Digital Nurture 3.0
ServiceNow

Week 2

Module 2 Report

Topic : ServiceNow Administration Fundamentals

Name: P Vainavi

Superset ID: 5097830

College: Dayananda Sagar Academy of Technology and Management

USN: 1DT21IS100

Date:06/09/2024

Platform Overview and Architecture

Platform Overview:

ServiceNow is not just an ITSM tool but a comprehensive cloud-based platform designed for digitizing and automating business processes across various functions. The platform integrates IT services, customer support, and operational management into a single cohesive system.

- Core Features:

— Single Data Model: Centralized data repository for all applications, ensuring consistency and integration.

— Process Automation: Automates repetitive tasks and workflows to enhance efficiency.

— Service-Oriented Architecture (SOA): Provides modular and scalable services that can be tailored to meet specific organizational needs.

Architecture:

- Instance Structure:

— Production Instances: Live environments where actual business processes run. Typically, changes are carefully planned and tested before deployment.

— Development and Test Instances: Environments for testing new features, customizations, and updates. They mirror the production environment but are used for validation.

— Upgrade Instances: Temporary instances created to validate new versions or patches before applying them to production.

- Database Schema:

— Tables: ServiceNow organizes data in tables. Each table consists of rows (records) and columns (fields). Examples include the `incident`, `change_request`, and `task` tables.

— Field Types: Fields in tables include standard types like string, integer, and date, as well as advanced types like reference fields (linking to other records) and choice fields (dropdown lists).

- Application Scope:

— Scoped Applications: These are custom applications developed within a specific scope to prevent interference with base system functionality. They contain their own tables, scripts, and UI elements.

— Namespace: Each application has a namespace that helps in identifying and isolating it from other applications.

User Interface and Branding

User Interface:

- Navigation:

— Application Navigator: The primary tool for accessing different modules and applications. Users can search for specific items using the filter navigator or browse through the module tree.

— Breadcrumbs: Help users track their location within the platform and navigate back to previous pages or higher-level modules.

- UI Elements:

— Forms: Used for data entry and record management. Forms can be customized with different field types, layouts, and sections.

— Lists: Display collections of records from tables. Users can configure columns, sort data, and apply filters to view specific records.

— Dashboards: Provide a visual representation of data through widgets, charts, and graphs. Dashboards can be tailored to display relevant metrics and KPIs.

Branding:

- Themes:

— Custom Themes: Modify the visual appearance of the Service Portal by changing colors, fonts, and layout elements. Themes ensure that the portal aligns with organizational branding.

— CSS Customizations: Advanced customization options allow for detailed styling changes using custom CSS.

- Logos and Branding:

— Custom Logos: Upload logos for display in the Service Portal and other interfaces to reinforce corporate identity.

— Branding Packages: ServiceNow offers tools to apply consistent branding across different modules and portals.

- UI Policies:

— Dynamic Behavior: Use UI policies to dynamically change the appearance or behavior of form fields based on specific conditions, enhancing user experience and ensuring data accuracy.

Lists & Filters and Forms

Lists:

- Creating Lists:

— List Layouts: Define which columns to display, their order, and their formatting. Users can customize lists to focus on relevant data.

— List Filters: Apply conditions to filter records based on criteria such as status, priority, or category. Filters can be saved as personal or global filters.

- Filtering:

— Basic Filters: Simple conditions to narrow down records, such as showing all incidents assigned to a specific user.

— Advanced Filters: Complex conditions using multiple criteria and logical operators (AND, OR). Save and share these filters for repeated use.

- Saving Filters:

— Personal Filters: Save filters for individual use, allowing users to quickly access commonly viewed subsets of data.

— Global Filters: Share filters with other users or groups, facilitating collaborative data analysis and reporting.

Forms:

- Form Layout:

— Field Arrangement: Customize the arrangement of fields on forms to improve usability. Group related fields into sections for a logical flow.

— Field Attributes: Configure field properties such as default values, help text, and read-only status.

