Sonata DevOps Competency

SONATA

SONATA SOFTWARE

Agenda

- About Sonata
  - Sonata Software at A Glance
  - Our Versatile Portfolio of Solutions
  - Sonata footprint across the globe
  - World Class Alliances
  - Marquee Client list
  - Sonata's Unique "PLATFORMATION" Approach
  - Digitizing Business Using Platforms
- Sonata Devops offerings
- Sonata Devops Framework
  - DEEEPS-DevOps Engineering Enablement & Execution Platform Solutions framework
  - DEEEPS Reference Architectures
- Sonata Devops Expertise & Skills
- Approach to Transformation with DevOps
  - Devops Maturity Assessment model
- Case Studies

Sonata Devops Services Overview

DevOps Services

DevOps Consulting

- Maturity assessment and roadmap
- Organizational change management(Review Process ,people, tools for as-is setup)
- Tools Suggestion
- Suggested Reference Architecture

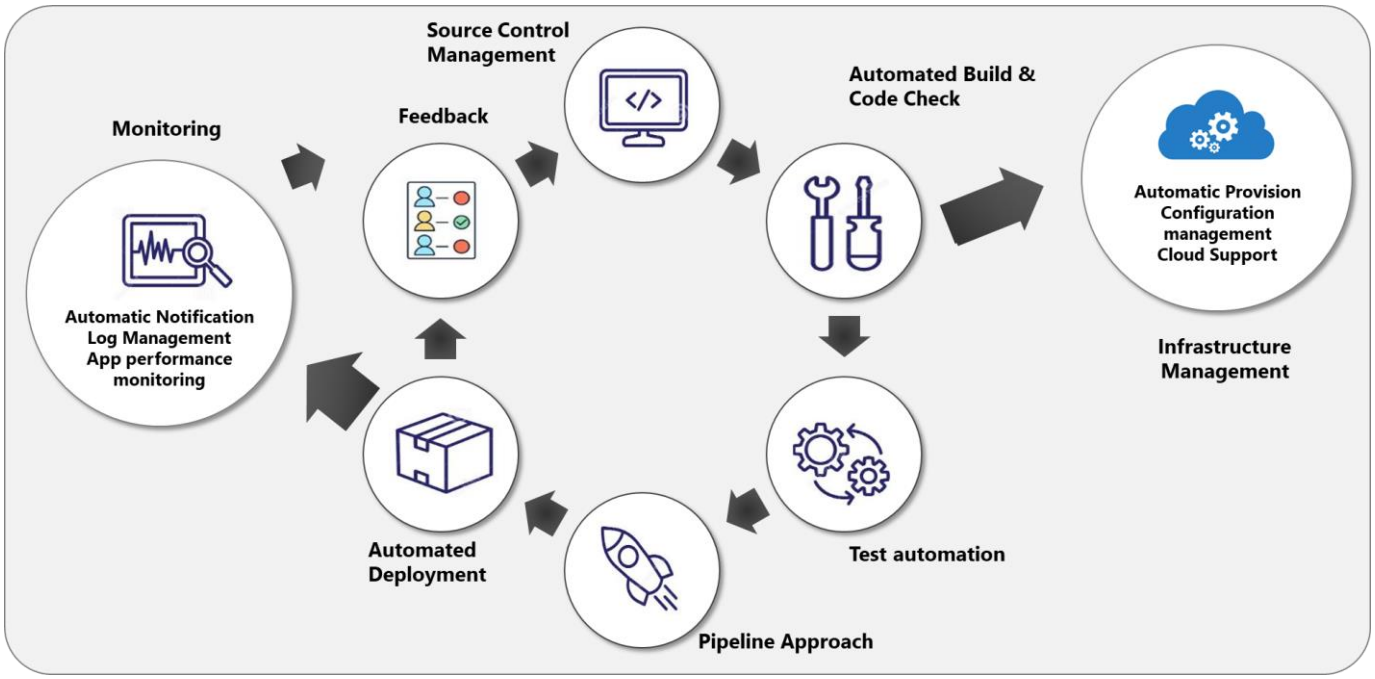
DevOps Implementation

- Branching Strategy
- Build Automation
- Pipeline orchestration
- Containerization & orchestration implementation.
- Code Coverage, Code Quality
- Automated Deployment & testing
- Monitoring setup.
- Cloud Native Solutions implementation
- Environments creation and setup

