ServiceNow Training With AI Agents

The most comprehensive ServiceNow AI Agents Course with hands-on labs to build intelligent ServiceNow applications, real-world projects, and certification preparation. From beginner to expert in one complete Future Reddy Programme.

ServiceNow Platform Admin

Master the Now Platform fundamentals and AI Agent Studio

ServiceNow Development

Build modern web applications with HTML5, CSS3, and JavaScript

ServiceNow Scripting

Create powerful automation with client and server-side scripts

ServiceNow UI Builder

Design responsive, modern user experiences

ServiceNow IDE

Code-first development with VS Code integration

ServiceNow Testing

Automated testing with ATF and AI-powered quality assurance

ServiceNow Integrations

Connect systems with REST, GraphQL, and Integration Hub

ServiceNow ITSM

Implement IT Service Management best practices

ServiceNow HRSD

Transform HR service delivery with digital workflows

ServiceNow CSM

Deliver exceptional customer service experience

Digital Edify

India's First AI-Native Training Institute

Learn AI. Build Agents. Lead Future.

hello@digitaledify

+91 6304982304

Digital Edify

About Digital Edify

India's #1 Training Institute for the AI Era

Established: 2016 **Headquarters:** Hyderabad, Telangana **Reach:** Global (Online + Offline)

The Transformation Narrative

Digital Edify has evolved from a premium training institute in the Automation Era to an AI-first organisation leading the Agentic AI revolution. Since 2016, we've transformed over 100,000 professionals and built partnerships with more than 1,000 industry leaders. Our journey reflects the technological evolution of our time—from traditional job placement to career transformation, and now to building AI-native professionals who will shape the future of work.

1

Automation Era (2016-2023)

Premium Training Institute focused on job placement with 100K+ students trained

2

AI Revolution (2024-2025)

AI-Powered Training with industry-AI integration and career transformation focus

3

Agentic AI Leadership (2026+)

AI First Institute building AI-Native Professionals with 1 Million AI-Native Vision

> "We started in the Automation Era. We evolved through the AI Revolution. Now, we're leading the Agentic AI Future—with 100,000+ professionals already transformed and 1,000+ industry partners trusting our graduates."

Vision & Mission

Vision

"To Create 1 Million AI-Native Professionals Who Will Build the Agentic Future of Work"

Mission

"We transform learners into AI-native professionals through industry-aligned programmes that integrate Agentic AI into every discipline—from development to data science to enterprise platforms."

Course Highlights

Section 1: ServiceNow Platform Admin

Understand the ServiceNow platform, core configurations, automation, analytics, and AI-driven administration.

Section 2: ServiceNow Development

Build essential web and JavaScript skills required for ServiceNow development.

Section 3: ServiceNow Scripting

Create powerful client and server-side scripts to customize and automate ServiceNow.

Section 4: ServiceNow UI Builder

Design modern, dynamic user interfaces using ServiceNow UI Builder.

Section 5: ServiceNow IDE

Develop, manage, and deploy applications efficiently using ServiceNow IDE.

Section 6: ServiceNow Testing

Test, validate, and optimize ServiceNow applications using manual, automated, and AI-based testing.

Section 7: ServiceNow Integrations

Connect ServiceNow with external systems using APIs and integration tools.

Section 8: ServiceNow ITSM

Manage end-to-end IT service processes with best practices and AI support.

Section 9: ServiceNow HRSD

Deliver seamless HR services with case management, automation, and analytics.

Section 10: ServiceNow CSM

Build efficient customer service workflows with insights and AI assistance.

SECTION 1

ServiceNow Platform Admin

Module 1: Introduction to ServiceNow and the Now Platform

This foundational module establishes a comprehensive understanding of ServiceNow as a cloud-based platform and its position within the broader technology landscape. Students explore the evolution from traditional Software as a Service (SaaS) to Application Platform as a Service (aPaaS), understanding how ServiceNow enables organisations to build custom applications on a unified platform. The module covers the complete technology stack and architectural principles that underpin the Now Platform, including instance management, environment configurations, and the fundamental data model that drives all ServiceNow applications.

Platform Concepts

- Understanding instances and environments
- Data model architecture and relationships
- Multi-tenancy and security isolation
- Platform scalability and performance

Product Portfolio

- IT workflows and service management
- Employee workflows and HR service delivery
- Customer workflows and service operations
- Business workflows and automation

01

Cloud Computing Fundamentals

Explore where ServiceNow fits in the SaaS to aPaaS spectrum

02

Technology Stack

Understand the architectural layers and components

03

Platform Capabilities

Discover the breadth of workflow solutions available

04

Certification Path

Navigate the CSA certification journey and learning resources

Module 2: ServiceNow UI & Navigation

Interface Components

- Banner: Global search, notifications, user menu
- Navigator: Application and module hierarchy
- Content Frame: Lists, forms, and workspaces
- Breadcrumbs: Navigation history and context

Working with Data

- Lists: Multiple records with filtering and sorting
- Forms: Single record viewing and editing
- Global Search: Cross-table record discovery
- Filters: Advanced query building and saved filters

Customisation Capabilities

- List configuration: Columns, sorting, grouping
- Form layout: Sections, fields, related lists
- Choice lists: Dropdown values and dependencies
- Reference fields: Table relationships and lookups

Navigation Organisation

Applications serve as containers for related modules, providing logical grouping of functionality. Modules represent individual menu items that link to specific tables, reports, or pages. Understanding this hierarchy enables administrators to create intuitive navigation structures that align with business processes and user roles.

UI Branding

ServiceNow supports extensive branding customisation, allowing organisations to apply their visual identity through logos, colour schemes, and custom themes. This capability ensures the platform aligns with corporate branding guidelines whilst maintaining usability and accessibility standards.

Module 3: ServiceNow Data & Database Fundamentals

Data Fundamentals

Understanding databases, tables, records, fields, and the critical role of sys_id as the universal unique identifier

Table Types

Base tables for core functionality, Extended tables for inheritance, Custom tables for business-specific requirements

Relationships

Extending creates inheritance hierarchies, Referencing establishes foreign key connections between tables

Exploring the Data Model

ServiceNow provides powerful tools for understanding and navigating the data model. The Tables module displays all available tables with their properties and relationships. The Dictionary reveals detailed field definitions including data types, constraints, and dependencies. The Schema Map offers a visual representation of table relationships, showing inheritance hierarchies and reference connections. These tools enable administrators to understand existing structures and plan new customisations effectively.

CMDB Architecture

The Configuration Management Database (CMDB) represents a specialised implementation of ServiceNow's data model, storing information about IT infrastructure components and their relationships. Understanding CMDB architecture is essential for implementing IT Service Management processes.

Data Integration

Import Sets provide a staging area for external data, allowing transformation and validation before loading into production tables. Update Sets capture configuration changes for migration between instances, ensuring controlled deployment of customisations.

Data Types

String, Integer, Boolean, Date, Reference, Choice, and specialised types for specific use cases

Custom Tables

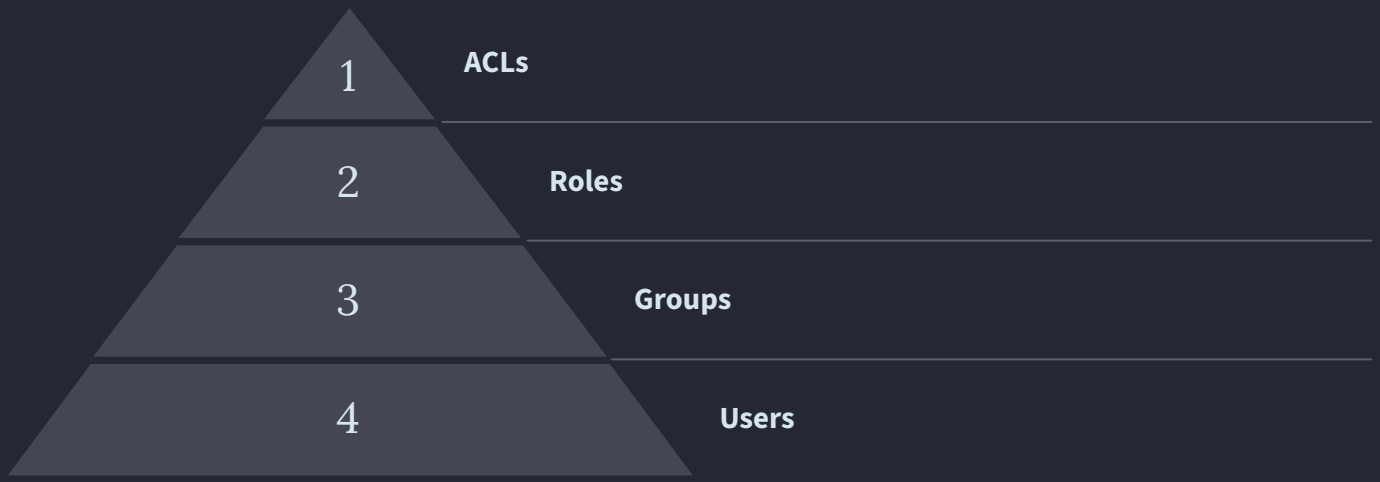
Creating tables with appropriate naming, fields, and relationships to support business processes

Import Sets

Staging external data with transform maps for validation and loading into production tables

Module 4: Users, Groups, Roles & Access Controls

Security architecture in ServiceNow operates on the principle of least privilege, ensuring users can access only the data and functionality necessary for their responsibilities. This module explores the four-layer security model that governs platform access. Roles serve as containers for permissions, bundling related access rights that can be assigned to users or groups. Access Control Lists (ACLs) represent the enforcement layer, defining precisely which users can perform specific operations on particular data.



