Custom Application Development (CAD) Process

The **Custom Application Development Process** is a structured, flexible, and iterative approach for developing enterprise-grade applications on **Web and Mobile platforms**. This process has been designed based on experiences from handling many such projects in open source environment. This CAD method can handle projects with **multiple modules**, each containing **multiple features** that are developed in distinct **phases**. The process promotes **collaboration with clients** throughout development, allowing for incremental feature releases and iterative enhancements. A key aspect of this process is its ability to **accommodate controlled scope creep** during the **User Acceptance Testing (UAT) phase**, recognizing that end-users may discover new tasks or system requirements during usage.

This process provides a **clear roadmap** from initiation to closure, ensuring that every phase, activity, and deliverable is well-defined. By enabling continuous stakeholder engagement and collaboration, it aims to achieve better alignment with client expectations and ensure the successful delivery of high-quality, custom-built software.

When Should This Process Be Used?

The Custom Application Development Process is best suited for **large, multi-module projects** that involve the following characteristics:

- **Modular Development**: When a project requires the development of **multiple modules** with distinct but interrelated features.
- **Collaborative Development**: When **client involvement is critical** for capturing business rules, functional requirements, and iterative feedback.
- **Phased Delivery**: When an application is developed and delivered in **multiple releases** or **iterations**.
- **Evolving Requirements**: When there is a possibility of **scope creep** during the UAT phase due to users discovering new tasks during system usage.
- **Custom-Built Applications**: When off-the-shelf solutions are not suitable, and a fully customized solution is required.

This process is typically used in large, enterprise-grade development initiatives where **client engagement is ongoing** and the ability to adapt to new requirements is crucial for success.

Key Players Involved

Several key players are responsible for executing this process and ensuring successful delivery. These roles include:

Role	Description of Responsibilities
Delivery Manager (DM)	Oversees the end-to-end delivery of the project, ensuring that timelines, budgets, and quality standards are met. Manages resources,

	coordinates teams, and ensures smooth communication across all phases.
Business Analyst (BA)	Works closely with stakeholders to gather, document, and analyse business requirements. Defines user stories, develops wireframes and creates functional specifications, facilitates communication between the technical team and clients, and ensures the product aligns with business needs.
BA Manager	Manages a team of business analysts, ensuring alignment with business goals, and coordinating the execution of BA activities across projects. Provides mentorship, reviews BA deliverables and maintains consistency in documentation and processes.
Architect	Defines the high-level system design and architecture, ensuring it meets functional and non-functional requirements. Ensures scalability, security, and performance across the solution.
Tech Lead (TL)	Leads the development team for one or multiple modules, ensuring that best practices are followed, and providing guidance and reviews on the architecture, design, and implementation of features. Coordinates with the team to ensure technical alignment with the project goals.
Developer	Designs, develops, unit tests, and debugs software components based on the specifications provided in FSDs. Writes clean, maintainable code and collaborates with the team on system design.
UI Developer	Focuses on the development of the user interface (UI), ensuring a responsive, user-friendly design. Works closely with UX/UI designers and integrates the front-end components with back-end services.
Test Manager	Leads the testing efforts, plans and manages testing activities, and ensures that the testing process is aligned with project goals. Manages test resources, defines testing strategies & plans, and ensures coverage of all requirements.
Tester	Executes test cases, logs defects, and ensures the quality of the product. Works closely with developers to verify that code changes meet the required functionality and that defects are resolved.
UAT Tester	Works with the client to test the product in a user acceptance testing (UAT) environment. Ensures the product meets the client's expectations, business requirements, and operational needs.
DevOps Lead	Manages the continuous integration/continuous deployment (CI/CD) pipeline, automating the process of code deployment. Oversees infrastructure management and ensures the smooth operation of cloud or on-premise environments.
Learning & Development Team (L&D Team)	Develops training materials and user manuals, provides training to the team and clients, ensuring they have the skills needed for the tools, technologies, and systems involved. Supports both technical and non-technical training throughout the project lifecycle.
Product Owner	Represents the client's interests, manages the product backlog, and defines the priorities for development. Ensures that the development team delivers value to the business and that the product aligns with the client's vision.

Client SMEs	Subject Matter Experts (SMEs) from the client side who provide domain expertise, answer questions about business rules, and validate product functionality to ensure alignment with business needs.
Client Owner	The main point of contact for the client side, who is responsible for use of the product in live environment, making high-level decisions regarding business functionality and priorities, and ensuring that the product is aligned to the business needs and expectations.

Each role plays a specific part in ensuring that activities are performed as planned, client expectations are met, and deliverables are completed on schedule.

Key Deliverables From CAD Method

The Custom Application Development Process produces key deliverables at every phase of development. These deliverables serve as tangible proof of progress and ensure that development activities remain on track. The key deliverables include:

Phase	Key Deliverables
Requirements Phase	Business Requirements Document (BRD), Functional Specification Document (FSD), Requirements Traceability Matrix (RTM)
Design Phase	System Architecture Document (SAD), High-Level Design (HLD), Low-Level Design (LLD), Test Strategy & Plan, Test Scenarios, Test Cases
Development & Test Iterations	Program Code, Defect Tracker, Test Reports
UAT Iterations	UAT Defect Tracker, UAT Progress Reports
Deployment Phase	Deployment Guidelines, Release Notes
Closure Phase	User Training Materials, User Manuals
All Phases - Task Management	Task Tracker, Iteration Plan & Tracker

Here is a brief description of the various deliverables:

Deliverable	Description
RACI Matrix	A document that defines **Roles, Accountability, Consultation, and Information** for key project stakeholders.
Business Requirements Document (BRD)	Captures the **business needs, goals, and objectives** for the project, along with high-level requirements.
UI Wireframes	Visual representations of the **user interface layout** for the application, showcasing screens and navigation flows.