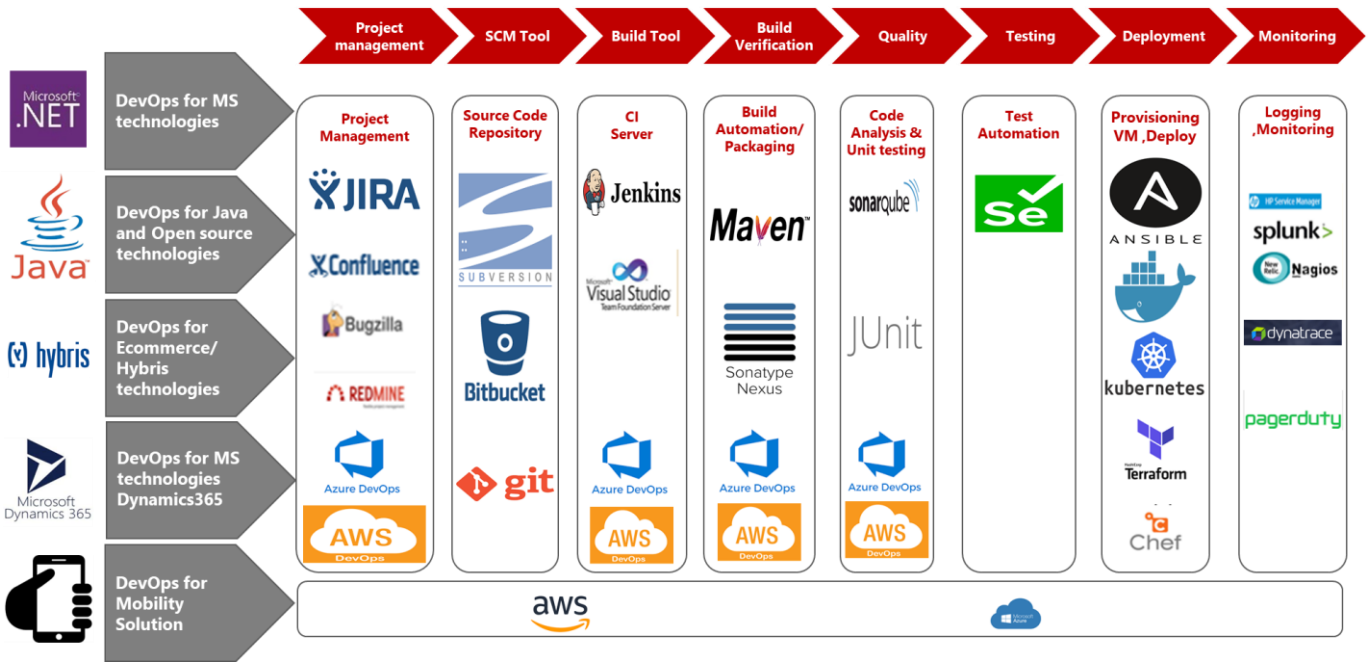
Support and Managed Services

- Environment Management
- Continuous improvement
- KPI/Metrics driven support delivery
- Environment & Application monitoring and issue resolution

