**ServiceNow: -**

**Headquarters: Based in Santa Clara, California.**

**Leadership:**

* Bill McDermott: Current CEO, formerly the CEO of SAP.
* Fred Luddy: Founder and chairman of the board, a programmer who created ServiceNow after the bankruptcy of Peregrine Systems**.**

**Why ServiceNow?**

* Problem Identification: IT departments often focus more on technology than solving business problems, leading to inefficiencies.
* Purpose: ServiceNow aims to create a platform where businesspeople can solve their problems independently, without relying on traditional IT departments.

**How ServiceNow Works**

* **NOW Platform**: A cloud-based Application Platform as a Service (APaaS) providing the necessary infrastructure, platform, applications, and workflows to support business IT needs.
* **Key Features**:
  + Secure and redundant infrastructure with global data centers.
  + A unified enterprise-wide data model and database.
  + Pre-built applications and workflows categorized into IT Workflows, Employee Workflows, Customer Workflows, and Creator Workflows.
  + Custom workflows and applications can be built within the platform.

**ServiceNow Definition**

* **Single Sentence Definition**: "ServiceNow is a software company, based in Santa Clara, California, founded by Fred Luddy in 2003, to solve problems large enterprises face with traditional IT delivery by providing a robust, simple-to-use cloud-based environment in which businesspeople can solve the business problems themselves."

1. **Now Platform Architecture:**
   * ServiceNow is described as an Application Platform as a Service (APaaS).
   * It combines infrastructure, platform, and software services, offering a cloud-based environment with a unified data model.
2. **Applications and Workflows:**
   * Divided into four categories: IT Workflows, Employee Workflows, Customer Workflows, and Creator Workflows.
   * Each category supports different business functions with a robust set of applications.

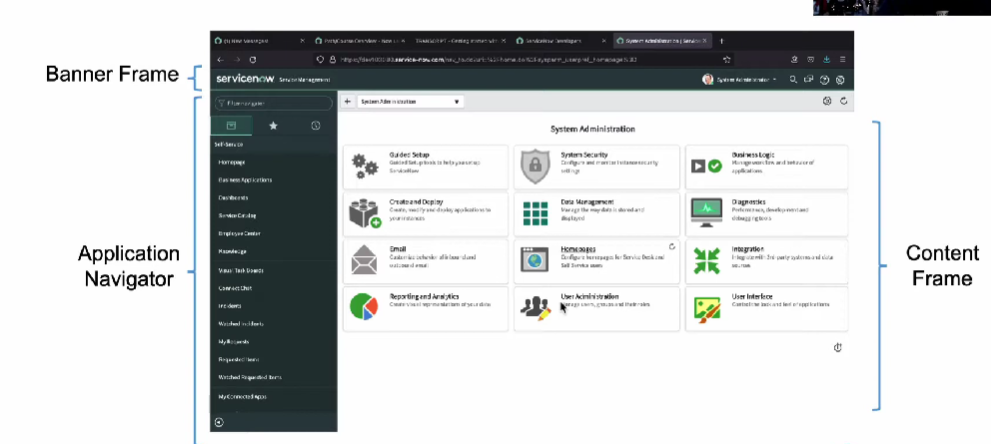


1. **User Interface Types:**
   * **Now Platform UI:** Primary interface used on desktops/laptops.
   * **Mobile Apps:** Includes ServiceNow Agent App, Now Mobile App, and Onboarding App.
   * **Service Portal:** A customizable web-based interface using widgets for specific user groups.
2. **Role-Based Access and Authentication:**
   * **Three Main Entities:** Users, Groups, and Roles.
   * Roles define the permissions assigned to users, determining what actions they can perform within the system.

