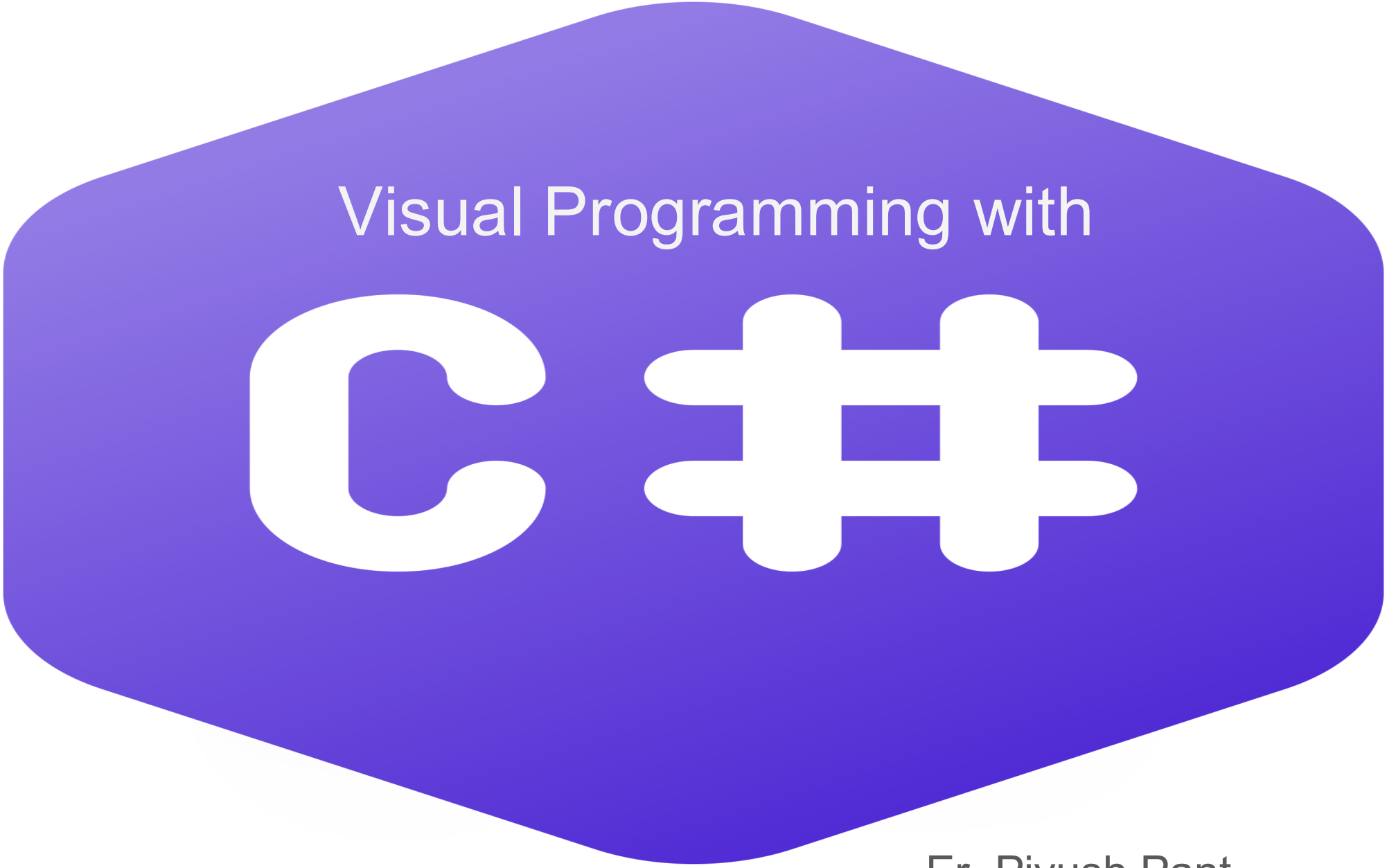
Visual Programming with C#



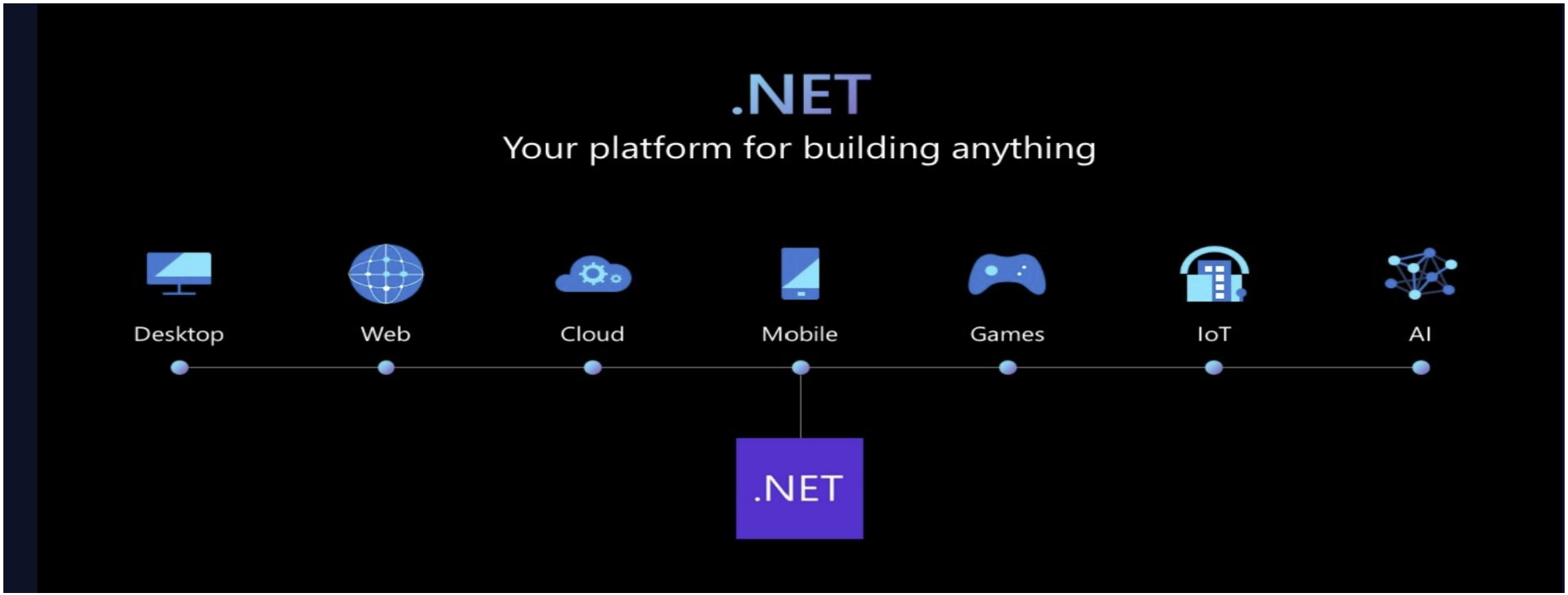
Er. Piyush Pant

Er. Piyush Pant

Senior Software Engineer (Verisk Nepal Pvt. Ltd.)