Access Control Operations

1

User Administration

Creating and managing user accounts with appropriate attributes and authentication methods

2

Group Organisation

Structuring teams and departments for work assignment and access control

3

Role Assignment

Bundling permissions and assigning roles to users and groups

4

ACL Configuration

Defining precise access rules for tables, fields, and operations

Module 5: Customisation & Scripting

UI Policies

No-code form behaviour control for field visibility, mandatory status, and read-only states based on conditions

Client Scripts

Browser-based JavaScript for real-time form validation, field auto-population, and user interaction handling

Business Rules

Server-side automation for data validation, field calculations, and related record updates on database operations

UI Actions

Custom buttons and links for forms and lists, enabling specialised workflows and user-initiated processes

Data Policies

Server-side validation enforcing data quality rules regardless of how records are created or modified

Script Includes

Reusable code libraries for common functions, promoting code reuse and maintainability across applications

Module 6: ITIL Foundations

7 Guiding Principles

Focus on value, Start where you are, Progress iteratively, Collaborate and promote visibility, Think and work holistically, Keep it simple, Optimise and automate

34 ITIL Practices

General, Service, and Technical management practices mapped to ServiceNow modules

Four Dimensions

Organisations and people, Information and technology, Partners and suppliers, Value streams and processes

Service Value Chain

Plan, Improve, Engage, Design and transition, Obtain/build, Deliver and support

Core ITSM Processes

01

Service Desk

Single point of contact for all user interactions and service requests

02

Incident Management

Rapid restoration of service following disruptions or degradations

03

Problem Management

Root cause analysis and permanent resolution of recurring issues

04

Change Management

Controlled implementation of changes with risk assessment and approval

ITIL Metrics and KPIs

Measuring service management effectiveness requires tracking key performance indicators aligned with ITIL practices. Common metrics include incident resolution time, first-call resolution rate, change success rate, and service availability. These measurements provide insights for continual service improvement.

ITIL Certification Path

ITIL certifications validate knowledge of IT Service Management best practices. The certification scheme includes Foundation, Practitioner, Intermediate, and Expert levels. Understanding ITIL principles enhances ServiceNow implementation effectiveness and career opportunities.

Module 7: Flows, Workflows, Automation & Autonomous Systems

Basic Automation

Simple rule-based actions triggered by specific events or conditions

Process Automation

Multi-step workflows with approvals, notifications, and task assignments

Intelligent Automation

AI-enhanced workflows with predictive capabilities and adaptive behaviour

Autonomous Systems

Self-managing processes that perceive, reason, act, and learn without human intervention

ServiceNow Automation Tools

Flow Designer

Visual workflow builder with drag-and-drop interface, pre-built actions, and natural language triggers.

Playbooks

Interactive workflows that guide users through complex processes with decision points, data collection, and task execution.

Identify Process

Analyse manual workflows to identify automation opportunities and define requirements

Design Flow

Map process steps, decision points, and integrations using appropriate automation tools

Build & Test

Implement automation with Flow Designer or Workflow Studio, testing thoroughly

Deploy & Monitor

Release automation to production and monitor performance for optimisation

Module 8: AI Agent Studio

Perceive

Gathering information from conversations, systems, and data sources to understand context

Reason

Analysing situations, evaluating options, and determining optimal courses of action

Learn

Improving performance through feedback, outcomes, and continuous optimisation

Act

Executing workflows, updating records, and orchestrating processes autonomously

Agentic AI Architecture

AI Agent Orchestrator

Central coordination layer routing requests to specialised agents

Specialised Agents

Domain-specific agents for IT, HR, Customer Service, Security

Workflow Data Fabric

Unified data access layer enabling seamless information retrieval

Now Assist

AI-powered productivity enhancement for human workers

Building and Configuring AI Agents

Agent Configuration

- Define agent purpose and scope
- Select tools and data sources
- Configure conversation flows
- Establish safety guardrails
- Train with knowledge and examples

Real-World Applications

- IT: Automated incident resolution
- HR: Employee onboarding assistance
- Customer Service: Intelligent routing
- Security: Threat detection and response
- Operations: Predictive maintenance

Module 9: Platform Analytics & Reports

7	3	100+
Report Types	Analytics Layers	Pre-built Reports
Bar, Pie, Line, List, Pivot, Gauge, and Score charts for diverse visualisation needs	Indicators, Breakdowns, and Data Collections forming the Performance Analytics foundation	Out-of-box reports covering common ITSM, HRSD, and CSM scenarios

Building Reports with Report Designer

Analytics Q&A

Natural language queries enable business users to ask questions and receive instant visualisations without understanding database structures or report configuration. This capability democratises data access, empowering users to explore information independently.

Dashboard Creation

Dashboards combine multiple reports and widgets into unified views, providing comprehensive insights at a glance. Effective dashboards balance information density with clarity, presenting the most critical metrics prominently whilst supporting drill-down for details.

Performance Analytics

1

Define Indicators

Identify key metrics aligned with business objectives and ITIL practices

2

Configure Breakdowns

Segment metrics by relevant dimensions for detailed analysis

3

Schedule Collections

Establish data gathering frequency and retention policies

4

Create Dashboards

Build executive scorecards and operational dashboards

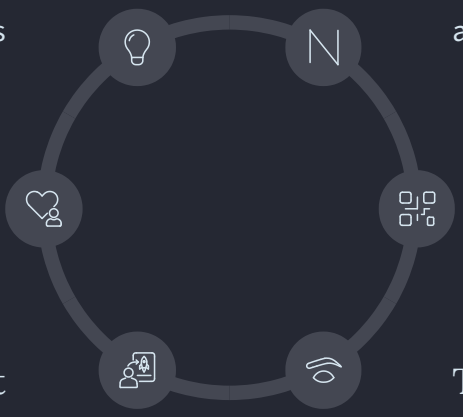
Module 10: Building Custom Applications

Ideation

Identify business needs and define application requirements

Design

Create data model, user interface, and workflow specifications



Maintenance

Monitor performance, gather feedback, and implement enhancements

Development

Build tables, forms, business rules, and integrations

Deployment

Migrate to production using Update Sets or source control

Testing

Validate functionality, performance, and user experience

Three Design Layers

Business Layer

Process definitions, business rules, workflow logic, and success criteria

Data Layer

Table structures, field definitions, relationships, and data integrity rules

UI Layer

Forms, lists, portals, mobile interfaces, and user experience design

ServiceNow Platform Architecture

Presentation Layer

User interfaces, portals, mobile apps

Customisation Layer

Custom applications and extensions

Application Layer

ITSM, HRSD, CSM modules

Foundation Layer

Database, security, core services

SECTION 2

ServiceNow Development

Module 1: HTML5 Complete Guide

Document Structure

DOCTYPE, html, head, body, and proper nesting of elements

Semantic HTML5

Header, nav, main, article, section, aside, footer for meaningful structure

Text & Links

Headings, paragraphs, emphasis, anchors, and text formatting

Media Elements

Images, audio, video, and responsive media embedding

Lists, Tables, and Forms

Accessibility Best Practices

Creating accessible HTML ensures all users, including those with disabilities, can interact with web applications effectively. Best practices include using semantic elements appropriately, providing alternative text for images, ensuring sufficient colour contrast, enabling keyboard navigation, and using ARIA attributes when necessary to convey additional meaning to assistive technologies.

HTML5 in ServiceNow

ServiceNow leverages HTML5 throughout the platform—in UI Pages, Service Portal widgets, and Email Notifications. Understanding HTML5 enables developers to create custom interfaces, enhance user experiences, and build responsive layouts that work across devices. The platform's UI Builder and Service Portal frameworks abstract much HTML complexity whilst still requiring foundational knowledge.

Module 2: CSS3 Complete Guide

1

CSS Fundamentals

Selectors, specificity, cascade, inheritance, and the box model

2

Typography & Colors

Font families, sizes, weights, line height, colour systems, and gradients

3

Layouts & Positioning

Display properties, positioning schemes, floats, and clearing

4

Flexbox

One-dimensional layouts with flexible containers and items

5

CSS Grid

Two-dimensional layouts with rows, columns, and grid areas

6

Responsive Design

Media queries, viewport units, and mobile-first approaches

Modern Layout Techniques

Responsive Design Principles

Modern web applications must work across devices from mobile phones to desktop monitors. Responsive design uses media queries to apply different styles based on screen size, orientation, and capabilities. Mobile-first approaches start with mobile layouts and progressively enhance for larger screens, ensuring core functionality works everywhere.

Transitions & Animations

CSS transitions smoothly animate property changes, enhancing user experience with visual feedback. Animations enable more complex, keyframe-based motion. Both capabilities should be used judiciously—enhancing usability without distracting or slowing interactions. Performance considerations include using transform and opacity properties for hardware-accelerated animations.

Module 3: JavaScript Fundamentals

JavaScript brings interactivity and dynamic behaviour to web applications. This module establishes a solid foundation in JavaScript programming, beginning with how JavaScript integrates with HTML through script tags and event handlers. Understanding variables and data types is fundamental—JavaScript supports primitives (strings, numbers, booleans, null, undefined, symbols) and objects (including arrays and functions). The let and const keywords introduced in ES6 provide block-scoped variable declarations, replacing the problematic var keyword. Operators enable calculations, comparisons, and logical operations, forming the building blocks of expressions and algorithms.

01

JavaScript Basics

Integration with HTML, script placement, and execution context

02

Variables & Data Types

let, const, primitives, objects, type coercion, and conversion

03

Operators

Arithmetic, comparison, logical, assignment, and ternary operators

04

Control Flow

if/else statements, switch cases, and conditional logic

05

Loops

for, while, do-while, for...in, for...of iteration patterns

Functions and Scope

Function Declarations

Traditional named functions hoisted to scope top, available before declaration

Function Expressions

Functions assigned to variables, not hoisted, enabling conditional definition

Arrow Functions

Concise syntax with lexical this binding, ideal for callbacks and functional programming

Module 4: JavaScript Functions & Arrays

Advanced Functions

- Arrow functions and lexical this
- IIFEs for scope isolation
- Higher-order functions
- Callbacks and event handlers

Closures

- Lexical scoping principles
- Data privacy patterns
- Factory functions
- Module patterns

Recursion

- Base cases and recursive cases
- Stack overflow prevention
- Tree traversal algorithms
- Divide and conquer strategies

Array Fundamentals and Manipulation

1

Mutating Methods

push, pop, shift, unshift, splice, sort, reverse—modify original array

2

Non-Mutating Methods

slice, concat, join, toString—return new values without changing original

Array Iteration and Functional Programming

Functional Array Methods

- forEach: Execute function for each element
- map: Transform elements into new array
- filter: Select elements matching condition
- reduce: Accumulate values into single result
- find: Return first matching element
- some/every: Test element conditions

Modern Syntax

- Array destructuring for element extraction
- Spread operator for array combination
- Rest parameters for variable arguments
- Default parameters for function arguments
- Template literals for string interpolation

Module 5: JavaScript Objects & Asynchronous Programming

Object Fundamentals

Creating objects with literals, constructors, and Object.create. Accessing and modifying properties with dot and bracket notation. Object methods for functionality encapsulation.

