



College of Engineering and Informatics  
Professional Experience Programme (PEP)  
Student Experience Programme (SEP)  
Project on Campus (PoC)  
V0.4 2015

## Formal Report

Submitted in fulfilment of the requirements for the B.E./BSc (Hons) Degree in

Computer Science & Information Technology

---

<b>Student Name</b>	<b>Student ID</b>	<b>Signature</b>	<b>Date</b>
Kevin Cusack	14102730	<i>Kevin Cusack</i>	13/07/18

---

Company Name:

SAP

Company Address:

Parkmore, Galway, Co Galway

---



<b>Authorised for submission:</b>	<b>Name</b>	<b>Signature</b>	<b>Date</b>
Company Supervisor	Eanna Hegarty	<i>Eanna Hegarty</i>	13/07/18

Reviewed by NUI Galway Reviewer

---

## Statement of Originality

I hereby declare that this report is my original work except where stated.

Signature: Kevin Cusack

Name: Kevin Cusack

Date: 13/7/18

## **Acknowledgements**

---

I would like to thank most sincerely Eanna Hegarty and Nadia Swords in their continuous support as an intern within SAP Galway.

# Table Of Contents

---

<b>Acknowledgements .....</b>	<b>iii</b>
<b>Table Of Contents .....</b>	<b>iv</b>
<b>Glossary v</b>	
<b>Chapter 1. Introduction.....</b>	<b>1</b>
1.1. Introduction.....	1
1.2 PEP Host Department .....	1
1.3 Mentoring.....	1
1.4 Testing.....	1
<b>Chapter 2. The Host Organisation/Company .....</b>	<b>2</b>
2.1. Introduction.....	2
2.2. About SAP.....	2
2.3. Galway Operations .....	2
2.4. Cloud Based Applications .....	2
2.5. Culture at SAP.....	3
<b>Chapter 3. Description of PEP Technical Activities and Technical Achievements.....</b>	<b>4</b>
3.1. Introduction.....	4
3.2. REST API.....	4
3.3. POSTMAN.....	5
3.4. Setup Installer – Inno Setup.....	6
3.5. SMT Updater for Linux .....	7
3.6. Documentation.....	7
<b>Chapter 1. Teamwork and Business Experience Acquired .....</b>	<b>8</b>
1.1. Introduction.....	8
1.2. Role with SAP SLS.....	8
1.3. SMT Updater EPIC.....	8
<b>Chapter 2. Achievement of Learning Outcomes and Programme Outcomes.....</b>	<b>9</b>
2.1. Introduction.....	9
2.2. LO1: Information Skills .....	9
2.3. LO2: Personal Management and Project Planning .....	9
2.4. LO3: Analysis, Interpretation and Problem-Solving Skills .....	10
2.5. LO4: Opportunity Awareness and Decision-Making Skills .....	10
2.6. LO5: Communications Skills.....	11
<b>Chapter 3. Support Provided by the Company.....</b>	<b>12</b>
3.1. Introduction.....	12
3.2. Online Learning.....	12
3.3. Code Review .....	12
3.4. iXp Summit - Dublin .....	12
<b>References .....</b>	<b>13</b>

## Glossary

---

API	Application Programming Interface
C#	C Sharp
ERP	Enterprise Resource Planning
IoT	Internet of Things
iXp	Internship Experience Project
JSP	Java Server Pages
OS	Operating System
PEP	Professional Experience Programme
REST	Representational State Transfer
SLS	Software Language Services
SME	Small to Medium Size Enterprises
SMT	Server Monitoring Tool
SWOT	Strengths, Weakness, Opportunities and Threats

# Chapter 1. Introduction

---

## 1.1. Introduction

This report will focus on information about the company, technical activities and achievements, teamwork experience (within a team and paired programming), learning outcomes attained using technologies such as Jira, C#, Java, RESTful API, Junit testing and creating an EPIC for automating the software upgrade of the SMT application autonomously.

