

OHIF Project Governance Framework

Version 1.2

Effective Date: December 12, 2025

Authors:

Gordon Harris, Ph.D.

Rob Lewis

Will Anderson

Alireza Sedghi, Ph.D.

James Hanks

Glossary	4
1. Introduction	4
1.1 Purpose of Governance	4
1.2 Mission and Vision	5
1.3 Core Principles	5
1.4 Governance Goals	5
2. Roles and Responsibilities	6
2.1 Acknowledgment of Founders	6
2.2 Advisory Board	6
2.3 Steering Committee	7
2.4 Roles	7
2.4.1 Project Maintainers	7
2.4.2 System Architect	8
2.4.3 Project Manager	8
2.4.4 Community Manager	8
2.4.5 Founding Members	8
2.4.6 Contributors	8
2.4.7 UI/UX Designer	9
2.5 Visualizing the Progression	9
3. Roadmap and Project Steering	10
3.1 Roadmap Prioritization	10
3.2 Project Steering and Conflict Resolution	10
4. Technical Decision Process	10
4.1 Informed Decision Making	10
4.2 Feature Requests	11
4.3 Documentation	11
5. Contribution Workflow	11
5.1 Code Contributions	11
5.2 Documentation Contributions	12
5.3 Community Contributions	12
5.4 Review Processes and Quality Control	12

5.5 Best Practices for Managing the Code	12
6. Funding and Sustainability	13
6.1 Grants and Sponsorships	13
6.2 Budget Oversight	13
6.3 Transparency in Funding	13
6.4 Attracting Funding for the OHIF Project	13
7.1 Governance Meetings	14
7.2 Community Meetings	14
7.3 Contributor Meetings	14
8. Release Management Strategy	15
8.1 LTS Support	15
8.2 Regular Releases	15
8.3 Branching Workflow for LTS and Regular Releases	15
8.4 Release Lifecycle and Workflow	16
8.5 Backporting Strategy	16
8.6 End-of-Life (EOL) Policy	16
Addendum: Implementing the Release Management Strategy	17
1. Define the Release Cadence:	17
2. Set Up Branching Strategy:	17
3. Automate Testing and CI/CD:	17
4. Establish Feature Freeze and Testing Phases:	17
5. Backporting Strategy:	17
6. Documentation and Communication:	17
7. Monitor and Iterate:	17
Version History	17

Glossary

- **Advisory Board** - A group of experts providing strategic advice and guidance on the direction of the project.
- **Backporting** - The process of applying a fix or patch from a newer software version (e.g., `master` branch) to an older version (e.g., LTS branch) to address security or critical issues without adding new features.
- **Branching Strategy** - A system used in version control to manage different versions of the code. Separate branches are created for new features, bug fixes, and releases to ensure stability and organized development.
- **Consensus-Based Decision Making** - A decision-making process where the community and stakeholders reach an agreement on a particular issue or change, rather than having a decision imposed by a single individual or group.
- **Contributor** - Anyone who contributes to the project by submitting code, improving documentation, testing, or participating in community discussions.
- **End-of-Life (EOL)** - The point at which a version of the software will no longer receive updates, patches, or support, prompting users to upgrade to a newer version.
- **Feature Freeze** - A phase in the release process where no new features are added to the software. The focus shifts to testing and bug fixes to ensure a stable release.
- **Hotfix** - A quick fix or patch that addresses critical issues such as security vulnerabilities. Hotfixes are typically applied to both `master` and LTS branches.
- **Long-Term Support (LTS)** - A release that is supported for an extended period (e.g., 1 year), receiving only critical bug fixes and security updates. LTS releases prioritize stability and reliability over new features.
- **Maintainer** - A core contributor responsible for managing and reviewing code submissions, ensuring quality standards, and overseeing the project's overall health.
- **Main Branch** - The primary development branch in a version control system where ongoing development occurs, including new features, improvements, and bug fixes.
- **Pull Request (PR)** - A request made by a Contributor to merge their changes into the project's codebase. Pull requests are reviewed by Maintainers to ensure the quality and alignment of the changes with project goals.
- **Steering Committee** - A group responsible for overseeing the strategic direction of the project, resolving conflicts, and making high-level decisions regarding partnerships, funding, and project growth.

