OpenEMR Weekly Meeting Detailed Summary

November 3, 2026

Attendees: Stephen Waite, Brady Miller, David Eschelbacher, Jerry Padgett, Simon Quigley, Prabhu Pathak (LogiCoy), Ansuman Nayak (LogiCoy), Sherwin Gaddis, Eric Stern (OCE), Michael Smith (OCE), Pedro Camellon. Stephen Nielson was absent.

> *Italicized indented passages are background, clarification, and commentary added after the meeting, not statements made on the call. Items from the meeting chat are folded in where they belong.*

- **Shape of the meeting:** Brady chaired in Stephen Nielson's absence, and the call ran about an hour and a half. It covered a LogiCoy update, Jerry's portal and audit work, FHIR as the project's busiest area, a first look at the HTI-5 final rule, and whether telemetry should be public. A long PR review turned into a discussion of how to tell when CodeRabbit is really finished with a PR. One PR was merged on the call.

LogiCoy — Pilots and a PDMP Module

- **Status:** Ansuman said LogiCoy is testing its OpenEMR module with a couple of vendors. Once the master services agreement is complete, it expects to take one or two live within about two weeks. It is collecting active feedback and implementing it.
- **Forum post:** asked about posting the Packagist package information discussed last week, Ansuman said LogiCoy is waiting until a client is live and the suggestions have settled.
- **A second module:** LogiCoy would like to contribute further, including integrating its prescription drug monitoring program (PDMP) service with OpenEMR as a separate module. It covers 11 states; other states would need a different vendor. LogiCoy noted that Bamboo Health already has an OpenEMR module, built by Sherwin.

> *LogiCoy runs the state PDMP system for Pennsylvania and handles data collection for Illinois, as noted on 10/13, and sells integrations that let EHRs query PDMPs in-workflow. Sherwin announced the Bamboo Health integration on the forum in September 2024, with Florida as the pilot state; it is a reporting module (oe-bamboo-pmp) built on Bamboo's PMP Gateway. A 2026 forum reply notes that it needs updating for OpenEMR 8, which replaced direct global-variable access with a globals wrapper. Because the two vendors cover different states, separate optional modules fit both the module policy settled on 10/13 — vendor-specific code ships as its own package — and the reality that PDMP access is state by state.*
>
> *A separate PDMP module also keeps e-prescribing and PDMP decoupled, which matters because practices may use one vendor's e-prescribing with another's PDMP access, and because several states require prescribers to check the PDMP before prescribing certain controlled substances regardless of which e-prescribing product they use.*

Jerry's Work — Portal Rebuild and Write Audits

- **Write audits for patient-facing apps:** Jerry is planning audits for patient-facing writes, mostly from SMART apps; Stephen Nielson sent useful comments that morning.
- **Removing the portal's old framework:** Jerry is pulling the Phreeze framework (heard as “freeze framework”) out of the patient portal. He started with the new portal audit, in preparation for the FHIR write audits, and has removed ORM data-access objects that are at least ten years old. He has already added five to seven new services to replace them, and the existing front-end JavaScript and its dependencies are coming out too. With the way AI has learned OpenEMR's patterns, he said, its output is now “almost spot on.”
- **Brady:** that will probably be a big win for PHPStan as well, removing a lot of baseline entries. Jerry agreed.
- **Why not move the portal into a module?** Jerry once started to, but got sidetracked into refactoring the portal's front end, and the portal is now more embedded in core. Until the project decides how to handle core modules, he is reluctant to move it. Brady agreed: keep optimizing what exists and building on it, since the module situation could still change.