https://www.linkedin.com/in/epiyushpant/

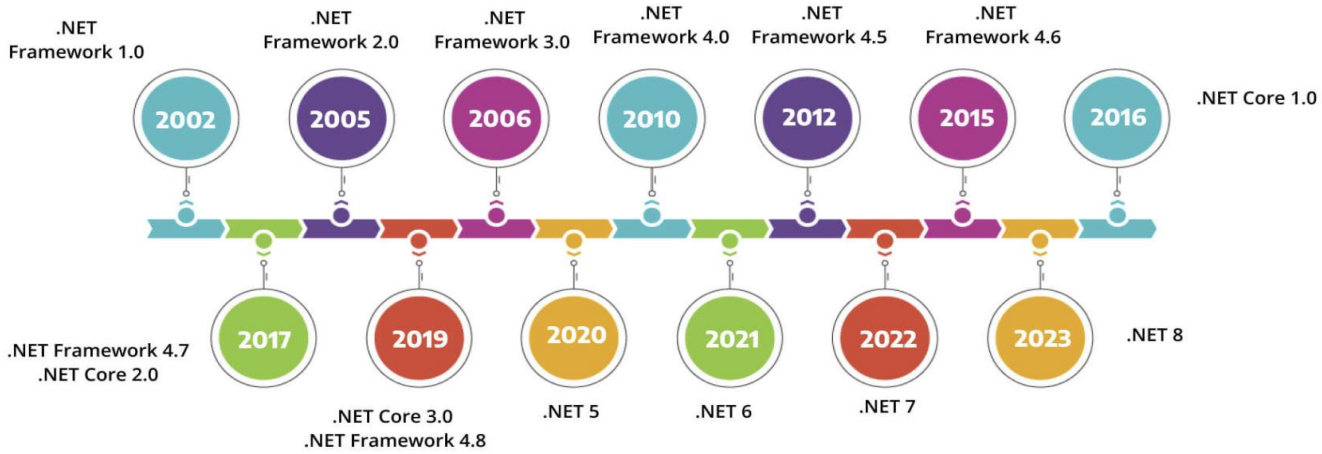
8 + Years of experience in development and implementation of .NET applications for Nepal governments and US Clients.