1. Introduction

1.1 Purpose of Governance

The purpose of the Open Health Imaging Foundation (OHIF) Governance Framework is to ensure that the project maintains its core principles while fostering an inclusive, collaborative, and sustainable community. The governance framework establishes clear guidelines for

decision-making, contribution, and growth to support the long-term success of OHIF and its Contributors.

1.2 Mission and Vision

OHIF's mission is to be a comprehensive, flexible, developer- and user-friendly open-source framework, free from copy-left restriction, for web-based medical imaging software applications. OHIF aims to simplify and accelerate the process of innovating and developing ubiquitously available tools and technology by encapsulating the complexity of DICOM, providing the most commonly needed components for building a wide range of interactive imaging tools, and the necessary flexibility to easily add novel tools and workflows.

Our vision for OHIF is to democratize medical imaging technology, significantly lowering the barrier to entry into what is otherwise a highly complex, often proprietary domain. For users, developers, academic institutions and companies alike, OHIF strives to enable and encourage collaboration, elevating the quality of, and increasing access to, cutting-edge medical imaging technology, advancing patient care and medical research everywhere.

1.3 Core Principles

The OHIF project is built on the following core principles:

- **Transparency:** Open discussions, decision-making, and processes to ensure community visibility into all aspects of the project.
- **Inclusivity:** Encourage participation from diverse stakeholders, including developers, healthcare professionals, researchers, and institutions.
- **Meritocracy:** Decisions are made based on the value and quality of contributions.
- **Sustainability:** Focus on long-term project sustainability by maintaining funding, partnerships, and community support.

1.4 Governance Goals

The goals of the governance framework are to:

1. Ensure smooth collaboration between developers, users, and stakeholders.
2. Maintain project stability by defining processes for decision-making and conflict resolution.
3. Foster innovation through continuous contributions from the community while ensuring that changes align with long-term project goals.

2. Roles and Responsibilities

2.1 Acknowledgment of Founders

The OHIF project would not be where it is today without the vision, dedication, and expertise of its founding members. We extend our heartfelt gratitude to:

- Gordon J. Harris, Ph.D. (Massachusetts General Hospital)
- Rob Lewis (Radical Imaging)
- Chris Hafey (Creator of Cornerstone.js)
- Erik Ziegler, Ph.D. (Technical Advisor, Formerly Radical Imaging)
- Trinity Urban (UX Advisor, Formerly Massachusetts General Hospital)

Their collective contributions have laid the foundation for a thriving, open-source platform that continues to empower the medical imaging community worldwide.

2.2 Advisory Board

The Advisory Board consists of a mix of industry and academic experts who provide high-level strategic advice, guidance, support, or other areas of expertise that further the aims of OHIF. The Advisory Board members meet with the Steering Committee at least quarterly in transparent sessions and provide non-binding guidance related to industry trends, user needs, emerging technologies, and other project concerns. Advisory Board members are appointed by the Steering Committee in a manner transparent to the community and serve a fixed term of two (2) years, optionally renewable or terminable by the Steering Committee. It is the responsibility of the Steering Committee to ensure that the Advisory Board is comprised of a broad range of project stakeholders with representation throughout the global OHIF community, including:

- Industry adopters
 - Perspective on industry trends
 - Feedback on technical implementation
 - Feature suggestions
- Academic and research adopters
 - Perspective on research trends
 - Feature suggestions
- Community representatives and end users
 - Contribute user feedback and experience
- OHIF Maintainers and technical Contributors
 - Validating technical concepts
- Other experts
 - Provide broader open-source sustainment experience

Advisory Board members must disclose any potential conflicts of interest to the OHIF Steering Committee and community. Advisory Board positions are voluntary, unpaid and receive no other forms of compensation from OHIF and its supporting entities.

2.3 Steering Committee

The Steering Committee actively manages the strategic direction of the OHIF project. It is responsible for approving significant architectural changes, securing funding, and managing high-level partnerships. The Steering Committee makes key decisions regarding project direction, resolves conflicts, and ensures core objectives are met.