## 1.2 PEP Host Department

Duration of PEP was with SAP Software Language Services (SLS). As previous employed with a chemical reagent plant where I localised software for different regional dialects, it made sense to go with the SLS department within SAP. SAP Galway is primarily a support centre, with software development groups.

## 1.3 Mentoring

I was mentored/trained and assigned to refactor code on an application called Server Monitor Tool (SMT), used to monitor remote Windows and Linux operating systems (OS) servers. SMT returns server system information such as CPU usage, disk space available, files, services and so forth to a backend database.

## 1.4 Testing

Created JUnit tests to familiarise with the inner workings of both the SMT front and backend. The backend been an Application Programming Interface (API), with access to a database containing server's data. The front end been the SMT client installed on the local server.

## Chapter 2. The Host Organisation/Company

---

### 2.1. Introduction

Introduce SAP and its position in the global market. What SAP's operations in Galway, Ireland are. Introduce primary products supported in Galway. Culture within SAP.

### 2.2. About SAP

SAP is the world leader in enterprise applications in terms of software and software-related service revenue. Based on market capitalization, they are the world's third largest independent software manufacturer (SAP, 2018).



**Figure 1: SAP Company Logo**

### 2.3. Galway Operations

SAP's operation in Galway is a multilingual service and support centre for SAP's Small to Medium Size Enterprises and Multinational Enterprise sectors. It also has teams responsible for supporting, Cloud Services, IT Support, Technical Writing, E-Learning, Translations, License Auditing and Partner Services. It covers 29 languages and has colleagues from 50 nationalities. (SAP, 2018).

One of SAP's strengths is on Enterprise Resource Planning (ERP) software used by enterprises to manage some part or all their systems such as monitoring their inventory, customers and suppliers.

### 2.4. Cloud Based Applications

Two SAP cloud based applications been supported

**SAP Hana:** Column orientated relational database management system, whose primary function is to store and retrieve data as requested by the application. It performs advanced analytics (predictive analysis), text searches and graph data.

**SAP Leonardo:** SAP have a focus to moving towards cloud based computing with the introduction of SAP Leonardo. Used to create new applications or extending existing applications in a secure cloud computing environment managed by SAP. It delivers software and micro-services that enable customers to leverage future-facing technologies like the Internet of Things, machine learning, block-chain, analytics, and Big Data. (SAP, 2018).

## **2.5. Culture at SAP**

SAP embraces an innovative culture throughout the entire company. With a network of interconnected programs and entities covering all dimensions of innovation, from research and development to customer implementation.



## Chapter 3. Description of PEP Technical Activities and Technical Achievements

---

### 3.1. Introduction

Discuss technologies and application of use used within the SMT application. Introduce a Windows setup installer application used to deploy software on Windows OS's. How options for SMT upgrades are presented and applied to the installer. Created documentation detailing steps in creating a software installer with Inno Setup so it can be used for later releases.

### 3.2. REST API

SAP gave free unlimited access to 'Lynda.com', which was used extensively throughout this placement. One of the first courses taken on Lynda.com was the JAX-RX course with RESTful (Representational State Transfer) API. It is used to make a connection from a client to a server over HTTP Protocol.

JAX-RS (Java API for RESTful Web Services).

The purpose of this JAX-RS is to provide a set of APIs that can simplify the development of REST-style Web services. Jersey is the reference implementation for JAX-RS, and contains the following parts.

- *Core Server:* By providing annotations and APIs standardized, you can develop a RESTful Web Service Core Client: The Jersey client API helps you to easily communicate with REST services
- *Integration:* Jersey provides libraries that can easily integrate with Spring, Guice, Apache Abdera...

For JAX-RS Web Service API to run on a server it will require a web distribution system such as Tomcat in Eclipse. This is a web container for RESTful web applications, that runs the API (War file). It unwraps the files from the War file and runs them. For testing purposes, it was ran locally using Apache Tomcat 8.