MSC in Computer System and Knowledge Engineering (Pulchowk Campus)


.NET
Your platform for building anything
Desktop
Web
Cloud
Mobile
Games
IoT
AI
.NET

.NET

- The .NET is a software development platform created by Microsoft.
- Free, cross platform , open source developer platform for building different types of application .
- It provides the tools and libraries needed to develop various types of applications, such as web, desktop, and mobile apps.



.NET FEATURES

Multi-Language Support : Supports multiple languages (C#, VB.NET, F#) with seamless interoperability.

Object-Oriented Programming: Promotes modular, reusable, and maintainable code.

ASP.NET for Web Development: Simplifies dynamic web application development.

Automatic Memory Management: Built-in garbage collection for better performance.

Secure Applications: Provides encryption, authentication, and code security.

.NET FEATURES

Rich User Interfaces : Supports advanced graphics and UI (Windows Presentation Foundation).

Database Connectivity : ADO.NET for easy access to databases.

Integration with Visual Studio : Seamless development with debugging and deployment tools.

Common Language Runtime (CLR) : Manages code execution (memory, threads, garbage collection).

Base Class Library (BCL) : Predefined reusable classes for common tasks like file handling, networking, and more.

PROJECT TYPES in .NET

Web APIs
APIs that allow different applications to communicate over HTTP, often used for backend services.

Blazor Applications
Web apps that use C# instead of JavaScript. Options include:

- **Blazor WebAssembly**: Runs in the browser.
- **Blazor Server**: Runs on the server with real-time updates to the UI.

Xamarin Applications

- Cross-platform mobile apps for iOS and Android using C#.

Microservices

- Small, independent services that work together to create scalable and distributed applications.

Azure Functions

- Serverless functions for cloud-based, event-driven applications.

Unit Test Projects

PROJECT TYPES in .NET

Console Applications

Simple, command-line programs without a graphical interface. Great for automation, utilities, or learning.

Web Applications

ASP.NET Core: Modern, cross-platform framework for building dynamic web apps.

ASP.NET Web Forms: Older framework for building web apps (less commonly used today).

Desktop Applications

- **Windows Forms**: Basic desktop apps with a simple user interface.
- **WPF (Windows Presentation Foundation)**: Desktop apps with advanced graphics and rich UI elements.
- **MAUI**: Cross-platform apps for Windows, macOS, iOS, and Android, using a single codebase.

Class Libraries

Reusable code that can be shared across different applications. Outputs as a .dll file.

Prerequires

- Download Visual Studio Community Edition **https://visualstudio.microsoft.com/vs/community/**
- Install git **https://git-scm.com/**
- Create github account and send me

NET FRAMEWORK , .NET CORE and .NET STANDARD

.NET Framework
The .NET Framework is a software development framework created by Microsoft in 2002. It provides a consistent programming environment for building and running applications on Windows. It includes a large class library (called the Framework Class Library or FCL) and a runtime environment (called the Common Language Runtime or CLR) for executing applications.

Key Features:

Windows-Only: Designed specifically for Windows-based applications.
Full API Support: Provides extensive libraries and APIs for desktop, web, and server-side applications.
Languages: Supports multiple languages like C#, VB.Net, and F#.
Application Types: Used for building desktop applications, web applications (ASP.NET), and web services.

NET FRAMEWORK , .NET CORE and .NET STANDARD

.NET Core
.NET Core is a modern, open-source, and cross-platform development platform introduced by Microsoft in 2016. It is designed for creating cloud-based, scalable, and high-performance applications that can run on Windows, Linux, and macOS. It is the successor to the .NET Framework and is built to meet the needs of modern development practices.

Key Features:

Cross-Platform: Runs on Windows, Linux, and macOS.
Open Source: Released under the MIT License and maintained by the .NET Foundation.
High Performance: Optimized for scalability and performance, especially for cloud and web applications.
Modular Architecture: Supports lightweight and modular development through NuGet packages.
Versatility: Suitable for web applications (ASP.NET Core), microservices, IoT, and more..

NET FRAMEWORK , .NET CORE and .NET STANDARD

.NET Standard is a set of APIs that defines a common functionality across different .NET implementations (such as .NET Framework, .NET Core, and Xamarin). It ensures that libraries built for one .NET platform can be used across others, providing cross-platform compatibility.

Purpose: Simplifies sharing libraries between platforms without platform-specific code.

In essence, .NET Standard helps make libraries portable across different .NET implementations.

Feature	.NET Framework	.NET Core
Platform Support	Windows only	Cross-platform (Windows, Linux, macOS)
Performance	Comparatively slower	Faster and more optimized
Deployment	Requires system-wide installation	Supports self-contained deployment
Application Type	Desktop and legacy applications	Web apps, APIs, cloud & microservices
Open Source	Mostly closed source	Fully open source
Future Support	Maintenance mode	Actively developed and recommended

Visual Programming

- Visual Programming is a programming approach where developers create applications using **visual elements such as forms, buttons, text boxes, and other components** instead of writing the entire program using text-based code.
- The programmer focuses more on **logic and user interaction** rather than syntax rules.
- In visual programming, predefined blocks or components already contain code internally.
- The user only needs to place and connect these components logically to perform a task.

