# Executive Summary

This document synthesizes insights from workshops held in May and June 2025, aimed at defining the roles, responsibilities, and essential skills of "Engineers" and "Developers" within the OneStream Software ecosystem. The primary objective is to address the current ambiguity surrounding these titles, ensuring a consistent understanding that will serve as a foundational pillar for the strategic planning and development of a robust Developer Education Program.

## Skill Mapping by Task Complexity

We identified three key dimensions for mapping skills:

* **Technical Expertise**: Beginner → Intermediate → Advanced
* **OneStream Experience**: From foundational to expert
* **Role Alignment**: Engineer/Developer, Consultant, Customer, etc.

This framework enables us to:

* **Design tiered learning paths** that support progressive growth
* **Tailor content** to specific personas (e.g., Engineers vs. Consultants)
* **Emphasize foundational skills**, including:
  + Workflow creation
  + Business rule development
  + Metadata management
  + BRAPI log debugging

## High-Impact Learning Categories

Tasks naturally grouped into the following learning categories:

* **Core Technical Skills** (e.g., data adapters, transformation rules)
* **Advanced Development** (e.g., Genesis blocks, custom dashboards)
* **Specialized Areas** (e.g., metadata, workflow design)
* **Soft Skills** (e.g., collaboration, communication)

These categories support:

* **Modular curriculum design** by skill domain
* **Balanced development** of technical and interpersonal skills
* **Certification pathways** aligned to these categories

## Common Developer Activities

Recurring tasks across roles include:

* Creating data sources and workflows
* Writing and reviewing business rules
* Managing security roles
* Building dashboards and cube views
* Testing and validating data

These will inform:

* Core curriculum modules
* Hands-on labs and sandbox exercises
* Skill validation checkpoints

## Role-Based Enablement

The task matrix revealed both overlap and divergence across roles:

* Engineers and Consultants share core development responsibilities
* Customers and Analysts focus more on testing, validation, and UI
* Implementation Partners span all categories

This insight drives:

* Persona-based learning journeys
* Role-specific enablement tracks
* Cross-role collaboration workshops

## Emerging and Advanced Topics

Advanced tasks surfaced, such as:

* External API integration
* Workspace assembly scripting
* XFBR development
* DOT MOVING (very advanced topic—haha!)

These may evolve into:

* Advanced learning modules
* Capstone projects
* Mentorship-driven learning experiences

# Overview

This paper synthesizes insights from two recent workshops (May and June 2025) aimed at precisely defining the roles, responsibilities, and essential skills of "Engineers" and "Developers" within the OneStream Software ecosystem. The primary objective is to address the current ambiguity surrounding these titles, ensuring a consistent understanding that will serve as a foundational pillar for the strategic planning and development of a robust Developer Education Program. By clarifying the technical and soft skills, key tasks, and relevant knowledge domains, OneStream can design targeted learning paths that empower its developer community to build reliable, efficient, and scalable solutions.

# 1. Introduction

In the rapidly evolving landscape of corporate performance management (CPM), the role of the developer is paramount to maximizing platform capabilities and delivering tailored solutions. At OneStream, the terms "Engineer" and "Developer" are frequently used interchangeably, leading to varied interpretations across internal teams, partners, and customers. This inconsistency poses a significant challenge for establishing a coherent and effective Developer Education Program. To address this, a series of workshops were conducted to collaboratively define the OneStream developer persona, ensuring that future educational initiatives are precisely aligned with real-world tasks and required competencies.

# 2. The Challenge: Bridging The Gaps

The initial premise for these workshops stemmed from a critical observation: asking five different individuals at OneStream "What is a OneStream Developer?" would likely yield five distinct answers. This ambiguity underscores a fundamental need for a unified definition. Without clear parameters for our target audience, any developer education strategy risks missing its mark, potentially leading to inefficient training, skill gaps, and suboptimal solution development.

# 3. Workshop Methodology

The insights presented in this paper are derived from collaborative brainstorming and categorization exercises conducted during virtual workshops in May and June 2025. Participants engaged in activities designed to:

* Identify roles and titles involved in creating, building, and developing on OneStream.
* Pinpoint specific tasks and activities performed by these individuals.
* Categorize activities based on technical expertise and OneStream experience.
* Map roles to associated activities to understand their distribution.

These workshops leveraged interactive platforms (Miro) to gather comprehensive data, which was subsequently analyzed to inform the developer definition and educational program planning.

