**ServiceNow Scripting Fundamentals Course**

**Module Overview**

Module 4 of the ServiceNow Scripting Fundamentals Course is designed to equip learners with the essential skills and knowledge necessary for effective scripting within the ServiceNow platform. This module serves a dual purpose: it not only introduces key scripting concepts but also aims to deepen the understanding of how these concepts can be applied in real-world scenarios.

The primary goal of this module is to provide a robust foundation in the various aspects of ServiceNow scripting, including client-side and server-side scripting. Learners will explore the differences and applications of each type, ensuring they develop a comprehensive understanding of where and how to implement scripts effectively. By the end of this module, participants should feel confident in their ability to write and troubleshoot scripts that enhance platform functionality.

Additionally, this module emphasizes best practices in scripting, focusing on maintaining code quality and performance optimization. By instilling these practices, learners are prepared not only to write effective scripts but also to ensure their scripts are scalable and maintainable over time.

Hands-on exercises and real-life examples are integral to this module, allowing learners to apply what they have learned in practical situations. These activities are designed to reinforce the concepts covered and promote active engagement with the material. Throughout the module, learners will also have access to resources and support that will aid in their scripting journey, helping them navigate challenges and enhance their learning experience.

In summary, Module 4 is crucial for anyone looking to master scripting on the ServiceNow platform, as it lays the groundwork for advanced scripting techniques and fosters a deeper understanding of the platform's capabilities.

**Learning Objectives**

By the end of Module 4, participants will achieve several key learning objectives that will enhance their scripting skills and problem-solving abilities within the ServiceNow platform. These objectives are designed to ensure that learners not only grasp theoretical concepts but also feel confident in applying them in practical scenarios.

**Understanding Scripting Concepts**: Participants will gain a solid understanding of core scripting concepts, distinguishing between client-side and server-side scripting. This foundational knowledge will enable them to identify when to use each type of script effectively, enhancing their ability to develop solutions tailored to specific business needs.

**Writing and Troubleshooting Scripts**: Learners will develop the skills necessary to write, test, and troubleshoot scripts within ServiceNow. This includes an understanding of common error messages and debugging techniques, allowing participants to identify issues within their code and resolve them efficiently. By engaging in hands-on exercises, learners will practice these skills, reinforcing their ability to produce functional scripts.

**Implementing Best Practices**: Participants will learn best practices for scripting within ServiceNow, including code organization, performance optimization, and maintainability. By adopting these practices, learners will ensure that their scripts are not only effective but also scalable and easy to manage as the platform evolves.

**Applying Real-World Solutions**: The module emphasizes the application of scripting skills to real-world problems. Participants will engage in case studies and practical exercises that mimic scenarios they may encounter in their roles. This hands-on experience will empower learners to apply their knowledge to solve complex issues that arise in the ServiceNow environment.

**Enhancing Problem-Solving Skills**: Through collaborative exercises and challenges, learners will enhance their problem-solving skills, equipping them to tackle scripting challenges creatively and efficiently. By working together to analyze problems and develop solutions, participants will gain valuable insights from their peers and instructors, fostering a collaborative learning environment.

By achieving these objectives, participants will emerge from Module 4 with a comprehensive skill set that prepares them for more advanced scripting tasks and the ability to contribute meaningfully to their organizations' ServiceNow implementations.

**Introduction to ServiceNow Scripting**

Scripting in ServiceNow is a powerful feature that enables users to customize and extend the platform's capabilities beyond its out-of-the-box functionalities. At its core, scripting involves writing code to automate processes, manipulate data, and enhance user interactions. Understanding the fundamentals of scripting is crucial for anyone looking to leverage ServiceNow effectively.

There are two primary types of scripting within ServiceNow: client-side scripting and server-side scripting. Client-side scripts run in the user's browser and are typically used to manage user interface behaviors, such as form validations and dynamic field updates. In contrast, server-side scripts execute on the server and are responsible for tasks such as data processing, business logic enforcement, and integration with other systems. Recognizing the differences between these two types of scripts is essential for designing efficient solutions that meet specific business needs.

Another foundational concept in ServiceNow scripting is the use of Glide APIs, which are built-in functions that facilitate interactions with the ServiceNow platform. These APIs allow developers to perform a wide range of operations, such as querying records, updating data, and managing user sessions. Familiarity with Glide APIs is critical for creating effective scripts, as they provide the necessary tools to harness the full potential of the ServiceNow environment.