Functional Specification Document (FSD)	Describes the **functional features and business logic** to be implemented in the system.
Traceability Matrix Document	Tracks the relationship between requirements and corresponding test cases, ensuring all requirements are tested.
System Architecture Document (SRD)	Describes the overall **technical design, components, and infrastructure** of the system.
High-Level Design (HLD)	Describes the **major components, data flow, and high-level structure** of the application.
Iteration Plan & Tracker	A plan for the **scope and timeline of an iteration**, along with tracking progress.
Low-Level Design (LLD)	Provides a **detailed design for individual modules and components**, including data structures, logic, and workflows.
Test Strategy & Plan	A detailed document that outlines the **test strategy** including the overall approach, scope, objectives, and methods for testing the application, as well as **test plans** including the testing activities, schedule, resources, roles, and responsibilities for a particular testing phase or iteration.
Test Scenarios	Describes the high-level **testing scenarios** to be validated during testing.
Test Cases	Describes **step-by-step test instructions** to validate the system's functionality and ensure defect-free releases.
Program Code	The **source code** for the application or module being developed.
Test Case Tracker	A tool used to track the **execution status** of test cases during testing.
Defect Tracker	Tracks the status and resolution of software **defects and bugs** during development and testing.
Daily Test Progress Dashboard	Provides **real-time visibility** into the progress of testing activities, including pass/fail rates and open defects.
QA Exit Report	A summary of **QA test results**, including any open defects, test coverage, and recommendations for release.
Performance Test Report	Details the results of **performance testing**, covering load, stress, and scalability tests.
Regression Test Report	Reports on the outcomes of **regression testing** to ensure new changes do not break existing features.
Regression Test Suite	A collection of **test cases that are re-executed** to verify system stability after changes.
Deployment Guideline	Detailed instructions on how to **deploy the application** to production or other environments.
Training Materials	**User guides, reference documents, and tutorials** to support end-user training.
User Manuals	**Comprehensive manuals** to help end-users understand system functionality and usage.
Task Tracker	A **list of tasks** (such that each task can be completed in 4-hours) required to complete each phase, linked to specific iteration.

These deliverables ensure the transparency of the project, offer documentation for future reference, and maintain alignment with client goals.

Suggested Tools for CAD Method

Jira	A project management and issue-tracking tool used for managing requirements, tasks, bugs in development projects. It enables teams to plan, track, and manage development workflows.
Confluence	A team collaboration tool used for creating, organizing, and sharing project documentation, knowledge bases, and meeting notes.
Figma	A web-based design and prototyping tool used for creating UI/UX designs, wireframes, and interactive prototypes collaboratively.
TestLink	A test management tool for planning, documenting, and tracking the execution of test cases and test plans. It helps ensure thorough testing and traceability.
JUnit	A unit testing framework for Java applications, enabling developers to write and execute repeatable tests to validate code functionality.
Jenkins	An open-source automation server that supports continuous integration and continuous delivery (CI/CD), automating the build, test, and deployment process.
Bugzilla	A bug-tracking system used for managing and tracking software defects, enabling teams to log, prioritize, and resolve issues.
JMeter	A load testing tool used to simulate user traffic and test the performance, scalability, and stability of web applications and services.
Selenium	An open-source testing framework used for automating web browser interactions, enabling automated functional and regression testing of web applications.
Postman	An API testing tool used to design, test, and document APIs, making it easier to send requests, inspect responses, and automate API testing.
Appium	An open-source mobile application testing tool that automates testing of native, hybrid, and mobile web apps on Android and iOS platforms.
ProofHub	A project and team collaboration tool used for task management, file sharing, project scheduling, and real-time team communication.

These tools play a critical role in **project management**, **design and prototyping**, **test management**, **test automation**, and **continuous integration and deployment (CI/CD)**, supporting every phase of the **Custom Application Development Process**.

Benefits of Using This CAD Method

The Custom Application Development Process is designed to **balance flexibility with control**, offering several key benefits:

1. **Enhanced Client Collaboration**: Frequent touchpoints with client users, stakeholders, and business analysts ensure a continuous feedback loop, increasing the likelihood of client satisfaction.
2. **Modular and Iterative Approach**: Breaking down development into modules and iterations ensures incremental delivery, reduces risk, and enables faster realization of value.
3. **Scope Creep Management**: By allowing controlled scope adjustments during the **UAT phase**, the process increases adaptability and ensures the system meets end-user needs.
4. **Increased Transparency and Accountability**: Key deliverables, such as progress reports, BRDs, and status updates, provide a clear view of project status and ensure all stakeholders remain informed.
5. **Quality Assurance and Risk Mitigation**: Testing at every iteration, combined with defect tracking and regression testing, ensures high-quality releases and reduces the risk of issues in production.
6. **End-to-End Traceability**: The use of an **RTM (Requirements Traceability Matrix)** ensures that all requirements are traceable from requirements to design, development, and testing.

This process minimizes risk, facilitates stakeholder buy-in, and ensures alignment with business goals at every stage.

This **Custom Application Development Process** is a comprehensive, collaborative approach for multi-module, multi-phase development projects. Its flexibility, combined with strong project controls, ensures high-quality application delivery while fostering a continuous improvement mindset. By adopting the recommended process enhancements, organizations can further strengthen project outcomes and achieve faster, more reliable results.