**Additional Insights:**

* **Fred Luddy, Founder of ServiceNow:** Motivated by the frustration of seeing IT staff make business people look foolish. He envisioned a platform allowing business people to solve their own problems.
* **Multi-Instance Architecture:** Provides each customer with their own instance, ensuring data is not intermingled with others, offering more control over upgrades and patches.
* **Backup and Security:** ServiceNow provides regular backups and has robust security measures certified by third-party organizations.
* **Domain Separation:** Allows different groups within a company to have segregated data access based on roles and permissions.

**Main Screen Elements:**



1. **Banner Frame:**
   * **Location:** Runs across the top of the interface.
   * **Components:**
     + **Logo:** Customizable; clicking it returns you to the homepage.
     + **User Menu:** Dropdown menu with options like Profile, Impersonate User, Elevate Roles, and Logout.
       1. **Profile:** Update profile attributes such as name, email, and time zone.
       2. **Impersonate User:** Allows viewing the instance as another user (admin only).
       3. **Elevate Roles:** Adds an extra step to perform high-impact actions (admin only).
       4. **Logout:** Exit the instance.
     + **Tools Section:**
       - **Global Search:** Search the entire instance for records.
       - **Connect Chat:** Integrated chat tool for communication within the platform.
       - **Help:** Contextual help related to your current context within the application.
     + **System Settings (Gear Icon):** Customizes the UI for personal preferences, including General, Theme, Accessibility, List, Form, Notification, and Developer Settings.
       - 1. **General:** Compact UI, keyboard shortcuts, date/time display, etc.
         2. **Theme:** Customizes the color and look of the UI.
         3. **Accessibility:** Adjusts settings for accessibility needs.
         4. **List/Form Settings:** Controls the appearance and performance of lists and forms.
         5. **Notification:** Manages notification preferences.
         6. **Developer Settings:** Application scope, update set selection, and developer tools.
2. **Application Navigator:**
   * **Location:** Sidebar on the left.
   * **Components:**
     + **Navigation Filter:** Filters the long list of applications and modules by keywords.
     + **Tabs:**
       - **All Applications:** Displays all available applications.
       - **Favorites:** Lists frequently used applications or records.
       - **History:** Shows the last 30 items you've accessed.
     + **Hierarchy:**
       - **Applications:** Top-level elements in the navigator.
       - **Modules:** Functions within each application.
       - **Separators:** Grouping mechanism for modules within an application.
3. **Content Frame:**
   * **Location:** The main area of the screen.
   * **Purpose:** Displays the content of the applications and modules you're working on.

**ServiceNow Branding Overview**

**1.** **Branding in ServiceNow:**

* Branding involves making the user interface appear and feel consistent with your company’s identity by using company-specific colors, fonts, logos, etc.
* Benefits include creating a shared identity, enhancing user comfort and confidence, and speeding up tool adoption.

2. **Guided Setup Wizards:**

* ServiceNow offers guided setup wizards that assist with branding and other setup tasks.
* Two primary types: IT Service Management (ITSM) Guided Setup and IT Operations Management (ITOM) Guided Setup.
* Focus on the **Company Guided Setup** wizard for branding, which includes tasks like system configuration and welcome page customization.

3. **Key Configuration Steps:**

* **System Configuration:** Adjust settings like time zones, date and time formats, upload logos, and customize banner frames and browser tab text.
* **Welcome Page:** Customize the login screen with messages, alerts, or welcome notes for users.

4. **Additional Customization Options:**

* ServiceNow Portal: A widget-based tool for building user interfaces by arranging widgets on the screen.
* UI Builder: A WYSIWYG editor for creating functional screens with buttons, headings, and other elements.

**ServiceNow Branding Introduction, Company Guided Setup, ServiceNow Portal, and UI Builder.**

1. **Introduction to Lists**: Lists are a central part of the ServiceNow platform, displaying records from database tables in a user-friendly format. The list view is a powerful tool, providing functionalities like sorting, searching, filtering, and analyzing data.
2. **Accessing Lists**: Lists can be accessed via the Application Navigator or by using the .list command. For example, typing incident.list will bring up the list view for the Incident table. If you don’t know the table name, sys\_db\_object.list can be used to view a list of all tables.
3. **List View Components**:
4. **Title Bar**: Displays the list's context, including the table name, saved views, and offers tools like the search function and a 'New' button to create new records.
5. **List Header**: Contains column names and provides tools to customize the list view, such as personalized list settings, condition builder, and multi-column search.
6. **Data Rows and Columns**: Each row represents a record, and each column represents a field from the table. Sorting can be done by clicking on column headers.