Moreover, scripting in ServiceNow is not just about writing code; it also involves adhering to best practices to ensure the scripts are maintainable and performant. This includes organizing code logically, using comments for clarity, and avoiding overly complex structures that can hinder future updates. Adopting these practices not only improves the quality of the scripts but also enhances collaboration among team members, making it easier to manage and update scripts as the platform evolves.

In summary, a solid understanding of ServiceNow scripting fundamentals is vital for anyone working with the platform. By mastering the basics of client-side and server-side scripting, becoming proficient with Glide APIs, and adhering to best practices, learners will be well-equipped to create effective solutions that enhance the overall functionality of ServiceNow.

**Scripting Fundamentals**

To effectively engage with ServiceNow scripting, it is essential to comprehend several fundamental programming concepts that underpin the platform’s scripting capabilities. These principles not only enhance the efficiency of scripts but also improve the maintainability and scalability of solutions built within ServiceNow.

One of the key programming concepts is understanding **variables** and **data types**. Variables serve as storage locations for data in scripts, allowing developers to manipulate information dynamically. ServiceNow supports various data types, including strings, integers, and arrays, each serving different purposes in script execution. Mastering the use of these data types is crucial for crafting effective scripts that can handle diverse data scenarios.

Another important principle is **control structures**, which dictate the flow of execution in scripting. Control structures include conditional statements (like if-else statements) and loops (such as for and while loops), allowing developers to implement logic that responds to specific conditions or iterates over collections of data. Understanding how and when to use these structures is vital for creating scripts that perform complex tasks efficiently.

**Functions** are another essential concept in scripting. Functions allow developers to encapsulate code into reusable blocks, promoting modularity and reducing redundancy. This practice is particularly important in ServiceNow, where similar operations may need to be performed across multiple scripts or modules. By leveraging functions, developers can enhance code readability and maintainability, making it easier to update scripts in the future.

Additionally, **error handling** is a critical aspect of effective scripting. Implementing error handling techniques, such as try-catch blocks, enables developers to manage exceptions gracefully, ensuring that scripts can recover from unexpected issues without crashing. This is particularly important in a production environment where reliability is paramount.

Lastly, understanding **object-oriented programming (OOP)** principles can significantly benefit ServiceNow scripting. OOP allows developers to create and manage complex systems by organizing code into objects that represent real-world entities. By utilizing OOP concepts, such as inheritance and encapsulation, developers can create more robust and reusable scripts that align with ServiceNow’s architecture.

By grasping these fundamental principles of programming, learners will be better equipped to navigate the intricacies of ServiceNow scripting, leading to more efficient, reliable, and maintainable solutions that effectively meet the needs of their organizations.

**ServiceNow Scripting Best Practices**

When writing scripts within ServiceNow, adhering to best practices is essential for optimizing performance, maintainability, and scalability. Here are several key guidelines to consider when crafting your scripts.

**Code Organization**

Organizing your code logically is vital for readability and maintenance. Use consistent naming conventions for functions and variables, making it easier for others (and future you) to understand the script's purpose. Group related functions together and utilize comments liberally to explain complex logic or the purpose of specific code blocks. This practice not only helps in troubleshooting but also enhances collaboration among team members.

**Avoid Hardcoding Values**

Whenever possible, avoid hardcoding values directly into your scripts. Instead, utilize configuration records or constants that can be easily modified without changing the code itself. This approach allows for greater flexibility and adaptability, especially when business requirements evolve or when deploying across different environments.

**Optimize Performance**

Performance optimization is crucial, especially in high-traffic environments. Favor client-side scripts over server-side scripts for operations that can be performed in the browser, as they reduce server load and improve response times. Additionally, use GlideRecord efficiently by limiting the number of records returned and applying filters that narrow down the dataset to only what is necessary.

**Error Handling**

Implement robust error handling to manage exceptions gracefully. Use try-catch blocks to capture errors and log them appropriately, allowing you to troubleshoot issues without disrupting user experience. Providing meaningful error messages can also assist users in understanding what went wrong and how to proceed.

**Reusability and Modularity**

Encapsulate commonly used logic within functions or classes to promote reusability. This modular approach not only reduces code duplication but also makes testing and updates easier. If a function needs modification, you can adjust it in one location rather than searching through multiple scripts.

