

CLINISYS LABORATORY SYSTEM

REPORTING CAPABILITY ANALYSIS

REPORT FUNCTIONALITY & PRODUCT SCOPE

PREPARED BY
ROSSANA BABAKHANI
UX RESEARCHER & DESIGNER

The taxonomy defines the reporting capability landscape. The analysis determines which capabilities are needed for patient, scientific, and management reporting.

01

PURPOSE

The Report Writer was intended to support increasingly sophisticated reporting without requiring laboratory users to become reporting or data specialists.

Rather than beginning with a predefined feature list, I first mapped the broader reporting capability landscape and then evaluated those capabilities against the types of reporting Clinisys needed to support.

The analysis addressed three questions:

What could a comprehensive reporting platform support?

Which capabilities become necessary as the underlying data grows more complex?

Which capabilities should users control directly, and which should be handled by the system or other data products?

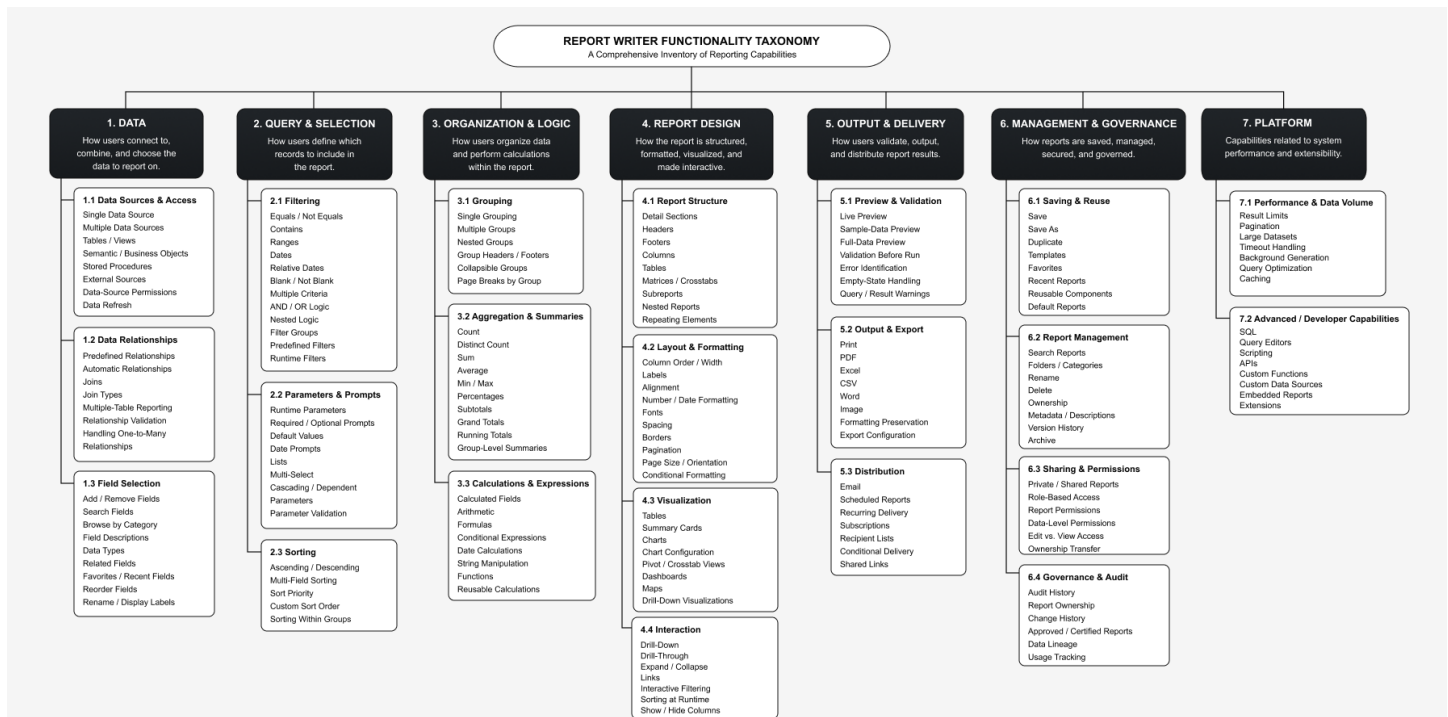
The goal was not to minimize reporting capability. It was to determine where complexity belonged.