Modern Object Syntax

Destructuring for property extraction. Spread operator for composition. Enhanced object literals with shorthand properties and computed property names.

Object Methods

Object.keys, Object.values, Object.entries for iteration. Object.assign for shallow copying. Object.freeze and Object.seal for immutability. Symbols for unique property keys.

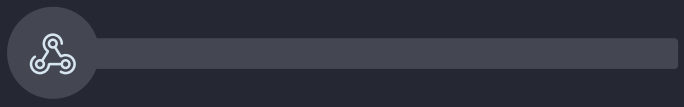
Asynchronous JavaScript

Callbacks and Callback Hell

Callbacks represent the original asynchronous pattern—passing functions to be executed upon operation completion. However, nested callbacks create "callback hell"—deeply indented, difficult-to-read code.

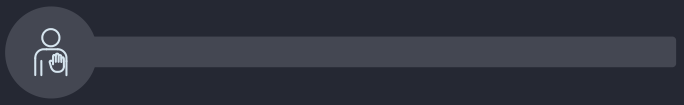
Promises

Promises provide a cleaner asynchronous pattern, representing eventual completion or failure of operations. Promises have three states: pending (initial), fulfilled (successful), or rejected (failed).



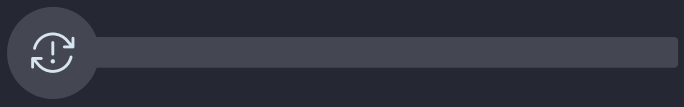
Callbacks

Original async pattern with nested functions



Promises

Chainable async operations with then/catch



Async/Await

Synchronous-looking asynchronous code

Fetch API

Modern HTTP requests with promises

SECTION 3

ServiceNow Scripting

Module 1: Scripting Overview

What is Platform Scripting?

Custom JavaScript code that extends ServiceNow functionality beyond declarative configuration

Why Not Script?

Scripts increase complexity, maintenance burden, and upgrade risk—use configuration first

When to Script?

Complex logic, calculations, integrations, or performance requirements exceed declarative tools

Where Scripts Execute?

Client-side (browser) for immediate feedback, Server-side (platform) for data access

Application Scope and JavaScript Engine

Condition Builder vs JavaScript

Condition Builder provides a visual interface for defining conditions without scripting. It's ideal for simple comparisons and logical operations. JavaScript conditions offer more flexibility for complex logic, calculations, and dynamic evaluations.

ServiceNow APIs

ServiceNow provides extensive APIs for common operations. GlideSystem (gs) offers utility methods for logging, user information, and date handling. GlideRecord enables database queries and updates.

Syntax Editor and Development Tools

The ServiceNow Syntax Editor provides context-sensitive help, autocomplete, and syntax highlighting for script development.

Module 2: Client Scripts

1	2
onLoad Form initialisation, default values, initial field states	onChange Field-level validation, auto-population, dynamic adjustments
3	**4**
onSubmit Final validation, confirmation dialogues, submission control	onCellEdit List view inline editing validation and processing

Client-Side APIs

g_form Methods

- getValue/setValue: Field value access
- setVisible/setReadOnly: Field state control
- setMandatory: Required field enforcement
- addInfoMessage: User feedback
- clearMessages: Message management
- getSections: Form section access

Best Practices

- Reference field names, not labels
- Minimise server lookups for performance
- Use g_scratchpad for server data
- Validate user input before submission
- Provide clear error messages
- Test across browsers and devices

Debugging Client Scripts

Effective debugging requires understanding browser developer tools. The Console tab displays log messages, errors, and warnings. The Sources tab enables breakpoint debugging, stepping through code line by line. The Network tab shows server requests and responses, useful for diagnosing AJAX issues. Common errors include referencing undefined variables, incorrect field names, and asynchronous timing issues. Console.log statements provide visibility into variable values and execution flow. Understanding the browser's JavaScript console is essential for troubleshooting client-side issues efficiently.

Module 3: UI Policies

01

Define Conditions

Use Condition Builder to specify when the policy should execute

02

Configure Actions

Set field visibility, mandatory status, and read-only states

03

Set Reverse Option

Determine whether to reverse actions when conditions become false

04

Test Behaviour

Verify policy executes correctly across different scenarios

UI Policy Configuration

Creating effective UI Policies requires understanding their configuration options. The Conditions field defines when the policy executes, using the visual Condition Builder to specify field values and logical operators. The "Reverse if false" checkbox determines whether actions reverse when conditions no longer match—enabling this option ensures fields return to their original state when conditions change. UI Policy Actions define specific field behaviours: Visible controls whether fields display, Mandatory determines whether fields require values, and Read-only prevents field editing. Each action targets specific fields, allowing granular control over form behaviour. The execution order matters when multiple policies affect the same fields—policies execute in order of their order field value.

Performance Benefits

UI Policies offer significant performance advantages over Client Scripts. They don't require JavaScript parsing or execution, reducing browser processing time. They're easier to maintain, as administrators can modify conditions and actions without understanding code. They're less prone to errors, as the visual interface prevents syntax mistakes.

Baseline Examples

The incident form demonstrates effective UI Policy usage. When Priority is set to 1 - Critical, the policy makes Assignment Group and Assigned To mandatory, ensuring critical incidents receive immediate attention. When State changes to Resolved, the policy makes Resolution Notes mandatory, ensuring proper documentation.

Module 4: Business Rules

Before Rules Data validation and field population before database save	**After Rules** Related record updates and workflow triggers after save
Async Rules Background processing without blocking user interactions	**Display Rules** Query optimisation and record filtering before display

Business Rule Configuration

Configuring Business Rules requires careful attention to timing and conditions. The "When to run" section specifies which database operations trigger the rule: Insert (new records), Update (modified records), Delete (removed records), or Query (record retrieval). Conditions determine whether the rule executes for specific records, using the Condition Builder or JavaScript for complex logic. The Advanced checkbox enables custom scripting, providing access to the current and previous objects.

The current and previous Objects

Business Rules access record data through the current object, which represents the record being processed. The current object provides methods for reading field values (current.field_name), setting values (current.field_name = value), and updating records (current.update()). The previous object contains field values before the current operation, enabling comparison of old versus new values.

Preventing Recursive Rules

Business Rules that update records can trigger themselves recursively, causing infinite loops. Preventing recursion requires checking whether the rule already executed using flags or comparing current versus previous values. The current.setWorkflow(false) method disables Business Rules for specific updates, breaking recursion chains.

Performance Considerations

Business Rules impact database performance, especially when processing large record sets. Minimising rule complexity, avoiding unnecessary GlideRecord queries, and using async rules for non-critical operations improves performance. Understanding execution order and consolidating related logic into single rules reduces overhead.

Module 5: GlideSystem API

Logging Methods

gs.log for debugging, gs.warn for warnings, gs.error for errors—all write to system logs

User Messages

gs.addInfoMessage for success feedback, gs.addErrorMessage for validation errors

User Information

gs.getUserID, gs.getUserName, gs.getUser for current user context and permissions

Date and Time

gs.now, gs.nowDateTime, gs.daysAgo for temporal calculations and comparisons

User Information Methods

GlideSystem provides methods for accessing current user information, essential for implementing role-based logic and personalised experiences. The gs.getUserID method returns the sys_id of the current user, useful for database queries and record assignments. The gs.getUserName method returns the username, often displayed in logs or messages. The gs.getUser method returns a GlideUser object with additional properties and methods, including role checking and preference access. These methods enable scripts to adapt behaviour based on who's executing them, implementing security rules or customising user experiences.

Date and Time Methods

Events in ServiceNow

The gs.eventQueue method triggers events, enabling event-driven automation. Events decouple triggers from actions, allowing multiple scripts to respond to the same event. This pattern promotes modularity and maintainability, as event handlers can be added or modified without changing triggering code.

Module Notes

GlideSystem methods can be included in any server-side script—Business Rules, Script Includes, Scheduled Jobs, and more. They provide consistent functionality across contexts, simplifying development and reducing code duplication. Understanding the full GlideSystem API enables efficient, maintainable server-side scripting.

Module 6: GlideRecord

1

Create

Instantiate GlideRecord object for target table

2

Query

Add conditions and execute database query

3

Loop

Iterate through result set with next method

4

Process

Read or modify each record as needed

CRUD Operations

Create

initialize, setValue, insert

Read

addQuery, query, next

Update

setValue, update, updateMultiple

Delete

deleteRecord, deleteMultiple

Query Building

Building effective queries requires understanding GlideRecord's query methods and operators. The addQuery method accepts three parameters: field name, operator, and value. Operators include equals (=), not equals (!=), less than (<), greater than (>), LIKE for pattern matching, IN for list membership, and many others. The addEncodedQuery method accepts complex conditions as encoded strings, useful for conditions built in the Condition Builder.

Dot-Walking and Advanced Techniques

Dot-walking enables accessing reference field properties without additional queries. For example, current.caller_id.email retrieves the caller's email address by traversing the reference from incident to user. GlideAggregate provides reporting queries with aggregation functions like COUNT, SUM, AVG, MIN, and MAX. The getRowCount method returns query result counts, though GlideAggregate is more efficient for this purpose.

Module 7: Script Includes

Script Include Overview

- Minimal configuration options
- Server-side execution only
- No default objects available
- Parameters passed from calling script
- Must be explicitly called to execute

Configuration Options

- Name: Script Include identifier
- API Name: Scoped name with namespace
- Client callable: GlideAjax enablement
- Application scope: Global or scoped
- Accessible from: Scope visibility

Script Include Types

Classless

Simple functions without OOP structure, ideal for utilities

Class-based

Object-oriented design with initialize method and state

Client-callable

GlideAjax integration for client-server communication

Calling Script Includes

Script Includes can be invoked from various contexts throughout the platform. Business Rules call Script Includes to encapsulate complex logic, keeping rules clean and maintainable. Other Script Includes can call each other, building layered functionality. Client Scripts use GlideAjax to call client-callable Script Includes, retrieving server-side data asynchronously. UI Actions invoke Script Includes to perform operations when users click buttons. Scheduled Jobs call Script Includes for batch processing and maintenance tasks. Understanding these calling patterns enables building modular, reusable code that serves multiple purposes.

