

Perspectives on Leading Effectively in Platform Trials: Leadership and Technical Approaches

Zhen (Laura) Li, AstraZeneca

ABSTRACT

Platform trials—run under a master protocol with sub-study-specific protocols—accelerate innovation by adaptively evaluating multiple therapies within a unified, evolving framework, presenting distinctive technical and leadership challenges. This presentation draws on my experience as a product lead programmer for an in-house platform trial and shares how I addressed these challenges while leading a study programming team.

On the technical side, establishing and maintaining programs and specifications that balance consistency with adaptability is foundational to high-quality deliverables. Practical examples will show how these approaches flex to diverse analysis requirements and evolving protocols. On the leadership side, effective study management is grounded in comprehensive planning, thoughtful resource allocation, and proactive, solution-focused communication, and supported by fit-for-purpose tools that streamline workflow and collaboration. Cultivating a collaborative, adaptive, and growth-oriented culture, alongside structured support for study programmers at varying experience levels, helps the study programming team navigate steep learning curves, resolve blockers, manage pressure, and sustain continuous learning. Real-world examples will illustrate these approaches in action.

The presentation shares practical experiences and approaches for leading a study programming team toward resilience, productivity, and continuous improvement in adaptive platform trials.

INTRODUCTION

Adaptive platform trials conducted under a master protocol are evolving how clinical evidence is generated. By enabling multiple sub-studies to run in parallel under shared governance and infrastructure, these designs accelerate learning cycles, reduce redundancy, and provide the flexibility to add or terminate treatment arms as science evolves. However, the same features that drive efficiency—heterogeneous data, frequent protocol amendments, and concurrent analyses—also introduce distinct operational and technical challenges for study statistical programming teams.

This paper presents a practitioner’s perspective based on an in-house oncology Phase II platform trial, which successfully delivered the first internally conducted interim analysis for the AstraZeneca Oncology Biometrics team. This paper focuses on two complementary dimensions critical to sustained high-quality delivery. The first is a scalable technical architecture that balances consistency with flexibility across data cuts, SDTM, ADaM, and TLFs, encompassing specifications, programming strategies, and directory structures that enable traceability, parallelization, and rapid adaptation to evolving requirements. The second is a leadership framework that emphasizes proactive planning, effective resource allocation, and structured communication, supported by study programming team management practices that enable efficient onboarding, continuous capability development, and resilience under pressure.

The scope of this paper is intentionally practical and focused on experience sharing. It presents concrete patterns, illustrative examples, and lessons learned, with the aim of providing useful insights for those leading study programming teams in platform trials.

MASTER PROTOCOL TRIAL

A master protocol is defined as a protocol that allows one or more interventions to be assessed within the same clinical trial, in one or more patient populations sharing common enrollment, follow-up, and assessment procedures. The key characteristics that distinguish master protocols include:

- Integration: Multiple sub-studies operate under a single governance structure, with each sub-study potentially pursuing different objectives.

- Shared Resources: Control arms, protocol elements, infrastructure, and oversight are shared among sub-studies.
- Efficiency: This approach delivers expedited drug development, operational efficiencies, reduced patient burden, and increased cost-effectiveness.

Examples of master protocol trial types include umbrella, basket, and platform trials:

