PC641 - MS-IT Project

Winter Internship Project Report - 2019

Internship Details

Start Date: 2nd Jan, 2019

End Date: 9th May, 2019

Duration: 17 weeks

Mentor

Niraj Chapla

Technical Lead at OpsHub Technologies Pvt. Ltd.

Company Details

OpsHub Technologies Pvt. Ltd.,

326-327 Ratna High Street,

Naranpura Cross Roads,

Ahmedabad, Gujarat 380013.

Website: https://www.opshub.com/

Submitted To

Prof. Manish Khare

Submitted By

201712001 - Archan Ranade

Submission Date

3rd May 2019, Friday

Index

1. Introduction

1.1 About OpsHub

OpsHub is the leading provider of integration and migration solutions for the ALM, CRM, and DevOps tool chains. OpsHub improves the efficiency and effectiveness of agile teams in ALM and DevOps environments by making necessary and current data available to each user, in that user's preferred system, with full context, in real-time.

OpsHub's solutions for integration and migration speed up development processes, reduce errors, improve decision-making, and result in delivering innovative products and services faster, with higher quality at lower cost.

1.2 About OpsHub's Products

OpsHub has currently two products:

1. **Opshub Integration Manager (OIM) -** OpsHub's flagship product, OpsHub Integration Manager seamlessly integrates 50+ ALM, ITSM, CRM, and DevOps systems. It ensures that accurate and current data is available to each user, in that user's preferred system, with full context, in real-time.
2. **OpsHub Visual Studio Migration Utility (OVSMU) -** is a migration utility exclusively designed to support data migration in Microsoft ecosystem. Using OVSMU, enterprises can migrate data between TFS-TFS (Team Foundation Server), TFS-Azure DevOps (Earlier VSTS), Azure DevOps-TFS, and Azure DevOps-Azure DevOps. OVSMU supports migration between all versions of TFS and Azure DevOps.

2. Scope

OpsHub supports more than 50 systems for integration, one of them is a popular desktop-based application called Enterprise Architect (EA) developed by Sparx Systems. EA is an enterprise wide solution to visualise, analyse, model, test and maintain all of your systems, software, processes and architectures. EA has different types of entities and OpsHub was supporting only one type of entity in EA known as Element. Elements are mainly Use cases, Requirements, Test cases, etc which are a part of system modelling and testing. My project was to support integration of Diagrams in EA along with all the other functionalities which OIM provides such as reconciliation and recovery. Diagrams can be UML as well as non-UML.

3. Use Case

Problem Statement: Customers who use OIM to synchronize and/or migrate their data from EA to some other system want to be able to sync/migrate Diagrams as well.
Solution: Add support for Diagrams in OIM by adding a new entity type in OIM for EA, create poller and adapter for Diagrams and support recovery for Diagrams. After this implementation, customers will be able to sync/migrate Diagrams as well. Since Diagrams contain Elements, this linkage between a Diagram and the Elements it contains can be synced as well. When Diagrams are migrated to systems other than EA, a PDF of the diagram will be attached to the synced Diagram in the target system as well.

4. Design Contribution

1. Restructured connector implementation: The previous implementation was structured to support only one entity type i.e. Element. My solution was to restructure existing code in such a way that current behaviour of Element does not get changed. To implement a new entity in existing code, *Factory pattern* was implemented where I created two entity handlers: ElementHandler and DiagramHandler. The existing business code for Element was moved to its handler, while new implementation was written in DiagramHandler for Diagram. Interfacing was used to enforce common business functions.
2. Resolve existing bugs and problems for EA connector: Before beginning with restructuring and implementation for Diagrams, automation tests were written which test all scenarios and functionalities for each connector. There were some issues with too many open connections being created during automation. The manner in which OIM connects with the EA system was restructured to restrict number of open connections which is only one per user. To communicate with EA, OIM has implemented a .NET service for it. This service can run through console as well as as a Windows service. Problems found in Windows service were addressed as well.

5. Programming Contributions

1. Implementing factory pattern: Factory pattern was implemented for supporting entity types in EA connector, where Interface was created to enforce similar behavioural pattern between two entity types. Two entity handlers were created for each entity type.
2. Enhancing EA Connector: Adapter was modified to implement Diagrams as well along with Elements. Poller was modified to query end system for Diagram as well.
3. Fixing automation failures: Solving existing Bugs and some infrastructure issues on automation servers. Since I followed TDD (Test Driven Development), my code was being tested everyday and failures were being addressed as they came.
4. Wrong HTML and RTF format was handled: The EA API was giving HTML and RTF data in a slight non-formatted manner. This was handled as well through the code.
5. Recovery support for EA Diagrams: There are no custom fields in EA for diagrams. Hence recovery scenario was slightly tricky to support, but it is supported now.
6. Linked Document support in EA Diagrams was added in OIM as well. To sync it as an RTF field in target system as well as to add it as an attachment was implemented.

6. Other Contributions

1. Documentation: Documentation was added in OpsHub Support Guide Doc which is helpful to customers to configure systems and troubleshooting, for Diagrams of EA.
2. Meetings with customer: There is one active customer for EA with whom weekly meeting were scheduled to discuss progress as they had to migrate their EA data to another system. I attended these meeting in order to know customers expectations.
3. QA and testing for OIM: Before releases, QA work is assigned as well. So test cases were assigned to perform QA on OIM.
4. Bug fixation: After the initial training period, two bugs were assigned for introduction to code base and getting to know how the code base. These bugs were fixed as well.

7. Tools, Technologies, APIs and Libraries used

Following tools and technologies were used:

1. Java: For writing test cases and enhancing Java side implementation
2. C#: For EA service code restructuring and Diagram implementation
3. EA APIs: For communicating with the EA app
4. Git: For managing source codes
5. Jenkins: For continuous integration
6. TestNG: For automation testing
7. Apache Ant: For building the code

8. Testing strategies

1. **Test Driven Development (TDD):** OpsHub primarily follows Test Driven Development as the strategy for testing in which a developer writes all test cases first and start the development. To follow this strategy, I enabled all automation test cases for my project and generated all the possible scenarios to test for integration/migration.
2. **Continuous Integration/Continuous Deployment:** The company also follows CI/CD approach to ensure the quality of deliverables. All the test cases are automated and executed daily, and for any reason if any test cases get failed, report has been generated and Failures on created on Azure Devops and then assigned to proper assignee which helps to improve the product quality.

9. Lessons Learnt

I learnt a lot of new things in these four months of my internship. Also the things which I had learnt in classroom and books were put to test with a lot of practical implementation.
Here are some of the things I learnt:

1. Code design and design patterns are a crucial part of a good product.
2. I learnt new things about the integration industry and why a lot of different organizations across the world use integration products to integrate their different tools to improve the efficiency of their teams.
3. I learnt how meetings with customers are scheduled and how to communicate with customers.
4. I learnt new ways of code testing and automation. I also learnt how important testing is to ensure quality of a product and how to write automation tests.
5. I learnt how using appropriate tools increases productivity.
6. I learnt how Agile methodologies are practiced in the industry and how useful they are in delivering on time with quality.
7. I learnt how important documentation is before starting with coding. Also how important User Manual is to the customer and it should be regularly maintained for the ease of customer’s usage.