World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(REVIEW ARTICLE)

Check for updates

CREATING VISIBILITY IN REMOTE CLINICAL PROGRAMMING TEAMS

Jyoti (Jo) Agarwal [1,*] and Neetika Khurana [2]

[1] Gilead Sciences.
[2] Loyola University Chicago.
* Corresponding Author

World Journal of Advanced Research and Reviews, 2022, 15(03), 698–701

Publication history: Received on 18 August 2022; revised on 20 September 2022; accepted on 28 September 2022

Article DOI: https://doi.org/10.30574/wjarr.2022.15.3.0935

Copyright © 2022 Author(s) retain the copyright of this article. This article is published under the terms of the Creative Commons Attribution License 4.0

ABSTRACT

Remote and hybrid work can make contributions to clinical programming less visible when progress is communicated mainly through final deliverables or scheduled meetings. This narrative review adapts practical ideas about trust, communication, and leadership presence to statistical programming and clinical data teams. It presents a six-practice framework: set context, make work visible, build trust through follow-through, choose communication channels deliberately, demonstrate presence through substance, and adapt based on feedback. Examples include concise status updates, documented handoffs, review-ready artifacts, and meeting practices that help distributed colleagues understand decisions and dependencies. The framework is a practical synthesis of the supplied presentation and published leadership resources; it is not an empirical study and makes no measured claims about productivity or performance. The practices are intended to help programmers, reviewers, project leads, and managers coordinate work while maintaining the documentation and confidentiality expectations of clinical development.

Keywords: Remote Work; Clinical Programming; Statistical Programming; Communication; Trust; Leadership Development

1 INTRODUCTION

Clinical programming depends on coordinated work across programmers, statisticians, data management, clinical teams, quality reviewers, and project leadership. A program, dataset, table, listing, or review response may pass through several people before it is accepted. In a remote or hybrid setting, much of this effort happens outside shared workspaces. Colleagues may see the final deliverable without seeing the assumptions clarified, issues resolved, or handoffs completed along the way.

Visibility in this context does not mean being constantly online or promoting every task. It means making the status, reasoning, and ownership of work understandable to the people who depend on it. The source presentation on creating visibility in a remote setting emphasizes trust, clear direction, active listening, resilience, audience awareness, and credibility (Southwick 2022). Neeley discusses the different ways trust develops in remote teams, while Goman describes leadership presence through credibility, confidence, composure, connection, and charisma (Neeley 2021; Goman 2020). This review translates those ideas into repeatable behaviors for clinical programming teams.

2 SCOPE AND APPROACH

This narrative review synthesizes on the supplied 2022 presentation and the cited leadership books, then applies those concepts to common clinical programming interactions. The six-practice framework and examples are the author's synthesis; they do not describe a measured intervention or a specific sponsor, study, or organization. Teams should adapt the practices to their operating procedures, communication norms, validated systems, and confidentiality requirements.

3 A SIX-PRACTICE FRAMEWORK FOR REMOTE VISIBILITY

The framework below treats visibility as a team coordination practice. Each behavior helps colleagues see what is happening, why it matters, and what is needed next, without creating unnecessary communication overhead.

Table 1: Six practices for creating visibility in remote clinical programming teams

Practice	What to make visible	Clinical programming example
1. Set context	Purpose, expected output, owner, dependencies, timing, and decision points.	At kickoff, confirm the analysis deliverable, source-data dependencies, reviewer, target date, and how changes will be communicated.
2. Show progress	Completed work, work in progress, next step, risks, and help needed.	Post a brief update with the program or output reviewed, outstanding issue, next action, and expected handoff.
3. Build trust through follow-through	Reliable commitments, early escalation, and clear ownership.	If an upstream data change affects a table, explain the impact promptly, identify the affected outputs, and agree on a revised review path.
4. Match the channel to the work	Whether a topic needs a durable written record, quick clarification, or live discussion.	Use controlled project documentation for decisions and specifications; use a short call for ambiguity, then record the outcome in the approved location.
5. Demonstrate presence through substance	Preparation, clear reasoning, listening, and constructive participation.	In a review meeting, summarize the decision needed, show the relevant evidence, invite the reviewer's concern, and confirm the next owner.
6. Adapt and learn	Feedback, changing constraints, lessons, and improvement opportunities.	After a difficult handoff or repeated review comment, ask what would make the next exchange clearer and update the working practice.

3.1 Set context before work begins

A clear "compass" reduces guesswork. At the start of a task, state the purpose, intended output, owner, contributors, dependencies, target date, and the decision or review that will close the work. Explain why the task matters when that context is not obvious. For example, a request to revise a table shell is easier to prioritize when the team also knows whether the change is needed for an internal review, a planned submission, or a downstream analysis.

Context should be concise and available where the team works. A kickoff note or task record can capture assumptions and responsibilities; a meeting can resolve questions that need discussion. When scope changes, update the same record so a new participant can understand the current plan without relying on a private message history.

3.2 Make progress visible without creating noise

A useful update answers four questions: What is complete? What is underway? What happens next? Is anything blocked or at risk? A brief, predictable update is often more useful than frequent messages that repeat activity without clarifying impact. When possible, link the update to the approved work item, review artifact, or project record.

Progress evidence can include a completed code review, a documented assumption, a reconciled output, a resolved comment, or a handoff accepted by the next owner. The point is to show work in a form that others can verify. In regulated or controlled environments, use approved repositories and procedures; do not move study information into informal channels to make it more visible.

3.3 Build trust through reliable follow-through

Remote trust grows when colleagues can understand one another's responsibilities and observe dependable behavior over time. A programmer can strengthen that trust by keeping commitments, stating uncertainty, asking for help early, and documenting a change in plan before it surprises downstream partners. When a deadline is at risk, a clear escalation should identify the cause, likely impact, options, and the decision needed.

