INNONE
innovation one

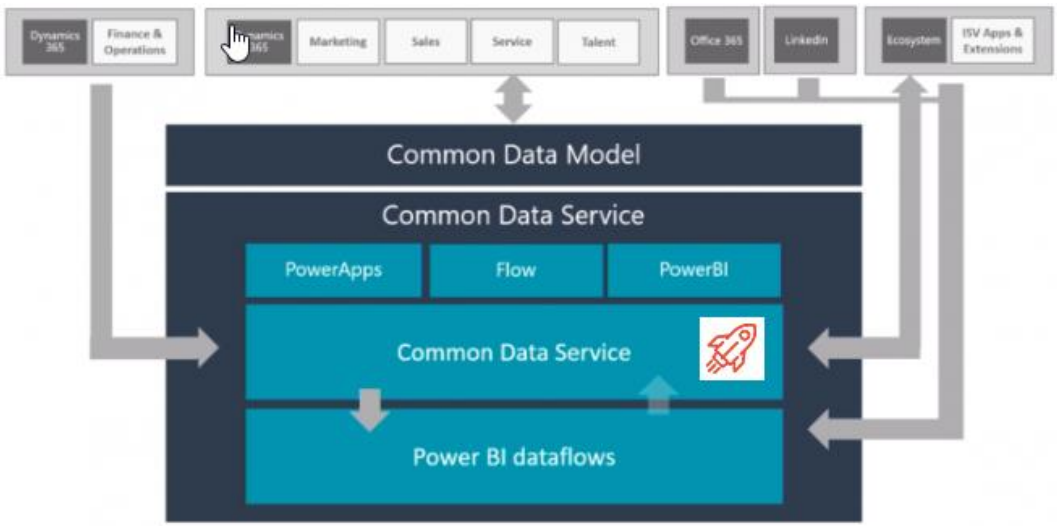
CDS Plugin Development Framework

Innovation One s.r.o

Version 1.3

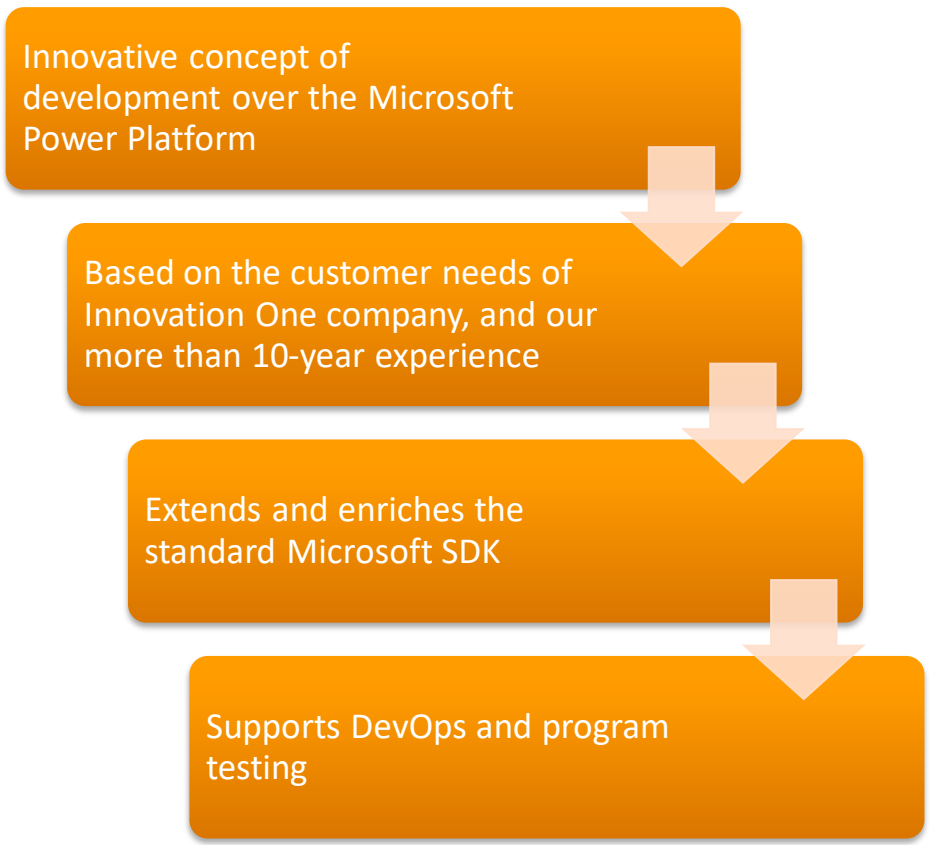
CDS Plugin Development Framework a part of Microsoft Common Data Service