> *Phreeze is a small open-source PHP framework from the mid-2010s that generates model, data-access, and reporter classes from database tables. OpenEMR's patient portal adopted it for its data layer when the portal was rebuilt around 2016. Generated DAO code of that era predates PHP's modern type system, which is why it accounts for so many PHPStan baseline entries. Replacing it with OpenEMR's own service classes puts the portal on the same patterns as the rest of the codebase.*
>
> *This is the work Jerry laid out in his three-phase audit epic on 10/6. Phase 1 is fixing the portal audit and its storage; phase 2 is hold-and-review for writes from patient-facing SMART apps. Rebuilding the portal's data layer is the groundwork both phases need.*

Getting PRs In — Two Fronts

- **Brady's framing:** the project is attacking PR bandwidth on two fronts. The first is working through the PRs already open. The bigger one is getting AI agents to produce PRs that are easy to bring in, by having the code already in the right places — controllers rather than interface/ — and continuing to tighten automated checks so agents have little room to produce junk. A couple of PRs coming soon will tighten things further.
- **Jerry's experience:** the way he and Seth have been working — their AI tools reviewing each other's PRs back and forth — works very well. He finds Claude's reviews better than CodeRabbit's, though the project would be lost without either. He also stressed that a lot of code still needs to come out of OpenEMR core into core modules.
- **Next release:** planned for about four weeks after 8.5.0. Brady has a couple of bugs to fix that can have domino effects, but the release mechanism is now smooth. The wiki updates and announcements are still manual, and some automation for the AWS packages remains to be done in the DevOps repository.

> *A four-week cadence from the October 27 release of 8.5.0 points to late November for the next release. It continues the monthly rhythm the project has kept since the summer: 8.2.0 in July, 8.3.0 in August, 8.4.0 in September, and 8.5.0 in October.*

FHIR — The Project's Busiest Area

- **Lots of activity:** Brady has noticed many active FHIR PRs, mostly fixes. His recent security-focused audit of OAuth 2 and the APIs added negative testing for the standard API and the patient portal API, so that a future change cannot quietly reintroduce a vulnerability. He has not yet done the same for FHIR, which is large and has many open PRs.
- **A focus area:** Brady suggested putting the FHIR PRs into a special category to work through. An existing issue where people can claim pieces of work is probably stale. Stephen Nielson has traditionally been the FHIR expert, but Brady thinks everyone will now need to learn FHIR, the way billing once became everyone's business.
- **Jerry:** he has been reviewing a lot of the FHIR work, and he and Seth are creating many of the FHIR PRs. Many of the bugs now surfacing come from the FHIR write work, because people are finally using it.
- **SMART scope versions — a decision for maintainers:** Jerry has reworked the scopes. One open question is whether to stop accepting the older SMART v1 scopes for new clients, rather than continuing to arbitrate between the two versions. Brady, from his audit, saw that v1 scopes are simply converted to v2 internally and kept for legacy clients, and that v2 is far more granular. Jerry thinks dropping v1 for new clients is logical but has held off because nobody commented, and it could break things. Some v2 scopes are not yet supported; Observation is a good example of where v2's granularity matters.
- **Best community support in years:** about five different authors are submitting FHIR PRs, and they look solid; passing CI removes much of the worry. FHIR is also low-hanging fruit for adding tests, and coverage is rising.

> *SMART App Launch version 1 used coarse scopes such as patient/Observation.read and .write. Version 2 uses permission letters — c, r, u, d, s for create, read, update, delete, and search — and allows query parameters, so patient/Observation.rs?category=vital-signs grants read and search on vital signs only. OpenEMR's API documentation already labels v1 "deprecated but supported" and recommends v2, mapping .read to .rs and .write to .cud internally. ONC's HTI-1 rule moved certification for the (g)(10) FHIR API criterion to SMART App Launch 2.0. The usual transition is the one Jerry describes: stop issuing v1 scopes to newly registered clients while continuing to honor them for existing ones, then retire v1 after a published notice period.*
>
> *Granular scopes connect directly to the 10/6 discussion of writes: category-level scopes are how an administrator could allow an app to write vital signs but not lab results. That only works if v2 scopes are fully supported and enforced, which is why the unsupported v2 cases Jerry mentions matter.*

HTI-5 — A First Look

Links: HTI-5 removals and revisions · HTI-5 information session slides · Regulatory update deadlines