Sonata Devops Services workflow



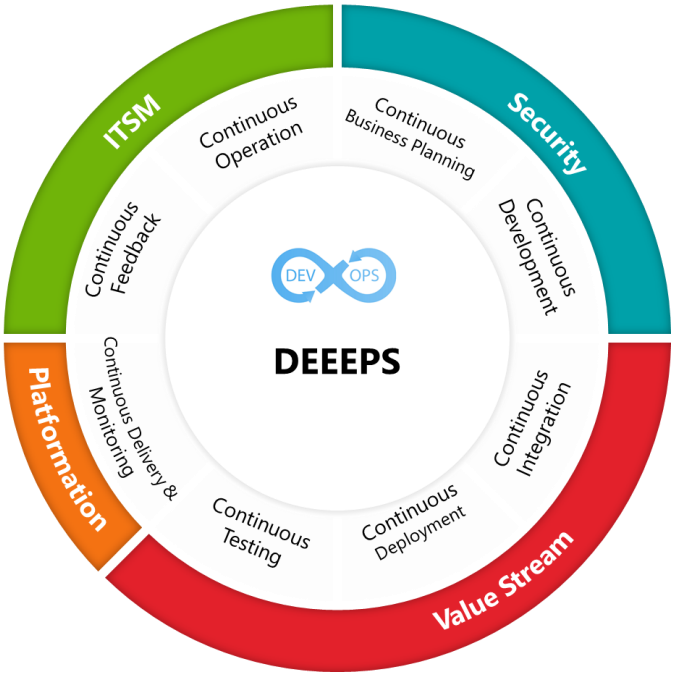
Sonata DevOps Expertise & Skills



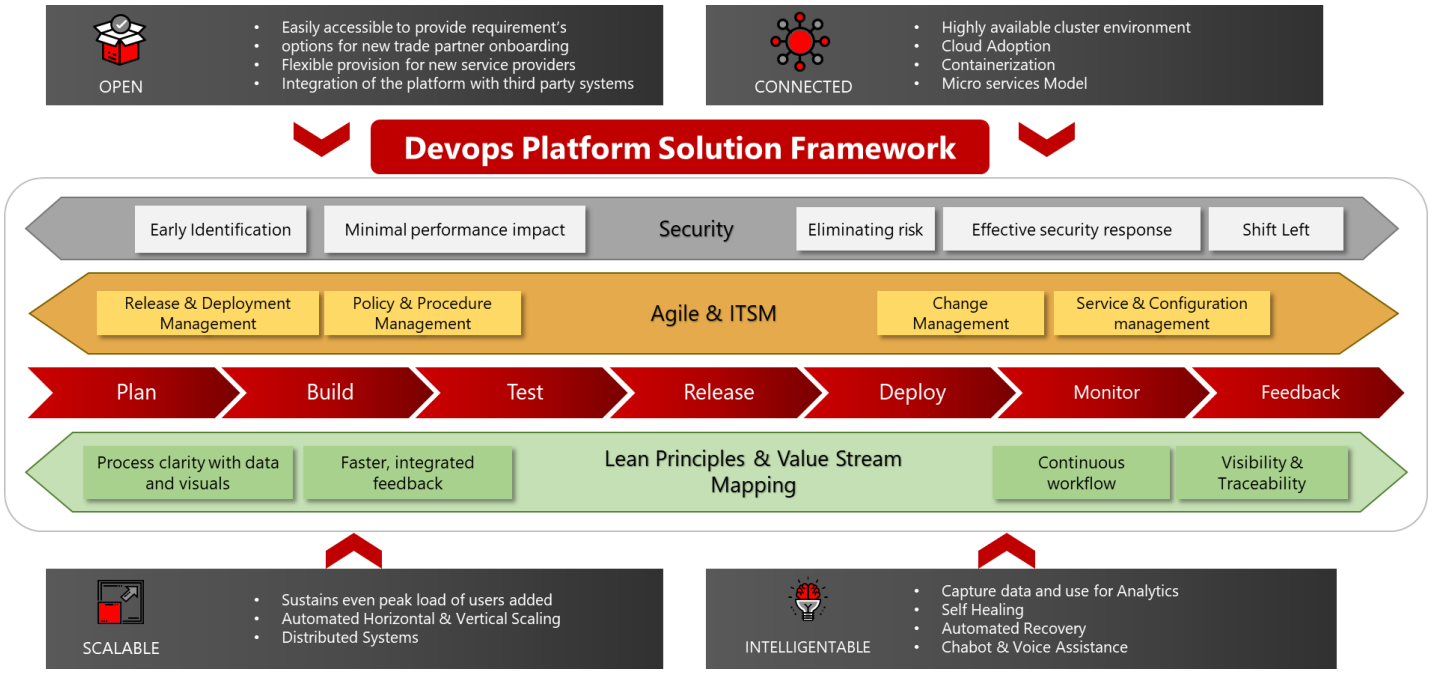
DevOps Engineering Enablement & Execution Platform Solutions framework

DEEEPS

Sonata Devops framework is built on various Devops pillars and in conjunction with ITSM ,Security , Value stream and platformation



Devops Platform Solution Framework



Approach (1/2)

"What" must be analysed, designed and developed!?

Determine Current state → **Define Future state attributes** → **Design DevOps Pipeline and template** → **Develop DevOps components** → **...Implementation planning and execution**

Current challenges and future expectations

- Environments and infrastructure
- Product / applications landscape
- Testing environment and process
- Release process
- Toolset
- Release process

Design | **Develop** | **Implement** | **Test**

- Template per pipeline
- Toolset, triggers / events, agents, automation and scripts
- Configuration (system, assets / packages required)

Approach (2/2)

"How and With whom"

Determine Current state → **Define Future state attributes** → **Design DevOps Pipeline and template** → **Develop DevOps components** → **...Implementation planning and execution**

How?

