"Software Asset Management (SAM) Tool Development"

Major Project Report

Submitted in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Darsh Maniar
(19BEC069)

Department of Electronics and Communication Engineering,
Institute of Technology, Nirma University,
Ahmedabad 382 481

May 2023

CERTIFICATE

This is to certify that the Major Project Report entitled "Software Asset Management Tool Development" submitted by Mr. MANIAR DARSH HIREN (19BEC069) towards the partial fulfillment of the requirements for the award of degree in Bachelor of Technology in the field of Electronics & Communication Engineering ofNirma University is the record of work carried out by him/her under our supervision and guidance. The work submitted has in our opinion reached a level required for being accepted for examination. The results embodied in this major project work to the best of our knowledge have not been submitted to any other University or Institution for award of any degree or diploma.

Date: 15/05/2023

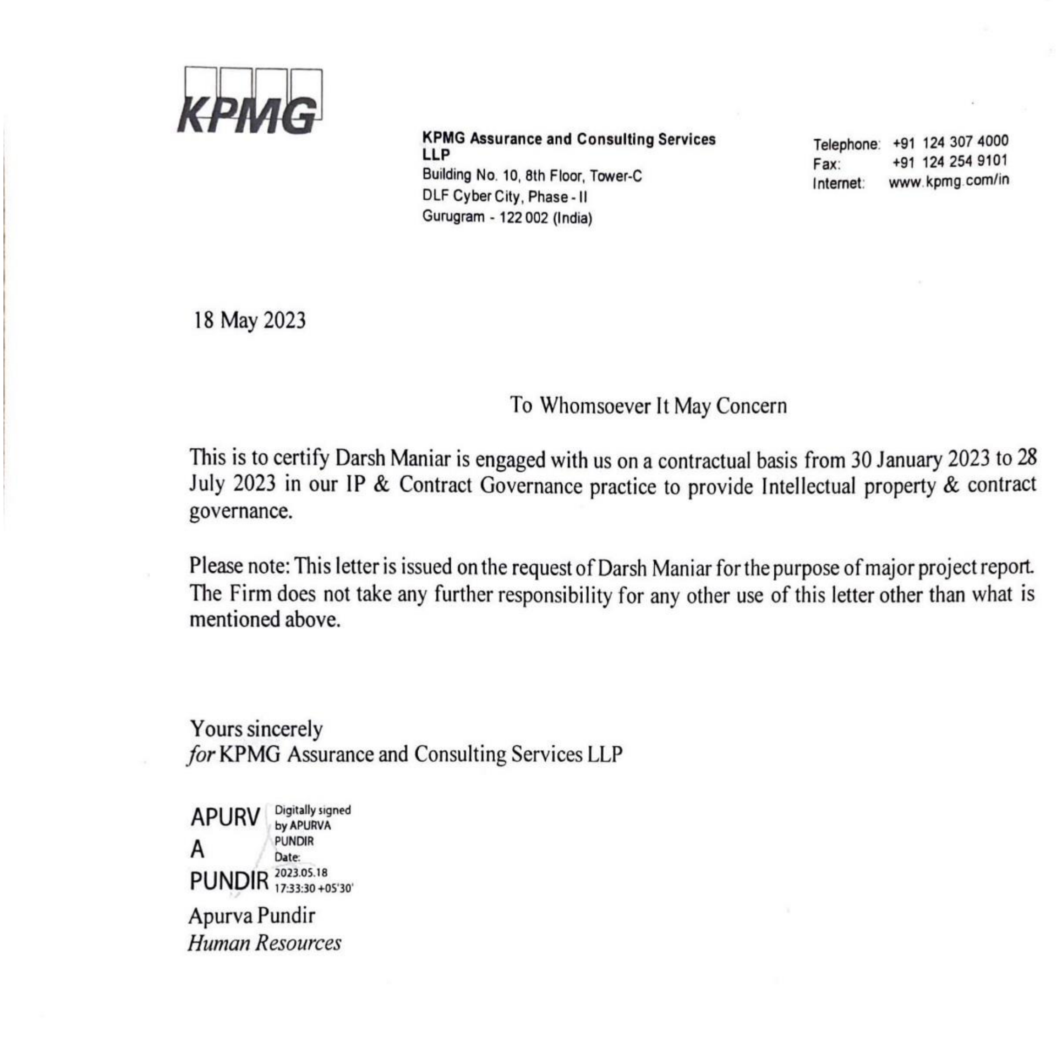
Industry- Guide
Ankit Kundu
Associate Director
KPMG,
16th floor NESCO,
Goregaon, Mumbai