- Middleware between standard Microsoft SDK and customer logic implemented in plugins
- Customer logic is structured into independent functional modules called Tasks
- The tool produces detailed information to diagnose errors and optimize performance issues within MS CDS
- It facilitates clear and sustainable codewriting



CDS Plugin Development Framework

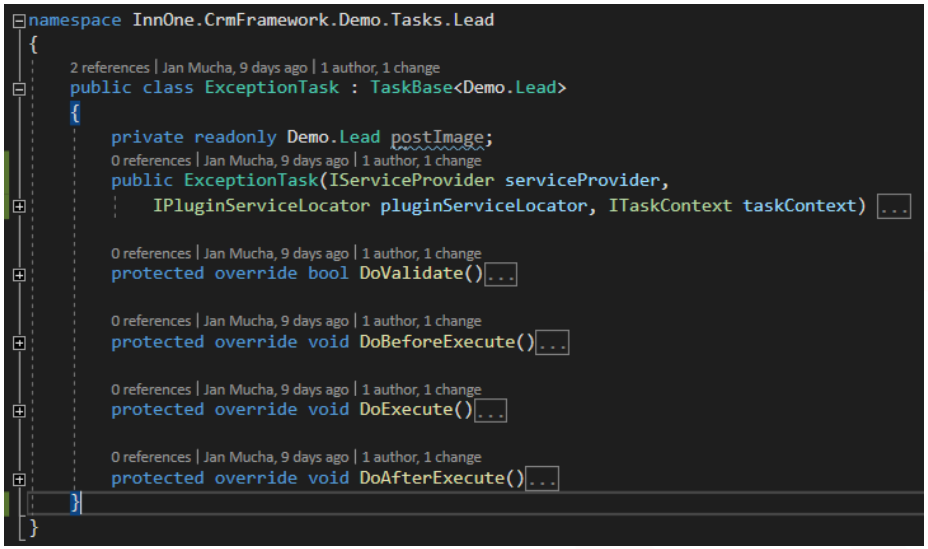
CDS Plugin Development Framework Benefits



- In-built logs of each Task execution
- Visualization of logged information
- Clearly defined, well-arranged structure of writen code
- Analysis linked with the implemented code
- Reduction of development and management costs
- Tasks are split into validation and execution part

Task – the esential element

- Task is the basic logical unit
- Task Name is a unique identifier across analysis, implementation, tests, and diagnostics
- Task is intended by an analyst at the time of functionality design
- Each Task consists of validation and execution parts
  - DoValidate
  - BeforeExecute
  - Execute
  - AfterExecute