- Data collection template and existing documents
- Discovery and design workshops
- Working sessions with part of the team (tech architect and operational leads)
- Weekly stakeholder alignment on design / planning decisions

With Whom?

- Operations / DevOps responsible
- Development
- Technical architect for the portfolio
- Infrastructure responsible
- Business stakeholder representative (part)

→ Working (core) team + 3rd party package specialists (if req.) – TBD. → Implementation team

Approach DevOps process maturity improvements

Sonata Devops Maturity Assessment model

	CI	CD		CT	CD
	Build and CI	Environments & Deployments	Release Management	Testing	Data Mgmt
Level 5	• Feedback loop, • All Steps Automated • All CI Steps at final goals	• All Environments managed and • Provisioning Automated	• Metrics used to reduce risks and cycle time • Minimal downtime	• Product rollbacks and issues are rare	• Release monitoring
Level 4	• Build Metrics Captured • Maturity Plans in place • Artifact repo in place	Orchestrated Deployments Release and Rollback automated	• App Health Monitored • Cycle Time Monitored	• Quality metrics monitored • NFRs Tracked	• DB Performance Monitored/optimized • Cycle Time Monitored
Level 3	• All builds automated • Every commit tested • Code quality tested	• 1-Click Deployments • Automated Deployment to All Environment	• Change mgmt. and approval process defined • Compliance managed	• Auto testing part of development work	• Databases changes are automated and deployments
Level 2	• Schedule Auto Builds • Code repo, Unit tests in place	• Automated to Some Env • Automated environment creation	• Painful infrequent releases • Limited traceability	• Automated Unit and Acceptance Testing	• Automated Scripts for Database releases and are versioned
Level 1	• Manual Builds, • No artifact mgmt • No Quality Checks	• Manual environments, • Manual Deployments • Environ specific artifacts	• Infrequent and unreliable releases	• Manual Testing	• Data and Databases deployed manually and unversioned

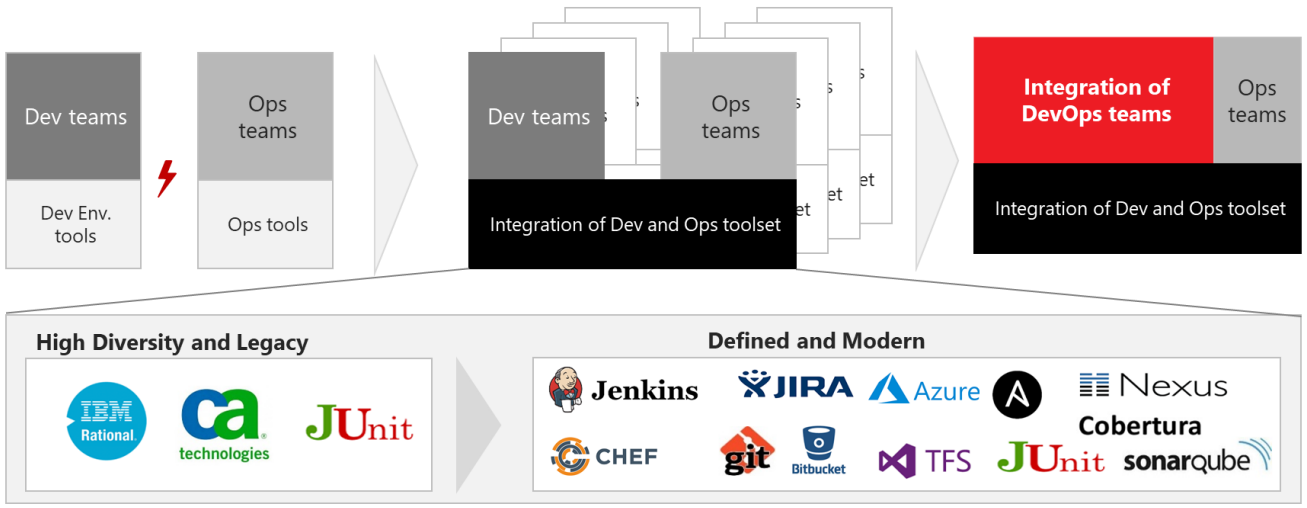
Approach

The definition of a common toolset is an important step in DevOps implementation (*Indicative toolset below)

