



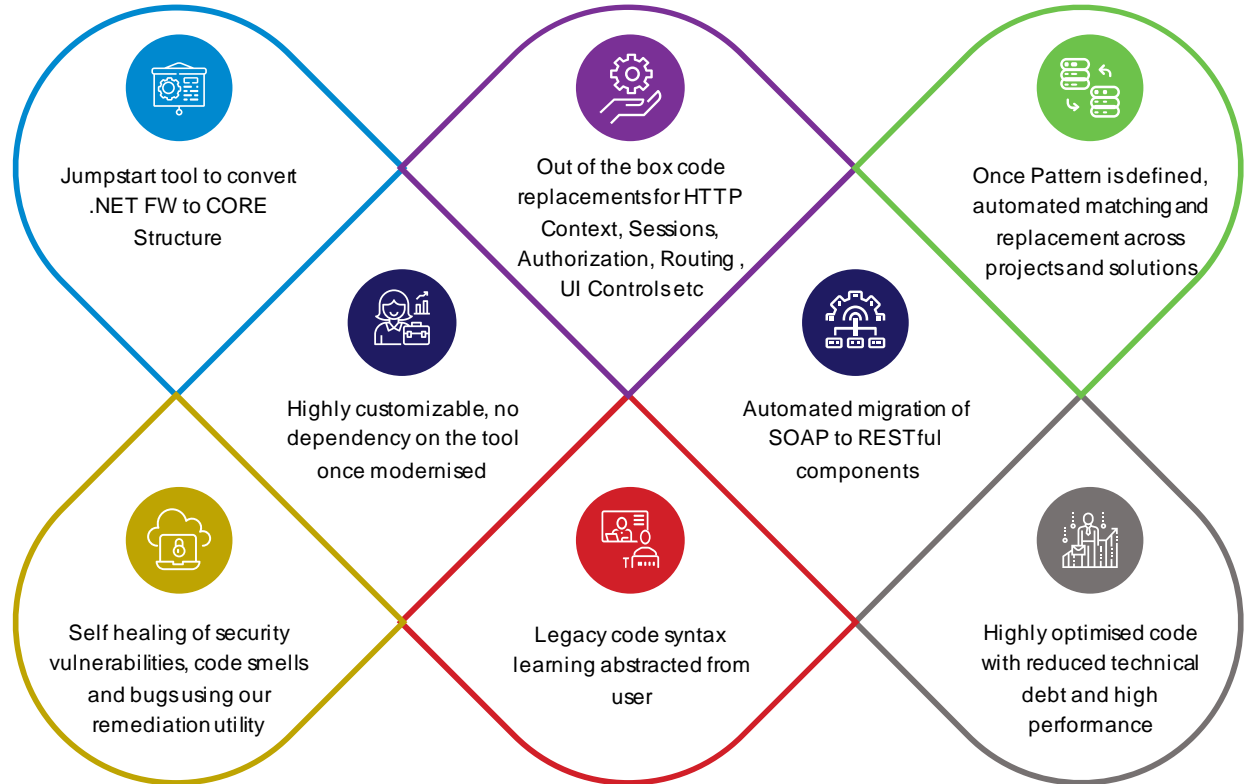
devNXT

AI based modernization platform
for Microsoft Applications

Problem statement and objective

When Customer is having large on-prem legacy application estate which span across .NET technology stacks and there is a need to make them cloud ready, devNXT can help in modernization of those legacy .NET applications, onboard the application to cloud thereby accelerating migration timelines.

devNXT automates the patterns required for modernization and helps developers upgrade legacy application versions (.NET, VB) to cloud ready versions to support digital transformation



devNXT features



Modernization

Features

- Conversion of .NET Framework (any version) to latest .NET core/.NET
- Conversion of Legacy .NET Framework to Latest .NET Framework (4.8)
- Supports - ASP.NET, MVC, Console, Entity Framework, Windows Apps, .NET Libraries, Windows Services, WCF
- Supports multi-project solutions
- Supports C#, ASP2.0 and Visual Basic solutions

Benefits

- Improved productivity
- Reduced modernization time by up to 35 to 45%,
- Reduced manual effort through the automation of repeatable patterns
- A Highly customizable solution based on the customer's environment and migration requirement



Remediation

- 100+ SonarQube and 30+ CAST rules automated for Auto Remediation & Correction of source code
- Supports C#, VB.NET, JavaScript, HTML, CSS languages for remediation
- Tool can be customized/enhanced with new rules based on customers' app requirement