Visual Programming

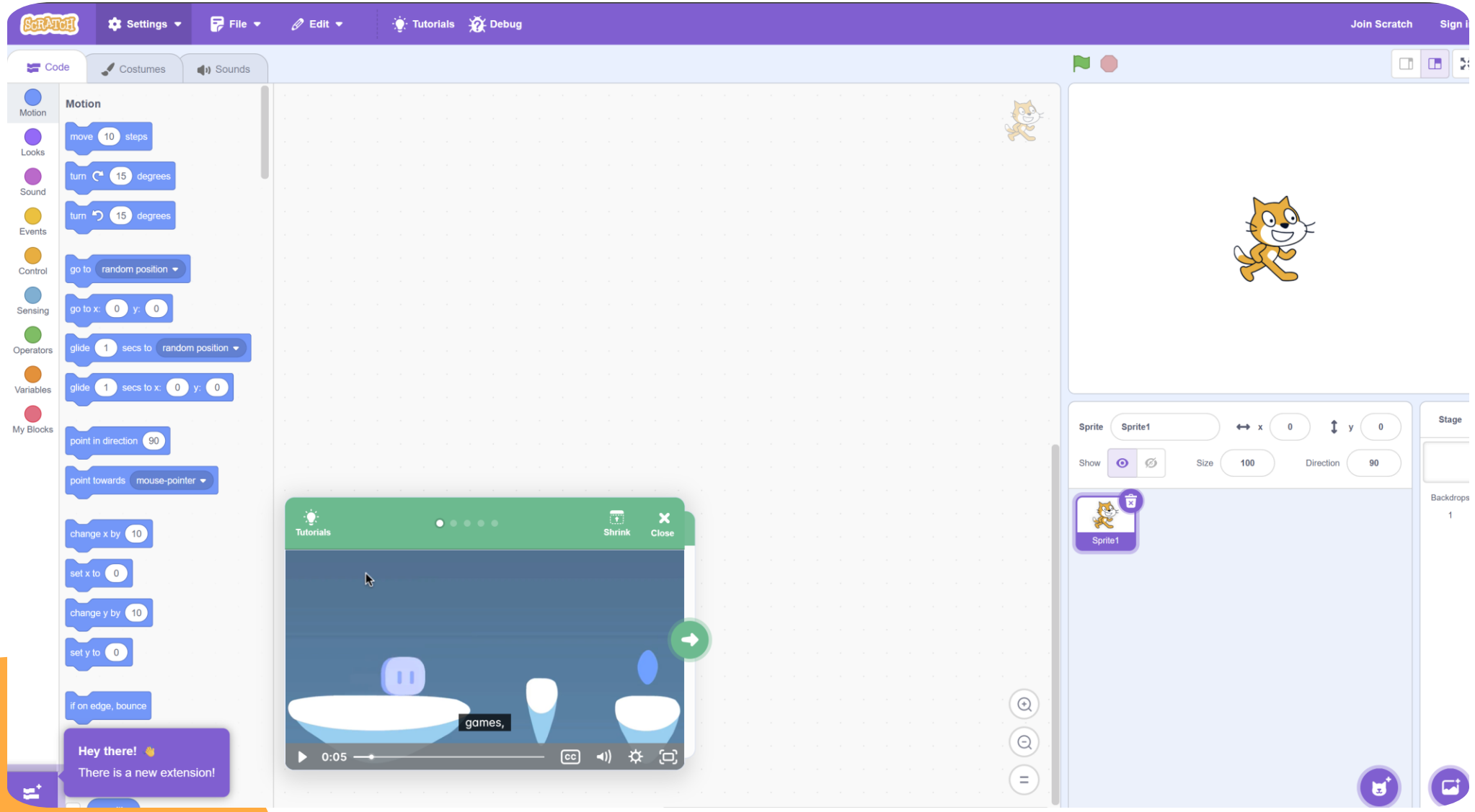
Text-Based Programming

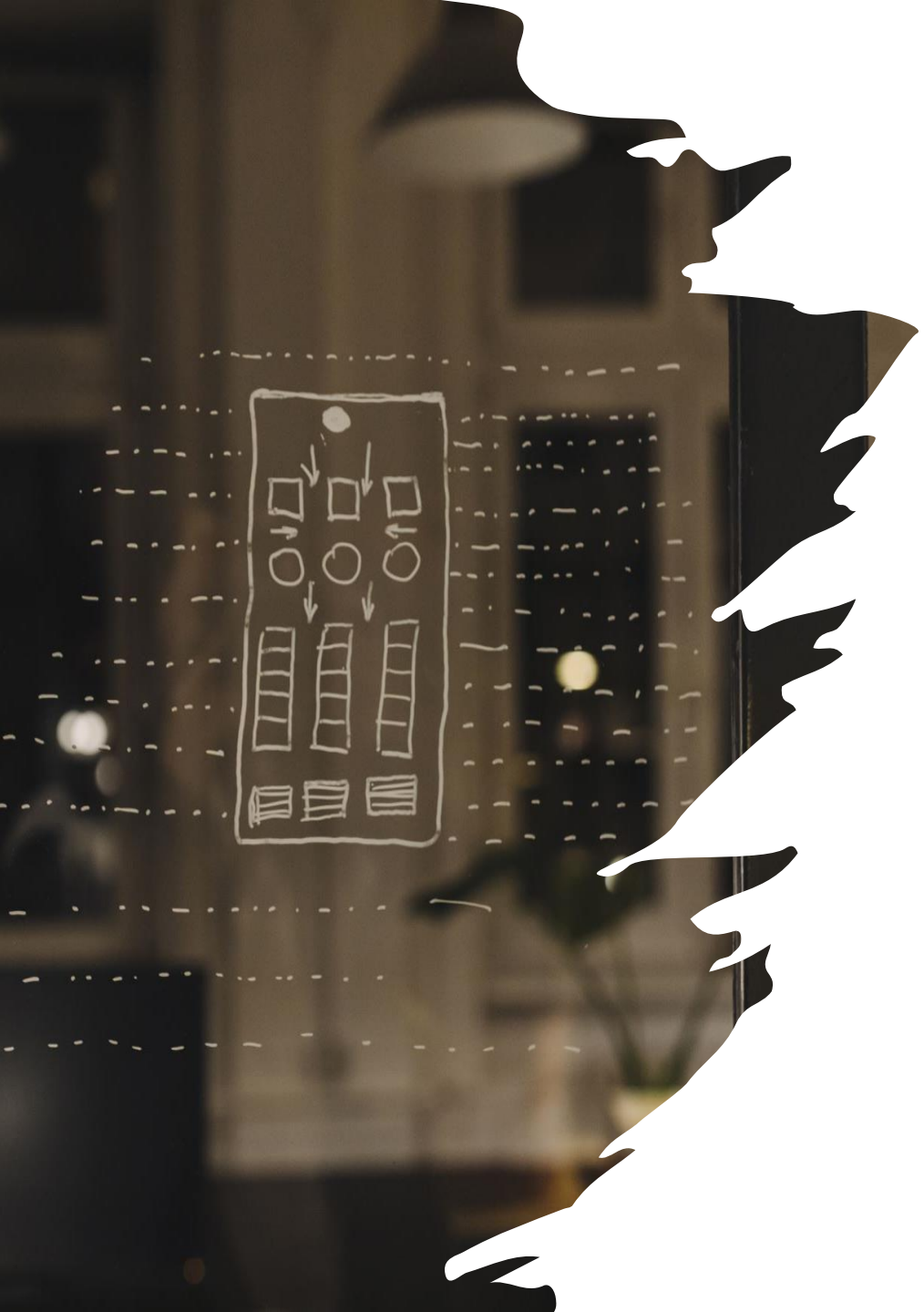
- Programmer writes code line by line
- Strict syntax rules must be followed
- Examples: C, C++, Java
- More chances of syntax errors

Visual Programming

- Uses drag-and-drop components
- Less dependency on syntax
- Focuses on logic and flow
- Easy to understand and use

Example of Visual Programming : SCRATCH





Advantage of Visual Programming

- The .NET framework provides strong support for visual programming through **Visual Studio IDE**.
- Developers can design graphical user interfaces (GUI) visually and then write minimal code to define the behavior of the application.

.NET mainly supports visual programming using:

- Windows Forms (WinForms)
- WPF (Windows Presentation Foundation)