Institute - Guide
Rutul Patel
Assistant Professor
Institute of Technology,
Nirma University,
Ahmedabad 382 481

Head of Department
Dr. Usha Mehta
Department of Electronics & Communication Engineering, Institute of Technology, Nirma University, Ahmedabad 382 481

Director
Dr. Rajesh Patel
Institute of Technology,
Nirma University,
Ahmedabad 382 481

CERTIFICATE



KPMG Assurance and Consulting Services LLP
Building No. 10, 8th Floor, Tower-C
DLF Cyber City, Phase - II
Gurugram - 122 002 (India)

Telephone: +91 124 307 4000
Fax: +91 124 254 9101
Internet: www.kpmg.com/in

18 May 2023

To Whomsoever It May Concern

This is to certify Darsh Maniar is engaged with us on a contractual basis from 30 January 2023 to 28 July 2023 in our IP & Contract Governance practice to provide Intellectual property & contract governance.

Please note: This letter is issued on the request of Darsh Maniar for the purpose of major project report. The Firm does not take any further responsibility for any other use of this letter other than what is mentioned above.

Yours sincerely
for KPMG Assurance and Consulting Services LLP

APURVA PUNDIR Digitally signed by APURVA PUNDIR Date: 2023.05.18 17:33:30 +05'30'

Apurva Pundir
Human Resources

Date: 18-05-2023

(Signature)

Mr. Ankit Kundu

Engagement Manager

Undertaking for Originality of the Work

I, Darsh Maniar, 19BEC069, give undertaking that the Major Project entitled "Software Asset Management Tool Development" submitted by me, towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in Electronics and Communication of Nirma University, Ahmedabad 382 481, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of any similarity found subsequently with any other published work or any project report elsewhere; it will result in severe disciplinary action.

Darsh

Signature of the Student

Date: 15/04/2023

Place: Ahmedabad, India

Endorsed by:

(Signature of External Guide)
Mr. Ankit Kundu

(Signature of Internal Guide)
Dr. Rutul Patel

Acknowledgement

I would like to express my sincere gratitude to everyone who has contributed to my enriching internship experience at KPMG.

I would like to express my heartfelt gratitude to Mr. Ankit Kundu, Associate Director at KPMG for helping me with the understanding of the topic and guiding me through the entire duration of the summer internship project. His guidance helped me in all the time of the Internship. I could not have imagined having a better advisor and mentor for Internship.

Besides my Immediate Supervisor,! would also like to extend my heartfelt appreciation to Mr. Mihir Kabani and Mr. Raj Mehta, members of my team at KPMG. Their unwavering support, insightful feedback, and willingness to share their knowledge have been pivotal in making this internship a truly fulfilling experience. Without their guidance, my journey would have been incomplete.

My sincere thanks also goes to The HR and Recruiting team at KPMG for providing me this opportunity for Internship at KPMG.

I am deeply grateful to Prof. Rutul Patel, my internal guide, for his constant support and belief in my abilities. His valuable insights, encouragement, and interest in my work have been tremendously motivating throughout my internship.

The successful completion of this Internship would not be possible without the dedicated support from all my mentors, family and friends.

Darsh Maniar

Abstract

Software applications have undergone rapid development, from web applications and mobile applications to cloud applications and software as a service (SaaS), indicating the start of "the software revolution". Software assets, such as licenses, installations, usage, and compliance, have become abundant and diverse, driving a revolutionary change in our information society. We are moving from the Software Ownership age (i.e., a single license per software) to the Software Optimization age in which an organization utilizes multiple software assets to access the required functionalities whenever and wherever needed.