The Discussion

- **Michael raised it:** ONC certification is a lot simpler now. The person who knows it best, Stephen Nielson, was absent, so the detailed conversation should wait for him. But OpenEMR probably carries "scar tissue" from earlier ONC requirements that are no longer needed, which could be cleaned up, removed, or moved to make room for more modern work.
- **Effective dates, posted in the chat:** the final rule is effective December 28, 2026, except for amendatory instructions 7, 10, and 12, which take effect January 1, 2027.
- **Jerry's first read:** he has mainly been working on the C-CDA side, and many of the requirements he worried about were removed. C-CDA itself remains, but the import requirement appeared to be going away, which he called a big one. There are revisions to transitions of care, clinical information reconciliation, and five other criteria.
- **Is anything due by January 1?** No. The changes are removals and relaxations: the project keeps its certification without doing anything, though, as Michael said, it is advantageous to act on them.
- **David's caution:** requirements are a minimum, so removing a criterion is not a reason to remove a feature — unless it is something genuinely "dumb" that gets in the way. In his view, C-CDA import is of little real use but will linger, like fax, because people are used to it. The final rule also walked back much of what had been proposed.
- **Michael's view:** there is no rush. But if something that used to be required is no longer required and could move into a module or be handled outside OpenEMR, core becomes much easier to manage. Much of that code is legacy in both location and style, and may be used by only a handful of installations; if so, let someone who is interested maintain it as a module. Jerry: that again needs the core-module decision first.
- **Open questions on the call:** Brady wondered whether (g)(9) was the criterion the project's Inferno testing covers, and whether removing the Direct criterion means OpenEMR no longer needs to tell people they must get EMR Direct. Jerry said Direct should still be supported as long as C-CDA exchange is. Someone asked whether, with the API criteria removed, OpenEMR is no longer bound by US Core; Jerry doubted it. The conclusion: more research is needed, Stephen Nielson should lead, and a published plan would help everyone.
- **HTI-4:** David posted that the HTI-4 deadlines remain December 31, 2027, so those changes still need to be made.

What HTI-5 Actually Changes

Compiled after the meeting from ONC's summary and the October 29 information session.

Change	Criteria
Removed — effective December 28, 2026	(a)(9) clinical decision support; (b)(7) and (b)(8) security tags for summary of care; (b)(9) care plan; (g)(6) C-CDA creation performance; (h)(1) Direct Project; (h)(2) Direct, Edge Protocol, and XDR/XDM.
Removed — effective January 1, 2027	(a)(12) family health history; (c)(4) CQM filter; (e)(3) patient health information capture; (g)(1) and (g)(2) automated numerator recording and measure calculation; (g)(7) application access — patient selection; (g)(9) application access — all data request.
Revised	(b)(1) transitions of care — C-CDA and Direct requirements removed effective January 1, 2030; (b)(2) clinical information reconciliation — now functional, without named content standards or simultaneous display; (b)(11) decision support interventions — source-attribute and predictive-risk requirements removed; also (c)(3), (e)(1), (f)(6), and (g)(3).
Not finalized	Removal of the privacy and security criteria (the (d) criteria stay as they are); proposed removal of (a)(14), (f)(4), (f)(7), (g)(4), and (g)(5); the information-blocking proposals, still under consideration.

The scale: ONC released the rule on October 26, 2026, and it was published in the Federal Register on October 28. It removes 14 of the program's 60 criteria and revises 7, against 34 removals in the proposed rule. HHS estimates savings of about $1.55 billion in present value. National Coordinator Tom Keane framed it as setting "the floor," and ONC describes it as a reset ahead of future rulemaking on read-write FHIR APIs.

Two points from the call can be cleared up. First, (g)(10) — the standardized FHIR API criterion that the project's Inferno testing in CI exercises — is not removed. The removed (g)(7) and (g)(9) were the older "application access" criteria for patient selection and all-data requests, which OpenEMR has met largely with C-CDA documents. Second, nothing in HTI-5 releases (g)(10) from US Core: changes to USCDI and US Core remain proposals under consideration, so Jerry's doubt was well founded. FHIR work still has to stay conformant for certified deployments.

On Direct and EMR Direct: the stand-alone (h)(1) and (h)(2) criteria are gone, but the transitions-of-care criterion (b)(1) keeps its C-CDA and Direct requirements until January 1, 2030, and ONC kept the Direct standards themselves. OpenEMR's current certification lists EMR Direct as relied-upon software for both (b)(1) and (h)(1), so the note on the certification page still applies to (b)(1) for now. On import, (b)(2) survives as a functional requirement — systems must still reconcile and incorporate incoming problems, medications, and allergies — but it no longer names C-CDA as the content standard or requires showing two sources side by side. That is close to, though not quite, Jerry's reading.