4. **Title Bar Tools**:

* **List Control Menu (Hamburger Icon)**: Offers options to manage the entire list, including changing views, setting filters, grouping data by columns, setting page size, refreshing the list, and adding favorites for quick access.
* **Search Tool**: Allows you to search within the table based on specific columns.
* **Activity Stream Icon**: Displays a sequential account of all activity related to the list, helpful for tracking changes and updates.

5. **List Header Tools**:

* **Personalized List Tool (Gear Icon)**: Customize the list by adding, removing, and reordering columns. These customizations are user-specific and do not affect other users.
* **Filter (Condition Builder)**: A robust tool for filtering list data, allowing the use of multiple conditions, sorting, and filtering by columns not currently displayed.
* **Multi-Column Search**: Toggle on the search row to filter by specific column values. Wildcards can be used to refine searches.

6. **Breadcrumbs**: Display applied filters in the list view, showing the filtering history and allowing for easy filter management. Clicking on breadcrumbs removes or adjusts filters, providing a clear view of how data is being filtered.

7. **Column Context Menu**: Provides options for viewing data in visual formats like bar charts and pie charts, importing/exporting data, and performing bulk updates on selected records.

8.**Field Context Menu**: Right-click on any field to access specific actions for that field, such as filtering by value, copying the record’s URL or sys\_id, and tagging the record for better organization.

9. **Additional Features**:

* **Checkboxes**: Allow selecting multiple records for batch actions.
* **Information Icon**: Provides a quick preview of a record without leaving the list view.
* **Linked Columns**: Clicking on a linked value, typically in the first column, opens the detailed form view for that record.

**ServiceNow Lists and Filters**

**Accessing Lists:**

* Lists can be accessed through the **Application Navigator** or by using the **dot list command** (e.g., incident.list, task.list).
* If you don't know the table name, use sys\_db\_object.list to view all tables.

**List Interface Components:**

1. **Title Bar:**
   * **List Control Menu**: Access options for view, filter, grouping, and more.
   * **Search Tool**: Search within specific fields using wildcard characters for flexible queries.
   * **Paging Controls**: Navigate through records using arrows.
2. **List Header:**
   * **Personalize List Tool**: Customize visible columns without affecting other users.
   * **Condition Builder**: Apply complex filters using AND/OR operators and multiple columns.
   * **Column Search Row**: Apply filters directly to visible columns.
   * **Breadcrumbs**: View and manage applied filters.
3. **Data Rows & Columns:**
   * Each row represents a record, and each column represents a field in the database.
   * **Column Context Menu**: Access specific options like sorting and grouping by column.
   * **Field Context Menu**: Right-click a field for options like filtering, copying URLs, and assigning tags.

**Additional Features:**

* **Activity Stream Icon**: Displays a sidebar with a chronological account of activities related to the list.
* **Quick Preview**: Use the information icon to preview a record without leaving the list.
* **Multiple Record Actions**: Use checkboxes and the "Actions on selected rows" dropdown to perform bulk actions.

**Key Commands:**

* tableName.list - Opens the list view for a specific table.
* tableName.form - Opens the form view for a specific record.
* tableName.config - Opens the configuration view for a table.

**Forms in ServiceNow**

**What is a Form?**

* A form in ServiceNow allows you to view, edit, or create records that are stored in the ServiceNow database. For example, viewing an incident record or creating a new user record.
* Forms can be accessed by selecting a record from a list or by entering the record ID directly in the global search.
* Every form corresponds to a specific record type in ServiceNow. There are thousands of different forms, each linked to a different table in the database.