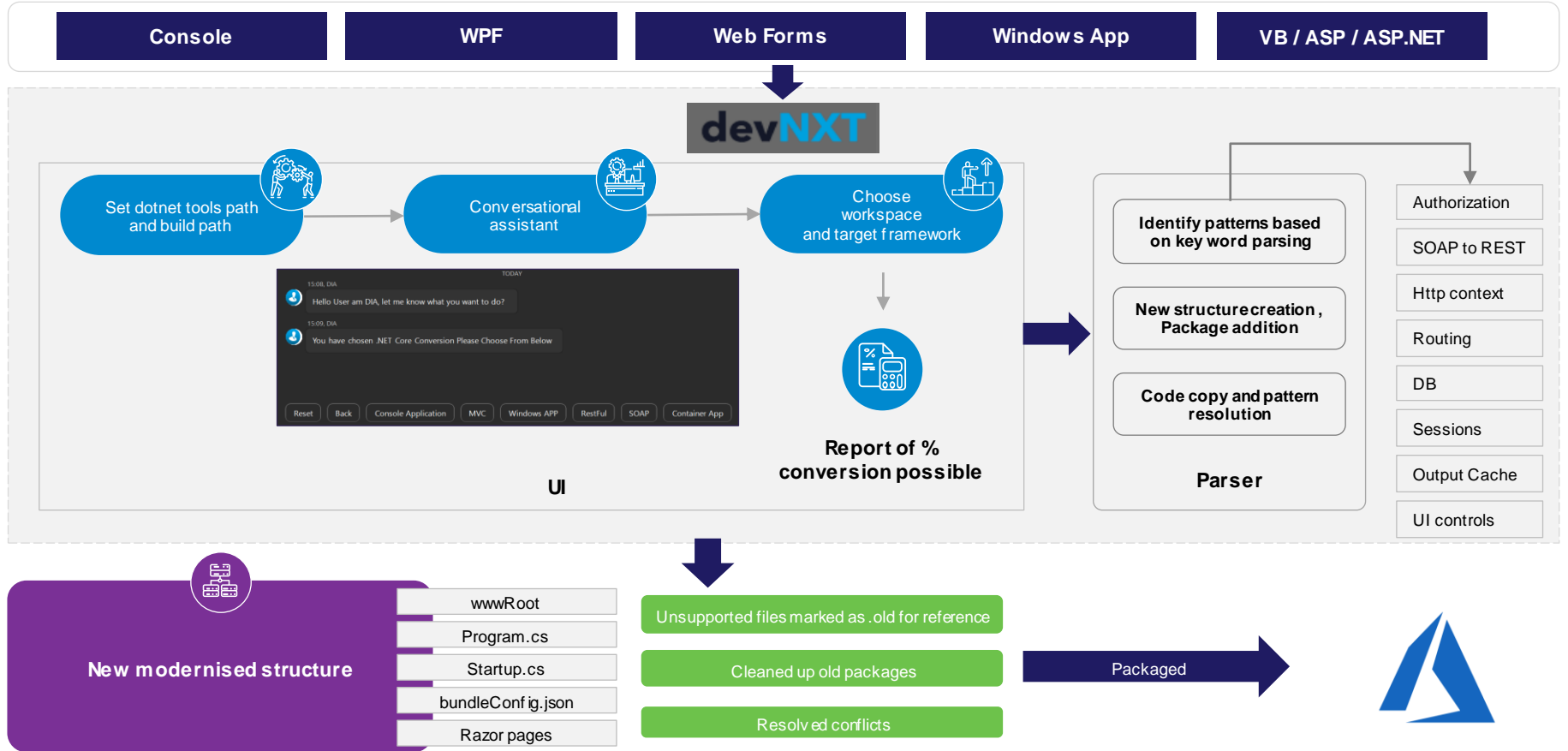
- Code remediated into technology compliant stack and compliant with Sonar/CAST rules



Azure Ops

- Easy creation and management of work items, repos, pipelines, test plan etc.
- Low code no code solution, where developer not required to use the Azure console to create Repos and pipelines

Solution overview



Case Study

Application modernization for one of the leading financial services company based in US using **Wipro's devNXT platform**



Background & Requirements

- Modernize 3000+ projects developed in legacy .NET Framework to .NET Core in an iterative model. Following 4 weeks Scrum cycle (3 weeks of dev & test + 1 week of build)
- These projects are part of 200+ core business-critical applications catering to Banking, Wealth Management, Insurance and Risk Management functions
- 2000+ internal business users and million+ users accessing the applications over internet
- Each project has 100+ internal /external and 3rd party integration / interfaces
- Replace all out-of-support legacy tools used in these applications with new age tools/ technologies (e.g., payment gateways, credit score check)
- Keep the applications cloud native and refactor them to a Microservices based architecture
- Enhance the applications to cater to new business requirements
- Enable API based integration with internal and external applications



devNXT Highlights

Support for core architecture structure creation and replacement of libraries

Support for migration of Auto mapper, conventional and attribute routing

Support for migration of Entity Framework to .NET Core

Support for Authorization (Ping Federate and channel secure)