Michael's point about scar tissue is concrete. Features built mainly to satisfy now-removed criteria — automated measure calculation for the old Meaningful Use reports, the C-CDA "all data request" API, patient health information capture — no longer carry certification test obligations. That makes them candidates for review once telemetry can show how much they are used. As Jerry said, the rule is a floor: removal from certification is permission, not an instruction.

HTI-4 is different. ONC's regulatory-update page lists two criteria with a December 31, 2027 deadline: (b)(3) electronic prescribing, moving to NCPDP SCRIPT 2023011 with real-time prescription benefit under HTI-4, and (b)(11) decision support interventions, under HTI-2. OpenEMR is not certified to (b)(3), so that deadline matters only if the project adds e-prescribing certification, as discussed with LogiCoy. The (b)(11) work may be smaller than first planned, since HTI-5 narrowed that criterion; worth confirming against the updated test procedure.

Core Modules, Dependencies, and Telemetry

Core Modules and Dependencies

- **Jerry's request for feedback:** pulling code out of OpenEMR core has long been a goal of his. He noticed a rule in the new architecture guidance saying, in effect, that a working dependency should not be removed, and he disagrees, but nobody has commented on the issue he opened.
- **Candidates waiting:** moving payments into a module was one of his original intentions, and several other features could come out entirely. He is holding off until the core-module approach is decided and until telemetry shows how much each feature is used.

Should Telemetry Be Public?

- **Jerry:** telemetry is important for deciding what to add and remove, but he has received little feedback on the data he publishes to the admins.
- **Is it public?** Michael asked. No — it is private to the admins, and publishing it is a board decision. Jerry said he has no problem with publishing it.
- **What feedback he wants:** help spotting trends. He has built charts and reports but would welcome help with the statistics. The data needs cleaning: demo installations have been reporting telemetry when visitors click "yes," and so have dead development branches.
- **Raw data or reports?** one view was that the goal was never to publish raw data, but to produce reports — possibly tiered — with raw data limited to those making decisions. Raw data is tricky to release because it includes geographic information that is not completely anonymous.
- **Jerry's response:** this was debated for years before telemetry went in, and the data collected was judged low-risk: the only location is geographic, at the level of a city, with no URL, address, or other identifier.
- **Can installations be identified?** David will look into whether the data could identify particular installations and write something up.
- **Using it:** it was pointed out that answering questions like "is this feature used?" needs the underlying data, not just a fixed report. The most important thing for now is using telemetry to improve OpenEMR, which is not yet happening. A short proposal on publishing some of it will be sent to Brady, and Jerry has a fresh batch of data ready to share.
- **Closing plug:** Michael's last words on the call were a plug for making the telemetry public.

> *Jerry introduced the telemetry design on the forum in March 2025. It is opt-in, and collects counts of menu clicks, country, region and city, language settings, and optionally browser and device information, described as non-personal and aggregated. An April 2025 forum poll asked the community what it was comfortable sharing. The data serves three audiences — developers deciding what to keep, implementers and vendors planning their work, and the Foundation describing the size of its community for grants — which is why the question of what to publish keeps returning.*
>
> *Other open source projects show workable models. Debian's opt-in popularity contest publishes aggregate package-usage statistics openly, and WordPress publishes the distribution of WordPress, PHP, and database versions across its installations. The usual safeguards are to publish aggregates rather than rows, coarsen geography (country or state rather than city), and suppress small counts — government health data releases commonly withhold any cell below 11 — so that a lone installation in a small town cannot be singled out. Version distribution and feature-adoption percentages are both low-risk and immediately useful to contributors deciding where to spend effort.*

PR Review — When Is CodeRabbit Really Done?

