

D4: Client Handoff Plan

This plan describes how we will deliver Media Markup to the client in a form that is immediately usable and also maintainable after the semester ends. The goal of the handoff is to leave the client with a working macOS application, the source code and supporting materials needed for future changes, and enough setup and release guidance to continue using and distributing the product independently.

Media Markup is a macOS desktop application for reviewing media files, placing and refining segment markers, and exporting CSV output compatible with the client's existing workflow. The system was designed around a Tauri desktop shell with a Svelte/TypeScript frontend and a Rust backend.

What the Client Will Receive

At final handoff, the client will receive a packaged macOS build of the application for normal day-to-day use, along with the project source repository for future maintenance and enhancement. The client will also receive concise in-repository quick-reference documentation, a longer developer handoff guide for maintenance and release workflows, a user manual for operating the application, and short video walkthroughs showing setup, access, and use.

This delivery model matches the intended use of the project: the packaged application is the primary way the client will run and distribute the tool, while the repository and developer documentation support future development and maintenance. The repository already contains documentation sections for both user and developer materials, as well as automated CI/CD workflows to support testing and release management.

Product Delivery Method

The primary delivery artifact will be a notarized macOS application bundle distributed as a .dmg or equivalent packaged release. This gives the client a ready-to-install version of the tool without requiring development setup. A secondary delivery artifact will be the GitHub repository, which will remain the source of truth for code, quick-reference technical documentation, and future changes.

Our release pipeline is already structured around GitHub Actions. The repository includes continuous integration for testing and validation and a release workflow that builds platform bundles from version tags, with support for macOS signing and notarization when the required credentials are configured.

Client Environment and Ownership

The client will control the environment in which the final product is used. For normal operation, this means installing and running the packaged macOS application on a client-controlled Mac. For future

development or modification, this means the client will also have access to the repository, build instructions, testing instructions, and release guidance needed to rebuild the application from source.

Because the project targets macOS and has already been validated on Apple Silicon Macs, the expected deployment and maintenance environment is a macOS system using Apple Silicon hardware. The functional scope of the delivered system includes opening local media files, waveform-based review, playback and marker refinement, validation rules, and CSV export.

Planned Handoff Steps

Before final delivery, we will prepare a release candidate build and verify that the final packaged application works correctly on a target Apple Silicon Mac. This verification will include basic launch/install confirmation and functional checks on the implemented workflow, including media loading, playback, marker placement and refinement, and CSV export.

We will then finalize the documentation package. This will include the user manual, the developer handoff documentation, and the shorter quick-reference documents located directly in the repository. Video walkthroughs will also be recorded and linked on the project website and shared directly with the client.

At final handoff, we plan to provide the packaged application, repository access, documentation links, and video links together. We also plan to schedule a live handoff session with the client to answer any questions the client has about the codebase or product itself, and potentially help troubleshoot any setup issues.

Maintenance and Future Development

For ordinary use, the client should rely on the packaged application. For future changes, the client will be able to build from source using the repository and the accompanying developer documentation. That documentation will cover environment setup, how to run the application locally, how to run tests, how to create build artifacts, and how to produce future distributable releases.

The repository already supports this maintenance path through existing documentation structure and automation. CI covers testing and validation, while CD supports tagged release generation for packaged builds.

Notarization and Release Signing

A short release-maintenance section will be included in the final developer-facing documentation covering signing and notarization for future macOS releases. This section will explain when notarization is required, what credentials or secrets are needed, and how the client can repeat the process when distributing an updated build.

Risks and Mitigation

The primary handoff risk is not application functionality, but release continuity. In particular, future distributed macOS builds must continue to follow the documented signing and notarization process to avoid installation friction for end users. This will be mitigated by including release guidance in the final developer documentation and preserving the existing automated release workflow.

A secondary risk is documentation drift after future changes. This will be mitigated by keeping both the quick-reference repository docs and the longer maintenance guide with the project deliverables so the client has a clear base for future updates.

Outcome

By following this plan, we will leave the client with a working packaged product for immediate use, the codebase and instructions needed for future maintenance, and clear supporting documentation and videos to reduce friction during adoption and later development.