REPORTING FUNCTIONALITY TAXONOMY

Before defining the Report Writer's scope, I created a taxonomy of functionality available across reporting platforms.

The taxonomy was intentionally **solution-neutral**. At this stage, functionality was not classified as required, unnecessary, user-facing, or system-managed. The purpose was to establish the full reporting landscape before making product decisions.

REPORTING CAPABILITY LANDSCAPE



WHAT THE TAXONOMY ESTABLISHED

The taxonomy exposed an important distinction early in the project:

A capability may be necessary for reporting without needing to become a feature the user has to understand or configure. For example, producing a report across related laboratory data may require joins and relationship validation. That does not mean a laboratory user needs a join editor.

The taxonomy therefore established what could exist. The next step was determining what the Report Writer actually needed to expose.

03

REPORTING COMPLEXITY MODEL

The planned development stages were based on the structure and scale of the underlying data. Reporting was expected to progress through three increasingly complex contexts:

PATIENT REPORTING

Single-record context: Patient reporting begins with a known patient and follows the clinical information associated with that patient.

PATIENT → ENCOUNTER → ORDER / TEST → RESULT → RELATED CLINICAL DATA

The data is comparatively bounded. Relationships are predictable, and the report primarily communicates information about an individual patient.

SCIENTIFIC REPORTING

Case and batch context: Scientific reporting expands beyond the single patient-centered record. Reporting may involve an individual scientific case or a batch of cases/specimens processed together.

**CASE / BATCH → SPECIMENS → TESTS / PROCEDURES → PROCESSING DATA → RESULTS
→ SCIENTIFIC DATA**

The reporting problem now involves:

- larger datasets
- more complex relationships
- cross-record reporting
- grouping
- aggregation
- comparison
- calculations
- more sophisticated filtering

MANAGEMENT REPORTING

Population and operational context: Management reporting expands beyond individual records and scientific work to examine activity across an organization. Reporting may span:

DISCIPLINE · LABORATORY · EMPLOYEES · CASES · TESTS · WORKLOAD · TURNAROUND · QUALITY · TIME

The reporting problem now involves large populations of records, organizational dimensions, time periods, aggregation, trends, operational measures, and recurring reporting.

Increasing Data Complexity

PATIENT

Single record + related data

→

SCIENTIFIC

Case or batch + larger related datasets

→

MANAGEMENT

Population + organizational dimensions + aggregated data

The development stages therefore weren't simply a roadmap for adding more features.

Increasing data complexity created the need for increasing reporting capability.

04 — Capability Analysis

With the capability landscape and reporting contexts established, I evaluated how each capability changed across Patient, Scientific, and Management reporting.

Analysis Key

- **Characteristic / required** — expected to be important at this reporting level
 - **Situational** — required depending on the report
 - **Not characteristic** — generally not part of this reporting context or not appropriate for the target user
-

Data Sources & Access

Capability	Patient	Scientific	Management	Product Direction
Select reporting data	●	●	●	Report Writer
Multiple data sources	○	●	●	Simplified for user
Tables/views	●	●	●	System-managed
Semantic/business objects	●	●	●	Present recognizable laboratory concepts

Capability	Patient Scientific Management			Product Direction
External sources	—	○	○	Platform capability
Data-source permissions	●	●	●	System-managed
Data refresh	○	●	●	Primarily system-managed

Data Relationships

Capability	Patient Scientific Management			Product Direction
Predefined relationships	●	●	●	System-managed
Automatic relationships	●	●	●	System-managed
Multiple related entities	●	●	●	Expose related information
One-to-many relationships	●	●	●	System-managed
Relationship validation	●	●	●	System-managed
Manual joins	—	—	—	Do not expose technical mechanics
Join types	—	—	—	System-managed

Finding: Users need access to related information. They do not need to understand or construct the joins required to retrieve it.