PR 15012 — FHIR Insurer Organization

Pull request: https://github.com/openemr/openemr/pull/15012 (by anun333)

- **The queue:** about 306 open PRs, or below 300 not counting several held up by flaky CI runs.
- **What the PR fixes:** creating an insurance company through the FHIR API — a POST of an Organization with the insurer type — returned a server error and saved nothing, because the parser called a method on a plain string. The fix reads both forms, maps the fields the way the insurance service expects, runs the validator, and returns 201 with the new record's ids. It removes 12 PHPStan baseline entries and adds API and service tests. The author lists follow-ups: PUT still fails, and a lowercase type code routes to a facility.
- **Jerry on CodeRabbit's reach:** CodeRabbit turns even small bug fixes into fixing many more bugs, which makes PRs take longer. By his research it is only about 58% accurate, though Brady has found it better than that. Another attendee agreed that reviews spiral: the bigger a PR gets, the more surface the AI has to find something new each time. Jerry digs until everything is fixed, because deep, buried bugs in OpenEMR tend to fire at the worst time, but maintainers may have to decide where the cutoff is.
- **Brady's problem as a reviewer:** from the PR page, he often cannot tell whether CodeRabbit is finished — whether it paused, stopped because a new commit arrived, or left issues unaddressed. On 15012, two CodeRabbit comments were unresolved, so the PR was judged not ready. Jerry will have it re-reviewed and push back on CodeRabbit where warranted.
- **A proposal:** require that CodeRabbit comments be resolved before merging — either fixed, or answered with a reason, which an agent can do. It was noted that a branch protection rule can require conversations to be resolved.
- **Jerry's caution:** telling CodeRabbit an issue is out of scope teaches it, and it may not flag a legitimate version of that issue in future. He prefers to have Claude assess each CodeRabbit finding and push back with reasons — CodeRabbit often agrees — and to judge minor findings himself. He also noted that AI tools sometimes miss CodeRabbit comments that are not attached inline to the diff.
- **Brady's technique:** ask an AI agent to go through all of one author's PRs — for example, the eight or ten from doctorman — and summarize which look ready, which still have open issues, and why.
- **Michael on verification:** Claude is better than CodeRabbit, but not perfect; at some point the human has to challenge the AI. Build checks into tooling wherever possible. Architectural mistakes are the hardest to catch. When an AI asserts something, ask for a verification protocol — a test or other proof — because “verification is the only way we get anything sane past the AI.” The project still has only about 20% test coverage, and test quality matters: he knows of people whose AI generated 10,000 tests, only for the first real request to the application to break.
- **Where AI does well:** with a clear spec and clear guardrails, AI does a very good job; when it has to make an architectural decision itself, it tends to choose wrong — one attendee described a side project that spent months on refactors repairing the AI's own early damage. Jerry pointed to the C-CDA conversion as work that would never have been finished so quickly by hand.
- **From the chat:** Pedro: “it comes back to how good is CodeRabbit.” Michael: CodeRabbit is “middle-of-the-road,” and has a command-line tool that shows what its review would say before a PR is opened.
- **Other review comments:** Jerry's review raised several data-mapping and consistency issues, and the author has fixes queued.

PR 15015 — Portal Login Wording (Merged)

Pull request: https://github.com/openemr/openemr/pull/15015 (by doctorman)

- **What it does:** display text only. First-time patients could not tell the read-only account ID from the username they chose, and entered the wrong one at login. “Account Name” becomes “Account ID (cannot be changed),” “Use Username” becomes “Choose your login username (minimum 8 characters),” and hints under the login fields say which value to enter. The failure message now names the fields checked, while remaining a single message for every failure.
- **Review:** CodeRabbit had no actionable comments, and Brady had also run Claude's review on it — the verdict was clean. Jerry agreed that simple PRs like this should not be over-reviewed. Brady merged it during the call.
- **Next step Brady suggested:** have AI go through the queue to find more low-hanging fruit like this.

Also Looked At

- **PR 15001:** Brady found it with AI as one that had been approved but not merged. The bug fix itself is simple, but a test method may have a CodeRabbit issue, which Seth is expected to address.
- **Copyright headers:** Claude reviews can be over the top — Brady once softened a Claude-written review before posting it. One PR came in despite a missing author and copyright header; one view was that a PR should be held for that, or it never gets fixed. Michael argued that license and copyright headers should not be an AI's job at all: a tool such as Apache RAT can check them, and anything deterministic should be moved off the AI's plate. Older code, such as the decade-old portal, has inconsistent headers that a tool would need tuning to handle.