The Steering Committee is composed of:

- **Project Maintainers:** Responsible for contributing to technical development in the form of codebase management.
- **System Architect:** Provides oversight to the technical design and development of project systems.
- **Project Manager:** Ensures roadmap alignment, delivery, and stakeholder communication.
- **Community Manager:** Maintains community support and reports community needs to Steering Committee

In addition, the following groups are invited to the Steering Committee as needed:

- **Founding Members:** To the extent available and interested, Founding Members noted above are welcome to participate, bringing historical context and strategic leadership to the project.
- **Contributors:** OHIF values the input of key community members, as such the Steering Committee invites these Contributors to relevant discussions as deemed necessary by the committee.
- **UI/UX Designer:** Provides community and market insights related to product design and user experience
- **Advisory Board:** Provides guidance for Steering Committee decisions regarding roadmap and sustainment

2.4 Roles

2.4.1 Project Maintainers

Project Maintainers manage day-to-day development activities. They review and merge pull requests, maintain code quality, and oversee release cycles. They work closely with Contributors to ensure that features align with the project roadmap and quality standards. Maintainers work directly with the Project Manager, System Architect, UI/UX

Designer, and other key project personnel as necessary to ensure code delivery expectations are met.

2.4.2 System Architect

The System Architect is responsible for technical oversight and design of the project. Working in collaboration with the Steering Committee, the System Architect devises technical solutions based on the stated goals of the project and ensures that Project Maintainers are able to execute the plans. The System Architect participates in reviewing and approving pull requests for changes, ensuring that new features and changes are compatible with the existing system and support long-term maintainability. The role actively collaborates with Maintainers to oversee the implementation of branching strategies and release workflows, keeping a pulse on community needs and technical trends related to the software ecosystem and informing the Steering Committee of related updates.

2.4.3 Project Manager

The Project Manager is responsible for overseeing the overall progress of the OHIF project, managing the roadmap, and ensuring timely delivery of releases through close collaboration with key Maintainers in areas such as UI/UX and requirements development, design documentation, and system validation. The Project Manager is also responsible for maintenance of funding commitments and project budgeting. They work closely with the Steering Committee, Project Maintainers, and the Community Manager to align project priorities and ensure that community and business needs are met.

2.4.4 Community Manager

The Community Manager is responsible for engaging with the community, managing outreach, and organizing events. They are also responsible for maintaining social media presence and fostering positive interactions within the OHIF community. The Community Manager ensures that community needs are communicated to the Project Maintainers for future releases, whether for long-term support (LTS) or regular versions.

2.4.5 Founding Members

Founding Members offer valuable insight into the long-term sustainment of OHIF. They are responsible for attending Steering Committee meetings as available to provide input on strategic decisions, maintain critical community partnerships, and manage overall project direction.

2.4.6 Contributors

Contributors are community members that participate in the development of the OHIF project by submitting code, documentation, or other contributions within the guidelines of the community contribution codes. They help identify bugs, propose new features, and engage in discussions on project direction. Contributors liaise with forward-facing members of the Steering Committee and the Community Manager through regular community feedback sessions and open collaboration on community development issues.

2.4.7 UI/UX Designer

The UI/UX Designer is responsible for development and management of front-end system components. Working closely with the Project Manager, Community Manager, System Architect, and Key Contributors, the UI/UX Designer ensures that front end changes to the system are driven by community needs, clearly and transparently documented to the community, and able to be executed within the technical design of the system. The UI/UX Designer is also responsible for maintaining a working knowledge of current design methodology and user experience of related products in the medical imaging landscape.

2.5 Visualizing Progression

Contributor → 2. **Regular Contributor** → 3. **Maintainer** → 4. **Steering Committee**

The following quantifies the progression from **Contributor** to **Steering Committee** membership:

1. **Contributor** (Entry Level):
 - Contributions: quality PRs in minor features/bug fixes.
 - Timeframe: 3-6 months of consistent engagement.
 - Engagement: Active in GitHub discussions, follows contribution guidelines.
2. **Regular Contributor** (Intermediate):
 - Contributions: leading medium complexity features, triaging issues.
 - Timeframe: 6-12 months.
 - Engagement: Leading discussions, mentoring newer Contributors.
3. **Maintainer** (Advanced):
 - Contributions: managing feature branches, leading major features.
 - Timeframe: 2+ years.
 - Engagement: Reviewing PRs, guiding project direction, collaborating with stakeholders.
4. **Steering Committee** (Leadership):
 - Selection: Maintainers demonstrating leadership, strategic vision, and active involvement in long-term planning.
 - Timeframe: 3+ years of sustained, impactful contributions.

This progression ensures that Contributors grow in responsibility, technical expertise, and leadership, aligning with OHIF's long-term goals.