**Standard Layout of Forms**

* Forms in ServiceNow follow a standardized layout:
  + **Header Bar**: Contains tools like submit, update, and save options.
  + **Main Section**: Displays fields with labels, including required (marked with an asterisk) and read-only (grey background) fields.
  + **Additional Sections**: Group related fields or display related lists and formatters.

**Form Field Types**

* Fields in the main section of the form vary depending on their data type:
  + **String Fields**: Simple input elements.
  + **Boolean Fields**: Displayed as checkboxes.
  + **Choice Fields**: Displayed as drop-down lists.
  + **Reference Fields**: Linked to records in another table, displayed with tools for validation and preview.

**Saving Changes**

* Changes are not automatically saved in ServiceNow; you must manually save them using:
  + **Submit or Update**: Saves changes and closes the form.
  + **Save**: Saves changes but keeps the form open.
  + **Insert and Insert & Stay**: Create copies of existing records to form new records, with "Insert & Stay" keeping the form open.

**Form Sections**

* Forms are built using sections to organize fields and related lists, which can be displayed as tabs or collapsible containers. Users can customize their view through preferences.

**Related Lists & Formatters**

* **Related Lists**: Show records from another table that are related to the current record.
* **Formatters**: Special elements displaying information like activity history or other non-field-related data.

**Form Views**

* ServiceNow allows different views of the same record depending on the user’s role or preferences. Users can switch between views and even create customized views to display records differently.

**Form Personalization**

* Users can personalize forms by toggling fields on or off through the form personalization tool. These changes only affect the current user’s view.

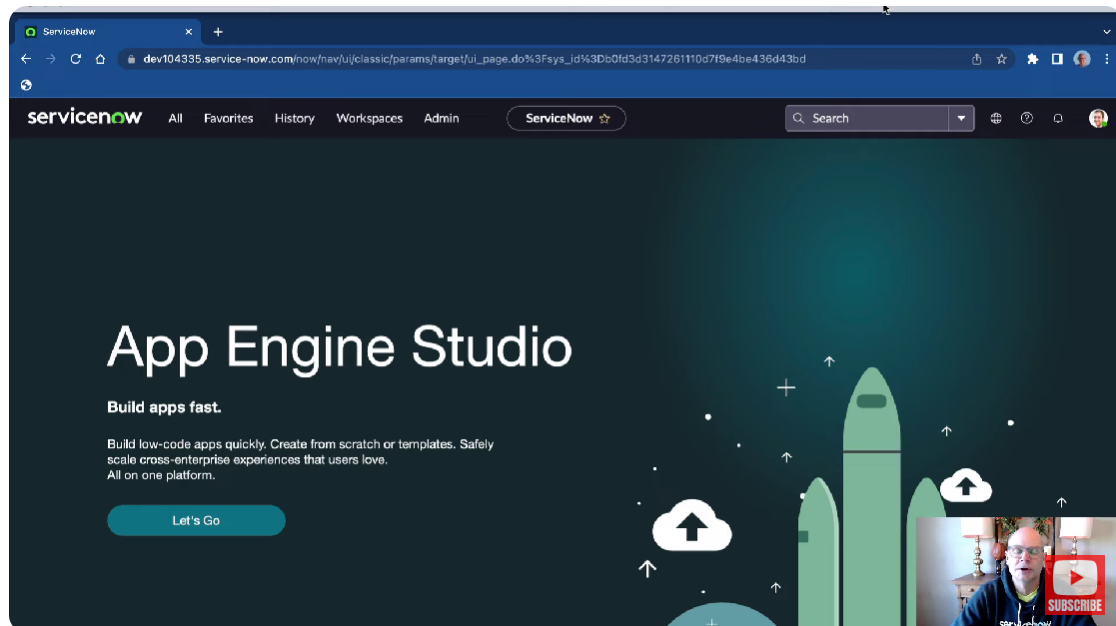
**Attachments and Templates**

* **Attachments**: Files can be attached to records, such as screenshots of error messages on incidents.
* **Templates**: Allow automatic population of fields for frequently used records. Templates can be created and managed via the template bar.