**Limit Script Size**

Keep scripts concise and focused. Large scripts can become unwieldy, making it difficult to identify issues or understand the logic. If a script grows significantly, consider breaking it into smaller, more manageable components. This practice enhances readability and makes it easier to maintain.

**Test and Debug**

Always test your scripts in a development or test environment before deploying them to production. Use the built-in debugging tools in ServiceNow to step through your code and identify any issues. Implement unit tests where applicable to ensure that your scripts perform as expected and to catch potential errors early in the development process.

By following these best practices, developers can create high-quality scripts that not only enhance the functionality of the ServiceNow platform but also stand the test of time, ensuring they are easy to maintain and adapt as business needs change.

**Hands-On Exercises Introduction**

Hands-on exercises are a critical component of the ServiceNow Scripting Fundamentals Course, providing learners with the opportunity to apply theoretical knowledge in practical scenarios. These exercises are designed to reinforce the concepts introduced throughout the module, allowing participants to gain confidence and competence in their scripting skills.

During the hands-on exercises, learners will engage in various activities that mirror real-world challenges they may face while working on the ServiceNow platform. These activities will require participants to write, test, and debug scripts, ensuring they become familiar with the scripting environment and the tools available within ServiceNow. By tackling practical problems, learners will not only solidify their understanding of scripting principles but also develop problem-solving skills that are essential for their future roles.

The exercises will cover a range of topics, including client-side scripting and server-side scripting. Participants will have the chance to experiment with different types of scripts, allowing them to explore the nuances of each and understand their appropriate applications. This hands-on approach ensures that learners can distinguish between the two scripting environments and choose the right one for specific tasks.

Moreover, learners will be encouraged to collaborate during these exercises, fostering a sense of community and teamwork. By working together to solve problems, participants can share insights and strategies, enhancing their learning experience. This collaborative environment not only aids in knowledge retention but also simulates real-world working conditions, where teamwork is often essential for successful project completion.

In summary, the hands-on exercises in this course serve as a vital bridge between theory and practice, empowering learners to transform their understanding of ServiceNow scripting into actionable skills that they can leverage in their careers.

**Real-World Scripting Scenarios**

Scripting within ServiceNow is not just an academic exercise; it plays a crucial role in addressing real-world challenges that organizations encounter. One common scenario involves automating incident management processes. In many organizations, incidents are logged manually, leading to delays in resolution and increased workload for IT support teams. By implementing server-side scripts that automatically assign incidents based on specific criteria—such as workload balance or expertise—organizations can streamline their incident management processes, reduce response times, and enhance customer satisfaction.

Another prevalent challenge is ensuring data integrity across various tables. Organizations often struggle with maintaining accurate and up-to-date information, especially when multiple users can create or modify records. Here, scripting can be employed to enforce data validation rules. For instance, client-side scripts can prevent users from submitting forms with missing or incorrect information, while server-side scripts can be used to validate data before it is saved to the database. This dual-layered approach ensures that only high-quality data enters the system, thereby improving reporting accuracy and decision-making.

In addition, there are scenarios where integration with third-party applications is necessary. Businesses frequently need to pull data from external systems or push updates back to them. Scripting plays a key role in these integrations, often leveraging REST APIs. For example, a server-side script can be written to automatically fetch user data from an HR system during onboarding, ensuring that ServiceNow has the most current employee information. This reduces manual data entry and minimizes errors.

Moreover, organizations often face challenges with user experience. Customizing forms and interfaces can significantly enhance usability. This is where client-side scripting comes into play. For instance, scripts can be used to dynamically show or hide fields based on user input, making forms more intuitive. Additionally, scripts can be written to provide real-time feedback, such as validating email formats as users type, which improves user satisfaction and reduces submission errors.

These examples illustrate how effective scripting solutions can address common challenges in ServiceNow environments, leading to improved operational efficiency, enhanced data integrity, and a better user experience.

**Customizing ServiceNow with Scripts**

Customizing ServiceNow through scripting is an essential practice that enhances the platform's functionality and tailors it to meet specific business needs. By leveraging scripts, administrators and developers can automate tasks, improve user interfaces, and enforce business rules, ultimately contributing to a more efficient and user-friendly experience.