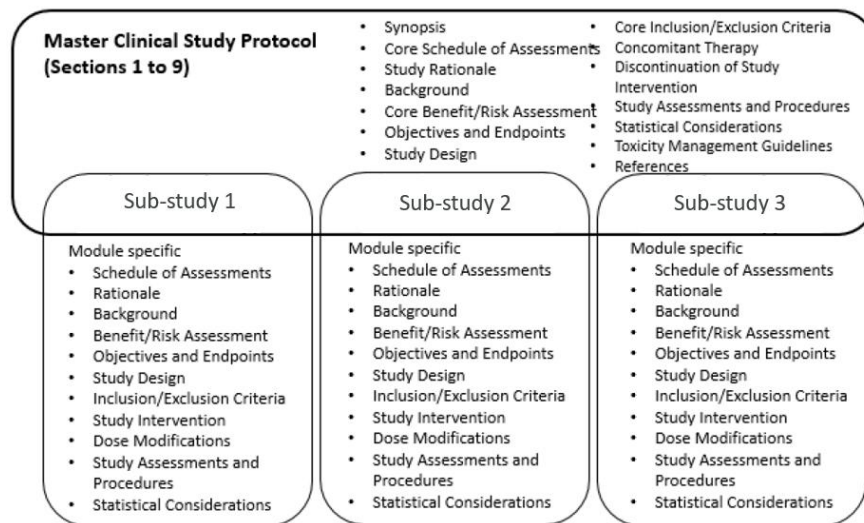
- Basket trial: A trial designed to evaluate a medical product for multiple diseases, conditions, or disease subtypes.
- Umbrella trial: A trial designed to evaluate multiple medical products concurrently for a single disease or condition.
- Platform trial: A trial designed to evaluate multiple medical products for a disease or condition in an ongoing manner, with treatments entering or leaving the platform over time.

A PLATFORM TRIAL

This paper focuses on an oncology Phase II platform trial managed entirely by an in-house study statistical programming team. The trial exemplifies the adaptive platform model: it was launched with two sub-studies in 2023 and later expanded to include a third sub-study in 2024 as the scientific strategy evolved. Each sub-study addresses a distinct indication and employs a distinct therapeutic approach. Sub-study 1 evaluates monotherapy across multiple dose levels. Sub-study 2 investigates combination involving the investigational product and various chemotherapies. Sub-study 3 assesses monotherapy through multiple parts.

The trial operates under a four-protocol structure, consisting of one overarching master protocol and three sub-study protocols. The master protocol defines the trial's foundational framework, including common governance structures, shared study conduct procedures, and core design principles that apply across all sub-studies. In contrast, each sub-study protocol provides detailed, indication-specific elements, such as, patient eligibility criteria tailored to the target indication, treatment administration schedules, and clinical evaluation timelines. This hierarchical design achieves a critical balance: the master protocol ensures operational consistency and enables integrated oversight, while the sub-study protocols preserve the flexibility necessary to address the distinct scientific questions of each indication. Figure 1 illustrates how these four protocols interrelate.

Figure 1: Protocol Structure



THE TECHNICAL APPROACH

THE CHALLENGE

Using this platform trial as an example, programming challenges exceed those of traditional designs, with data heterogeneity as a primary concern. A single electronic data capture (EDC) database consolidates data from all sub-studies, incorporating both universal case report forms and sub-study-specific forms or questions (e.g., eligibility criteria, exposure, and adverse events). This requires SDTM data-mapping rules that accommodate both shared and sub-study-specific logic, increasing the complexity of specifications and program implementation.

Analysis definitions introduce an additional layer of complexity. Patient allocation to sub-studies and treatment arms varies: some sub-studies involve randomization or assignment based on arm-specific criteria, while others assign all patients directly to treatment arms. This variability necessitates careful definition of key concepts, such as some baseline definitions and analysis populations. Other elements, including visit re-mapping rules and dose-limiting toxicity evaluable sets, also differ across sub-studies due to variations in treatment regimens and assessment schedules. These definitions may vary not only between sub-studies but also across treatment arms within a sub-study, further increasing programming complexity.

Frequent data migrations add further complexity. The introduction of new sub-studies or amendments to existing ones necessitates data migrations, with programming updates requiring careful assessment of downstream impacts on annotated CRFs, specifications updates, and SDTM, ADaM and TLFs programs. Coordinating these updates while supporting ongoing analyses demands rigorous planning and strong version control.