> *How accurate is CodeRabbit? There is no independent consensus figure. Jerry's 58% contrasts with a March 2026 study by Signal65 reporting 95.88% precision and the most high-severity bugs found among five tools — but that study was commissioned by CodeRabbit. The two numbers also measure different things: precision counts how many flagged issues are real, not whether each one is worth fixing in this PR. Michael's “middle-of-the-road” is a fair working summary: a useful first pass that still needs judgment.*
>
> *Several of Brady's puzzles have mechanical explanations. CodeRabbit reviews incrementally, looking mainly at new commits, so a full review (“@coderabbitai full review”) can surface issues the incremental passes missed. A new push while a review is running can interrupt it. And replies explaining why a suggestion does not apply can be saved as “learnings” that shape future reviews in the repository — Jerry's worry about teaching it to ignore issues is real, though learnings can be reviewed and removed. The command-line tool Michael mentioned runs the same review on local changes, so an author can clear findings before anyone else sees the PR.*
>
> *GitHub's “Require conversation resolution before merging” setting blocks a merge until every review thread on the PR is marked resolved, and the OpenSSF's source-control best practices recommend it. Two caveats: resolving a thread is a click, not proof of a fix, so it works best paired with an expectation that each thread gets either a fix or a reasoned reply; and it applies to every reviewer's comments, not only CodeRabbit's.*
>
> *Michael's “verification protocol” has established tools behind it. Mutation testing — Infection is the standard tool for PHP — deliberately breaks the code and checks that some test fails, which exposes the kind of hollow test suite in his 10,000-tests story. Conformance suites like Inferno, reproducible bug reproductions, and contract tests against the API are other forms of verification an AI's claim can be checked against.*
>
> *For headers, Apache RAT, the Release Audit Tool from the Apache Creadur project, scans a codebase for files missing license headers. It runs from the command line or through Ant and Maven, and its own documentation warns that it relies on heuristics and needs tuning. In the PHP world, the FSFE's REUSE tool, which checks SPDX license identifiers, is a common alternative. Either way, header checks become a CI job rather than a review comment. Claude Code's review command, which Brady used, can be pointed at a PR by number and reports its findings privately unless asked to post them.*

Priorities, Approvals, and a Possible Theme

- **Pedro's question, from the chat:** where can a contributor find the project's current top priorities, to help pick PRs to review?
- **Brady:** FHIR is one clear focus, and there is an older FHIR issue about adding tests for all the endpoints, though it is a little stale. Security is the top priority, but that work runs through a separate, private channel.
- **Please approve PRs you think are ready:** anyone can approve a PR, Brady said, using the Approve option in the GitHub review panel, and then leave a comment. Approved PRs go to the top of the maintainers' queue — he uses an agent to find approved-but-unmerged PRs. Pedro said he has been approving some, while missing context on the wider codebase. Jerry had noticed a couple and ran additional reviews on them.
- **A theme sprint?** The project has never had theme sprints. One idea was for the admins to pick a focus, such as FHIR through the end of November. Volunteers will still work on what interests them, and a good PR would not be turned away for being off-topic. Jerry thought inviting FHIR PRs could work, but could not see maintainers all concentrating on one area.
- **Brady's advice:** work on what interests you; each developer can offer a wishlist — his is security. Jerry has found that many new contributors simply want to help the project and are open to anything; Seth, for example, built the FHIR Task resource.
- **Right now, reviewing is the priority:** managing the PRs already in the queue matters most, and reviewing is work few people enjoy, so it is especially valuable. Jerry floated pausing new PRs until the queue catches up. Brady disliked discouraging contributors, and noted that security intake cannot be paused. Jerry's concern is that PRs go stale and their contributors drift away.
- **Pedro:** a visible list of priorities or epics would help people — and their agents — pick something, since many contributors simply want something useful to work on. He found OpenEMR while building his own EHR to learn about healthcare, chose it to study a mature codebase, and works on AI. Brady's suggestion: FHIR and HTI-5.
- **Core modules:** Jerry asked that a plan for core modules be worked out when Stephen Nielson is available, since he has code he wants to move out of core.