Field Selection

Capability	Patient Scientific Management			Product Direction
Add/remove fields	●	●	●	Direct
Search fields	●	●	●	Direct
Browse by category	●	●	●	Direct
Field descriptions	●	●	●	Support nontechnical users
Related fields	●	●	●	Simplify relationships
Reorder fields	●	●	●	Direct
Display labels	●	●	●	Direct
Favorites/recent fields	○	●	●	Progressive

Filtering

Capability	Patient Scientific Management			Product Direction
Basic field filters	●	●	●	Direct
Date/range filters	●	●	●	Direct
Blank/not blank	●	●	●	Direct
Multiple conditions	○	●	●	Direct
AND/OR logic	○	●	●	Progressive
Nested filter logic	—	○	●	Progressive
Predefined filters	●	●	●	Simplify common reporting
Runtime filters	●	●	●	Direct

Parameters & Prompts

Capability	Patient Scientific Management			Product Direction
Runtime parameters	●	●	●	Direct
Required/optional prompts	●	●	●	Direct
Default values	●	●	●	Direct
Date prompts	●	●	●	Direct
Lists	●	●	●	Direct
Multi-select	○	●	●	Direct
Dependent parameters	○	●	●	Progressive/system-assisted
Parameter validation	●	●	●	System-managed

Sorting & Grouping

Capability	Patient Scientific Management			Product Direction
Basic sorting	●	●	●	Direct
Multi-field sorting	○	●	●	Direct
Custom sort order	○	●	●	Progressive
Single grouping	○	●	●	Direct
Multiple groups	—	●	●	Direct
Nested groups	—	○	●	Progressive
Group headers/footers	○	●	●	Direct
Page break by group	○	●	●	Direct

Finding: Grouping becomes increasingly important as the reporting context expands from individual records to batches and populations.

Aggregation & Summaries

Capability	Patient Scientific Management			Product Direction
Count	○	●	●	Direct
Distinct count	—	●	●	Direct
Sum	○	●	●	Direct
Average	○	●	●	Direct
Min/max	○	●	●	Direct
Percentages	○	●	●	Direct
Subtotals	○	●	●	Direct
Grand totals	○	●	●	Direct
Running totals	—	○	●	Progressive

Calculations & Expressions

Capability	Patient Scientific Management			Product Direction
Calculated fields	○	●	●	Progressive
Arithmetic	○	●	●	Simplified
Date calculations	○	●	●	Simplified
Conditional calculations	—	●	●	Progressive
String manipulation	—	○	○	Minimize exposure
Functions	—	●	●	Simplified/predefined where possible
Formula language	—	○	○	Avoid exposing unless necessary

Finding: The need for calculated information does not automatically justify exposing a programming-like formula environment.

Report Structure

Capability	Patient Scientific Management			Product Direction
Detail sections	●	●	●	Direct
Headers/footers	●	●	●	Direct
Tables	●	●	●	Direct
Repeating elements	●	●	●	Direct/system-assisted
Multiple sections	○	●	●	Direct
Crosstabs/matrices	—	●	●	Progressive
Subreports	○	●	●	Progressive
Nested reports	—	○	○	Advanced

Layout & Formatting

Capability	Patient Scientific Management			Product Direction
Column order/width	●	●	●	Direct
Labels	●	●	●	Direct
Alignment	●	●	●	Direct
Number/date formatting	●	●	●	Direct
Page size/orientation	●	●	●	Direct
Pagination	●	●	●	Primarily automatic
Conditional formatting	○	●	●	Progressive
Detailed visual formatting	●	●	●	Direct where necessary

Visualization & Interaction

Capability	Patient Scientific Management			Product Direction
Tables	●	●	●	Report Writer
Basic report graphics	○	○	○	Evaluate within reporting
Charts	—	○	●	Data visualization ecosystem
Dashboards	—	○	●	Data visualization ecosystem

Capability	Patient Scientific Management			Product Direction
Maps	—	○	○	Data visualization ecosystem
Drill-down	—	○	●	Data visualization ecosystem
Drill-through	—	○	●	Data visualization ecosystem
Interactive filtering	—	○	●	Data visualization ecosystem

Finding: Report Writer needed to coexist with future data-visualization products, not absorb the entire analytics problem into report authoring.