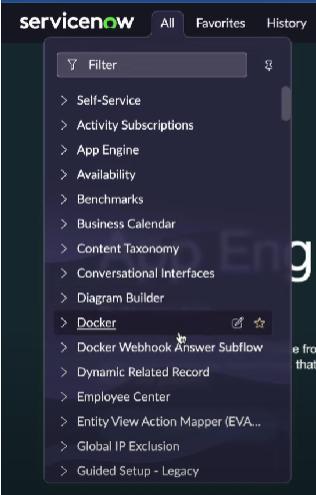
**What is ServiceNow? (A Hands-on ServiceNow Tool Demo)**

After login to the instance there is Next Experience UI

**Navigating the Next Experience UI**



1. **The Navigation Bar:**
   * **User Menu:** View your logged-in account and set up user-specific configurations or preferences to tailor the UI.
   * **Show Notifications:** View and set up notifications within the platform, allowing you to stay informed about events and tasks.
   * **Contextual Help:** Access knowledge-base articles or help tools relevant to the platform section you are currently in.
   * **Application Scope Picker:** Mainly for administrators or developers, it limits access and controls deployment of applications.
   * **Global Search:** A powerful search tool that allows you to search for anything within the platform, like a user's name or incident number.
   * **Contextual App Pill:** Displays a textual representation of your current location within the platform.
   * **Favorites:** Use the star icon to mark screens or modules as favorites for quick access later.
   * **Admin Menu:** Contains specific tools for administrators, visible only to those with admin roles.
   * **Workspaces:** Single screens with multiple widgets designed for specific types of work, such as managing the CMDB.
   * **History:** Keeps track of the last 30 places you visited within the platform, allowing you to quickly return to previous screens.
   * **All Menu:** Contains all the applications available in ServiceNow, both out-of-box and custom-developed. This is where you’ll spend most of your time exploring applications.
2. **Application Navigator**

****

**Key Application Categories:**

* + Self-Service: Allows users to access various self-service modules.
  + App Engine: Provides tools for creating and customizing applications.
  + Employee Center: Offers tools and applications designed for employee tasks.
  + Integration Hub: Facilitates data import from legacy systems and integrates with other tools.
  + Predictive Intelligence: Includes modules for intelligent automation.
  + Process Automation: Supports automation of business processes.
  + Service Desk Suite: Known for managing incidents, problems, and changes.
  + CMDB (Configuration Management Database): Manages configuration data across the platform.
  + Service Catalog: Provides a list of services that users can request.
  + Knowledge Management: Allows the creation of knowledge articles to guide users.
  + Asset & Inventory Management: Manages assets and inventory within the platform.
  + Contract Management: Han dles contract-related activities.
  + Reporting: Offers various reporting tools and dashboards.
  + Artificial Intelligence: Includes AI modules for automating tasks.
  + Automated Test Framework: Supports testing and automation within the platform.
  + User Administration: Manages user accounts and permissions.
  + Workflow: Enables the creation and management of workflows.

**ServiceNow Store:** If a needed application is not available out of the box, users can access the ServiceNow Store.

**Lists Overview**

* **Definition**: A list in ServiceNow displays multiple records from a database table.
* **List Control Menu**: Located at the top of the list view, it allows you to:
  + **View**: Create multiple views tailored to different needs (e.g., mobile, specific groups).
  + **Group By**: Group list items by any field within the table.
  + **Filter**: Apply sophisticated filters using the Condition Builder, which combines conditions with "AND" or "OR".
  + **Personalization**: Customize the list view for the logged-in user, choosing which fields to display and their order.
  + **Actions on Selected Rows**: Apply bulk actions like archive, delete, etc., to selected rows.