GlideAjax Pattern

GlideAjax enables client-side scripts to call server-side Script Includes asynchronously, retrieving data without page refreshes. Creating client-callable Script Includes requires extending the AbstractAjaxProcessor class and implementing methods that return values. Client-side code instantiates GlideAjax objects, adds parameters, and provides callback functions to handle responses. This pattern enables sophisticated client-server interactions whilst maintaining security—server-side code validates requests and enforces permissions. Understanding GlideAjax is essential for building responsive, data-driven user interfaces that leverage server-side logic without compromising performance or security.

Module 8: Flow Designer Scripting

Core Concepts

Actions (reusable logic), Subflows (reusable sequences), Triggers (flow initiation), Flow Variables (data storage)

Scripting in Flows

Custom script steps for complex logic, accessing flow variables, evaluating core actions first

Error Handler

Catching errors, running recovery actions, managing error states and messages

Best Practices

Evaluate core actions before custom scripts, real-world use cases, appropriate usage frequency

Scripting in Flow Designer

When to Script

Custom scripts become necessary when requirements exceed available actions—complex calculations, specialised data transformations, or unique integration patterns. However, scripts increase maintenance burden and reduce flow readability. Always evaluate whether existing actions can meet requirements before adding custom code.

Error Handler

Error Handlers catch exceptions during flow execution, enabling graceful failure handling and recovery. Configuring Error Handlers involves specifying which actions or subflows to run when errors occur. Error status details include error codes and messages, providing context for troubleshooting. Understanding error states—Completed (error caught), Completed (error skipped), or Error—enables appropriate recovery strategies.

SECTION 4

ServiceNow UI Builder

Module 1: UI Builder Fundamentals & Getting Started

01

Understand UXF

Learn component architecture and Now Experience Framework principles

02

Explore Interface

Navigate Canvas, component panel, and configuration options

03

Create Pages

Build responsive pages with variants for different contexts

04

Configure Components

Set properties, bind data, and style components appropriately

UI Builder Interface

The UI Builder interface consists of several key areas that enable visual development. The Canvas displays the page being designed, showing components in their rendered state with responsive preview modes for desktop, tablet, and mobile devices. The Component Panel provides access to the component library, organised by category—Layout components for structure, Data Display components for presenting information, Input components for user interaction, and Navigation components for wayfinding.

Pages and Variants

Pages represent individual screens or views within an application. Variants enable creating different versions of pages for specific contexts—authenticated versus anonymous users, different roles, or device types. URL routing connects pages to specific paths, enabling navigation and deep linking.

Component Library

The component library provides dozens of pre-built components covering common UI patterns. Layout components include containers, sections, and grids. Data display components show information through lists, cards, and detail views. Input components enable data entry through forms, fields, and buttons. Navigation components provide menus, breadcrumbs, and tabs.

Module 2: Components, Pages & Data Binding

Component Properties Configurable attributes controlling appearance and behaviour	**Data Bindings** Connections between components and data sources
Events Interaction handlers for clicks, changes, and custom events	**Styles** Visual appearance through CSS and design tokens

Data Resources and Query Configuration

Data Resources provide declarative interfaces for database operations, eliminating the need for custom GlideRecord scripts in most scenarios. Configuring Data Resources involves selecting the target table, defining query conditions using the visual condition builder, and specifying which fields to retrieve.

Data Binding Syntax and Expressions

Client State Management

Client State provides application-level data storage, enabling components to share information and maintain application context. Creating state variables involves defining names, types, and initial values. Managing state requires understanding immutability principles—state should be updated through setter functions rather than direct mutation, ensuring predictable behaviour and enabling change detection.

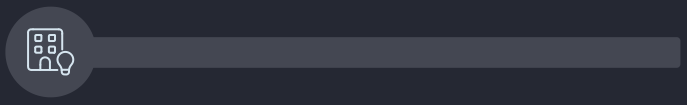
Data Brokers

Data Brokers enable executing server-side scripts from UI Builder pages, providing flexibility for operations that exceed Data Resource capabilities. Configuring Data Brokers involves creating Script Includes that accept parameters and return results. Client-side code invokes Data Brokers asynchronously, handling responses through promises or async/await syntax.

Dynamic Lists and Forms

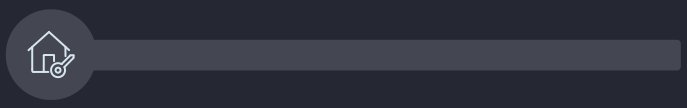
Lists and forms represent common UI patterns that UI Builder handles elegantly. Dynamic Lists display collections of records with filtering, sorting, and pagination. Repeaters enable custom layouts for list items, providing flexibility beyond standard list components.

Module 3: Advanced Component Development



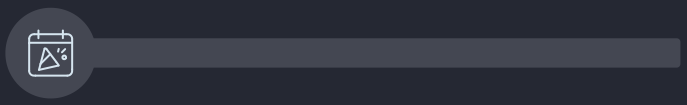
Component Architecture

HTML/JSX templates, JavaScript logic, CSS styles



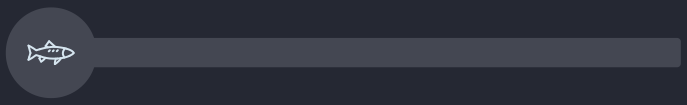
Properties Schema

Define configurable attributes with types and validation



Component Events

Dispatch custom events for parent communication



Lifecycle Hooks

Mount, update, unmount for resource management

Component Lifecycle and State Management

Styling Components

Component-scoped CSS ensures styles don't leak to other components, preventing conflicts and enabling modular design. The Now Design System provides design tokens for colours, spacing, typography, and other visual properties, ensuring consistency across applications. Using design tokens rather than hard-coded values enables theming and maintains visual coherence.

Working with Data

Components interact with data through HTTP effects, which handle asynchronous operations like API calls. Configuring HTTP effects involves specifying endpoints, methods, headers, and request bodies. Error handling requires try-catch blocks or promise rejection handlers, providing user feedback when operations fail. Loading states improve user experience by indicating when operations are in progress.

Advanced Patterns and Testing

Advanced component patterns enable sophisticated designs. Compound components combine multiple related components into cohesive units, like form groups with labels, inputs, and validation messages. Render props enable components to share logic without inheritance, promoting composition over inheritance.

Module 4: Client State Management & Event Handling

Component State

Local state for individual component concerns and internal logic

Page State

Shared state coordinating related components within a page

Session State

User-specific state persisting across page loads

Application State

Global state accessible across pages and components

Form State Management

Async State Management

Asynchronous operations require managing loading states, error states, and success states. Loading states indicate operations in progress, providing user feedback through spinners or progress indicators. Error handling displays meaningful messages when operations fail, enabling users to understand and resolve issues. Optimistic updates improve perceived performance by updating UI immediately whilst operations complete in the background, rolling back if operations fail.

Event Bus Pattern

The Event Bus pattern enables cross-component messaging without direct coupling. Components publish events to the bus, whilst other components subscribe to events they're interested in. This pattern implements the publish-subscribe pattern, promoting loose coupling and enabling complex component interactions without tight dependencies.

URL State and Performance Optimisation

URL State Management enables deep linking and shareable application states. Encoding state in URL parameters allows users to bookmark specific application states or share links with colleagues. URL state synchronisation keeps URL parameters in sync with application state, enabling browser back/forward navigation. Performance optimisation techniques include memoization to cache expensive calculations, debouncing to limit function execution frequency, and throttling to ensure functions execute at most once per time period. Understanding these techniques enables building performant applications that remain responsive even with complex state and frequent updates.

Module 5: Integration, Deployment & Best Practices

Platform Integration

Agent Workspace, Employee Center, and portal integration patterns

Flow Designer Integration

Triggering workflows and handling execution results

Service Catalog Integration

Custom request forms and catalog item workflows

REST API Integration

Consuming external APIs and creating custom endpoints

Application Deployment

Performance Optimisation

Performance optimisation techniques include lazy loading components to reduce initial page load, code splitting to separate large applications into smaller bundles, and caching strategies to reduce server requests. Monitoring performance metrics identifies bottlenecks and opportunities for improvement.

Accessibility (a11y)

Accessibility ensures applications work for all users, including those with disabilities. WCAG compliance requires meeting Web Content Accessibility Guidelines standards. Keyboard navigation enables users to interact without a mouse. ARIA attributes provide additional context for assistive technologies like screen readers.

SECTION 5

ServiceNow IDE

Module 1: Introduction to ServiceNow IDE & Module 2: IDE Setup

1

Platform UI

Browser-based development with form editors and script fields

2

Studio

Integrated development environment within ServiceNow platform

3

IDE

Code-first development with VS Code, Git, and local tooling

IDE Architecture and Development Workflow

Benefits for Teams

- Local development with familiar tools
- Git version control and branching
- TypeScript for type safety
- Modern IDE features and extensions
- Offline development capabilities
- Automated testing integration

Infrastructure Requirements

- Visual Studio Code installation
- Git version control system
- ServiceNow CLI tool
- Node.js runtime environment
- ServiceNow instance access
- Appropriate user permissions