Preview & Validation

Capability	Patient Scientific Management			Product Direction
Live/report preview	●	●	●	Direct
Sample-data preview	●	●	●	Direct
Validation before run	●	●	●	System-managed
Relationship validation	●	●	●	System-managed
Error identification	●	●	●	Identify and explain
Empty-result handling	●	●	●	System-managed
Data-volume warnings	—	●	●	System-managed

Output & Distribution

Capability	Patient Scientific Management			Product Direction
Print	●	●	●	Direct
PDF	●	●	●	Direct
Excel	○	●	●	Direct
CSV	—	●	●	Direct
Email	●	●	●	Direct
Scheduled reports	○	○	●	Management emphasis
Recurring delivery	○	○	●	Management emphasis
Subscriptions	—	○	●	Progressive
Recipient management	○	○	●	Direct

Saving, Reuse & Management

Capability	Patient Scientific Management			Product Direction
Save report	●	●	●	Direct
Save As / duplicate	●	●	●	Direct
Templates	●	●	●	Simplification opportunity
Favorites/recent	●	●	●	Direct
Search reports	●	●	●	Direct
Folders/categories	●	●	●	Direct
Ownership	●	●	●	Direct/system-managed

Capability	Patient Scientific Management			Product Direction
Version history	○	○	●	System-managed

Permissions, Governance & Audit

Capability	Patient Scientific Management			Product Direction
Private/shared reports	●	●	●	Direct
Role-based access	●	●	●	System-managed
Data-level permissions	●	●	●	System-managed
Report ownership	●	●	●	Visible
Change history	○	●	●	System-managed
Data lineage	○	●	●	Primarily system-managed
Usage tracking	—	○	●	System-managed

Performance & Platform

Capability	Patient Scientific Management			Product Direction
Result limits	○	●	●	System-managed
Pagination	●	●	●	System-managed
Large datasets	—	●	●	Platform capability
Background generation	○	●	●	System-managed
Query optimization	●	●	●	Platform capability
Caching	○	●	●	Platform capability
SQL editor	—	—	—	Do not expose
Scripting	—	—	—	Platform/developer capability
APIs	—	○	●	Platform/ecosystem
Extensibility	—	○	●	Supports future applications

05 — Findings

Complexity was driven by the data—not by an arbitrary feature roadmap.

Patient, Scientific, and Management reporting represented progressively broader reporting structures:

Single record → Case/batch → Population/organization

The Report Writer needed to expand because the questions being asked of the underlying data became more complex.

Necessary capability did not mean necessary user complexity.

Sophisticated reporting requires data relationships, validation, permissions, query processing, calculations, and other technical capabilities.

The analysis distinguished between **what the product needed to do** and **what a laboratory user needed to configure**.

Client requests represented only part of the reporting need.

Laboratory professionals understood the questions they needed their data to answer. They could not reasonably be expected to specify the reporting architecture required to answer them.

The analysis therefore translated reporting outcomes into enabling capabilities rather than using feature requests alone to define scope.

Reporting and data visualization were related but distinct.

The Report Writer was intended to work within a broader, application-agnostic reporting strategy. Interactive analytics, dashboards, exploratory visualization, and similar capabilities could be supported by future visualization products without turning Report Writer into an all-purpose analytics platform.

The objective was not to build less.

The objective was to **put complexity in the right place**.

User-facing capabilities remained accessible to report authors. Technical complexity that could be reliably managed by the system remained behind the interface. Capabilities better suited to analytics and visualization remained available to the broader product ecosystem.

06 — Product Direction

The analysis established a capability model that could expand with the underlying reporting problem.

PATIENT REPORTING

Start with a bounded, patient-centered reporting experience.

↓

SCIENTIFIC REPORTING

Expand to individual cases, batches, larger datasets, complex relationships, grouping, aggregation, comparison, and calculations.

↓

MANAGEMENT REPORTING

Extend reporting across disciplines, laboratories, employees, populations, operational measures, and time.

↓

DATA VISUALIZATION ECOSYSTEM

Enable future analytics and visualization applications to work alongside the reporting platform rather than forcing every data capability into Report Writer.

The resulting design challenge was not simply to make an existing reporting tool easier to use:

How might we give laboratory users access to increasingly sophisticated reporting capabilities without requiring them to become reporting or data specialists?