**Filtering and Searching**

* **Condition Builder**: Allows for complex filtering of records by selecting fields and applying conditions.
* **Breadcrumbs**: Show the filters applied to the list and allow easy manipulation or removal of filters.
* **Column Context Menu**: Offers options like sorting, grouping, and visualizing data (e.g., pie charts).

**Forms Overview**

* **Definition**: A form in ServiceNow displays a single record from a table and allows you to view and modify its details.
* **Form Context Menu**: Provides actions specific to the record, such as saving, inserting new records, and exporting.
* **Views**: Forms can have multiple views tailored to different user needs, displaying different attributes and related lists.
* **Personalization**: Similar to lists, forms can be personalized for individual users.

**Knowledge Management in ServiceNow**

* **Knowledge Bases**: Libraries of important articles or documentation divided into categories for easy access.
  + **Articles**: Can be searched across the entire knowledge base.
  + **Categories**: Articles are organized within specific categories for easier navigation.

**Introduction to Importing Data in ServiceNow**

**components of a ServiceNow import process: -**

* Data Sources
* Import Sets
* Transform Maps
* Field Maps
* Data Import Scheduling

1. **Source Data Entity**: The original data set you want to import into ServiceNow.
2. **Target Data Entity**: The destination within ServiceNow where you want the data to be stored after the import.
3. **Staging Table (Import Set Table)**:

* An intermediary data entity automatically created by ServiceNow during the import process.
* Also referred to as an "Import Set Table" in ServiceNow, but in these notes, it’s referred to as a "Staging Table" for simplicity.

**Three Main Data Entities in the Import Process**

1. **Source Data**: The data you want to import.
2. **Staging Table (Import Set Table)**:
   * Automatically created by ServiceNow when the import begins.
   * Serves as a temporary holding area for the data before it moves to the target entity.
3. **Target Data Store**: The final location within ServiceNow where the imported data will be stored.

**Creating a Data Source in ServiceNow**

1. **Navigate to the Data Source Table:**
   * You can use the filter navigator to enter the table name (sys\_data\_source.list) to see existing data sources.
   * Alternatively, you can go through **System Import Sets > Administration > Data Sources**.
2. **Create a New Data Source:**
   * Click **New** to add a record to the data source table.
   * Name your data source (e.g., "Test Import").
   * Set the label for the staging table (e.g., "Test Import"), and ServiceNow will generate a table name (e.g., u\_test\_import).
   * Select the **type** of data source (e.g., file, JDBC, LDAP, etc.).
   * If using a file, specify the format (e.g., Excel, CSV) and attach the file to the data source.
3. **Advanced Setup:**
   * If using JDBC, provide additional connection details such as database type, server name, port, and SQL query if needed.
   * For file sources, attach the file to the data source record.

**Understanding Import Sets in ServiceNow**

**Testing the Data Source:**

* Now, we are ready to test the data source, checking that the connection works, the staging table is created, and data is loaded into it.
* We reviewed the data source record stored in the sys\_data\_source table. The staging table was labeled as "Test Import" with a database table name U\_test\_import.

**Running the Import:**

* When running the import, ServiceNow first checks if the staging table exists. If not, it creates the table based on the parameters set in the data source.
* We loaded the data from an Excel file containing five rows into the staging table.
* After the import, the table U\_test\_import was created, and the five rows were successfully inserted.

**Viewing the Staging Table:**

* The staging table was configured as expected, with columns labeled according to our data source.
* Custom columns were added to match the headers in the source file, including columns for address, city, name, state, and zip.

**Running the Import Again:**

* Running the import a second time resulted in five more records being added to the staging table, making a total of ten records.
* Each set of records in the staging table is tied to a specific import run, managed by the sys\_import\_set table.

**Understanding Import Sets:**

* The sys\_import\_set table in ServiceNow helps manage records loaded into the staging table by grouping them based on import runs.
* Each row in the sys\_import\_set table represents a group or set of records inserted during an import run.
* The staging table records are linked back to these import set groups, allowing for organized management of imported data.