Software Installation and Workspace Setup

01

Install Software

VS Code, Git, Node.js, ServiceNow CLI, and extensions

02

Configure Instance

Set up authentication and test connectivity

03

Create Workspace

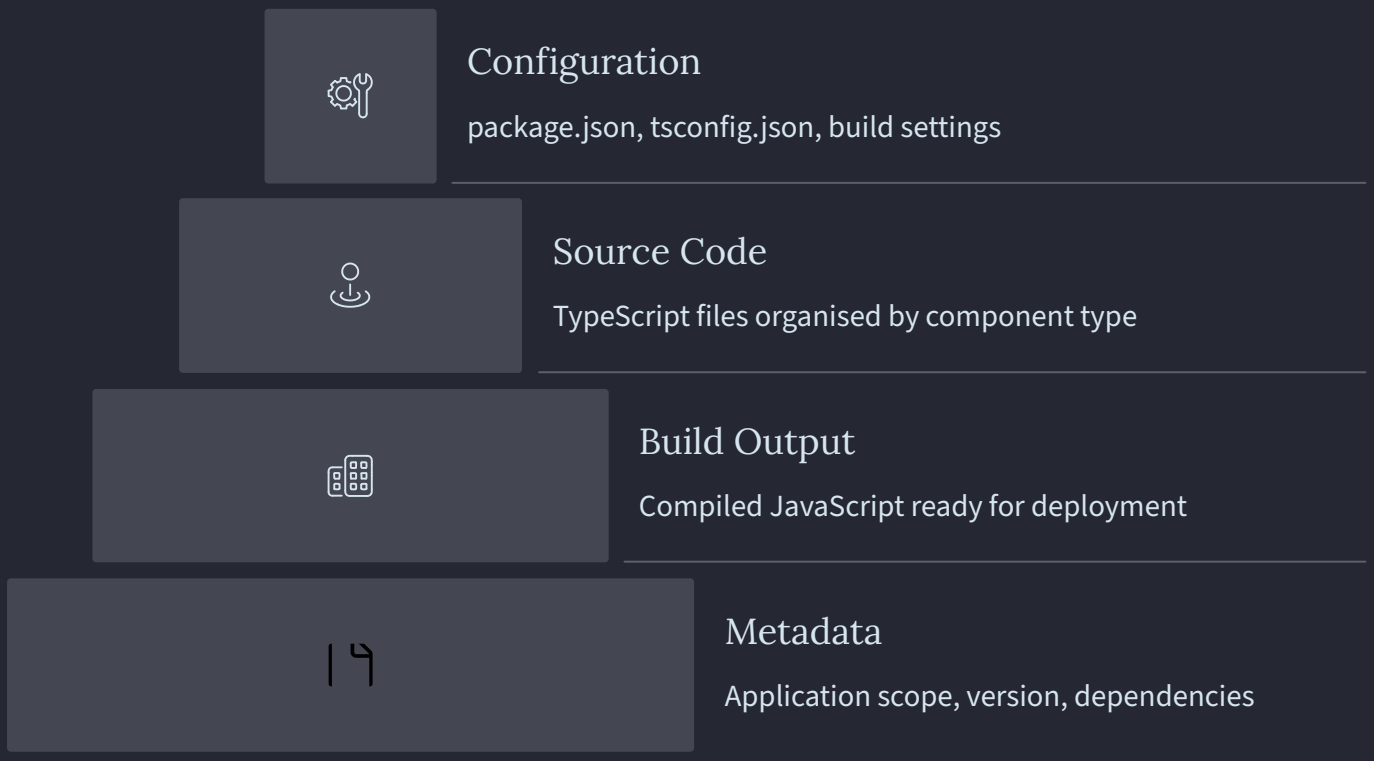
Initialise project structure and configuration files

04

Verify Setup

Build and deploy test application to confirm functionality

Module 3: Application Architecture & Structure



The index.ts File and Table Definitions

The index.ts file serves as the application entry point, defining tables, fields, and metadata. This file uses TypeScript syntax to declare table structures with type safety, ensuring field definitions are consistent and validated at compile time. Defining tables involves specifying table names, labels, and extension relationships. Field definitions include field names, types, labels, and validation rules.

Server-Side and Client-Side Code

Using Third-Party Libraries

The IDE supports npm packages, enabling use of third-party JavaScript libraries. Installing packages involves npm commands, whilst importing requires TypeScript import statements. Type definitions (@types packages) provide IntelliSense and type checking for libraries, improving development experience and code quality.

Metadata Explorer

The Metadata Explorer enables browsing ServiceNow instance metadata, pulling records into the IDE, and converting UI-configured components to code. This capability bridges the gap between UI-based and code-based development, enabling teams to migrate existing applications to IDE gradually.

Module 4: Building Applications with IDE & Module 5: Source Control

Design

Define requirements and data model

Develop

Write TypeScript code for components

Iterate

Incorporate feedback and enhance

Build

Compile and validate code

Test

Verify functionality and integration

Deploy

Push to ServiceNow instance

Building the Safety Incident Application

The course project—a Safety Incident Management Application—demonstrates IDE development practices through a real-world scenario. Requirements include tracking workplace safety incidents, managing investigations, and generating reports. The data model includes tables for incidents, investigations, and corrective actions with appropriate relationships.

Git Version Control and Deployment

Multi-Instance Deployment

Professional development requires deploying applications across multiple instances—Development for active development, Test for quality assurance, and Production for end users. The IDE supports this workflow through instance profiles and deployment scripts, enabling consistent, repeatable deployments.

CI/CD Integration

Continuous Integration/Continuous Deployment (CI/CD) automates build, test, and deployment processes. Pipeline architecture includes source control triggers, automated builds, test execution, quality gates, and deployment automation. GitHub Actions, Jenkins, and other CI/CD tools integrate with ServiceNow CLI, enabling sophisticated automation workflows.

SECTION 6

Module 1: Introduction to ServiceNow Testing & Module 2: ATF Essentials

Unit Testing

Testing individual components like scripts and business rules in isolation

Functional Testing

Validating business logic and workflows meet requirements

Integration Testing

Verifying interactions between modules and external systems

Regression Testing

Ensuring existing features work after changes or upgrades

Performance Testing

Assessing speed and reliability under load conditions

Security Testing

Identifying vulnerabilities and ensuring data protection

User Acceptance Testing

Business validation that requirements are met

Agentic AI Testing

Testing AI agents and autonomous workflows

Automated Test Framework (ATF) Architecture

Pre-built Test Step Library

Benefits of ATF

- 90% reduction in testing effort
- Faster release cycles
- Higher quality deployments
- Regression testing automation
- Consistent test execution
- Documentation of expected behaviour

ATF Evolution

- Istanbul: Initial ATF release
- Subsequent releases: Enhanced capabilities
- Yokohama: Reusable Tests feature
- Continuous improvements in each release
- Growing test step library
- AI-powered test generation

Module 3: Advanced ATF Techniques & Module 4: AI-Powered Testing

1

Reusable Tests

Modular test components reducing duplication and improving maintainability

2

JavaScript Coding

Custom test logic for complex scenarios exceeding pre-built steps

3

Complex Workflows

Testing approvals, routing, SLAs, and multi-step processes

4

Workspace Testing

Handling workspace URLs and navigation patterns

Testing Complex Scenarios

Testing workflows and Flow Designer requires validating approval chains, task routing, and process completion. Test steps can trigger flows, wait for completion, and verify outcomes. Service Catalog testing involves validating catalog items, variables, order guides, and fulfilment workflows. SLA testing verifies start, pause, and stop conditions work correctly,

AI-Powered Testing with Now Assist

Benefits of AI-Powered Testing

- Reduced test creation time
- Accessibility for non-technical users
- Intelligent test maintenance
- Improved test coverage
- Faster identification of issues
- Continuous test optimisation

Writing Effective Prompts

- Be specific about scenario
- Describe expected behaviour
- Include validation criteria
- Specify test data requirements
- Mention edge cases
- Provide context about process

Module 5: AI Agent Testing & Module 6: Performance Testing

01

Validate Understanding

Test AI agent's ability to interpret user requests correctly

02

Verify Reasoning

Confirm agent selects appropriate actions based on context

03

Test Execution

Validate agent executes workflows and updates records correctly

04

Assess Learning

Verify agent improves performance based on feedback

Testing Agentic Workflows and Knowledge

Agentic workflows combine autonomous decision-making with structured processes, requiring testing that validates both aspects. Testing supervised versus autonomous execution modes ensures agents behave appropriately based on configuration—supervised mode requires human approval for actions, whilst autonomous mode executes independently. Verifying AI agent tool integrations confirms agents can access necessary systems and data sources. Optimising knowledge base data for AI Search ensures agents retrieve relevant information efficiently. Monitoring performance with analytics dashboards provides insights into agent effectiveness, user satisfaction, and areas for improvement. Understanding these testing approaches ensures AI agents deliver reliable, high-quality service.

Performance Testing in ServiceNow

Performance Testing Tools

- Performance Profiling: Built-in ServiceNow capability
- Gatling: Scalable load testing framework
- BlazeMeter: Cloud-based performance testing
- Tricentis NeoLoad: ServiceNow-specific testing

Performance Metrics

- Response time: Time to complete operations
- Throughput: Operations per second
- Resource utilisation: CPU, memory, database
- Error rate: Failed operations percentage

Module 7: Integration Testing & Module 8: CI/CD Integration

REST API Testing Validating endpoints, authentication, payloads, and responses	**Data Integrity** Ensuring information transfers correctly without corruption
Transform Maps Testing import sets and data transformation logic	**MID Server** Validating connectivity and integration functionality

Testing Transform Maps and Third-Party Systems

Transform maps convert external data formats into ServiceNow records, requiring thorough testing to ensure accuracy. Testing involves validating field mappings, data type conversions, and error handling for invalid data. Import sets provide staging areas for external data, enabling validation before loading into production tables. Third-party system integrations require testing connectivity, authentication, data exchange, and error handling. MID Server connectivity validation ensures the integration infrastructure functions correctly, as MID Servers facilitate communication between ServiceNow and on-premises systems. Error handling and retry mechanisms must be tested to ensure integrations recover gracefully from transient failures.

CI/CD and DevOps Integration

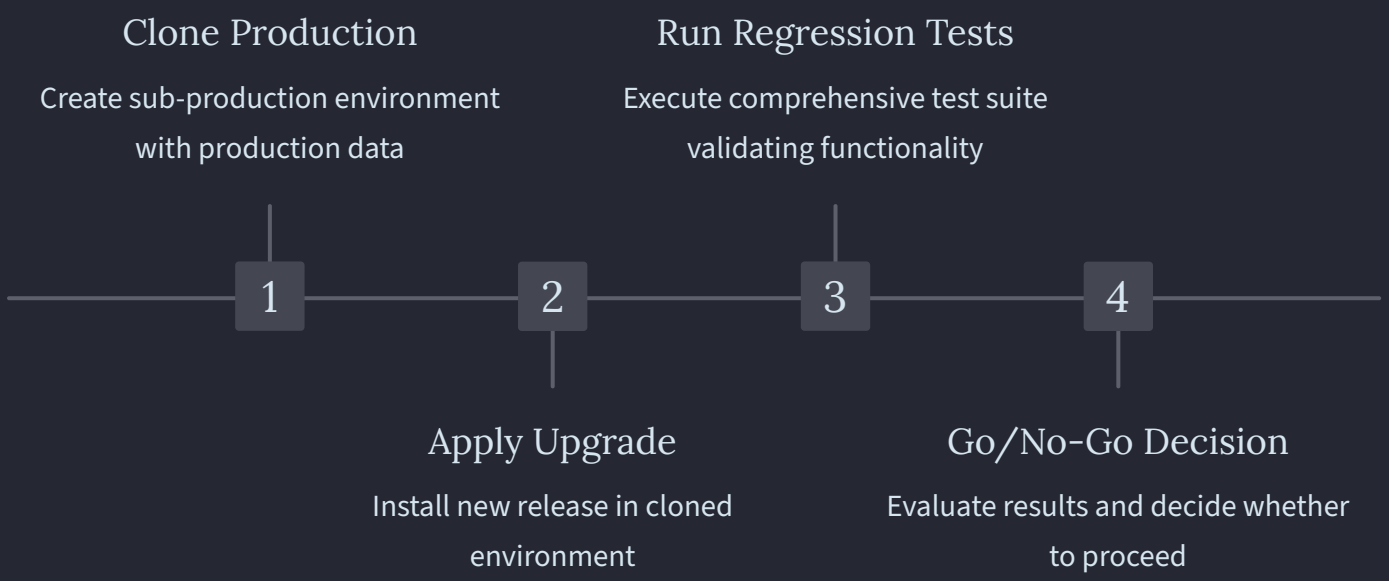
Automated Test Triggers

CI/CD pipelines trigger automated tests on code commits, pull requests, or scheduled intervals. This automation ensures continuous quality validation, catching issues immediately rather than discovering them during manual testing cycles. Test results gate deployments, preventing broken code from reaching production.