One of the primary areas for customization is form behavior. For instance, client-side scripts can be employed to dynamically hide or display fields based on user inputs. This not only simplifies the user interface but also guides users through the form completion process, reducing errors and enhancing the overall experience. For example, a script could be written to display additional fields only when a user selects a particular category, ensuring that users are only presented with relevant information to fill out.

Another significant customization opportunity lies in workflow automation. Server-side scripts can be used to create automated processes that enhance efficiency. For instance, an organization might implement a script that automatically assigns tasks to different teams based on predefined criteria, such as workload or expertise. This not only speeds up response times but also ensures that the right personnel handle the appropriate tasks, leading to more effective resolution of issues.

Moreover, scripts can be utilized to validate data integrity before it is stored in the database. A common example is implementing server-side validation scripts that check for duplicate records or validate data formats. Such scripts can prevent inaccurate data entry, ensuring that the information within ServiceNow remains consistent and reliable. This is particularly crucial for reporting and analytics, where data quality directly impacts decision-making.

Additionally, integrating ServiceNow with other applications often requires scripting. For example, a server-side script may be used to pull information from an external database or API, ensuring that ServiceNow reflects the most current data. This can streamline processes such as onboarding, where user data needs to be synchronized across multiple platforms.

In summary, customizing ServiceNow with scripts not only enhances the platform's capabilities but also significantly improves user experience by automating tasks, ensuring data integrity, and creating intuitive interfaces. As organizations continue to evolve, the ability to adapt ServiceNow through scripting becomes increasingly vital.

**Debugging ServiceNow Scripts**

Debugging is a crucial skill for any ServiceNow developer, as it allows for the identification and resolution of errors in scripts, ensuring smooth functionality within the platform. Effective debugging techniques can significantly enhance the performance and reliability of the scripts that power business processes.

ServiceNow provides several built-in tools that assist developers in troubleshooting and resolving script issues. One of the primary tools is the **Script Debugger**, which enables real-time debugging of server-side scripts. This tool allows developers to set breakpoints, step through code line by line, and inspect variable values at different execution points, making it easier to identify where issues arise. Utilizing the Script Debugger can save time and reduce frustration by providing immediate feedback on script execution.

Another essential debugging tool is the **System Logs**, which captures logs for various actions within ServiceNow. Developers can access the logs to view error messages, warnings, and informational messages generated by their scripts. By filtering logs based on severity or script name, developers can quickly locate relevant entries, allowing for efficient diagnosis of issues. The logs can also provide insights into the execution flow, helping to trace the path taken by the script and identify bottlenecks or errors.

In addition to these tools, leveraging **GlideSystem APIs** for logging during script execution can be extremely beneficial. By incorporating gs.info(), gs.warn(), or gs.error() statements in the code, developers can generate custom log messages that provide insights into script behavior. This practice not only aids in identifying issues but also enhances understanding of the script’s performance in a production environment.

Furthermore, implementing **unit tests** can serve as a proactive debugging strategy. By writing tests for individual script components before deployment, developers can catch errors early in the development process, allowing for corrections before the script impacts end-users. Unit testing frameworks available within ServiceNow facilitate this practice, ensuring that scripts perform as expected under various scenarios.

In summary, mastering debugging techniques in ServiceNow scripting is essential for developers. By effectively utilizing the Script Debugger, System Logs, GlideSystem APIs, and unit testing, developers can ensure high-quality scripts that enhance the overall functionality and reliability of the ServiceNow platform.

**Performance Optimization of Scripts**

Performance optimization of scripts in ServiceNow is a critical consideration for developers and administrators alike. As organizations increasingly rely on ServiceNow for their IT service management needs, ensuring that scripts run efficiently becomes paramount. Poorly optimized scripts can lead to slow response times, increased server load, and an overall degraded user experience, making it essential to adopt strategies that enhance script performance.

One of the primary reasons script performance is crucial is the impact it has on user satisfaction. Users expect applications to respond quickly and efficiently; delays caused by slow scripts can lead to frustration and decreased productivity. Furthermore, in environments with a high volume of transactions, such as incident management, the cumulative effect of slow scripts can significantly affect overall system performance. Thus, optimizing scripts not only improves individual user experiences but also enhances the platform's reliability and scalability.