Diverse analysis requirements further amplify the challenge. Most analyses are performed at the sub-study or treatment arm level, including Safety Review Committee evaluations, sub-study- or arm-specific interim analyses, and sub-study-specific clinical study reports. Multiple interim analyses often require distinct data cuts for a specific sub-study, treatment arm, or combinations of sub-studies. Subgroup analyses also differ due to variations in patient populations, disease characteristics, and treatment histories.

Together, these challenges require a programming infrastructure that achieves two key objectives: consistency and flexibility.

TECHNICAL APPROACHES

To balance consistency and flexibility, strategic design decisions were implemented at each layer of the programming infrastructure: data cuts, SDTM and ADaM specifications and programs, and TLFs programs.

Data-Cut Approach

A single database supports all sub-studies, yet analysis requests may require different data-cut requirements. The data-cut approach was designed to accommodate various data-cut requirements.

Two independent, validated data-cut programs were developed:

1. Patient-level cut: Selects patients by sub-study or treatment arm across datasets.
2. Observation-level cut: Selects observations based on criteria defined on data-cut specification.

This design allows programmers to apply patient-level, observation-level, or combined cuts depending on analysis needs. For example:

- Analyses that include data from all sub-studies and share a common cut-off date require only the observation-level cut.
- Analyses that select patients by a specific sub-study or treatment arm without a cut-off date use only the patient-level cut.
- Interim analysis for a specific sub-study typically requires both cuts.

In addition to raw data, the patient-level cut program can also be applied to SDTM and ADaM datasets, efficiently enabling subgroup analyses.

Traceability and quality control are maintained through structured folders for pre-cut data, post-cut datasets, and selected patient lists.

SDTM Data Specification and Programs

Most SDTM mapping rules are shared across sub-studies, with some sub-study-specific definitions. CRF updates and data migrations require frequent specification maintenance. To ensure consistency and efficiency, a unified SDTM specification and centralized program set were implemented. This approach simplifies maintenance, harmonizes updates across sub-studies, and improves overall efficiency. Additional variables are included, where needed, to capture sub-study identification information (e.g., T1, TV).

ADaM Data Specification and Programs

Efficiency in ADaM dataset generation is critical to meeting analysis deliverable timelines, as complex analysis definitions are implemented at the ADaM level. Two programming approaches were considered. The first approach uses a single, unified program set across all sub-studies. This approach facilitates maintenance of definitions shared across sub-studies but introduces increased program complexity. The second approach involves separate program sets for each sub-study. While this simplifies individual programs and improves readability, it makes it more challenging to maintain consistency for definitions shared across sub-studies.

The second approach better supports parallel programming, enabling concurrent analyses across sub-studies. It also allows programming efforts to focus on variables and definitions specific to each sub-study and aligns with the production of separate CSRs and corresponding CRT packages. Based on these considerations, the second approach was adopted.

To further support this approach and improve efficiency in implementing variables and definitions shared across sub-studies, a unified ADaM specification structure was developed. A single specification file is used, with separate derivation columns for each sub-study (unused cells left blank). Existing macros were slightly modified—for example, the macro used to read variable attributes from the specification—to support this multi-column specification structure. Programs are stored in sub-study-specific central folders to maintain traceability.

TLF Programs

TLF outputs vary across sub-studies, and the approach mirrors that used for ADaM: TLF programs are maintained separately for each sub-study. Programs are centrally stored within sub-study-specific directories to ensure consistency and maintainability. Within the AstraZeneca Oncology Programming department, a wide range of user-friendly, high-quality template programs is available. Using these templates allows for efficient program setup and significantly reduces the time required to develop initial TLF programs for each sub-study.

This infrastructure enables the team to efficiently deliver complex, concurrent analyses while maintaining high quality.

DELIVERABLE MANAGEMENT

THE CHALLENGE

Leading the study programming team within this platform trial environment required balancing operational intensity with strategic complexity. The high volume and concurrency of analysis requests—since some sub-studies or treatment arms required multiple analyses throughout the trial—demanded careful planning and coordination. Over 50% of ad hoc requests required rapid turnaround or needed to be executed in parallel with analyses from other sub-studies or treatment arms. Each request involved unique analysis scopes, data requirements, and stakeholder considerations, providing opportunities for the team to demonstrate adaptability and deliver high-quality outputs under dynamic conditions.