1 Original Operating Model

2 Toolset platforms and scaling

3 DevOps team integration



Reference Cases

Case – Leading US Mortgage Provider

Context and need

Re-engineered Product

- Expedite testing of the Re-engineered, Cloud enabled product on the new platform

Process

- Centralized Governance and Streamlined Process
- Optimize Effort & Costs

Key project outcomes

- Improved Resource Rationalization by 20%
- 30% reduction in TCO
- 30% reduction in time to release
- 100% test automation for the business critical scenarios

Project Summary

Tool Consolidation & Landscaping

- Centralized , Cloud deployed Test Management Platform
- Test Requirements & Management, Automation, Performance, Functional Regression
- Integration with Redline

Governance & Best Practices

- Process Standardization & Metrics Driven Governance
- Reusable test components & Template Repository

Testing & Automation

- Designed and implemented Test Automation strategy the re-engineered product
- Automated 200 scenarios (> 1,000 test cases)
- Deployed platform for 4 business groups, with automation covering 100+ Critical business flows

Optimized Operational Model

- KPI & Metrics driven governance model
- Optimized tool stack for testing - Test Mgmt. & Execution

Reference Cases

Case – Largest Leisure Travel company in the world

Context and need

- As part of a new set of platform programs TUI wanted to move to a modern Cloud based operation stack.
- It was decided to build a DevOps platform that could serve as a template across all development streams.

Key project outcomes

- Setup of a team of 30 DevOps engineers
- Significant decrease in manual intervention
- Deploy to Cloud and full automate wherever possible (>30% of process)

Project Summary

What was implemented

Continuous integration and delivery

- Purpose designed pipelines
- Automated pipeline configuration
- Automated (docker) deployment to AWS

Automated testing

- Testing automation
- Test libraries with service virtualisation
- Automated defect reporting

Cloud deployment

- Dynamic environment provisioning
- auto-tearing down environment after test
- Automated recovery

Tech stack

- SCM - BitBucket
- Pipeline Manager - Jenkins
- Orchestrator - Ansible
- Shared Data store – Json Data
- Code Quality analyzer – SonarQube
- Service Virtualization – CA Lisa
- Static Assets - Docker
- Current build and deployment environment - AWS

Case Study - Managed DevOps Services for a World's largest Insurance company (Crawford)

Background

Customer

Crawford & Company is the world's largest independent providers of claims management to the risk management and insurance industry as well as self-insured entities, with clients in more than 70 countries.

Objective

- DevOps platform Design and Setup.
- DevOps environment provisioning and Maintenance.
- Automated Build and Release engineering.
- DevOps implementation & Service

Engagement Highlights

- Implemented DevOps platform comprises of Azure DevOps, Azure Container Registry, Azure Kubernetes Services, SonarQube, Selenium etc.
- Automated Continuous Integration & Continuous Deployment process
- Monitoring, Alerts & Feedback to support Development & Testing Teams
- Automated Pipeline & orchestration workflow approach across every phase

Benefits

75% Decrease in build & integration time	**40%** Reduce Implementation Failure, Reflections and Recovery Time
55% Cost saving due to automation	**35%** Faster time to market
30% Shorter Development Cycles, Faster Innovation	**50%** Increase Communication & Cooperation

Reference Cases

Case – Largest Insurance company in the world

Context and need

- As part of a new platform programs Crawford wanted to Replicate their multiple UK Business Model into a newer region & as One Business Model programs
- It was decided to implement a DevOps Culture so that it could serve faster & efficient development streams.

Key project outcomes

- Setup of a team of 30 DevOps engineers
- Significant decrease in manual intervention
- Deploy to Cloud and full automate wherever possible (>30% of process)

Project Summary

What was implemented

Continuous integration and delivery

- Custom Designed Pipelines
- Approval Based Automated Build Configuration
- Environment specific application Configuration
- Containerization & Orchestration Services
- Automated container deployment to Azure AKS

Cloud deployment

- Azure Container Registry & Azure Kubernetes
- Automated Recovery

Tech stack

- SCM – Azure Repos
- Pipeline Manager – Azure Pipeline
- Shared Data store – JSON Data
- Code Quality analyzer – SonarQube
- Static Assets – Azure Container Registry
- Container Orchestrator - Azure Kubernetes Services
- Current build and deployment environment – AKS Cluster