Conversion of Web Forms (.aspx) to razor pages

Support for migration of HTTP handler & Modules to .NET Core



Solution Provided by Wipro

- **Wipro devNXT modernization platform** was used for automatic code conversion
- Discovery phase to understand technical and functional details of the applications
- Analyzed templates, dependencies, packages configurations, code and manual migration specifics
- Develop quick proof-of-concept to test critical aspects
- Design and implement a Microservices based architecture
- Work closely with 3rd party service provider to ensure working of all integration points
- Decision Tree based approach to determine re-platform Vs. re-factor (depending upon .NET version and MVC)
- Prepare backlog and prioritize, enhance devNXT to support backlog items migration



Benefits

Installation and configuration time of < 10 mins

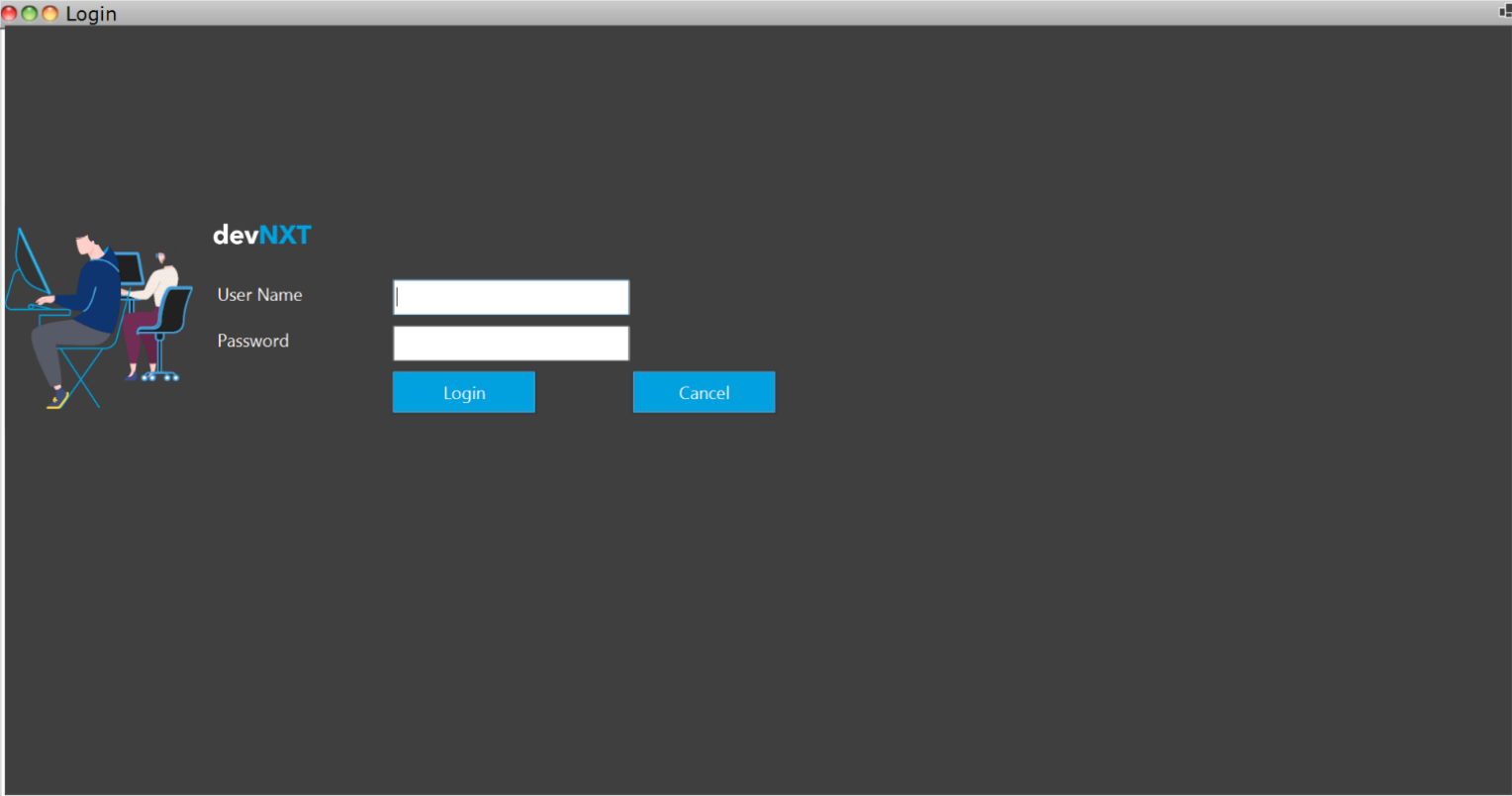
Migration using devNXT tool was less than 2 mins