Shifting efficiently across sub-studies highlighted the team's flexibility. Programmers supported analyses concurrently across sub-studies and arms, managing competing priorities and deadlines while maintaining accuracy and consistency. This environment fostered strong multi-tasking skills, situational awareness, and collaborative problem-solving within the team.

Collaboration across multiple stakeholder groups added both complexity and richness to the process. Diverse perspectives on deliverable objectives, scope, and timelines encouraged thoughtful discussion and alignment, fostering shared understanding and robust decision-making.

Collectively, these factors create a dynamic and demanding operating environment. Successfully navigating this environment requires a structured leadership approach that emphasizes adaptability, proactive communication, and the cultivation of team resilience to maintain high performance under continuous and evolving demands.

STRATEGIC FRAMEWORK: THREE PILLARS

Deliverable management is built on three interconnected pillars: planning, resource allocation, and communication. Together, these pillars provide a structured framework to ensure consistent, high-quality delivery in a complex and dynamic platform trial environment.

1. Planning

Planning focuses on establishing clear objectives for the study programming team while balancing milestone commitments with the flexibility needed to accommodate dynamic and evolving ad hoc requests. At the beginning of each year, an integrated plan is developed to align high-level objectives for all planned analyses. This plan provides clear visibility into periods of peak workload as well as opportunities for preparation, enabling proactive resource and timeline management. Based on this plan, cross-functional alignment meetings foster shared understanding of key delivery timelines and clarify input requirements across contributing functions.

Operationalizing the plan in Teams Planner is achieved through two structured tabs: study-level objectives, which capture all analyses in progress—including ad hoc requests—to support effective tracking of team progress and expectations; and compound-level objectives, which focus on milestone commitments while excluding ad hoc requests to maintain clarity for high-level planning. High-priority items—such as the first delivery of any sub-study or milestone analyses with evolving inputs—are clearly highlighted, and time buffers are incorporated to effectively accommodate expected variability.

2. Resource Allocation

Resource allocation emphasizes strategic role assignment and dynamic adjustment, rather than task-by-task delegation. In this study, the study programming team comprises two to three study programmers supporting three sub-studies, with each programmer assigned a defined role within each sub-study. These role assignments are informed by the technical approach, programmers' experience and time zones, the high-level deliverable plan, and the anticipated volume of ad hoc requests. This structured approach minimizes unnecessary context switching, strengthens accountability, and preserves flexibility to respond to time-sensitive needs.

Prioritization under competing timelines is managed through adaptive resource coordination. When simultaneous requests arise, the team is strategically split—for example, enabling parallel processing at the ADaM and TLF levels to accelerate delivery. These decisions are guided by defined roles, the progress of each request, risk considerations, and alignment with the overall technical approach.

By aligning resources with both individual strengths and specific analysis requirements, this approach supports efficient quality control, reduces operational friction, and provides team members with clear and stable responsibilities.

3. Communication

Communication is critical to aligning expectations across study teams and within the study programming team. Providing ongoing guidance on standard processes and realistic timelines, along with early clarification of requirements, supports effective cross-functional and cross-study collaboration—particularly in a complex platform trial with high programming demand.

Each programmer, or sub-group of programmers, is provided with a prioritized work list that reflects concurrent deliverables and ongoing preparation activities. Regular team meetings, described later, enhance transparency and support timely issue resolution.

Clear and proactive communication reduces misalignment, strengthens stakeholder confidence, improves operational efficiency, and reinforces a culture of transparency.