In the following code, some points need to be highlighted. This is the Core Server from above, which provides annotations and APIs standardisation. The following are examples of annotations.

- **@Path:** This annotation defines the resource base URI. Formed with context root and hostname, the resource identifier will look like  
<http://localhost:8080/ServerMonitor/rest/json/files>  
When calling from the Web, it uses this path to get content.
- **@GET:** The following method responds to the HTTP GET method. Using POSTMAN web client, and running the GET statement will return objects from the web application. Calling a GET statement with the following would return a plain text string "Hello World".  

```
@GET
@Produces(MediaType.TEXT_PLAIN)
public String getMessage(){
    return "Hello World";}
```

- @Produces: Defines the response content MIME type such as TEXT\_PLAIN, Json or XML. when called, this would return an XML formatted object.  
@Produces(MediaType.APPLICATION\_XML).

### 3.3. POSTMAN

**POSTMAN:** Was used to test REST API's with the use of GET, and POST requests. Objects were created using the POST request, and primarily in Json format. Objects were retrieved from the API using the GET request, and this too was primarily in Json format.

POSTMAN can also be used to test the response time of a request to a database.

tests ["Response time is less than 80ms"] = responseTime < 80;

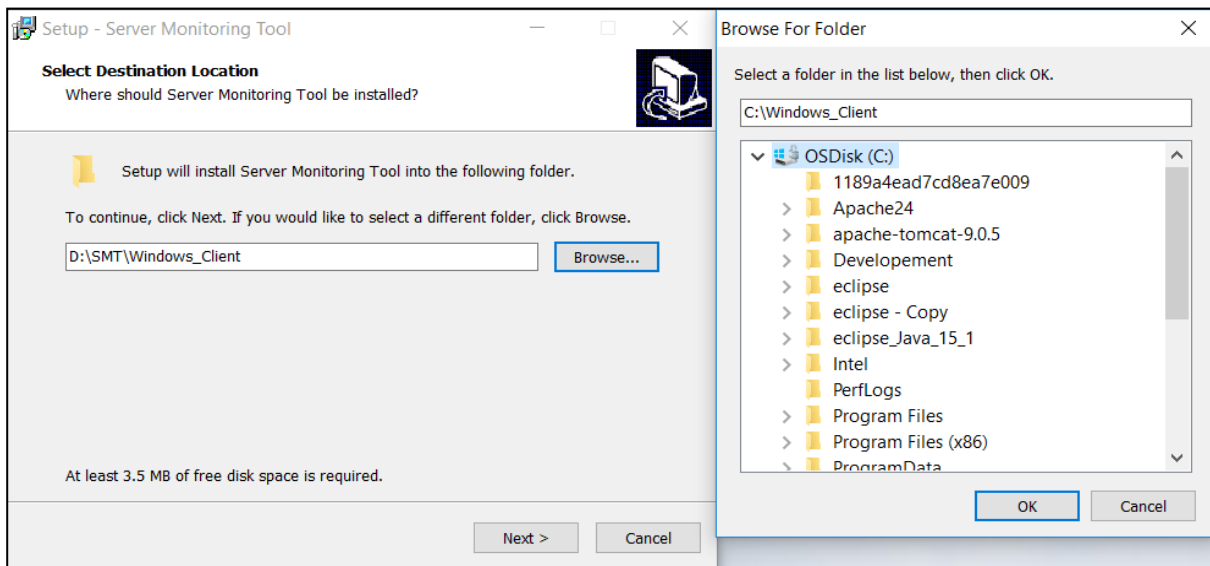
I was assigned an EPIC, where I re-factored the SMT C sharp (C#) code (for Windows OS system) so should a new SMT upgrade available, then the SMT will download and install the upgrade without user interaction.

Researched online on how other developers approached this problem. The following is how it was to be coded and compiled.