**ServiceNow Transform Maps & Field Maps**

**Field Maps:**

Field Maps in ServiceNow define how data should be moved from the staging table to the target table, on a field-by-field basis. Each field mapping is stored in the Field Map table (CIS\_transform\_entry).

To see the existing field maps:

1. Navigate to the CIS\_transform\_entry table.
2. Run a .list command.
3. You should see the records that correspond to each field mapping.

**Transform Maps:**

Transform Maps group Field Maps together and define how the import will be executed. The Transform Maps are stored in the Transform Map table (CIS\_transform\_map).

To create a Transform Map:

1. Go to the CIS\_transform\_map table and click "New".
2. Name your Transform Map.
3. Specify the Source Table (the staging table) and the Target Table (where the data will be imported).

**Field Mapping Process**: You can either create Field Maps manually or use the Map Assist feature:

1. Start with the Transform Map.
2. Use the Map Assist tool to map fields from the staging table to the target table.

**Coalesce Field:**

* The Coalesce field helps avoid inserting duplicate rows by identifying a unique match during the import process.
* While you can set any field as the Coalesce field, it’s advisable to use a field that uniquely identifies each record.

Target Table Example: In this demonstration, we created a custom table called "my\_table" with fields that match the imported data (username, address, city, state, zip code).

**ServiceNow Incident Management Tutorial and Task Administration**

**ServiceNow Task Structure**

* Tasks in ServiceNow are represented as records in the task table.
* Common task types include Change Request, Incident, and Problem.
* These are implemented using a hierarchical database design, where specialized tables (e.g., incident, change request) extend from the task table, inheriting common attributes like description, status, and due date.

**Task Creation and Management**

* Tasks aren't created directly in the task table but through extended tables like Incident or Change Request.
* ServiceNow enables building processes around tasks, such as assignment rules, approvals, service level agreements (SLAs), and inactivity monitors.

**Task Assignment**

* Tasks are assigned to users or groups using the “Assigned to” and “Assignment group” fields.
* Assignment rules automate the process, with rules being checked in order of execution to assign tasks based on predefined conditions.

**Demonstration**

* Example: A new assignment rule for hardware incidents assigns them to the hardware support group and a specific user.

**User Interaction with Tasks**

* Users access their tasks through Service Desk applications, where “My Work” shows tasks assigned to them, and “My Group’s Work” shows tasks assigned to their group.
* ServiceNow facilitates task collaboration through features like user presence, real-time editing, and activity streams, enhancing teamwork and communication.

**Visual Task Boards**

* Visual task boards provide a graphical way to view tasks, allowing users to drag and drop tasks between lanes representing different task categories or statuses.

**ServiceNow Reporting Tutorial**

ServiceNow’s reporting functionality is supported by a simple data model:

1. **Report Table (sys\_report):** Stores each report as a record. When you create or view reports, you’re interacting with this table.
2. **Report Source Table:** Stores saved queries for retrieving data from source tables.
3. **Scheduled Email of Reports Table (sys\_auto\_report):** Manages automatic report execution and emailing.
4. **Report Users and Groups Table:** Shares reports with specific users or groups.
5. **Dashboard Table:** Allows reports to be added to dashboards.

**Exploring the Report Table**

To view the fields in the Report Table:

* Go to the application navigator and enter sys\_report.list.
* You’ll see a list of reports with fields like title, table, type, and more.

**Commonly Used Reporting Fields**

1. **sys\_id:** Unique identifier for each report.
2. **Title**: The report's title.
3. **Source Type:** Specifies if the report uses a data source or table.
4. **Report Source:** Specifies the data source record (if applicable).
5. **Table**: Primary database table for the report data.
6. **Field Name**: Field for grouping data in the report.
7. **Filter:** Conditions to filter the data.
8. **Type:** Specifies the type of report visualization (e.g., pie chart, bar chart).