Software Asset Management (SAM) tools are crucial for optimal management of software assets, which is not feasible with traditional manual methods that are prone to errors, inefficiencies, and inconsistencies. SAM tools are software applications that help organizations manage their software assets, such as licenses, installations, usage, and compliance. SAM tools help organizations track their software inventory, monitor their software usage, optimize their software spending, ensure their software compliance, and plan their software strategy. SAM tools are vital for fintech institutions, as they help them optimize their software spending by avoiding over-licensing or under-licensing, reduce security and legal risks by preventing unauthorized or unlicensed software use, and improve their operational efficiency by enhancing their software performance and availability. SAM tools also enable fintech institutions to leverage cloud computing, such as scalability, flexibility, and cost-effectiveness, by helping them manage their cloud-based software assets and services.

Software development of SAM tools involves using various technologies and methodologies that are suitable for creating robust, reliable, and user-friendly software applications. Some of the common technologies and methodologies used for software development of SAM tools are web development frameworks (such as ASP.NET or Angular), cloud services (such as Azure or AWS), databases (such as SQL Server or MongoDB), API’s (such as RESTful or Graph), agile software development (such as Scrum or Kanban), and DevOps (such as CI/CD or automation). Software development of SAM tools requires a high level of expertise and innovation, as the SAM domain is dynamic and complex. The SAM domain involves various challenges and opportunities that require constant research and development, such as evolving software licensing models (such as subscription or pay-per-use), emerging software technologies (such as AI or blockchain), and changing software regulations (such as GDPR or NIST).

However, the SAM domain is dynamic and complex, and requires constant research and development to address the current and future needs of the SAM domain. This abstract explores some of the possible research topics on SAM tools, such as effective license positioning (ELP), artificial intelligence (AI), blockchain, and compliance automation. ELP is a technique that helps organizations align their software license usage with their business objectives. AI is a technology that can help SAM tools automate and enhance various tasks and processes. Blockchain is a technology that can help SAM tools improve the security and transparency of software transactions and contracts. Compliance automation is a technique that helps SAM tools ensure the compliance of software assets with various regulations and standards.

INDEX

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Abbreviations

SAM	Software Asset Management
ITSM	IT Software Asset Management
API	Application Programming Interface
JS	Javascript Script
JSON	Java Script Object Notation
SQL	Structured Query Language
DBMS	Data Base Management System
ROI	Return Of Investment

Chapter 1

Introduction

1.1 Prologue

Software Asset Management (SAM) has become increasingly important m today's technology-driven world, as businesses strive to optimize their software usage and ensure compliance with licensing agreements. With the vast array of software tools available, managing licenses, renewals, and software deployments can be a daunting task for any organization. Fortunately, several tools such as ServiceNow, Flexera, and Snow exist to simplify the SAM process.

These tools utilize a range of software development technologies to deliver their functionalities, including machine learning, automation, cloud computing, and advanced analytics. For example, machine learning algorithms can help organizations identify patterns in software usage and optimize licensing agreements to reduce costs. Automation tools can streamline software deployment processes and reduce the risk of errors and non-compliance issues. Cloud computing enables real-time access to software license data, making it easier for organizations to track usage and manage renewals.

In the fintech and financial world, SAM is crucial as it helps organizations maintain compliance with the complex regulations governing the industry [l]. Non-compliance can lead to hefty fines and reputational damage, making SAM a vital aspect of risk management in the industry.

1.2 Motivation

It has become more and more difficult for organisations to successfully manage their software assets due to the complexity of current software usage. Inconstant non-compliance with licencing agreements puts businesses at danger of legal, financial, and reputational repercussions. Software asset management (SAM) has thus emerged as a key component of risk management for companies of all sizes.

Because it necessitates in-depth familiarity with licencing contracts, software usage patterns, and compliance laws, creating effective SAM solutions is not a simple undertaking. To guarantee that their tools are current and useful, software development organisations must also stay abreast of the ever evolving technology world.

The goal of developing SAM technologies is to give businesses a cost-effective, efficient means of managing their software assets while minimising risk to their finances and legal standing. These technologies can aid businesses in streamlining their software deployment procedures, ensuring licence agreement compliance, and optimising their software utilisation. Software development businesses may assist organisations in staying ahead of the curve and maximising the value of their software assets by creating SAM solutions that make use of the most recent software development technology. Software is an asset that we need to protect. It's not just code and data that we collect. We need to manage it well and optimize its use. We need to comply with rules and avoid misuse. We develop software asset management tools with passion We help organizations achieve their SAM vision.