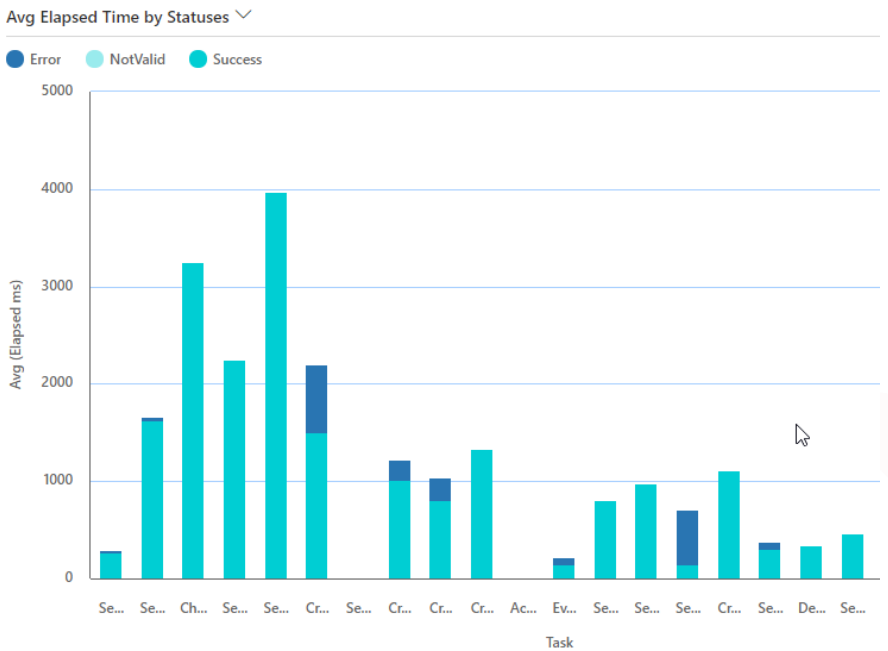
CDS plugins do only Tasks execution

```
public class GeneralConsentExecutor : PluginExecutor
{
    public GeneralConsentExecutor(string unsecureConfig, string secureConfig) : base(unsecureConfig, secureConfig)
    {
        RegisterEvents((int)CrmFramework.Core.Enums.SdkMessageProcessingStepStage.Preoperation,
        new[] { "Create", "Update" }, asc_generalConsent.EntityLogicalName, typeof(SetIdConsent));

        RegisterEvents((int)CrmFramework.Core.Enums.SdkMessageProcessingStepStage.Preoperation,
        new[] { "Create", "Update" }, asc_generalConsent.EntityLogicalName, typeof(GeneralConsentName));


        RegisterEvent((int)CrmFramework.Core.Enums.SdkMessageProcessingStepStage.Postoperation,
        "Create", asc_generalConsent.EntityLogicalName, typeof(GenralConsentValidation));

        RegisterEvents((int)CrmFramework.Core.Enums.SdkMessageProcessingStepStage.Postoperation,
        new[] { "Create", "Update" }, asc_generalConsent.EntityLogicalName, typeof(UpdateSubstitueLead));

        RegisterEvent((int)CrmFramework.Core.Enums.SdkMessageProcessingStepStage.Postoperation,
        "Create", asc_generalConsent.EntityLogicalName, typeof(UpdateConsentLead));

        RegisterEvents((int)CrmFramework.Core.Enums.SdkMessageProcessingStepStage.Postoperation,
        new[] { "Create", "Update" }, asc_generalConsent.EntityLogicalName, typeof(ResolveConsentDependence));

        RegisterEvents((int)CrmFramework.Core.Enums.SdkMessageProcessingStepStage.Postoperation,
        new[] { "Create", "Update" }, asc_generalConsent.EntityLogicalName, typeof(DeleteUselessConsent));

        RegisterEvents((int)CrmFramework.Core.Enums.SdkMessageProcessingStepStage.Postoperation,
        new[] { "Create", "Update" }, asc_generalConsent.EntityLogicalName, typeof(CountAllowAddressingAttributesTask));

        RegisterEvents((int)CrmFramework.Core.Enums.SdkMessageProcessingStepStage.Postoperation,
        new[] { "Create", "Update", "Delete" }, asc_generalConsent.EntityLogicalName, typeof(ConsentLog));
    }
```

- Plugins don‘t hold customer logic any more
- Every plugin has the same structure
- Tasks are associated with plugins based on either the entity or the functionality
- Tasks are executed in a defined order
- Information about registration of plugins and their steps is stored and versioned right in the code