# 4. Defining the OneStream Developer: A Unified Perspective

While "Engineer" and "Developer" are often used synonymously within OneStream, it is important to acknowledge their distinct descriptions in the broader market:

* **Engineers**: Typically focus on designing, building, and maintaining software systems, ensuring reliability, efficiency, and scalability through coding, testing, and debugging.
* **Developers**: Primarily responsible for writing code, creating software applications, translating requirements into functional code, implementing features, and managing the entire software development lifecycle (planning, design, testing, deployment).

**For OneStream, the converged "Engineer/Developer" role exhibits a primary focus on technical development and implementation.** This involves hands-on solution building, including writing code, creating formulas, debugging issues, and constructing custom applications. Their responsibilities span a wide array of technical tasks aimed at solving problems, optimizing performance, and ensuring software functionality, often in close collaboration with other technical teams.

# 5. Key Technical Domains and Activities

The workshops identified a comprehensive set of tasks critical to the OneStream developer's role, which can be grouped into distinct technical domains:

## 5.1. Environment & Architecture

This domain encompasses foundational tasks related to solution structure and deployment within the OneStream environment.

* Creating Applications within OneStream Environment
* Setting Up OneStream Solutions (on Marketplace)
* Setting Up a Cube
* Determine Solutions Architecture
* Determine Solutions Packaging
* Determine Solutions Requirements
* Dynamic dashboard creation
* Solution creation (PLN, TXM, RCM, etc.)
* Workspace assembly coding

## 5.2. Data Management & Integration

Tasks focused on data flow, transformation, and connectivity, crucial for effective data utilization.

* Importing Data
* Creating Data Management / Building Data Management
* Creating Data Management Jobs
* Creating Setup Tables
* Creating a Data Source
* Creating Tables
* Creating Transformation Rules
* Creating Data Adapters
* Creating Repeatable Processes
* Load/Extract Solution
* Setting Up Cash Flow
* Create SQL Scripts (CRUD)
* Data integration creation
* Data integration from other systems
* Database schema development
* Develop Connector, Parser, Conditional, and Derivative BRs
* Integration to internal data + systems
* Seed data

## 5.3. Security Configuration

Activities related to securing applications and managing user access within the OneStream platform.

* Setting up Security Roles
* Managing Security
* Setting up Community Solutions
* Setting up Partner Solutions
* Prevent currency translation
* Prevent data input
* Security of app/artifacts

## 5.4. Testing & Validation

Tasks that ensure solution accuracy, performance, and adherence to requirements.

* General Testing
* Validating/Testing Calculations
* Testing functionality
* Building Validation Files
* Validating Data
* Performance testing
* Rapid prototyping/POC

## 5.5. Debugging & Tools

(Note: While the May workshop explicitly listed "Debugging with BRAPI Error Log" and "Toggle Design Mode," the June data did not explicitly categorize tasks under this heading, suggesting debugging is often an inherent part of other development activities.)

* Debugging with BRAPI Error Log
* Copy OneStream Code Editor files to Visual Studio

## 5.6. Workflow & Metadata

Tasks involving business logic, metadata management, and process automation.

* Creating a Workflow
* Managing Metadata
* Building Workflow
* Creating Confirmation Rules
* Calculation engine (derivation/finance rules) / Calculation engine creation / finance rules
* Develop Spreadsheet BRs
* Develop custom eliminations
* Develop event handlers
* Query data/metadata via returned tables for export
* XFBR Development (formatting, parameter helpers, etc.)

## 5.7. Collaboration & Deployment

Activities supporting teamwork, documentation, and the rollout of solutions.

* Collab with Other Engineer/Dev
* Promoting Builds to Different Environments
* Attend Partner/Technical Webinars
* Attend Splash
* Attend Tech Talks
* Automate the mundane
* Document parameters
* Knowledge Transfer
* Send emails (+ HTML attachments)
* Team-based development

# 6. Essential Skills for Success

Beyond the specific tasks, OneStream developers require a blend of technical acumen and crucial soft skills:

## 6.1. Technical Skills

* **Programming Languages**: Proficiency in C#, SQL, VB.NET, and other relevant languages.
* **Data Management**: Skills in creating and managing data tables, importing data, and building data management jobs.
* **Debugging**: Ability to debug code and troubleshoot issues using relevant tools and logs.
* **Dashboard Development**: Experience in developing and customizing dashboards.
* **Security Management**: Setting up and managing security roles within OneStream.
* **API Integration**: Fetching data from external APIs and integrating it into OneStream solutions.
* **Business Rules**: Writing and implementing business rules to automate processes.
* **UI/UX Design**: Building user interfaces and ensuring a seamless user experience.