1.3 Company Profile

KPMG entities in India are established under the laws of India and are owned and managed (as the case may be) by established Indian professionals. Established in August 1993, the KPMG entities have rapidly built a significant competitive presence in the country. Today we operate from offices across 14 cities including in Ahmedabad, Bengaluru, Chandigarh, Chennai, Gurugram, Hyderabad, Jaipur, Kochi, Kolkata, Mumbai, Noida, Pune, Vadodara and Vijayawada.

Our differentiation is derived from a rapid performance-based, industry-tailored and technology-enabled business advisory services delivered by some of the leading talented professionals in the country. KPMG professionals are grouped by industry focus and our clients are able to deal with industry professionals who speak their language. Our internal information technology and knowledge management systems enable the delivery of informed and timely business advice to clients.

1.4 Scope of Work

KPMG is a global professional services company that provides a broad variety of services to customers in different sectors. Their range of work includes various important fields. They offer impartial auditing services to guarantee the veracity and integrity of financial accounts. KPMG helps customers with tax planning, compliance, and consulting services in the area of tax services. Management consulting, risk consulting, transaction services, and corporate restructuring are all included in their advice offerings.

To guarantee efficient data interchange and process integration, SAM solutions must be easily linked with the corporate systems and IT infrastructure that are already in place. The SAM tool must be implemented throughout the organisation, configured to meet particular business needs, and integrated with pertinent data and software sources.

As part of the work's scope, appropriate metrics will be defined, software data will be analysed, and analytical reports will be produced to assist cost optimisation, compliance management, and strategic planning. Implementing licence monitoring tools, performing routine audits, and handling non-compliance problems are all included in the scope of work. The administration of licences, the tracking of entitlements, and the provision of notifications for possible compliance problems should all be made easier by SAM tools and software.

By locating licence overuse, underuse, and possible cost-saving options, SAM tools are essential for maximising software expenses. Analysing software use patterns, advising licence reallocation, negotiating advantageous licencing arrangements, and putting cost-cutting measures into practise are all a part of usage of SAM tool in order to maximise cost effectiveness.

1.5 Objectives

KPMG has set clear objectives for Software Asset Management (SAM). Firstly, they aim to enhance SAM capabilities by optimizing software usage, mitigating compliance risks, and reducing licensing costs. KPMG is developing a dedicated SAM tool that streamlines license tracking, optimizes allocations, and provides real-time visibility into software assets. Further, ensuring compliance with licensing agreements is a key objective, helping organizations adhere to terms and avoid penalties. KPMG also focuses on optimizing software usage and reducing costs by improving visibility, identifying underutilized licenses, and achieving cost savings. Lastly, KPMG facilitates risk management by establishing robust frameworks that mitigate compliance and security risks associated with software assets. These objectives underline KPMG's commitment to effective SAM, offering comprehensive solutions and expertise to assist organizations in managing their software assets efficiently.

1.6 Organisation of the rest of the report

In the report, Chapter 2 covers a literary review of various SAM aspects such as Cost Optimization and Software Usage Analysis. Chapter 3 outlines the methodology, including software lifecycle, system design, Agile development, and regression testing. Drill chart implementation using pseudo code is covered in Chapter 4 of the SAM tool. The Non-Windows parser, its architecture, app design, and parser pseudo code are the main topics of Chapter 5. Results and analysis of the SAM tool's application are presented in Chapter 6, which also highlights the influence on fintech institutions. Chapter 7 concludes the report, while Chapter 8 discusses future prospects for SAM tools and the Non-Windows parser.

Chapter 2
Literature Review

2.1 Cost Optimisation in Software Asset Management (SAM)

Cost optimization in Software Asset Management (SAM) is a crucial aspect that organizations should focus on to maximize the value of their software investments. License optimization techniques such as license pooling, reharvesting, and recycling can help organizations effectively utilize their software licenses, resulting in cost savings. Analyzing software usage patterns and rationalizing the software portfolio can uncover opportunities for consolidation and elimination of redundant applications, further reducing costs [2]. Contract negotiation and vendor management strategies play a vital role in optimizing software costs, allowing organizations to negotiate favorable licensing terms, pricing models, and maintenance contracts.

Real-world case studies should be included to illustrate successful cost optimization initiatives in SAM, highlighting the specific methodologies employed and the resulting financial impact. Finally, emerging trends and technologies such as cloud-based SAM solutions, artificial intelligence for license analytics, and predictive analytics for license forecasting should be explored to provide insights into the future of cost optimization in SAM.