> *On approvals: GitHub records an approval from anyone who reviews a public PR, but only approvals from people with write access count toward a required-review rule. A newcomer's approval is therefore a signal rather than a gate — exactly the use Brady describes. Maintainers can find these PRs with a search such as is:pr is:open review:approved.*
>
> *Pedro's question has now come up two weeks running. A short, pinned priorities issue — FHIR fixes and tests, the legacy-code cutoff, HTI-5 cleanup, review of approved PRs — with GitHub labels to match, would answer it. It would also give the AI triage discussed on 10/27 something to rank against, and let contributors point their agents at real needs rather than the top of the issue list.*

Items Discussed and Proposed

Recorded as topics raised and directions favored, not as assigned work or commitments.

Area	Discussed / Proposed
LogiCoy	Pilots under way; first clients live within about two weeks of the services agreement. Forum post once live. A separate PDMP module planned.
Portal	Phreeze framework and old DAOs being replaced with services; write-audit plan for patient-facing apps in progress.
PR intake	Work the existing queue while tightening automated checks; more restrictive PRs coming.
Release	Next release about four weeks after 8.5.0.
FHIR	A focus area for review and testing. Decision pending on refusing SMART v1 scopes for new clients.
HTI-5	Removals are optional; no action required by January 1. Plan to be led by Stephen Nielson.
Core modules	A decision is needed before moving the portal, payments, or other features out of core.
Telemetry	Publication debated; a proposal on publishing some data to go to Brady.
PR 15012	Not ready; unresolved CodeRabbit comments and review fixes queued.
PR 15015	Merged on the call.
CodeRabbit workflow	Proposal to require that review comments be resolved; caution about suppressing valid findings.
License headers	Automate checks with a tool such as Apache RAT rather than relying on AI review.
Approvals	Anyone may approve PRs they believe are ready; approvals move PRs up the queue.

Future Discussion and Next Meeting Agenda

Proposed for upcoming meetings, drawn from this week's discussion, chat, and notes; not a fixed agenda.

Item	Details
Telemetry	What can be made public and what should not; the proposal to Brady.
HTI-5 rule changes	Stephen Nielson's review: what can be cleaned up or moved to modules, and confirming that (g)(10) and US Core are unchanged.
HTI-4 deadlines	Changes still needed by December 31, 2027, especially (b)(11) and, if pursued, (b)(3).
SMART v1 scopes	Maintainer decision on refusing v1 scopes for newly registered clients.
Core modules	A plan for how core modules will work, with Stephen Nielson.
Dependency rule	Jerry's objection to the architecture guidance on keeping working dependencies.
CodeRabbit resolution	Whether to require resolved conversations before merging.
License headers	A CI check, using Apache RAT, REUSE, or similar.
Theme focus	Whether to name FHIR as a focus through the end of November.
Priorities list	A pinned list of current priorities for contributors (Pedro's request).
AI triage	Agreed on 10/27; only partly covered this week.
LogiCoy	First client go-live, the forum post, and the PDMP module.
Carried over	Multisite and subdirectory deprecation; FHIR procedure writes; logging #2119; Ensora outreach; contributor limits; the legacy-code cutoff.

Notes

- **The connecting thread:** evidence over assertion. Michael's argument about verifying what AI reviewers claim, the debate over whether telemetry should inform what stays in core, and Jerry's reminder that certification rules are a floor rather than a feature list all point the same way. Decisions should rest on tests, data, and the actual text of the rules, not on what a tool, a habit, or a first read suggests.
- **Stephen Nielson's absence was felt:** HTI-5, the SMART scope decision, and the core-module plan were all explicitly held for him.
- **Reviewing is now the bottleneck, and anyone can help:** a newcomer's approval is a useful signal, and Brady is already using AI to find approved PRs and quick wins.
- Ways to help: review open PRs — especially FHIR ones — and approve those you believe are ready, with a comment. Anyone with statistics experience could help Jerry make sense of the telemetry trends, and anyone who has worked through HTI-5's details is welcome to share an analysis. Comments on the forum or at the weekly call are welcome.