1. The refactored SMT tool and new Updater service would initially have to be installed manually, as the Updater service must be initially installed on the local system. But future updates would install autonomously.
2. The Updater Epic/User stories/Sprints was coded in the following order
  - Check what version of the SMT is installed on the local machine
  - Given a version.html URI, check the [version] tag value within this file
  - Compare installed SMT version with that of the Live version. If live version is greater than the version installed then do the following
    - Given a URL, get the value of the [URL] tag
    - Download .zip file from the above value
    - Extract the files from this .zip file to the SMT installation directory
    - Delete this .zip file
    - Launch the SMT application.

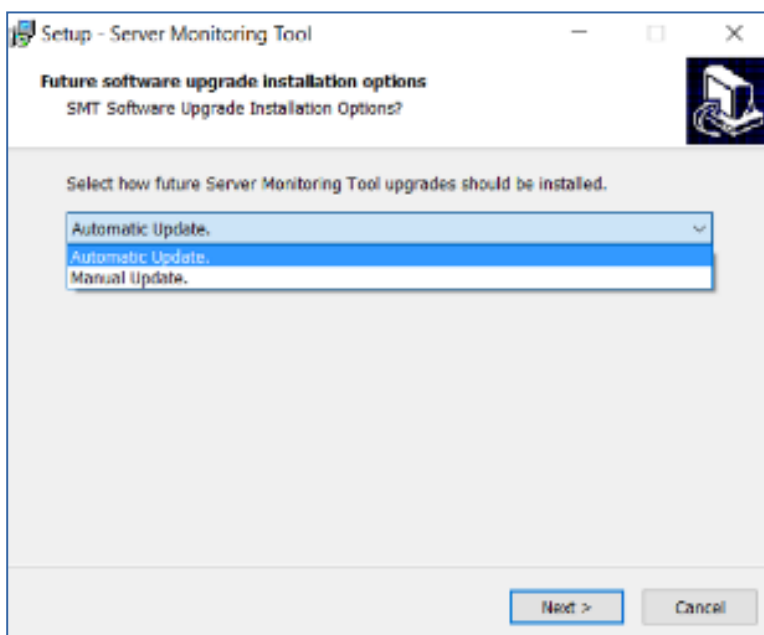
### 3.4. Setup Installer – Inno Setup

**Point 1:** Researched online and Used Inno Setup software to create the initial Software Installer Setup. This allows the user to install the SMT tool and Updater service manually on the local machine. The user will have the option to install it to a Default directory or browse a directory of choosing.



**Figure 2:** option for the installer to select an installation directory

The software installer has the option to allow future updates to install automatically or to disable this feature completely.



**Figure 3:** Option to select if SMT should be upgraded autonomously or not

When the setup installer installs the SMT source files on the local machine, it adds two registry keys containing the Installation directory, and whether upgrades will be manual or automatic. After the installation is completed Inno Setup runs a batch file that launches the application on install. The application will have an entry in the 'Add or Remove programs'. This is where the user has the option to uninstall this application.

**Point 2:** Refactored C# code for Windows OS's using Eclipse IDE. Created an ExecutorService to run daily comparisons of the Installed version of SMT with that of the Live version. Configure to run the SMT after a reboot to ensure SMT is always running.

Conducted local and remote testing on a Windows OS server to test proof of concept.

### 3.5. SMT Updater for Linux

The Updater service above was also created for a Linux OS but using Java as opposed to C#. This is where VMWare Player was installed and ran an Ubuntu Linux OS image for coding/testing. Linux is more technical, as it extensively uses command prompts when creating/moving files/folders, and running .jar files, and will require elevated permissions. This is due to Linux been a more locked down system.

### 3.6. Documentation

Created a SMT installation instruction document detailing the manual steps to install this application. Created Inno Setup document indicating how to create a software installer setup using Inno Setup, so as this information is not lost, and can be used for future software installer builds.