Pipeline Visibility

Modern CI/CD tools provide dashboards showing pipeline status, test results, and deployment history. This visibility enables teams to identify issues quickly, understand deployment status, and maintain confidence in release quality. Traceability links code changes to test results and deployments, supporting audit and compliance requirements.

Module 9: Upgrade Testing & Module 10: UAT



Upgrade Testing Process

Effective upgrade testing follows a structured process. Cloning production to sub-production environments provides realistic testing conditions with actual data and configurations. Instance backup and rollback strategies ensure recovery options if issues arise. Running regression test suites validates that existing functionality continues working after the upgrade. Test results inform go/no-go decisions—if critical issues are discovered, the upgrade is postponed until fixes are implemented. If testing passes, the upgrade proceeds to production with confidence. Understanding this process ensures smooth upgrades that minimise disruption and maintain service quality.

User Acceptance Testing (UAT)

UAT Execution

- Prepare testers with training and documentation
- Execute test scenarios systematically
- Capture test evidence with screenshots
- Document actual results versus expected
- Record pass/fail for each scenario
- Manage defects through resolution

UAT Completion

- Review all test results
- Ensure critical scenarios pass
- Resolve or accept known issues
- Obtain business sign-off
- Document acceptance criteria met
- Prepare for production deployment

SECTION 7

ServiceNow Integrations

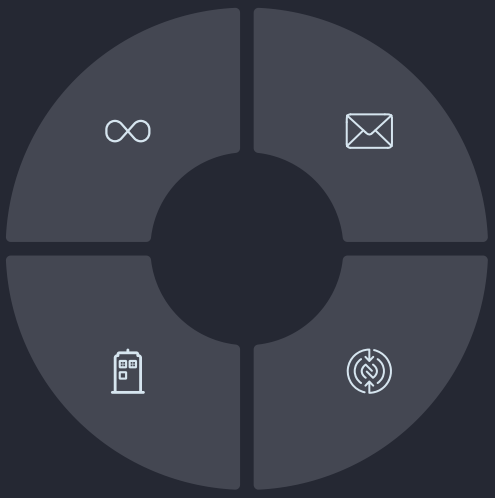
Module 1: Integration Fundamentals & Module 2: REST API Integration

Inbound Integration

External systems sending data into ServiceNow for processing and storage

Real-time Integration

Immediate data exchange as events occur, ensuring current information



Outbound Integration

ServiceNow sending data to external systems for consumption or processing

Bidirectional Integration

Two-way data exchange maintaining synchronisation between systems

Integration Components and Data Formats

Understanding integration components is essential for building effective integrations. Endpoints represent the URLs or addresses where systems communicate, defining where requests are sent and responses are received. Payloads contain the data being exchanged, structured according to agreed formats. Headers provide metadata about requests and responses, including authentication tokens, content types, and custom parameters.

REST API Integration

Error Handling

Robust integrations handle errors gracefully, implementing retry logic for transient failures, logging errors for troubleshooting, and providing meaningful error messages to users. Understanding HTTP status codes enables appropriate error handling—2xx indicates success, 4xx indicates client errors, and 5xx indicates server errors.

Performance Optimisation

Integration performance impacts user experience and system load. Rate limiting prevents overwhelming external systems with too many requests. Caching reduces redundant requests by storing frequently accessed data. Pagination retrieves large datasets in manageable chunks. Understanding these techniques enables building performant integrations.

Module 3: ServiceNow REST API & Module 4: GraphQL Integration

Table API

CRUD operations for ServiceNow tables with query capabilities

Import Set API

Bulk data loading with staging and transformation

Aggregate API

Calculations and summary statistics without large data transfers

Scripted REST APIs

Custom endpoints with specialised business logic

Real-World ServiceNow API Use Cases

Understanding real-world applications demonstrates API value. Employee onboarding automation integrates HR systems with ServiceNow, automatically creating user accounts, assigning equipment, and provisioning access when new employees join

GraphQL Integration

GraphQL in ServiceNow

ServiceNow supports GraphQL through the GraphQL API, enabling efficient data retrieval for complex queries. The GraphiQL Explorer tool provides an interactive interface for building and testing GraphQL queries, with autocomplete and documentation. Understanding GraphQL syntax enables building sophisticated queries that retrieve exactly the needed data.

Real-time Data with Subscriptions

GraphQL subscriptions enable real-time data updates, pushing changes to clients as they occur. This capability is ideal for dashboards, monitoring tools, and collaborative applications requiring current information. Mobile app optimisation benefits from GraphQL's efficient data retrieval, reducing bandwidth usage and improving performance.

Module 5: Integration Hub

Microsoft Teams

Send messages, create channels, manage teams, and integrate collaboration workflows

Slack

Post messages, create channels, manage users, and automate team communication

JIRA

Create issues, update tickets, manage projects, and synchronise development workflows

AWS

Manage EC2 instances, S3 storage, Lambda functions, and cloud infrastructure

Azure

Provision resources, manage virtual machines, and integrate Microsoft cloud services

Salesforce

Synchronise accounts, contacts, opportunities, and customer relationship data

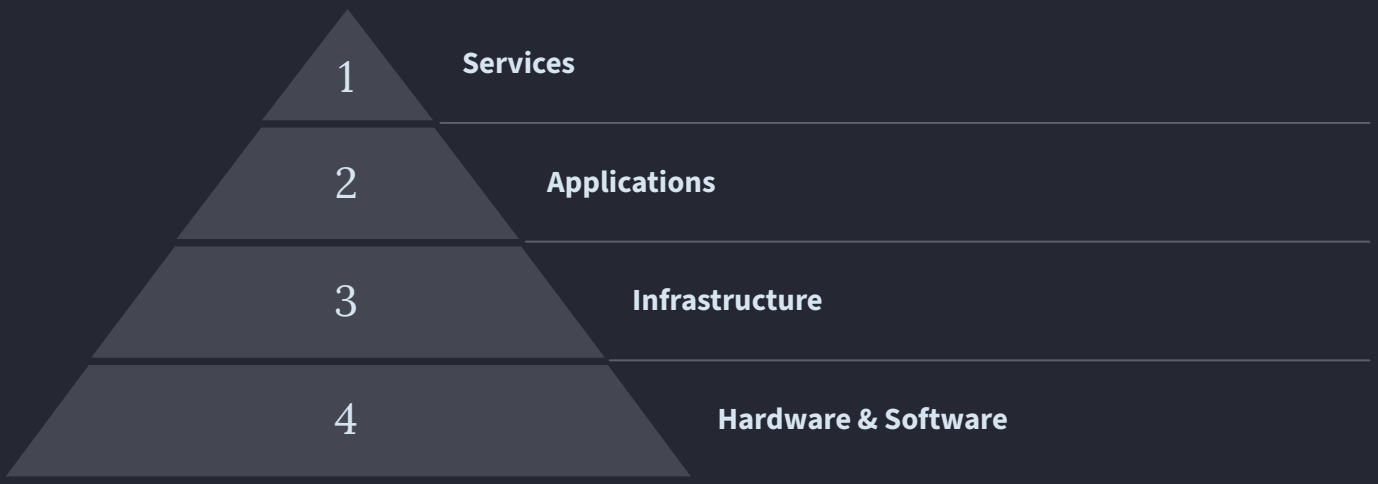
Spokes, Actions, and Connection Management

Understanding Spokes and Actions is essential for leveraging Integration Hub effectively. Spokes represent collections of related actions for specific systems—the Microsoft Teams Spoke contains actions for sending messages, creating channels, and managing teams. Actions represent individual operations like "Send Message" or "Create Channel". Pre-built spokes cover popular systems including collaboration tools (Microsoft Teams, Slack), project management (JIRA, ServiceNow), cloud platforms (AWS, Azure), and CRM systems (Salesforce). Connection and credential management secures integration authentication, storing credentials centrally and enabling reuse across multiple flows. Creating custom actions extends Integration Hub capabilities for organisation-specific requirements, encapsulating specialised logic in reusable components.

SECTION 8

ServiceNow ITSM

Module 1: CMDB & Configuration Management & Module 2: CSDM



Configuration Item Relationships and Discovery

Configuration Item relationships enable understanding how infrastructure components connect and depend on each other. Dependency mapping reveals which services rely on which infrastructure, enabling impact analysis when components fail or require maintenance. Visual Service Maps provide graphical representations of service dependencies, showing how business services connect to applications, which connect to servers, which connect to network devices.

Common Service Data Model (CSDM)

Service Modelling Best Practices

- Start with business services
- Map to supporting applications
- Connect to infrastructure components
- Maintain accurate relationships
- Regular data quality reviews
- Align with CSDM standards

CSDM Maturity Assessment

The CSDM Data Foundations Dashboard provides visibility into CSDM implementation maturity, showing which domains are populated, data quality metrics, and recommendations for improvement. Understanding maturity levels enables planning CSDM adoption and measuring progress.