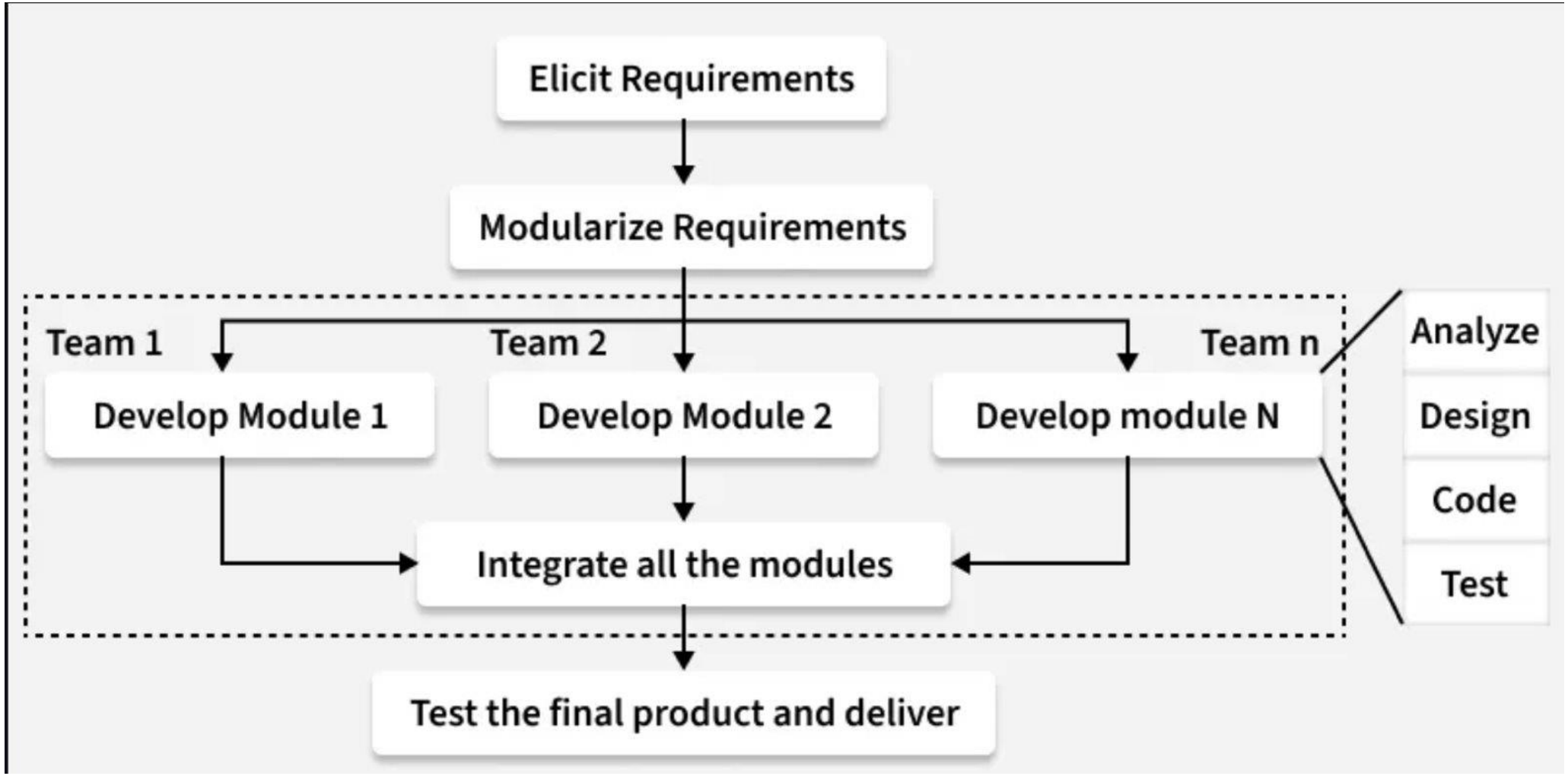
Rapid Application Development Model

- RAD (Rapid Application Development) is an **incremental software development model**
  - Proposed by **IBM in the 1980s.**
  - Focuses on **fast development** using **short time-boxed cycles (60–90 days)**
  - Uses **prototyping** and **continuous user feedback.**
  - System is divided into **small modules developed in parallel.**
  - Unlike traditional models such as the Waterfall Model, RAD adapts quickly to changing requirements and user needs.