# **Chapter 1. Teamwork and Business Experience**

## **Acquired**

---

### **1.1. Introduction**

Role within SAP SLS. Discuss paired programming and the benefits it had on the placement. Discuss a software management tool used by SLS for software development, and keeping track of projects so other team members can view the status of projects.

### **1.2. Role with SAP SLS**

My internship title was a software tester/developer within SAP SLS department. During placement there was another intern (Daniel Regan) where we worked side by side. After three weeks we were assigned to do Junit tests on the SMT application. This was done using paired programming where both of us sat in front of a single computer. This was beneficial as it showed both our strengths and weakness. If one of us missed something then the other could possibly see where we need to modify something else. If both of us were stuck, we could troubleshoot by doing tests in stair step fashion, or reverse engineer it to see where we were making mistakes. Searched online for code examples, and at times found code that was more efficient than other code. This was a good lesson learned on this placement. With paired programming the coding time was reduced as it was two people coding on a single computer.

### **1.3. SMT Updater EPIC**

After two months each of us were assigned different Epic's, and teamwork was reduced. But it did increase my confidence where I was given a problem and come up with a solution. My role was to refactor an existing application SMT. In this case the live application SMT required an option to update automatically or to update manually. The Epic was documented in a management tool used by the SLS group called 'JIRA', and used for ticketing and keeping track of projects. Tickets are visible to all team members. If the epic, user story or sprint required a different member to work on it, then it could be assigned to them. Should another member of the team require to see the status of this epic, user story or sprint they could view it all through Jira.

Changes to code was committed to GitHub using software called 'SourceTree'. GitHub is used as a store by developers to push and pull code. Any change to the code you would leave a comment indicating what you changed. This code could then be reviewed by my supervisor and any change requests I would then refactor and re-commit the updated code. Other team members can also view this code and do a comparison on this code to previously committed code.

## **Chapter 2. Achievement of Learning Outcomes and Programme Outcomes**

---

### **2.1. Introduction**

Discuss the achievements and the learning outcomes of the PEP programme.

1. Information skills
2. Personal Management and Project Planning
3. Analysis, Interpretation and Problem-Solving Skills
4. Opportunity Awareness and Decision-Making Skills
5. Communications Skills.

### **2.2. LO1: Information Skills**

Assigned an Epic, consisting of checking if the installed SMT is the latest version, and if not, then the latest version should be downloaded and installed on the local machine. The Epic was then split to user stories and sprints.

For this Epic I researched online to see how other developers approached this issue and how they planned it out. Researched my Epic, User stories, and Sprints online, searching for C# and Java code examples online.

Researched online examples of UML diagrams, and created a UML diagram showing the connection between methods and what variables are been passed between them. Documented and uploaded this in Jira.

### **2.3. LO2: Personal Management and Project Planning**

An Epic is a top-level project that must be broken down to user stories and broken down further to sprints. By splitting the problem into smaller problems, this gives structure to finding solutions to the problem.

After researching online how developers solved the problem to autonomously update software, I then formulated a plan.

Created a UML diagram to show how methods will interact with other methods. What variables and their data types will be passed/returned to other methods. The UML diagram was uploaded to Jira as a central location for the Epic's progress and store of documentation and emails. As the project was advancing I was updating the UML document as variables and their data types were changing, along

with creating addition methods. Tried to reduce changing the code and updating the UML document to a minimum.

During the day my personal 'Notes' document was getting updated with any information found useful or containing steps on how to use technologies used for development. Saved code samples in a separate document.

My supervisor was always available should I have any queries, and the Epic was discussed at the daily scrum meetings.

## 2.4. LO3: Analysis, Interpretation and Problem-Solving Skills

Testing was conducted on the local machine with a locally installed Tomcat web server to emulate the version.html and url.html web pages containing information on what the latest version of SMT is available and the URL to where the upgrade can be downloaded from.

The Updater service in development was ran locally, and used to check what version of SMT is currently installed to that on the local web server.