- UI Policies:

— Field Visibility: Show or hide fields based on user input or record conditions.

— Field Mandatory Status: Make fields mandatory or optional depending on specific criteria.

- Form Sections:

— Collapsible Sections: Organize form fields into collapsible sections to manage large forms and improve user experience.

Task Management

- Tasks:

— Types: Includes incidents, problems, changes, and service requests. Each type has specific processes and workflows.

— Task Lifecycle: Track the lifecycle of tasks from creation to resolution, including stages such as assignment, investigation, and closure.

- Workflows:

— Creating Workflows: Use the Flow Designer or legacy Workflow Editor to design automated processes for task management. Define actions, approvals, and conditions.

— Workflow Stages: Include stages such as initiation, review, approval, and completion. Each stage can trigger specific actions or notifications.

- Assignment Rules:

— Automated Assignment: Configure rules to automatically assign tasks based on criteria like task type, priority, or location.

— Rule Conditions: Define conditions that determine when and how tasks are assigned to users or groups.

Notifications

- Notification Types:

— Email Notifications: Send email alerts to users based on specific triggers, such as task updates or status changes.

— SMS Notifications: Notify users via SMS for critical alerts or updates.

— In-Platform Notifications: Display notifications within the ServiceNow interface, such as pop-ups or banner alerts.

- Creating Notifications:

— Templates: Design notification templates with dynamic content. Include placeholders for variables like user names, task details, and links.

— Notification Conditions: Define triggers and conditions for sending notifications. For example, send a notification when an incident status changes to “Resolved.”

- Notification Rules:

— Recipient Selection: Specify recipients based on roles, user groups, or individual users.

— Content Customization: Customize notification content to include relevant information and instructions.

Knowledge Management

- Knowledge Base:

— Creating Knowledge Bases: Set up knowledge bases to store and organize knowledge articles. Create multiple bases for different topics or departments.

— Article Types: Include FAQs, how-to guides, troubleshooting tips, and procedural documentation.

- Article Lifecycle:

— Stages: Articles go through stages such as draft, review, publish, and retire. Define workflows for article creation and approval.

— Review Process: Implement review processes to ensure articles are accurate and up-to-date.

- Categories and Subcategories:

— Organizing Content: Use categories and subcategories to organize knowledge articles, making it easier for users to find relevant information.

— Category Management: Create and manage categories to reflect organizational structure and information hierarchy.

Service Catalog

- Catalog Items:

— Creating Items: Define catalog items such as request forms, services, or products. Configure item details, pricing, and fulfillment processes.

— Request Forms: Customize request forms to capture relevant information from users.

- Order Guides:

— Designing Guides: Create order guides to streamline the process of requesting multiple items or services. Include steps, dependencies, and conditional items.

— Guided Workflows: Use guided workflows to lead users through the order process, ensuring all required information is collected.

- Request Fulfillment:

— Automated Fulfillment: Set up automated workflows to handle the fulfillment of catalog requests. Include steps for approvals, provisioning, and notifications.

— Service Delivery: Monitor and manage the delivery of services to ensure timely and accurate fulfillment.

Tables and Fields

Tables:

- Table Structure:

— Standard Tables: Includes tables like `incident`, `change_request`, and `task`. Each table has predefined fields and relationships.

— Custom Tables: Create custom tables to store additional data. Define fields, relationships, and access controls.

- Field Types:

— Basic Fields: String, integer, date/time, and boolean.

— Advanced Fields: Reference fields (link to other records), choice fields (dropdown lists), and currency fields.

Fields:

- Field Properties:

— Default Values: Set default values for fields to simplify data entry.

— Help Text: Provide help text to guide users on how to fill out fields.

— Validation Rules: Implement validation rules to ensure data accuracy and consistency.

Access Control List

- Purpose: Control who can view or modify records and fields. ACLs are essential for maintaining data security and privacy.
- Configuration: Set up ACLs for table-level access (entire table) and field-level access (specific fields within a table).

- Types of ACLs:

— Table-Level ACLs: Restrict access to entire tables based on roles or conditions.

