public Alert and Warning System, FEMA's system to disseminate public alerts.

LOC (Lines of Code) - Metric representing the number of executable source code lines.

NATS - Next-generation open-source messaging system replacing ActiveMQ.

SA2IPAWS - Creates and transmits CAP format alerts to IPAWS / WEA for public alerting.

ShakeAlert - U.S. Geological Survey's Earthquake Early Warning system.

Solution Aggregator (SA) - Combines earthquake location and magnitude solutions from EPIC, FinDer and GFAST. Uses same code as DM with different configuration parameters

USGS - U.S. Geological Survey

WEA - Wireless Emergency Alerts delivered to the public through IPAWS.