When developing the updater service, one method was coded at a time and tested. If successful, then the next method was coded as per the UML diagram.

As I was coding for an autonomous software upgrade, testing would be extensive and should I come across an exception then a method was created to solve this exception as the application would fail to upgrade, and the monitoring tool would not upload a server's system state to the REST API.

Most of the code was exception handling and ensuring the SMT is always running. One such exception handling was if there is no network access to the version.html file, then the Updater service would ping this URL three times and if still no connection, then it would stop checking for the version installed and leave the SMT running as normal. The Updater service required to run daily, and as such a scheduler was created.

## 2.5. LO4: Opportunity Awareness and Decision-Making Skills

The following applications were used to assist in decision making in regards what Java code to use, and best practices.

**JUnit** (Viewed course on Lynda.com): JUnit is a framework to write repeatable tests, and is used in test driven development. Started out by testing a very simple test and ensuring it passes. This was built on with more advanced testing. This promotes the idea of "first testing then coding". This approach is like "test a little, code a little, test a little, code a little." It increases the productivity of the programmer and the stability of program code.

**SonarQube**: Was used for continuous inspection of java code quality, and used to perform automatic reviews with static analysis of code to detect bugs, code smells and security vulnerabilities. null-pointers dereferences, logic errors, resource leaks. This also gives hints on what to change and if

some variables are not used. It can give alternative code and examples, such as if your code has multiple same string text, it will prompt you to assign this text to a String and use the variable where the text is used.

Code was also reviewed by my supervisor Eanna Hegarty, who at times gave alternative / better options.

## **2.6. LO5: Communications Skills**

Daily scrums every day at 11:30 with my supervisor, another intern and I. Discussed what we were working on and if there is anything in our way. Asking questions such as what code is more performance optimised than others.

Trained in SAP BUILD which is used to prototype an interface. Shortly afterwards training was rolled out to members in the SLS team where they would have to build a web app. I was assigned to assist anyone at this training if they were having difficulties and assist them in getting their web app up and running.

Attended the SAP iXp summit in Dublin for three days and met other interns from Galway and Dublin. One speaker noted that interns need to increase their LinkedIn connections. With the interns I met I added to my LinkedIn profile. Some of these interns I meet on a daily and weekly basis.



## **Chapter 3. Support Provided by the Company**

---

### **3.1. Introduction**

Support supplied by the host PEP company, both online and by a supervisor. Code reviews by the supervisor. Outcome of a SAP intern summit in Dublin.

### **3.2. Online Learning**

SAP supplied considerable support. Gave free unlimited access to 'Lynda.com', which was used extensively throughout this placement.

### **3.3. Code Review**

Eanna Hegarty was the supervisor and any queries about code I could go to him. Communication with Eanna was through daily scrum meetings, Slack communicator, email and was in close proximity. Initially last thing on a Friday Eanna would take Daniel (intern) and I to a conference room for two plus hours, and we would do code review along with coding for specific tasks. Took turns in plugging in our laptop to the projector and discuss the pros and cons of different code.

### **3.4. iXp Summit - Dublin**

Attended SAP iXp summit in Dublin for three days, and used to introduce interns from SAP Galway with SAP Dublin. The summit consisted of

- SAP employees speaking from life experience of facing adversaries
- Making connections through Linked-in and Facebook
- Introduce to SAP Leonardo (Cloud computing)
- SWAT analysis on a problem
- Team Work: building a bike.

## References

---

- [1] SAP 'SAP Company Information'.  
Available at <https://www.sap.com/corporate/en/company.html> (accessed 22th June 2018)
- [2] SAP 'SAP in Ireland'.  
Available at <https://www.sap.com/uk/about/careers/who-we-are/locations/dublin-galway.html>  
(accessed 28th June 2018)
- [3] SAP 'What Is SAP Leonardo'.  
Available at <https://news.sap.com/what-is-sap-leonardo-2/> (accessed 28th June 2018)