Standard patterns which repeats across applications were automated

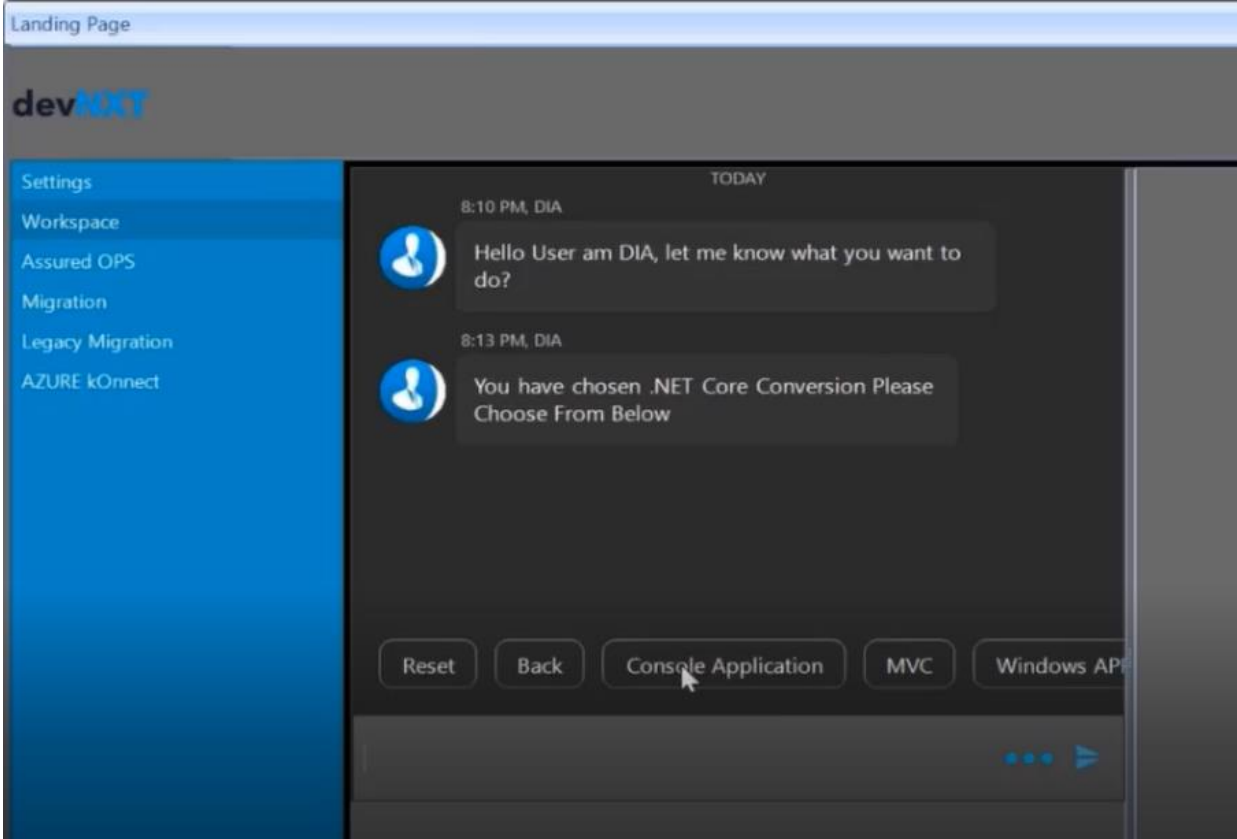
Built a strong technical and architecture governance process

Minimal compilation errors compared to manual process

Application Snapshots



Application Snapshots



Application Snapshots

The screenshot displays the devNXT application interface. On the left, a chat window shows a conversation with a user named DIA. The main workspace configuration area includes fields for 'Workspace' (D:\devNXT\workspace\vstudio\ASPMVCWebApplication - Copy (3)), 'Target Framework' (.NET 6.0), and buttons for 'Browse', 'Compatibility Test', 'Migrate', and 'Open Report'. A 'Wiser Mode' indicator is present, along with a message stating 'devNXT can convert the solution 100%, please proceed by clicking the Convert button...'. Below this is an 'Analysis Report' table.

Assembly Name	Target Framework	.NET Core,Version=...	.NET,Version=v5.0
Antr3.Runtime, Ve...		100	100
ASPMVCWebAppli...	.NETFramework,Ve...	75.24	75.24
Newtonsoft.Json, V...	.NETPortable,Veri...	100	100
Newtonsoft.Json, V...	.NETFramework,Ve...	100	100
Newtonsoft.Json, V...	.NETFramework,Ve...	100	100
Newtonsoft.Json, V...	.NETStandard,Veri...	100	100
Newtonsoft.Json, V...		100	100
Newtonsoft.Json, V...	.NETStandard,Veri...	100	98.65

Assembly Type Summary

Legend

- UnresolvedAssembly
- SkippedAssembly
- 0.015625
- 0.984375



Thank you
for your time