STUDY PROGRAMMING TEAM MANAGEMENT

The study programming team consists of one study lead programmer and two to three study programmers with varying levels of experience. A tailored onboarding program is implemented across three domains to accelerate capability development. First, programmers build proficiency in company standards, macros, and analysis systems through a dedicated test workspace and shared reference materials. Second, oncology domain knowledge is strengthened through exposure to oncology-specific derivations and datasets, along with hands-on assignments such as spot checks of efficacy datasets and the generation of efficacy TLFs. Third, the in-house programming workflow is introduced through the analysis & reporting internal handbook, detailed planning materials outlining end-to-end programming activities, and lessons learned from initial deliverables.

Development objectives are structured across three progressive stages. In Stage 1, the study programming team works under close guidance to build familiarity and confidence. Stage 2 emphasizes smooth and consistent delivery under oversight, with minimal direct involvement in detailed execution. Stage 3 focuses on independence, enabling the study programming team to manage study-level programming activities autonomously.

To maintain alignment and foster a supportive team environment, recurring touchpoints are established. Weekly team alignment meetings review overall priorities and progress, with the rationale behind planning decisions shared and potential risks discussed collectively to enhance transparency. This approach ensures that all programmers remain closely engaged in planning, strengthening ownership and proactive risk management. In addition, weekly one-on-one meetings with the study lead programmer and bi-weekly one-on-one meetings with study programmers provide opportunities to identify potential blockers and discuss support needs. When parallel support for multiple concurrent deliverables is required, the team is organized into sub-groups, with focused discussions addressing technical topics and emerging considerations via ad hoc calls. These structured interactions promote open communication, build trust, and strengthen team cohesion, enabling the team to perform effectively in a complex and dynamic environment.

CONCLUSION

Platform trials amplify both the opportunities and the operational complexity of statistical programming. The experience presented here highlights three key conclusions:

First, a modular technical architecture—separating patient- and observation-level data cuts, centralizing SDTM specifications and programs, and organizing ADaM and TLFs by sub-study—provides a disciplined backbone. This approach preserves shared standards while accommodating legitimate heterogeneity, ensuring both consistency and flexibility across analyses.

Second, reliable delivery depends as much on leadership systems as on technical design. A strategic framework anchored in comprehensive planning, role-based resource allocation, and transparent communication enables teams to absorb flexibility, manage concurrent ad hoc and milestone analyses, and sustain high-quality delivery under pressure.

Third, early investment in the study programming team—including structured onboarding, staged development, and recurring touchpoints—accelerates learning, builds confidence, and develops a resilient, independent team. These practices strengthen cohesion, foster trust, and enable the team to deliver complex analyses across multiple sub-studies while consistently meeting quality and timeline expectations.

In practice, these elements reinforce one another: the technical design makes work decomposable and traceable; leadership practices align stakeholders, clarify priorities, and protect focus; and a strong team

culture sustains performance amid evolving requirements. Together, they enabled an in-house programming team to deliver complex analysis requests across multiple sub-studies while consistently meeting quality and timeline expectations.

REFERENCES

1. US Food and Drug Administration (FDA). Master Protocols for Drug and Biological Product Development: Guidance for Industry. Silver Spring, MD: FDA; 2023.
2. George Washington University School of Medicine & Health Sciences. Overview of Modern Clinical Trial Designs. Washington, DC: GW SMHS; 2023.
3. US Food and Drug Administration (FDA). Master Protocols: Efficient Clinical Trial Design Strategies to Expedite Development of Oncology Drugs and Biologics: Guidance for Industry. Silver Spring, MD: FDA; 2022.
4. Roustit M, Demarq O, et al. Platform trials. Therapies. 2023;78:29-38.

ACKNOWLEDGMENTS

I extend my sincere appreciation to all study programmers for their outstanding collaboration and dedication over the past two years. I also thank the AstraZeneca Early Oncology Programming leaders for their support, trust, and for championing my continued professional growth.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name: Zhen (Laura) Li

Company: AstraZeneca

Email: zhen.li@astrazeneca.com

Any brand and product names are trademarks of their respective companies.