Reliability does not require pretending that every estimate will hold. It requires communicating when assumptions change. This is especially useful when upstream data, specifications, or review feedback affect a sequence of dependent outputs. A timely update gives the team a chance to adjust workload or sequencing while preserving a shared understanding of the work.

3.4 Choose communication channels deliberately

Not every issue belongs in a meeting. Written updates work well when the information is stable, needs an audit trail, or must be reviewed across time zones. A synchronous conversation can help when a question is ambiguous, several perspectives need to be reconciled, or a decision is stuck. After a live discussion, record decisions, rationale, owners, and due dates in the team's approved location.

For meetings, share a short agenda and identify the desired outcome. Be direct about the question to resolve, leave room for others to contribute, and adjust the plan if the discussion reveals a new risk. Listening includes noticing when participants are confused, silent, or working from different assumptions; it does not require inferring agreement from silence.

3.5 Demonstrate presence through substance

Leadership presence can be practiced at any role level. In a technical discussion, credibility comes from accurate, well-supported work and a clear explanation of its limits. Confidence can mean offering a reasoned recommendation while remaining open to correction. Composure helps keep a review focused when priorities shift. Connection means recognizing the expertise and workload of partners. Charisma, in a work setting, can be understood as communicating with enough energy and clarity to help others engage (Goman 2020).

These behaviors can be visible in a short design review: summarize the issue, point to the supporting evidence, explain the options and tradeoffs, ask for input, and state the next action. This approach allows technical contributors to be recognized for judgment and collaboration, not just for speaking often or appearing online.

3.6 Adapt and learn across handoffs

Remote teams benefit when the people doing and reviewing work share knowledge in both directions. A senior programmer can explain a pattern and invite questions; a newer programmer can surface a simpler method or identify an unclear instruction. Coaching and learning are not one-way roles. After a handoff, ask whether the next owner had enough context, whether the artifact was understandable, and whether any repeated friction can be removed.

Resilience in this setting is practical: face issues early, stay engaged with the work, and revise the approach when evidence changes. It does not mean accepting unsustainable workloads or treating every delay as an individual failure. Teams should examine dependencies and capacity as well as individual execution.

4 ILLUSTRATIVE EXAMPLE

Consider a remote team preparing a set of review outputs after a late clarification to an analysis specification. A low-visibility approach would be to make the requested edits, wait for the next scheduled meeting, and report only that the outputs are complete. A higher-visibility approach begins by confirming the clarification and identifying affected outputs. The programmer records the assumption in the approved project location, updates the expected sequence, and tells the statistician and reviewer which deliverables may change. The review meeting then focuses on the decision needed, evidence available, and any remaining uncertainty. The team records the agreed next steps and confirms the handoff owner.

This example does not claim that a particular turnaround time or quality improvement will result. It illustrates how context, progress, reliability, channel choice, presence, and learning can work together to reduce avoidable uncertainty in a distributed workflow.

5 PRACTICAL STARTING PLAN

A team can begin with a small, observable change rather than adding a new reporting process. For one recurring workstream, agree on a short status format with owner, current state, next step, dependency or risk, and decision needed. Use it for two or three cycles, then ask contributors and reviewers whether it improved handoffs or created unnecessary overhead. Retain only the fields that help the team coordinate.

Individual contributors can also choose one behavior to practice: clarify an assignment before beginning, send an early risk update, prepare a concise meeting summary, or ask a reviewer what information would make a future handoff easier. Managers can make the behavior safer by modeling honest status reporting and recognizing clear collaboration as well as completed deliverables.

6 TEAM CHECKPOINTS

Before work begins, confirm purpose, expected output, owner, dependencies, and review path. During execution, share progress at an agreed cadence and raise risks with enough context to support a decision. At handoff, identify what is ready, what remains open, where the supporting record lives, and who owns the next step.

After two or three cycles, ask the team whether updates made work easier to coordinate, whether reviewers had enough context, and whether any new reporting burden outweighed the benefit. Adjust the practice with the people who use it. The goal is a shared, lightweight view of work that supports decisions and dependable handoffs.

7 A REUSABLE STATUS FORMAT

A short status note can make progress easier to review while avoiding a new reporting layer. The following illustrative format can be adapted to the team's existing work item or project record:

- **Work item:** Create and review the planned analysis output.
- **Complete:** Specification review and initial programming are complete.
- **Next:** Run the agreed checks and send the output for independent review.
- **Dependency or risk:** Confirm the interpretation of one unresolved analysis rule with the statistician.
- **Handoff:** Reviewer and target date are recorded in the approved project location.

These example statements are generic and should not include confidential study details in an unapproved channel. The status format can be shortened when there is no dependency or decision to report.

8 CONCLUSION

Remote visibility is most useful when it helps colleagues coordinate, review, and trust the work. Clinical programming teams can build it through clear context, concise progress updates, dependable follow-through, deliberate channel choices, substantive participation, and learning across handoffs. These practices support connection without equating visibility with constant availability. Teams can test them in one workflow, gather feedback, and adjust them to fit their quality system and communication norms.

STATEMENTS ON COMPLIANCE WITH ETHICAL STANDARDS

Disclosure of conflict of interest

The author declares no conflicts of interest regarding this manuscript.

Disclaimer

The views are the author's own; no proprietary or confidential employer information is included.

REFERENCES

[1] Goman, C. K. 2020. Stand Out: How to Build Your Leadership Presence. London, UK: Kogan Page.

[2] Neeley, T. 2021. Remote Work Revolution: Succeeding from Anywhere. New York, NY: Harper Business.