Module 3: Service Portfolio Management & Module 4: Incident Management

01

Service Definition

Define service capabilities, features, and target customers

02

Service Design

Design service delivery, support model, and technical architecture

03

Service Transition

Implement service, train staff, and prepare for launch

04

Service Operation

Deliver service, monitor performance, and support users

05

Continual Improvement

Analyse metrics, gather feedback, and enhance service

Service Commitments and Cost Management

Service commitments define what customers can expect from services, including availability targets, response times, and support hours. SLAs (Service Level Agreements) formalise these commitments, providing measurable targets and consequences for non-compliance. Cost management and pricing models enable understanding service costs, allocating expenses appropriately, and pricing services to recover costs or generate profit. Performance tracking and analytics measure service delivery against commitments, identifying trends and opportunities for improvement. Integration with ITSM processes ensures service portfolio information informs incident, problem, and change management.

Incident Management

SLA Management

SLA definitions specify response and resolution time targets based on priority. SLA tracking monitors time remaining, alerting teams when targets are at risk. Breach notifications escalate incidents approaching or exceeding SLA targets, ensuring appropriate attention.

Major Incident Management

Major Incident Management procedures handle high-impact incidents requiring coordinated response. VIP user handling ensures executives and critical users receive priority support. Incident categorisation strategies enable consistent classification, supporting reporting and trend analysis.

Module 5: Problem Management & Module 6: Change Management

1

Problem Identification

Detect recurring incidents or patterns indicating underlying issues

2

Root Cause Analysis

Investigate problems using structured methodologies to identify causes

3

Known Error Documentation

Document problems with workarounds in Known Error Database

4

Permanent Resolution

Implement fixes through change management to prevent recurrence

Root Cause Analysis and Known Errors

Root Cause Analysis (RCA) methodologies provide structured approaches to identifying problem causes. Common techniques include the Five Whys (asking "why" repeatedly to drill down to root causes), Fishbone Diagrams (visualising potential causes across categories), and Fault Tree Analysis (mapping logical relationships between failures). The Known Error Database manages problems with identified root causes and documented workarounds, enabling support teams to resolve related incidents quickly whilst permanent fixes are implemented.

Change Management

Risk Assessment

Risk assessment evaluates change impact and likelihood of issues, informing approval decisions. Risk conditions define factors that increase risk—changes to critical systems, changes during peak hours, or changes without testing. Risk calculation and scoring combine multiple factors into overall risk ratings, enabling consistent evaluation.

Change Advisory Board

The Change Advisory Board (CAB) reviews and approves changes, providing governance and coordination. The CAB workbench provides a unified interface for reviewing pending changes, viewing risk assessments, and making approval decisions. CAB meeting management coordinates regular reviews, ensuring timely decisions.

Module 7: Knowledge Management & Module 8: Service Catalog

Creation

Author articles with clear, accurate content

Review

Validate accuracy and completeness

Retirement

Remove outdated or obsolete content

Approval

Obtain stakeholder approval for publication

Maintenance

Update articles as information changes

Publication

Make articles available to users

Service Catalog & Request Management

Variable Types and Sets

Variable types define input fields for catalog items—text, numbers, dates, choices, references, and more. Variable sets group related variables for reuse across multiple catalog items, promoting consistency and reducing duplication. Understanding variable configuration enables building intuitive request forms that gather necessary information whilst maintaining usability.

Catalog Builder

Catalog Builder provides a visual interface for business users to create and modify catalog items without technical knowledge. This capability democratises catalog management, enabling service owners to maintain their offerings directly. Request fulfilment workflows automate service delivery, routing requests to appropriate teams and tracking progress.

Request Management and Analytics

Approval routing configuration defines who must approve requests based on cost, risk, or other factors, ensuring appropriate governance. Shopping cart functionality enables users to request multiple services simultaneously, improving efficiency. Portal customisation tailors the catalog interface to organisational branding and user preferences. Employee Center integration provides a modern, consumer-grade experience for service requests. Request analytics and reporting track request volume, fulfilment time, and user satisfaction, providing insights for service improvement.

Module 9: Now Assist for ITSM & Module 10: Integration and Reporting

Incident Management with AI

AI-powered summarisation, resolution notes, categorisation, and similar incident detection

Virtual Agent Enhancements

Conversational AI, natural language understanding, context-aware responses, sentiment analysis

Knowledge with AI

AI-powered article generation, automated updates, gap detection, search understanding

Case Management with AI

Automated summarisation, next-best-action recommendations, workload prediction, agent assist

Virtual Agent and Knowledge Enhancements

Case Management with AI

Automated case summarisation provides concise overviews of case history and current status. Next-best-action recommendations suggest appropriate steps based on case details and historical patterns. Workload prediction forecasts case volume and complexity, enabling resource planning. Agent assist in real-time provides suggestions and information whilst agents work cases. Performance optimisation suggestions identify process improvements based on case data analysis.

Predictive Intelligence

Predictive categorisation suggests categories based on incident descriptions. Priority prediction recommends priorities based on impact and urgency indicators. Assignment prediction suggests appropriate assignment groups or individuals. Resolution time estimation predicts how long incidents will take to resolve. Trend analysis and forecasting identify patterns and predict future incident volume.

SECTION 9

ServiceNow HRSD

Module 1: HRSD Overview & Module 2: Manage & Secure HR Access

Employee Self-Service

Intuitive portals enabling employees to access HR services, submit requests, and find information

HR Case Management

Unified case management for all HR inquiries, requests, and issues with intelligent routing

Employee Journey

Guided experiences for onboarding, transitions, and offboarding with automated tasks

HR Knowledge

Comprehensive knowledge base with policies, procedures, and FAQs for self-service

HR Organisational Structure and Skills

Security Best Practices

- Implement least privilege access
- Separate HR and IT roles
- Encrypt sensitive data
- Audit access regularly
- Train users on security
- Monitor for anomalies

Compliance Considerations

- GDPR data protection requirements
- Employment law compliance
- Data retention policies
- Right to be forgotten
- Audit trail maintenance
- Privacy by design principles

Module 3: HRSD Configure & Module 4: HR Case Management

01

Case Creation

Employees or HR create cases through multiple channels

02

Case Routing

Intelligent assignment to appropriate HR teams or specialists

03

Case Resolution

HR professionals investigate, resolve, and document cases

04

Case Closure

Verification, feedback collection, and case closure

Agent Workspace and Employee Relations

Agent Workspace for HR Case Management provides a unified interface for HR professionals to manage cases efficiently. The workspace includes case lists, forms, related information, and productivity tools. Employee Relations functionality handles sensitive employee issues including grievances, investigations, and disciplinary actions. Anonymous Report Center enables employees to report concerns confidentially, with protections ensuring anonymity whilst enabling investigation. I18N (internationalisation) and localisation considerations ensure HRSD works effectively for global organisations, supporting multiple languages, date formats, and cultural norms.

HR Case Management

Case Assignment Options

Assignment Rules route cases based on conditions like topic, location, or employee attributes. Matching Rules consider agent skills, availability, and workload. Advanced Work Assignment provides sophisticated routing with real-time workload balancing, skill matching, and capacity management. Understanding these options enables choosing appropriate assignment methods for different case types.

Case Lifecycle

HR cases progress through states from New to In Progress to Resolved to Closed. State transitions can be automated based on conditions or require manual progression. Understanding case lifecycle enables designing workflows that balance automation with human judgment, ensuring cases receive appropriate attention whilst maintaining efficiency.

Module 5: Employee Engagement & Module 6: Journey Management

HR Knowledge

Self-service knowledge base with policies, procedures, and FAQs

HR Catalog

Service catalog for HR requests with intuitive request forms

Learning Posts

Embedded training and development content

Listening Posts

Employee feedback collection at key journey moments

HR Catalog and Service Requests

HR Catalog major components include catalog items representing services employees can request, variables collecting necessary information, and workflows automating fulfilment. Catalog item creation using Manage HR Catalog provides a visual interface for building request forms without coding. Variables and variable sets define input fields—text, numbers, dates, choices, references—with validation ensuring data quality. Record producers enable creating records in any table through catalog-like interfaces, extending self-service beyond traditional HR requests. Understanding these components enables building comprehensive HR service catalogs that improve employee experience whilst maintaining appropriate controls.

Employee Journey Management

Journey Best Practices

- Start with critical lifecycle events
- Involve stakeholders in journey design
- Balance automation with personal touch
- Gather feedback and iterate
- Measure journey effectiveness
- Maintain journey templates regularly

Common Journey Scenarios

- New hire onboarding
- Role changes and promotions
- Manager transitions
- Leave of absence
- Return from leave
- Offboarding and exit

Module 7: HR Integrations & Module 8: Track HR Progress

1

ESM Integrations Framework

Standardised framework for building and managing integrations

2

Spokes and Capabilities

Pre-built connectors for popular HR systems and platforms

3

Sources and Web Services

Connection definitions and authentication configuration

4

Schema Mappings

Field mappings between ServiceNow and external systems

Track HR Progress

HR dashboards provide visibility into service delivery performance and employee engagement. The Overview Dashboard presents high-level metrics for HR leadership. The Manager Dashboard shows team-specific information for people managers. The Agent Dashboard displays case workload and performance for HR professionals. Case Spotlight highlights critical cases requiring attention.

HR Surveys and Pulse Analytics

Key HR Metrics

- Case resolution time
- First-contact resolution rate
- Employee satisfaction scores
- Knowledge article usage
- Journey completion rates
- Service catalog adoption

Analytics Best Practices

- Define clear success metrics
- Establish baseline measurements
- Track trends over time
- Segment data by relevant dimensions
- Share insights with stakeholders
- Use data to drive improvement

Module 9: HR Now Assist & Module 10: HR Reports and Analytics

40%	85%	95%
Reduction in Case Handling Time	Self-Service Resolution Rate	Categorisation Accuracy
AI-powered suggestions and automation reduce time spent on routine cases	Virtual Agent and knowledge enable employees to resolve issues independently	Predictive Intelligence ensures consistent, accurate case categorisation

HR Reports and Analytics

Advanced report creation enables building sophisticated reports that provide insights into HR service delivery, employee engagement, and workforce trends. Custom dashboards present relevant metrics for different audiences—executives need high-level summaries, managers need team-specific information, and HR professionals need operational details. Performance Analytics content packs provide pre-built analytics for common HR scenarios, accelerating implementation. KPI definition and tracking establish key performance indicators aligned with HR strategy, measuring progress towards goals. Trend analysis and forecasting identify patterns in HR data, enabling proactive planning and resource allocation. Understanding these capabilities enables building comprehensive analytics programmes that inform HR decision-making and demonstrate service delivery value.