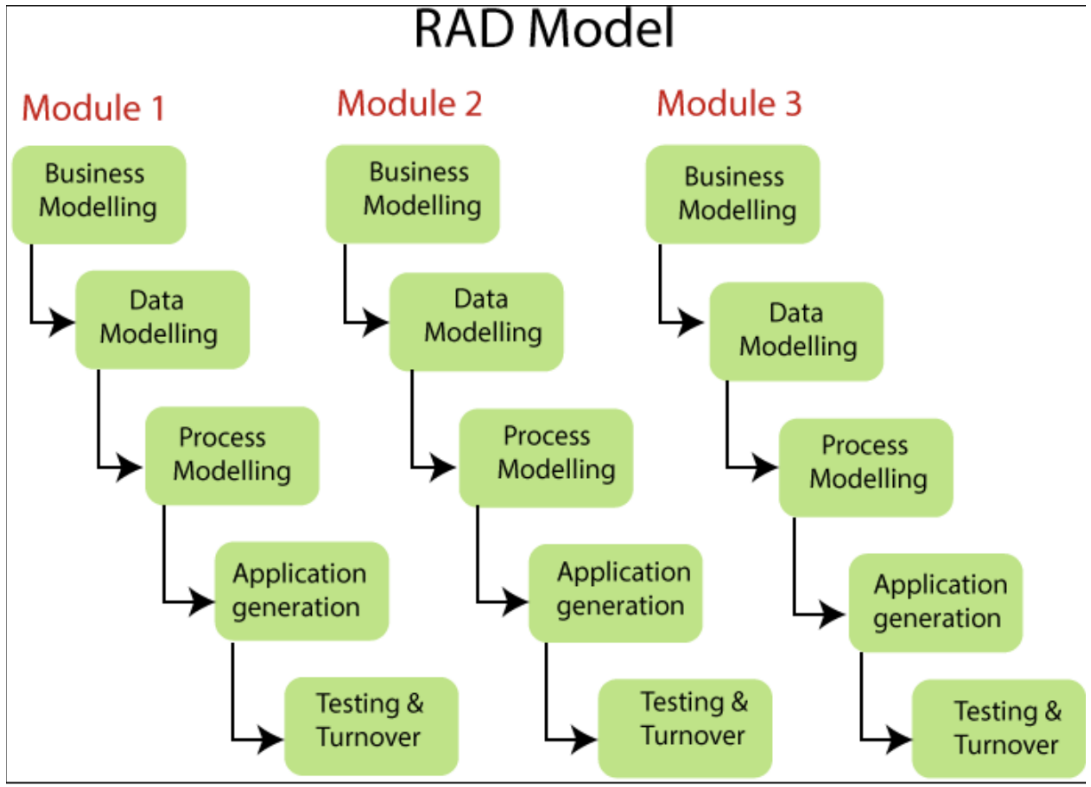
Rapid Application Development Model

- RAD (Rapid Application Development) is an **incremental software development model**
  - Proposed by **IBM in the 1980s.**
  - Focuses on **fast development** using **short time-boxed cycles (60–90 days)**
  - Uses **prototyping** and **continuous user feedback.**
  - System is divided into **small modules developed in parallel.**
  - Unlike traditional models such as the Waterfall Model, RAD adapts quickly to changing requirements and user needs.

Rapid Application Development Model



RAD



1. Business Modeling (What to build)

- Understand **what the client wants**.
- Identify **modules** (Login, User, Reports, Billing, etc.).
- Example: *What screens and features are needed in the .NET app.*

2. Data Modeling (What data to store)

- Decide **tables, fields, and relationships**.
- Design **SQL Server database**.
- Example: *User table, Order table, Foreign Keys.*

RAD MODEL

3. Process Modeling (How it will work)

- Decide **how data moves between UI, API, and database**.
- Define **business logic**.

Example: *User clicks Save → Controller → Service → Database.*

4. Application Generation (Build fast)

- Quickly develop using **ASP.NET, MVC/Core, Web API, Visual Studio**.
- Use **ready components**, scaffolding, and reusable code.
- Example: *Generate CRUD pages, APIs, and forms quickly.*

5. Testing and Turnover (Check & deliver)

- Test using **unit testing, debugging, and user testing**.
- Fix issues and **deploy** the application.

Example: *Deploy to IIS / Cloud and hand over to client.*

Advantage of RAD model

Fast development: Reusable components reduce development time.

Early customer feedback: Users see prototypes early and give feedback.

Lower development effort: Fewer developers are needed due to automation.

Better quality: Powerful tools help build quality software quickly.

Easy progress tracking: Development progress is visible in each phase.

Flexible to changes: Changing requirements are easy to handle due to short cycles.

High productivity: Small teams can deliver results faster.

Disadvantage of RAD model

Needs skilled professionals: Developers must be experienced with advanced tools.

Depends on reusable components: Lack of reusable parts can cause failure.

High coordination needed: Team leader must closely manage developers and customers.

Not suitable for all systems: Systems that cannot be modularized cannot use RAD.

Continuous customer involvement required: Lack of user commitment can fail the project.

Costly for small projects: Tool and automation costs may exceed the budget.