3. Roadmap and Project Steering

3.1 Roadmap Prioritization

The prioritization of the project roadmap is determined by the Steering Committee. Consideration is taken to meet overall project funding aims, technical direction, and community needs. Community feedback collected via GitHub issues, discussions, and regular surveys is a valuable input when setting priorities. Regular roadmap reviews are conducted to assess progress, adjust priorities, and address any emerging needs or challenges. Each milestone on the roadmap is assigned clear deadlines and deliverables. The Steering Committee reserves the right to amend the project roadmap based on determined needs, including changes to project funding, technical or competitive landscape, collaborations, and feedback from the OHIF community. Roadmap changes are publicly documented on the OHIF website.

3.2 Project Steering and Conflict Resolution

The Steering Committee meets regularly to discuss progress towards critical project aims and milestones. In the event that a roadmap item or other project aim is in danger of failure, the Steering Committee discusses mitigation and delegates implementation of mitigation steps to the core maintainer team. As necessary (including, but not limited to, the acceptance of roadmap changes, community advisory role input, and priority conflict resolution), the Committee uses a simple majority vote to resolve decisions. Steering Committee meetings are summarized and released as public documentation.

4. Technical Decision Process

4.1 Informed Decision Making

OHIF employs a data-driven decision-making process, utilizing feedback from community discussions held on GitHub Issues or through other community outreach, such as meetings, conferences, etc. Once an issue is understood, the Project Maintainers discuss the implementation strategy and timeline. It is the responsibility of the project maintaining team to determine if an issue requires escalation to the Steering Committee for further review. Decisions regarding LTS vs regular releases are made based on the criticality and stability requirements of the proposed changes.

For critical issues such as security vulnerabilities, a fast-track decision-making process is used. Project Maintainers approve urgent fixes without full community discussions. For standard decisions, votes are held by the Steering Committee, and a simple majority is used for approval. Fast-tracked decisions can impact both the LTS and regular release branches depending on the issue's scope. All technical decisions are publicly documented in GitHub issues.

4.2 Feature Requests

Proposals for new features or project changes are submitted through GitHub Issues or collected through survey and other community input. Feature requests are reviewed and discussed with the community, and approved by the core Project Maintainers, including the System Architect and Project Manager. Where necessary, such as major or breaking changes, requests are escalated by the Project Maintainers to the Steering Committee for review and may require further action, such as request for comment, prior to approval. When reviewing proposals, consideration is given to whether the changes should be integrated into regular or LTS branches. New features typically target regular releases, while critical updates are backported to LTS branches.

4.3 Documentation

OHIF maintains publicly available documentation for community reference. The following is a non-exhaustive list of available documentation:

- Product changes are documented publicly through a detailed change log
- New features are provided with user guidelines and video/other tutorials as deemed necessary
- Processes and procedures governing the Steering Committee and core sustainment team's activities are available in a public repository
 - Request for comment procedure
 - Software development and validation procedure
 - Process for attraction and management of funding agreements
 - Commitment to open source
- Funding agreements that are not subject to confidentiality or other prohibitive agreements are publicly available alongside the OHIF roadmap
- Code of Conduct is available on GitHub

5. Contribution Workflow

5.1 Code Contributions

Contributors submit code through pull requests. Project Maintainers review the code for quality, adherence to project standards, and alignment with the project roadmap. Once approved, the

code is merged into the appropriate branch: either the `master` branch for regular releases or the appropriate LTS branch for backported fixes. Contributions to OHIF are licensed under the MIT License, a permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

5.2 Documentation Contributions

Contributors can submit documentation improvements and additions. Documentation is critical to the usability of OHIF, and a dedicated Project Maintainer ensures that contributions meet quality and accuracy standards. LTS branches have their own dedicated documentation to ensure that long-term users have access to stable feature documentation.

5.3 Community Contributions

Non-code contributions, such as organizing events or contributing to community discussions, are essential to the project's growth. Contributors propose community initiatives which are then reviewed by the Community Manager and the Steering Committee.

5.4 Review Processes and Quality Control

Every contribution, whether code or non-code, goes through a rigorous review process. Code is tested, and documentation is verified for accuracy. This ensures that contributions maintain high quality and consistency. Quality control is especially important in LTS branches, where stability is paramount.