Logging

- Every Task execution is automatically logged with detailed information
- LogService is available for developers, making it easy to log required information
- The required logging level can be set for a specific environment
- Logged information is available in clear charts on the dashboards
- The specific log can be drill down from the graph for detailed diagnosis



Dashboards – Logs Overview

All Logs Last 5 days

- The dashboard contains comparison of the number of executions and the average execution time according to the status of the task
- This report helps the user identify any performance issues and recommend effective optimization

Errors and Fatals Last 5 Days

- The dasboard displays the current status of the application and reports errors or performance issues
- This data provides the user with detailed information about the errors and helps him to clearly identify the cause and come up with an effective remedy

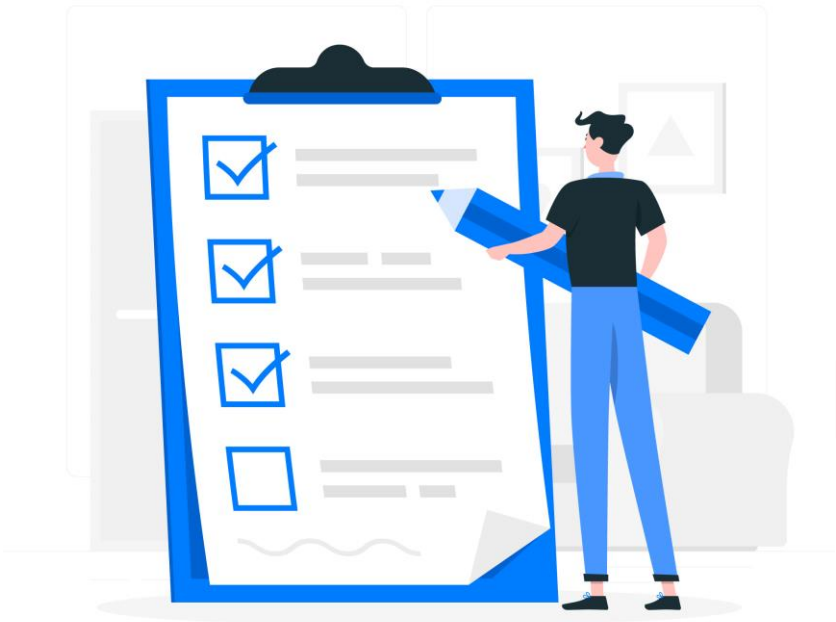
Program tests

InnOne Framework offers full support for program testing

The InnOne Framework includes a set of components that speed up and streamline the implementation of program tests

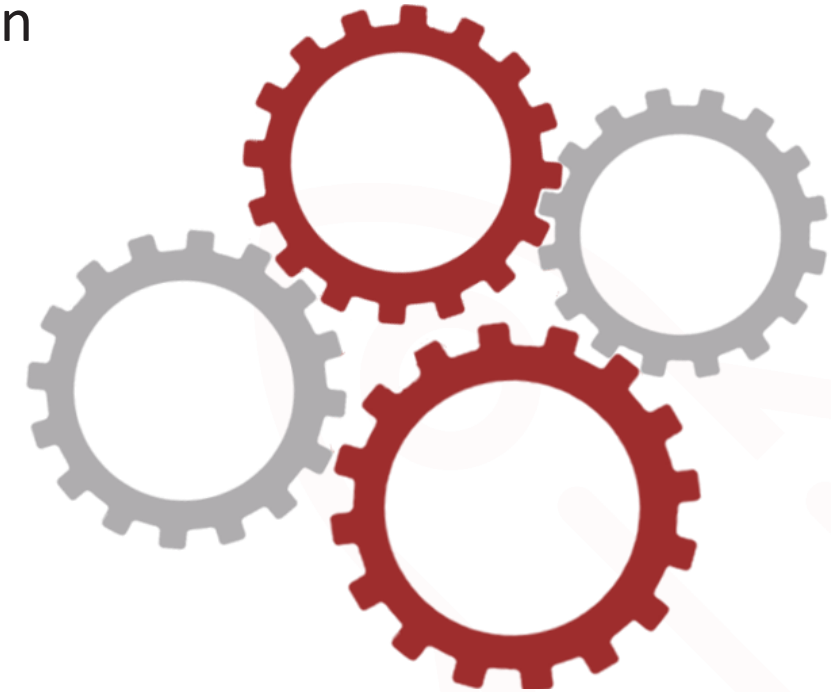
Software testing significantly increases the quality of delivery and reduces errors