To ensure scripts run efficiently, developers should consider several optimization strategies. First, minimizing the use of server-side scripts when client-side scripting suffices can help reduce server load. For instance, performing simple validations and UI manipulations on the client-side can lead to faster response times as these operations do not require a round trip to the server.

Additionally, efficient use of GlideRecord is vital for optimizing database interactions. Developers should be judicious in their use of queries, applying filters to limit the number of records returned and only retrieving fields necessary for their operations. Utilizing the get() method for single record retrieval can also improve performance over query(), particularly when only a specific record is needed.

Another effective strategy is to cache data where appropriate. For instance, frequently accessed data can be stored in memory rather than querying the database each time it is needed. This reduces the number of database calls and speeds up script execution.

Finally, regularly reviewing and refactoring scripts to eliminate redundant or unnecessary code can lead to significant performance improvements. Keeping scripts concise and focused not only enhances readability but also reduces the processing time required for execution.

By implementing these performance optimization strategies, ServiceNow developers can ensure that their scripts run efficiently, providing a seamless and responsive experience for users while maintaining system integrity and performance.

**Developing Advanced Scripts**

As learners progress beyond the foundational scripting concepts in ServiceNow, the opportunity to explore advanced scripting techniques becomes essential. These techniques not only enhance the functionality of the platform but also empower developers to create innovative solutions tailored to their organization's unique needs. This section delves into various advanced scripting concepts that can inspire creativity and drive impactful development.

**Utilizing Advanced Glide APIs**

While basic Glide APIs are foundational, advanced usage can unlock powerful functionalities. For instance, leveraging GlideAggregate allows developers to perform complex queries that summarize data, such as counting occurrences or calculating averages directly from the database. By mastering these advanced APIs, learners can streamline their scripts and enhance data processing efficiency, making scripts not just functional but also optimized for performance.

**Asynchronous Scripting**

In many cases, tasks within ServiceNow need not be executed synchronously. Advanced scripting techniques such as using asynchronous calls via GlideAjax or Scheduled Jobs can enhance user experience by allowing scripts to run in the background. This approach enables users to continue interacting with the application without waiting for lengthy processes to complete, thereby improving overall performance and satisfaction.

**Scripted REST APIs**

As organizations increasingly rely on integrations, the ability to create Scripted REST APIs becomes invaluable. This advanced technique allows developers to expose ServiceNow functionalities to external systems, enabling seamless data exchange. By mastering the creation of these APIs, learners can foster innovation through integrations, allowing ServiceNow to interact with various applications and services efficiently.

**Client-Side JavaScript Libraries**

For those working with client-side scripts, exploring JavaScript libraries such as jQuery can significantly enhance the user interface. By incorporating these libraries, developers can implement advanced UI features like animations, dynamic content loading, and enhanced form validations. This not only improves the user experience but also sets the stage for more engaging and interactive applications.

**Scripting for Performance and Scalability**

Advanced scripting techniques also include strategies for ensuring that scripts are not only effective but also scalable. Techniques such as modular design, where scripts are broken into reusable components, allow for easier maintenance and updates. Additionally, leveraging caching strategies and optimizing database interactions can lead to improved performance, particularly in high-demand environments.

By encouraging learners to explore these advanced scripting techniques, we foster a culture of innovation and creativity in script development. As they implement these concepts, they will not only enhance their technical skills but also contribute to the overall success of their ServiceNow implementations.

**Review of Key Topics Covered**

Throughout the ServiceNow Scripting Fundamentals Course, various key topics have been explored, providing learners with a comprehensive understanding of scripting within the ServiceNow platform. The foundation of this module rests on the vital distinction between client-side and server-side scripting. Client-side scripts operate within the user's browser, managing user interface behaviors, while server-side scripts run on the server, handling data processing and business logic. Understanding these differences is crucial for effective scripting, allowing learners to select the appropriate scripting approach based on specific scenarios.

Another significant topic covered is the importance of Glide APIs. These built-in functions serve as essential tools for interacting with the ServiceNow platform, enabling developers to perform operations such as querying records and updating data efficiently. Mastery of these APIs enhances script effectiveness and allows for sophisticated interactions within the ServiceNow environment.

Best practices in scripting were also a focal point of the module. Emphasizing code organization, performance optimization, and error handling ensures that scripts are not only functional but also maintainable and scalable. By adhering to these best practices, learners are better equipped to produce high-quality scripts that stand the test of time.