— Field-Level ACLs: Control access to specific fields within a table.

- Testing ACLs:

— Verification: Test ACLs by simulating different user roles and access scenarios. Ensure that permissions are enforced as intended.

Data Import

- Import Sets:

— Creating Import Sets: Use import sets to bring data into ServiceNow from external sources such as CSV files or databases.

— Mapping Data: Map source data fields to ServiceNow table fields to ensure accurate data import.

- Data Transformation:

— Transformation Rules: Define rules to convert and format data during the import process. This includes filtering, merging, and adjusting data.

- Scheduled Imports:

— Automation: Set up scheduled imports to automate the process of updating ServiceNow with data from external systems at regular intervals.

CMDB (Configuration Management Database)

- Configuration Items (CIs):

— Types of CIs: Includes hardware, software, network devices, and services. Each CI has attributes and relationships with other CIs.

— CI Records: Maintain detailed records of each CI, including attributes such as location, owner, and status.

- CI Relationships:

— Visualizing Relationships: Use relationship diagrams to understand how different CIs are connected and dependent on each other.

— Impact Analysis: Analyze the impact of changes to CIs on other related CIs and business services.

- Discovery:

— Automated Discovery: Use ServiceNow Discovery to automatically detect and update CIs in the CMDB. Discovery tools scan network devices, servers, and applications.

— Data Accuracy: Ensure the CMDB remains accurate and up-to-date through regular discovery scans and reconciliation processes.

Integration

- IntegrationHub:

— Integration Capabilities: Use IntegrationHub to connect ServiceNow with external systems and applications. IntegrationHub provides pre-built connectors and integration tools.

— Flow Designer: Design and manage integration workflows using the Flow Designer interface.

- REST and SOAP APIs:

— REST APIs: Use RESTful APIs for lightweight, stateless interactions with ServiceNow. Ideal for integrating with web and mobile applications.

— SOAP APIs: Use SOAP APIs for more complex, stateful interactions. Useful for integrating with legacy systems.

- Data Exchange:

— Data Synchronization: Facilitate data synchronization between ServiceNow and other systems. This can include real-time data exchange or batch updates.

Update Sets

- Creating Update Sets:

— Purpose: Track and move customizations between ServiceNow instances. Capture changes such as new fields, forms, and business rules.

— Configuration: Define and manage update sets to ensure that all necessary customizations are included.

- Managing Update Sets:

— Review and Testing: Review update sets to verify that changes are accurate and complete. Test updates in non-production instances before applying to production.

— Committing Changes: Apply update sets to target instances. Ensure that changes do not conflict with existing configurations.

- Merging Update Sets:

— Combining Sets: Merge multiple update sets to consolidate changes from different sources. Resolve conflicts and ensure consistency.

Events

- Event Management:

— Event Collection: Collect events from various sources, including system logs, external monitoring tools, and user interactions.

— Event Analysis: Analyze events to identify patterns, trends, and potential issues.

- Event Rules:

— Rule Configuration: Define rules to trigger specific actions based on events. For example, create an incident automatically when a critical error is detected.

— Action Automation: Automate responses to events, such as sending notifications or initiating workflows.

- Alerting:

— Alert Configuration: Set up alerts to notify users or system administrators of critical events. Customize alert content and delivery methods.

Platform Stats

- Performance Metrics:

— Monitoring: Track key performance metrics such as response times, transaction throughput, and system load.

— Optimization: Use performance data to identify bottlenecks and optimize system performance.

- Usage Analytics:

— User Activity: Analyze user activity to understand how the platform is being used. Identify popular features, frequent issues, and areas for improvement.

— Reporting: Generate reports on system usage and performance to support decision-making and continuous improvement.

- Logging and Monitoring:

— Log Management: Collect and analyze logs to track system events, errors, and user actions. Use logs to troubleshoot issues and ensure system stability.

— System Monitoring: Implement monitoring tools to track system health and performance. Set up alerts to proactively address potential problems.