Continuous writing of program tests significantly increases the sustainability of the solution and simplifies further development

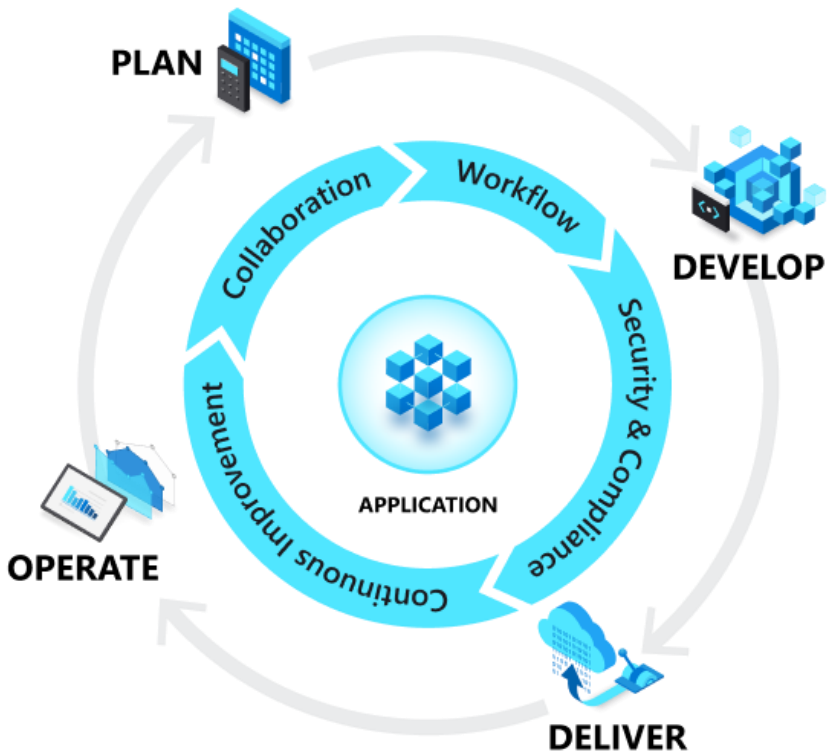


CDS Plugin Development Framework Components

- CDS Plugin Development Framework Solution
  - InnOne Setting Entity
  - InnOne Log Entity
  - InnOne Framework Application
    - Error Log Dashboard Last 5 Days
    - All Log Dashboard Last 5 Days
- NuGet package with a set of DLL libraries
  - Support for writing plugins
  - Support for development of applications that communicate with CDS
  - Support for writing program tests
- Full support of *SPKL Framework*



CDS Plugin Development Framework fully supports DevOps



- One-click deployment
- Automatic deployment in all environments
- Program tests are run automatically at the deployment time
- The risk of human error is minimized thanks to the automated processes

Implementation Process

Free preliminary consultation with an Innovation One ltd. specialist

Analysis of customer problems

Identification of the key benefits of the *CDS Plugin Development Framework*

Design of the implementation process according to the customer´s needs

Deployment of *CDS Plugin Development Framework* and staff training

Easy and reusable development, increasing quality and sustainability on the Microsoft Power Platform

INNONE innovation one

Partner for

digital transformation

Koněvova 2660/141, Žižkov,
130 00 Praha 3
Po-Pá: 8-17 hod
+420 736 213 603
obchod@innone.cz