The hands-on exercises integrated within the module provided practical applications of the theoretical concepts discussed. Through real-world scenarios, learners had the opportunity to write, test, and debug scripts, reinforcing their understanding and enhancing their problem-solving skills. This experiential learning approach is vital for developing confidence in scripting capabilities.

Finally, the exploration of advanced scripting techniques revealed pathways for further growth. Topics such as asynchronous scripting, scripted REST APIs, and the use of client-side JavaScript libraries offer learners the tools needed to create innovative and efficient solutions tailored to their organization's needs.

By synthesizing these key topics, participants are well-prepared to navigate the complexities of ServiceNow scripting, equipping them to contribute effectively within their respective roles.

**Future Learning Paths**

As learners complete Module 4 of the ServiceNow Scripting Fundamentals Course, several future learning paths are available to deepen their knowledge and skills in ServiceNow scripting and its applications. These paths can help enhance their expertise, making them valuable assets in their organizations.

**Advanced ServiceNow Scripting**

One option is to pursue advanced scripting courses that delve deeper into the intricacies of ServiceNow. These courses can cover advanced Glide APIs, asynchronous scripting techniques, and the development of scripted REST APIs. By mastering these advanced topics, learners can create more complex and efficient solutions tailored to specific business needs.

**Integration and Automation**

Learners may also consider exploring integration and automation within ServiceNow. This path can involve courses focused on integrating ServiceNow with external systems using APIs, as well as automating workflows to enhance efficiency. Understanding integration techniques will empower learners to connect ServiceNow with various applications, creating seamless data exchanges that drive operational excellence.

**ServiceNow Certified Developer Program**

For those looking to formalize their skills, pursuing the ServiceNow Certified Developer program can be an excellent choice. This certification validates a developer’s ability to build applications on the ServiceNow platform, including scripting, UI design, and integration. This credential can significantly enhance a learner's career prospects and demonstrate their proficiency to potential employers.

**Community and User Groups**

Engaging with the ServiceNow community and participating in user groups can also provide valuable learning opportunities. These platforms allow learners to share knowledge, ask questions, and collaborate with peers and experts in the field. Networking with other ServiceNow professionals can uncover best practices and innovative approaches to common scripting challenges.

**Resource Recommendations**

Additionally, various online resources can support continued learning. Recommended resources include:

* **ServiceNow Documentation**: Comprehensive guides and references for all aspects of ServiceNow, including scripting.
* **Online Forums and Communities**: Platforms such as ServiceNow Community where learners can ask questions and share insights.
* **YouTube Tutorials**: Video content that provides visual demonstrations of scripting techniques.
* **Books and E-Books**: Titles focused on ServiceNow development and scripting best practices.

By following these suggested learning paths and utilizing additional resources, learners can further their understanding and application of ServiceNow scripting, positioning themselves for success in their careers.

**Course Conclusion**

As we conclude Module 4 of the ServiceNow Scripting Fundamentals Course, it is essential to reflect on the learning journey undertaken throughout this module. This module has equipped participants with a solid foundation in scripting within the ServiceNow platform, emphasizing both theoretical understanding and practical application. By exploring the intricacies of client-side and server-side scripting, learners have gained insights into how to effectively utilize these tools to enhance platform functionality.

The hands-on exercises provided throughout the module have been instrumental in solidifying knowledge and fostering confidence in scripting skills. Participants have had the opportunity to engage in real-world scenarios, allowing them to apply learned concepts and troubleshoot common issues. This practical experience is vital as it lays the groundwork for tackling more complex challenges in the future.

Moreover, the emphasis on best practices in scripting has prepared learners to write high-quality, maintainable code. By understanding how to organize scripts, optimize performance, and handle errors effectively, participants are well-equipped to create robust solutions that align with organizational needs.

Looking ahead, we encourage participants to seek feedback on their experiences within this course. Reflecting on what worked well and what could be improved is crucial for personal growth and the evolution of future training. The ServiceNow platform is continuously evolving, and so are its capabilities. Therefore, ongoing exploration of new features and advanced scripting techniques will be essential for staying current in the field.

As you continue your journey with ServiceNow, remember that the skills and knowledge gained in this module are just the beginning. With a solid foundation in scripting, you are now poised to delve deeper into the platform’s capabilities, unlocking new possibilities for innovation and efficiency within your organization.