Common HR Reports

- Case volume and trends
- Resolution time by category
- Employee satisfaction scores
- Knowledge article effectiveness
- Journey completion rates
- Service catalog adoption
- Agent productivity metrics

Advanced Analytics

- Predictive case volume forecasting
- Sentiment analysis from feedback
- Root cause analysis for trends
- Workforce planning insights
- Service delivery benchmarking
- ROI calculation for HRSD
- Employee experience scoring

SECTION 10

ServiceNow CSM

Module 1: Introduction to CSM & Module 2: Business Models

B2C (Business-to-Consumer)

Consumer records, households, consumer addresses, and Consumer Service Portal

B2B (Business-to-Business)

Customer Accounts, contacts, account hierarchy, and Business Portal

B2B2C & B2B2E

Multi-level data models with Account Consumer relationships

Configurable Portals

Customisable portal widgets and Engagement Messenger

Business Models

Business models and customer data models overview reveals how CSM supports different business types. B2C (Business-to-Consumer) model serves individual consumers with Consumer records storing personal information, Consumer Users enabling portal access, Households grouping family members, and Consumer addresses tracking locations. Consumer Service Portal access and self-registration enable consumers to access services independently. B2B (Business-to-Business) model serves corporate customers with Customer Accounts representing companies, Contacts representing individuals within accounts, Account hierarchy showing parent-subsidiary relationships, and Account addresses tracking business locations. Business Portal access and self-registration enable business customers to access services. B2B2C & B2B2E models support complex scenarios where organisations serve consumers through intermediaries or serve employees of business customers. Multi-level data models and Account Consumer relationships enable modelling these complex structures. Configurable Portal widgets enable customising portal experiences for different customer types. Engagement Messenger provides modern chat interfaces for customer interactions.

Module 3: Interaction & Module 4: Products and Services

Email

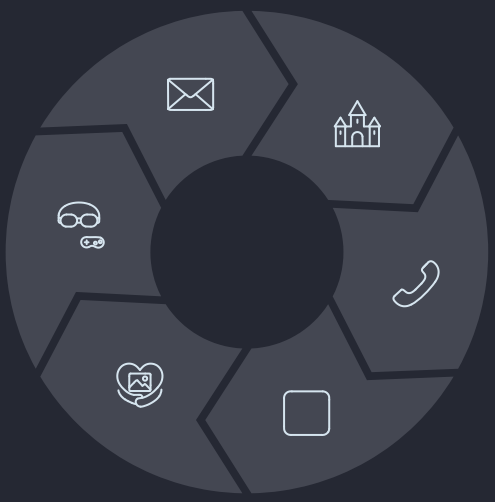
Email-to-Interaction process flow

Virtual Agent

AI-powered conversational support

Social Media

Social media monitoring and response



Chat

Agent Chat and Engagement Messenger

Phone

Voice integration and call logging

Portal

Self-service portal interactions

Virtual Agent and Chat Capabilities

Virtual Agent with NLU (Natural Language Understanding) provides AI-powered self-service, understanding customer questions and providing relevant answers or creating cases. Agent Chat capabilities enable real-time text conversations between customers and agents. CSM Configurable Workspace for live agents provides a unified interface for managing interactions, cases, and customer information. Public and private chat conversations enable agents to communicate with customers (public) or consult with colleagues (private) within the same interface. Quick actions and Conversation Autopilot automate routine responses and actions, improving agent efficiency. Response Templates provide pre-written responses for common questions, ensuring consistency and speed.

Products and Services

Product Model Benefits

- Accurate entitlement verification
- Proactive product support
- Warranty tracking
- Upgrade and renewal management
- Product performance analytics
- Customer asset visibility

Entitlement Workflows

Entitlement workflows automate entitlement verification, ensuring cases receive appropriate priority and service levels based on customer contracts. Understanding entitlement configuration enables building sophisticated service delivery models that balance customer commitments with operational efficiency.

Module 5: Case & Module 6: Case Initiation

Case Creation Multiple channels and methods for creating cases	**Case Routing** Intelligent assignment to appropriate teams or agents
Case Work Investigation, resolution, and customer communication	**Case Closure** Resolution verification and feedback collection

Case Types and Configuration

Case Types definition and architecture enable creating specialised case handling for different scenarios. When to use Case Types depends on whether different case categories require distinct workflows, fields, or SLAs. Case Type configuration involves defining case type attributes, workflows, and forms. Case Type hierarchy enables inheritance, where child case types inherit properties from parents whilst adding specialised attributes. Understanding Case Types enables building flexible case management that adapts to diverse customer service scenarios. Case creation and work activity tracking provide visibility into case progress. Form Header configuration displays key case information prominently. Form Ribbon configuration provides quick access to common actions. Contact Time Zone and Follow the Sun features enable global service delivery, routing cases to appropriate regions based on time zones.

Routing and Assignment

Routing and Assignment determine how cases reach appropriate agents. Manual versus automatic routing options balance control with efficiency—manual routing enables supervisors to assign cases based on judgment, whilst automatic routing uses rules to assign cases immediately. Advanced Work Assignment (AWA) provides sophisticated routing with workload balancing, skill matching, and capacity management. AWA components and configuration include work queues, assignment rules, and agent capacity settings. Agent Affinity improves customer experience by routing cases to agents who previously helped the customer (Historical), worked related cases (Related task), or are part of the customer's account team (Account Team). Case Initiation as an Agent involves agents creating cases on behalf of customers during interactions. Service Definitions and Case Types determine case structure and workflow. Service Selector and Case Type Selector enable agents to choose appropriate case types. Case through Service Catalog enables customers to create cases through self-service catalog items.

Module 7: Case in Progress & Module 8: Case Resolution and Closure

1

Case Tasks

Breaking complex cases into manageable work items with assignments

2

Action Status

Tracking case progress and identifying blocking issues

3

Case Summaries

Communicating case status and actions to stakeholders

4

Escalations

Elevating cases requiring management attention or intervention

Case Summaries and Escalations

Case Summaries provide concise overviews of case status, actions taken, and next steps. Case Action Summary creation involves documenting key information and decisions. Publication and notification options enable sharing summaries with customers, managers, or other stakeholders. Escalations elevate cases requiring management attention, additional resources, or policy exceptions. Escalation benefits and outcomes include faster resolution for critical issues, management visibility into service delivery challenges, and customer satisfaction improvement. Case and Account escalations distinguish between escalating individual cases versus escalating all cases for specific accounts. Escalation process flow defines how escalations are initiated, reviewed, and resolved. De-escalation process returns cases to normal handling once escalation reasons are addressed. Roles for managing escalations define who can escalate cases, review escalations, and approve resolutions.

Major Case Management

Major Issue Management (MIM) handles high-impact cases affecting multiple customers or critical services. Benefits and outcomes include coordinated response, stakeholder communication, and faster resolution. Major Case definition establishes criteria for major case designation—typically based on customer count, business impact, or service criticality.

Module 9: Now Assist for CSM & Module 10: CSM Analytics

Case Summarisation

AI-generated summaries of case history and communications

Response Suggestions

Intelligent recommendations for customer responses

Similar Cases

Identification of previously resolved similar issues

Sentiment Analysis

Detection of customer emotions and intent

Automation and Configuration

Automation and Productivity capabilities streamline routine tasks and improve efficiency. Automated case categorisation and routing use AI to suggest appropriate categories and assignment groups based on case descriptions. AI-powered workflow automation identifies opportunities to automate routine processes, reducing manual effort.

CSM Analytics

Key CSM Metrics

- Case resolution time
- First-contact resolution rate
- Customer satisfaction scores
- SLA compliance percentage
- Agent utilisation rates
- Case backlog trends
- Channel distribution

Advanced Analytics

- Predictive case volume forecasting
- Sentiment trending from surveys
- Root cause analysis for escalations
- Product quality insights
- Agent performance benchmarking
- Customer lifetime value analysis
- Service delivery ROI calculation

CONGRATULATIONS!

Your ServiceNow Journey Begins

10

Major Sections Completed

Platform Admin, Development, Scripting, UI Builder, IDE, Testing, Integrations, ITSM, HRSD, CSM

59

Comprehensive Modules

From fundamentals through advanced topics, covering every aspect of ServiceNow

100%

AI-Native Training

Integrated AI agent development and Now Assist capabilities throughout

Your Next Steps

01

Apply Your Knowledge

Start building real-world applications, implementing what you've learned in practical scenarios

02

Pursue Certification

Validate your expertise with ServiceNow certifications—CSA, CAD, CIS, and specialised credentials

03

Join the Community

Connect with other ServiceNow professionals, share knowledge, and continue learning

04

Build Your Career

Leverage your AI-native ServiceNow skills to advance your career and lead digital transformation

> "The future belongs to those who can build AI agents, work alongside them, and lead teams that deploy them. You're now equipped to be that leader. Welcome to the Agentic AI future of work."
>
> — Digital Edify Team

Stay Connected

Your learning journey doesn't end here. Digital Edify provides ongoing support, advanced training, and career services to help you succeed. Join our community of 100,000+ professionals who are building the future of work with AI-native skills. Together, we're creating 1 million AI-native professionals who will shape the Agentic Era.

Thank You

Your Future Starts Now

Thank you for choosing Digital Edify for your ServiceNow training journey. You've invested in yourself, your career, and your future. The skills you've gained position you at the forefront of the AI revolution in enterprise technology. As organisations worldwide adopt ServiceNow and integrate AI agents into their workflows, your expertise will be in high demand. You're not just trained—you're transformed into an AI-native professional ready to lead in the Agentic Era.

Remember, this is just the beginning. The ServiceNow platform continues evolving, AI capabilities expand with each release, and new opportunities emerge constantly. Stay curious, keep learning, and embrace the exciting challenges ahead. You have the foundation, the skills, and the support to achieve remarkable things. Go forth and build the future.

> "The best way to predict the future is to create it. You now have the tools to create an AI-powered future of work."

From all of us at Digital Edify, congratulations on completing this comprehensive programme. We're proud of your achievement and excited to see what you'll build. Welcome to the community of AI-native ServiceNow professionals. Your journey has just begun, and the possibilities are limitless.

Digital Edify
India's #1 Training Institute for the AI Era
Building 1 Million AI-Native Professionals