5.5 Best Practices for Managing the Code

To maintain OHIF with high-quality standards, long-term stability, and the trust of the community, all OHIF core Maintainers adhere to the following best practices:

1. **Interface Stability:** Once an interface is published, it should never be broken. If changes are needed, introduce a new version while maintaining backward compatibility. For backward-compatible changes, include warnings to notify users that the deprecated interface will be removed in the future version.
2. **Code Review:** All code contributions should undergo thorough peer review to ensure quality, consistency, and adherence to project standards.
3. **Testing and Validation:** Ensure comprehensive automated and/or manual tests for all code changes, especially for critical components and interfaces.
4. **Documentation:** Maintain clear and up-to-date documentation, especially for APIs and interfaces, for all components, including any new features or changes.
5. **Versioning:** Follow semantic versioning to clearly communicate updates, including major changes, minor improvements, and patches.

6. **Continuous Integration:** Use CI/CD pipelines to automatically test and verify code before merging, ensuring no regressions are introduced.
7. **Deprecation Policy:** If an interface or feature needs to be deprecated, provide clear communication, documentation, and a transition period for users to adapt.

5.6 Contributor Recognition

Notable contributions to the project are discussed regularly by the OHIF core Maintainers and Steering Committee. The core Maintainers regularly acknowledge notable contributions through public announcement in the OHIF newsletter, website, or other community channels. Emphasis is given to recent and relevant contributions, and all past notable contributions are catalogued for public reference.

6. Funding and Sustainability

6.1 Grants and Sponsorships

The project relies on grants and sponsorships to fund its development and community activities. The Steering Committee, working together with other key personnel such as the Community Manager, is responsible for identifying grant opportunities and working to secure funding.

6.2 Budget Oversight

The Project Manager is responsible for maintaining and budgeting allocated funds to the project to ensure all funding commitments are met. The Project Manager reports the status of project funding to the Steering Committee and executes Steering Committee directives related to dispersal and usage of available funding.

6.3 Transparency in Funding

All financial decisions, including the allocation of grants and sponsorships, are transparent and publicly documented, with the exception of funds committed under a confidentiality agreement with the funding party. Funding to OHIF that results in contribution to the platform is conditional upon open-source code release. This enhances accountability and trust within the community. The following section outlines potential sources for sustainment funding.

6.4 OHIF Funding Mechanisms

1. **Grant Support:** OHIF receives direct grant funding from federal and private sources and participates as an open-source collaborator on community member grants.

2. **Tax Benefits for Donations:** OHIF is structured as a nonprofit through affiliation with Massachusetts General Hospital to enable tax-deductible donations for supporting open-source development. Companies and individuals donating to OHIF could benefit from tax incentives.
3. **Collaborative Projects:** Partner with healthcare organizations, universities, or research institutions to fund specific features, integrations, or research-driven initiatives related to OHIF.
4. **Corporate Sponsorships:** Approach medical imaging equipment manufacturers, healthcare software vendors, and technology firms for sponsorships, in exchange for custom integrations or on-demand consulting within the OHIF ecosystem.
5. **Raising Awareness:** Solicit public donations on a marketing webpage or portal. Launch crowdfunding campaigns targeting healthcare professionals and the tech community to raise awareness and financial support for OHIF's development. Attend relevant trade shows and other events to attract opportunities to OHIF.
6. **Foundations and Charitable Organizations:** Partner with philanthropic organizations that focus on healthcare, technology, or innovation to provide sustainable funding, as these entities often have a vested interest in improving public health through open technology solutions.
7. **Direct Development Contribution:** Offer the opportunity for companies or individuals to pay developers to implement specific features, fix bugs, or enhance performance, enabling them to contribute financially to project improvements.

7. Governance and Community Meetings

7.1 Governance Meetings

Regular governance meetings are held between the Steering Committee, Project Maintainers, and key stakeholders. These meetings cover strategic planning, resource allocation, financial reporting, and community management. The release management strategy, including LTS and regular release schedules, is regularly reviewed and adjusted based on community feedback and project needs.

7.2 Community Meetings

Community meetings provide an open forum for all Contributors and users to discuss project updates, upcoming features, and potential collaborations. The Community Manager facilitates these calls at a regular cadence, subject to availability, to ensure transparency and encourage feedback.

7.3 Contributor Meetings

Weekly meetings are hosted by the core Contributor team to interface directly with community members about technical issues, answer implementation questions, and document bug reports. The results of these meetings are documented and available for all community members to review, and noteworthy discussions are escalated to the Steering Committee as necessary.

8. Release Management Strategy

The OHIF project follows a structured release management strategy similar to Ubuntu's Long-Term Support (LTS) model. This ensures stability while allowing for innovation through regular updates.