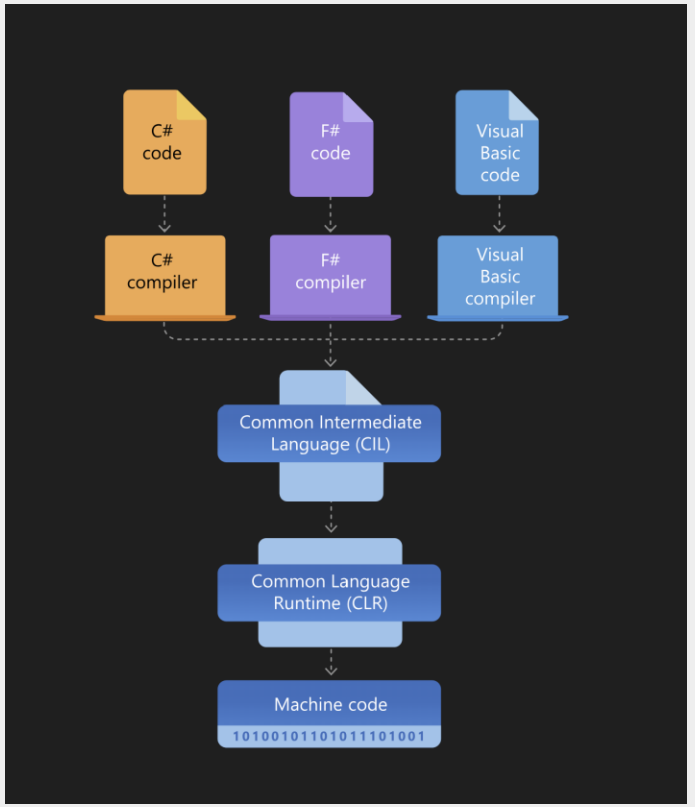
Not suitable for every application: Some projects do not fit RAD structure.

.NET ARCHITECTURE

Major Components : CLR and BCL

CLR (Common language runtime) : is the execution engine that handles running applications. It provides services like thread management, garbage collection, type-safety, exception handling, and more

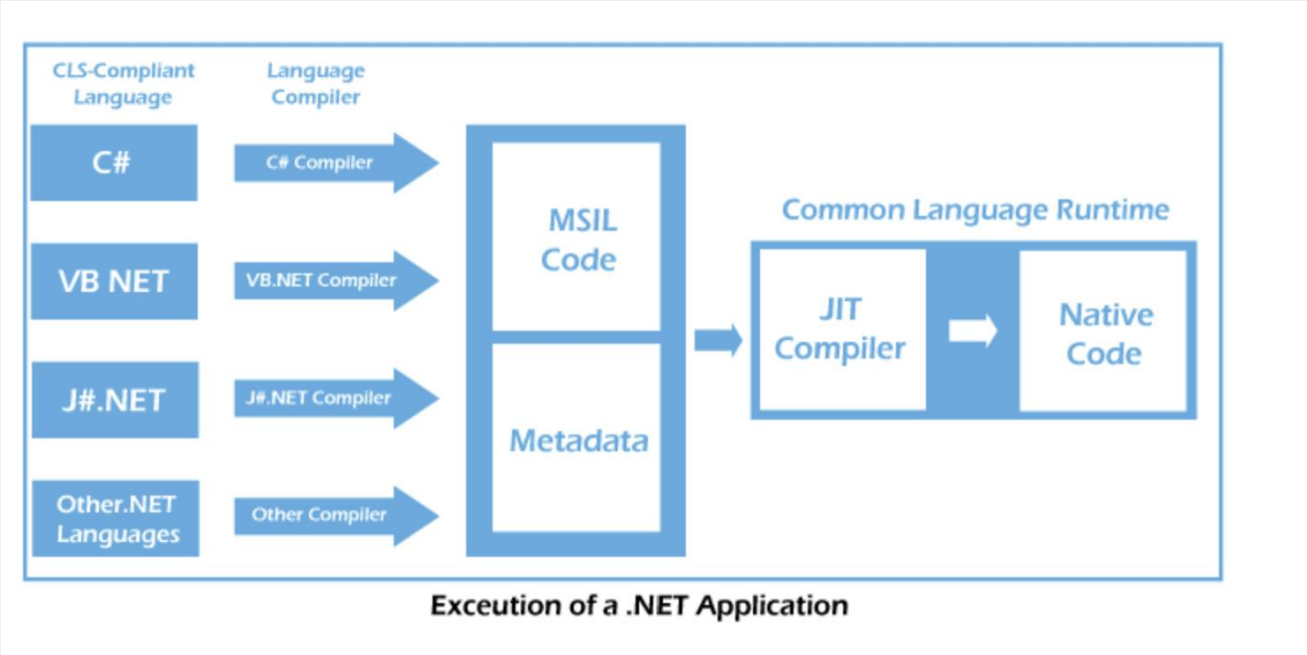
Base Class Library provides a set of APIs and types for common functionality. It provides types for strings, dates, numbers, etc.
The Class Library includes APIs for reading and writing files, connecting to databases, drawing, and more.



.NET ARCHITECTURE

.NET applications are written in the C#, F#, or Visual Basic programming language. Code is compiled into a language-agnostic Common Intermediate Language (CIL). Compiled code is stored in assemblies—files with a .dll or .exe file extension.

When an app runs, the CLR takes the assembly and uses a just-in-time compiler (JIT) to turn it into machine code that can execute on the specific architecture of the computer it is running on.



Exceution of a .NET Application