**Creating a New Report**

You can create reports in three ways:

1. **Reports Application:** Use the “Create New” module to set up a report.
2. **ServiceNow Studio:** Create a report within an application scope.
3. **From Existing List View:** Create a report directly from a list view by selecting visualization options.

**Scheduling Reports**

To schedule a report:

* Open the report, click the sharing icon, and select “Schedule”.
* Set up details like report name, users, groups, frequency, and email content.

**Sharing Reports**

To share a report:

* Open the report, click the sharing icon, and choose “Share”.
* Share with individual users, groups, or globally.

**Adding Reports to Dashboards**

To add a report to a dashboard:

* Open the report, click the sharing icon, and select “Add to Dashboard”.
* Choose the dashboard and tab where you want the report displayed.

**What is Low Code No Code Development?**

**Exploring Low Code/No Code Development**

Are you curious about low code/no code software development? You've come to the right place! Welcome to *Tear Down That Wall: A Journey to Low Code/No Code*, brought to you by the team at ServiceNow.

Our journey begins in the familiar land of traditional software development. This journey will take you across the IT landscape, seeking something new and better. You'll discover what low code/no code development is, how it works, its pros and cons, and the career opportunities it offers. Sit back, relax, and enjoy the show!

**Meet Our Characters**

First, let’s introduce our key players:

* **The Savvy Business Person:** You might know this individual as the marketing guru, sales phenom, customer support specialist, technician, or staffing manager. Your company likely has many of these characters. They possess deep knowledge of the company’s products, services, and business processes. They’re always on the lookout for cost-cutting and productivity-boosting opportunities.
* **The IT Superstar:** This character has extensive education and training in technology. They’re experts in analysis, design, and creating complicated acronyms. With certifications in SQL, JDBC, CSS, AJAX, XML, ITSM, ITIL, and even CPR from the YMCA, they can code in almost any language but struggle with clear communication. Their role is to support the Savvy Business Person’s needs.
* **The Wall:** The antagonist in our story, the Wall, stands between the Savvy Business Person and their desired transformation. It serves as a barrier over which ideas are tossed and frustration is vented.

**Traditional Software Development**

Our story starts in the land of traditional software development, where things have been done the same way for years. The Savvy Business Person has a transformative idea but needs servers, databases, and coding to make it a reality. This involves SQL, HTML, JavaScript, integrations, design, and support—just another brick in the wall.

Frustration builds as the Savvy Business Person writes up requirements, tosses them over the wall to the IT Superstar, who then designs a system. But often, the result doesn’t meet expectations—leading to a cycle of feedback, revisions, and more feedback until a solution is finally delivered.

**The Low Code/No Code Revolution**

Enter the hero: Low Code/No Code. This approach breaks down barriers, allowing the Savvy Business Person to tackle digital transformation without relying solely on IT. Low code/no code simplifies the process, hiding complexity and empowering business people to solve problems independently.

With tools like ServiceNow’s App Engine Studio for application creation, the Now Experience UI Builder for workspace creation, and Flow Designer for automating workflows, the low code/no code approach streamlines development. Companies like Microsoft (PowerApps), Zoho, Appian, Salesforce, and ServiceNow offer similar capabilities.

**Pros and Cons**

Low code/no code development brings benefits like improved agility, faster market entry, lower costs, and increased automation. However, it’s not perfect. Simplified tools often come with constraints, less flexibility, and the reality of code still being present—just pre-written for you.

**Career Opportunities**

Before we wrap up, a word about career opportunities: Savvy Business People, embrace your new empowerment. Look for ways to simplify, automate, and improve your work. Continuously learn technical skills and collaborate with IT professionals. Remember, with power comes responsibility—start small, test thoroughly, and avoid blaming IT.

Developers, don’t worry—your jobs are safe! Embrace the low code/no code approach, and remember, in many companies, IT supports the business. Our goal should be to understand the business better and seek opportunities to make processes more efficient.