8.1 LTS Support

LTS releases are issued approximately every year and focus on stability, security, and backward compatibility for that period. LTS versions receive critical security updates and bug fixes for an extended period (e.g., 1 year). These releases are ideal for institutions needing long-term reliability.

8.2 Regular Releases

Regular releases are issued more frequently (e.g., every 3-6 months) and contain the latest features and improvements. These releases allow for rapid innovation and community feedback but may not be as stable as LTS versions. These releases are supported for one subsequent release cycle, until replaced by the latest regular release.

8.3 Branching Workflow for LTS and Regular Releases

To support the LTS and regular releases, OHIF adopts a Git-based branching model that provides flexibility for both stable and fast-moving development environments. Below is a detailed breakdown of how the branching strategy integrates with the release management strategy:

- **`master` Branch:** The `master` branch is the bleeding-edge of development (e.g., `3.10-beta1.x`). All new features and enhancements are merged here. Each regular release (e.g. `release/3.9`) is branched off `master` before a new version is finalized and goes through comprehensive QA. This branch receives critical and high severity security patches weekly, with medium and low severity patches as deemed necessary by the core Maintainers.
- **`release` Branch** - The `release` branch (e.g., `release/3.9`) is created from `master` when the feature set for the upcoming version is complete. It focuses on bug fixes, testing, and final tweaks, leading to the final release version. This branch receives critical and high severity

security patches weekly, with medium and low severity patches as deemed necessary by the core Maintainers.

- **LTS Branches:** Separate LTS branches (e.g., `lts/4.0` after 3.9, 3.10, etc.) are maintained independently from the `master` and `release` branches, receiving only critical bug fixes and security patches.

8.4 Release Lifecycle and Workflow

Each release, whether LTS or regular, is subject to the OHIF software development and validation procedure to ensure quality and stability. Key steps include the development phase, feature freeze, testing, and release. Critical updates for LTS branches are backported from the main branch after thorough testing.

8.5 Backporting Strategy

Backporting plays a key role in maintaining security for LTS releases without introducing instability. Critical patches are selectively backported from the `master` branch to the LTS branches and tested extensively before being merged.

8.6 End-of-Life (EOL) Policy

LTS releases are maintained annually. Once an LTS version reaches End-of-Life, it no longer receives updates, and users are encouraged to upgrade to the latest LTS version.

Addendum: Implementing the Release Management Strategy

1. Define Release Cadence:

- Establish timelines for both Long-Term Support (LTS) releases (e.g., every year) and regular feature releases (e.g., every 3-6 months).

2. Set Up Branching Strategy:

- Create dedicated branches for `master`, LTS versions (e.g., `lts/3.0`), and regular releases (e.g., `release/4.0`).
- Implement the process for backporting critical fixes to LTS branches.

3. Automate Testing and CI/CD:

- Integrate continuous integration (CI) pipelines to automatically test and deploy branches.
- Ensure automated tests are run for each branch to maintain quality.

4. Establish Feature Freeze and Testing Phases:

- Create clear deadlines for feature freeze and thorough testing prior to LTS or regular releases.
- Ensure all new features are merged into the `main` branch before creating the release branches.

5. Backporting Strategy:

- Define criteria for backporting critical security and bug fixes from `main` to the LTS branch.
- Implement additional testing for backported changes to ensure stability.

6. Documentation and Communication:

- Document the new release management process and share it with Contributors via GitHub.
- Regularly update the roadmap and release notes, providing clarity on upcoming releases and features.

7. Monitor and Iterate:

- After a few release cycles, review the process, gather feedback from Contributors, and refine the release management strategy as needed.

Version History

Version 1.0 (June 2025)

- Initial release.

Version 1.1 (August 2025)

- Amended section 3.1 to include the possibility for roadmap changes in the interest of important community collaborations.

Version 1.2 (November 2025)

- Added effective date to versioning.
- Updated Advisory Board requirements in section 2.2
- Added clarity about public documentation of roadmap changes to section 3.1
- Added clarity about public documentation of Steering Committee meetings to section 3.2
- Added clarity about public documentation of technical decisions to section 4.1
- Clarified that all OHIF core Maintainers are responsible for adhering to the best practices listed in section 5.5
- Added section 5.6 to outline Contributor recognition guidelines
- Added detail to the funding transparency commitment in section 6.3
- Added details about security patching OHIF branches to section 8.3