## 6.2. Soft Skills

* **Problem-Solving**: Strong analytical skills to identify and resolve technical issues.
* **Collaboration**: Ability to work effectively with other developers and technical teams, including team-based development.
* **Attention to Detail**: Precision in coding and testing to ensure robust solutions.
* **Communication**: Effective interaction with clients to understand needs and provide solutions.
* **Continuous Learning**: Engagement with webinars, workshops, community forums, technical blogs, and documentation (e.g., Read developer/platform documentation).
* **Knowledge Transfer**: Actively sharing knowledge within the team and community.

# 7. Implications for the Developer Education Program

The detailed understanding of the OneStream developer's role and associated tasks directly informs the structure and content of the Developer Education Program. Key learning outcomes should align with these findings to ensure comprehensive and practical training:

1. **Introduction to OneStream**: Comprehensive overview of the platform's capabilities, initial setup, and basic navigation.
2. **Core Technical Skills**: Foundational capabilities in programming languages, data management, and fundamental rule/formula creation.
3. **Advanced Development Skills**: Higher-order skills for extensibility, optimization, custom logic, and advanced dashboarding.
4. **Specialized Areas**: Domain-specific knowledge in data validation, metadata management, workflow management, and integrations.
5. **Strategic Implementation**: Project management, client interaction, and strategic planning for complex projects.
6. **Practical Experience**: Hands-on projects, case study analysis, and mentorship opportunities.
7. **Continuous Learning**: Encouraging participation in community engagements, webinars, and self-study.
8. **Soft Skills Development**: Cultivating problem-solving, collaboration, and leadership abilities.
9. **Resources and Support**: Providing access to study guides, online platforms, and mentorship.
10. **Evaluation and Feedback**: Implementing regular assessments and feedback mechanisms for continuous improvement.
11. **Career Development**: Support for job placement and networking opportunities.

This structured approach ensures that participants gain a holistic understanding of OneStream development and are equipped with the necessary skills to excel in their roles.

# 8. Recommendations and Next Steps

To build upon these foundational findings and further enhance the Developer Education Program, the following recommendations are proposed:

1. **Formalize Definitions**: Disseminate the refined definition of a OneStream Developer across the organization to ensure consistent understanding.
2. **Curriculum Mapping**: Directly map all identified tasks and skills to specific modules or courses within the Developer Education Program.
3. **Tiered Learning Paths**: Develop distinct learning paths (e.g., Beginner, Intermediate, Advanced) for each technical domain, reflecting the varying complexity of tasks.
4. **Integrate Soft Skills**: Ensure that soft skills development is not a separate track but is woven into technical training modules where applicable (e.g., team projects for collaboration).
5. **Address Debugging Tools**: Explicitly define tasks and provide training on debugging methodologies and tools within OneStream, given its under-representation in the June workshop's categorization.
6. **Continuous Feedback Loop**: Establish a mechanism for ongoing feedback from developers, partners, and customers to continuously update the developer definition and education program to reflect evolving needs and platform capabilities.

# 9. Conclusion

The collaborative effort to define the OneStream developer has yielded invaluable insights, moving us closer to a unified understanding of this critical role. By clearly outlining the required technical expertise, key responsibilities, and essential soft skills, OneStream is now better positioned to design and implement a world-class Developer Education Program. This initiative will not only empower individual developers but also strengthen the overall OneStream ecosystem, fostering innovation and enabling the delivery of high-quality solutions that meet the complex demands of corporate performance management.

# 10. Appendix

## [June Paper](https://onestreamsoftware-my.sharepoint.com/:w:/r/personal/mlatin_onestreamsoftware_com/Documents/Virtual%20Workshop%20What%20is%20a%20OS%20Developer/Workshop_Defining%20OS%20Developers%20-%20June%202025.docx?d=w2c95f759703741cf9ff8b0c169dac9b5&csf=1&web=1&e=PfzPiQ) [May Paper](https://onestreamsoftware-my.sharepoint.com/:w:/r/personal/mlatin_onestreamsoftware_com/Documents/Virtual%20Workshop%20What%20is%20a%20OS%20Developer/Workshop_Defining%20OS%20Developers%20-%20May%202025.docx?d=wa9f363c02b0c4801a4633d9811857e85&csf=1&web=1&e=J1e5fl)