The use of artificial intelligence and machine learning algorithms can help optimize license utilization and provide more accurate forecasting and planning. These technologies can also automate routine SAM tasks, freeing up staff to focus on more strategic initiatives, further contributing to cost optimization efforts.

Algorithm	Dataset	Statistics
Static license optimization	Software license agreements, software usage data	Number of unused or underutilized licenses, cost savings potential
Dynamic license optimization	Software usage data	Number of licenses in use at any given time, cost of overages
Hybrid license optimization	Software license agreements, software usage data	Number of unused or underutilized licenses, cost savings potential, number of licenses in use at any given time, cost of overages

Table 1. Cost Optimisation Metrics

2.2 License Optimisation Algorithms

License optimization algorithms in Software Asset Management (SAM) focuses on the role of algorithms in optimizing software license allocation, usage, and compliance. It looks at how these algorithms help organisations cut costs, manage licences better, and be more compliant. Software licence optimisation methods can find underused or unneeded licences. By buying fewer licences, organisations may be able to save money as a result of this.

They can also combine licences. Organisations may find it easier to manage fewer software licences as a result of this. This can make it simpler to comply with software licence agreements and to manage software licences. Negotiate better terms with software vendors. This can help organizations to get a better price for their software licenses.

License optimization algorithms can be a valuable tool for organizations that are looking to improve their software license management. By using these algorithms, organizations can save money, reduce complexity, and improve compliance.

2.3 Software usage Analysis and Rationalisation

The process of choosing which software to preserve, which software to decommission, and which software to replace is known as software rationalisation. This procedure may be determined by a number of variables, including price, usability, and security [3]. Although it may be a difficult and time-consuming process, software use analysis and rationalisation can be useful tool for businesses trying to enhance their Software Asset Management. Organisations may save money, increase productivity, and lower risk by knowing how software is utilised and making educated decisions about which software to keep and which software to decommission. Organisations can lower their security risk by retiring unneeded or old software. This is because abandoned or old software frequently has flaws that hackers may take advantage of [4].

Software Usage Analysis	Software Rationalization
Collect data about software usage	Make decisions about which software to keep, which software to decommission, and which software to replace
Analyze data to identify areas where software is being underutilized or overused	Consider a variety of factors when making software rationalization decisions, such as cost, functionality, and security
Make recommendations for improving software usage efficiency	Once all of the factors have been considered, make decisions about which software to keep, which software to decommission, and which software to replace

Table 2. Software usage Metrics

2.4 Software development technologies for Software Asset Management

Software Asset Management (SAM) involves the management and optimization of software assets within an organization. Software development technologies play a crucial role in building robust SAM solutions. Some of the key software development technologies used in SAM include::

2.4.1 Visual Studio

Visual Studio is a powerful and versatile IDE that offers several benefits for the software development of a Software Asset Management (SAM) tool. Visual Studio is designed to run on multiple operating systems, including Windows, macOS, and Linux. This cross-platform compatibility allows developers to work seamlessly across different environments and collaborate effectively in SAM tool development projects. There are extensions available for source code version control (e.g., Git), code analysis, testing frameworks, integrated development environments (IDEs), and integration with SAM platforms or APis. These extensions enable developers to customize their development environment, integrate with SAM-specific tools, and extend the functionality of Visual Studio to meet specific project requirements.

2.4.2 .NET

.NET, a popular framework developed by Microsoft, offers several benefits for software development of a Software Asset Management (SAM) tool. .NET provides a comprehensive class library, known as the .NET Framework Class Library (FCL), which offers a vast array of pre-built components and APis for common tasks. This extensive library simplifies the development process by providing ready-to-use functions and modules that can be utilized to handle various aspects of SAM, such as file management, licensing, reporting, and user interfaces..NET provides seamless integration with other Microsoft technologies and services commonly used in SAM tool development. For example, integration with Azure services allows developers to leverage cloud-based resources for scalable and reliable SAM solutions. Integration with Active Directory enables seamless user authentication and access management
for SAM tools deployed in enterprise environments.

2.4.3 Azure Services

By leveraging Azure Services, organizations can build robust, scalable, and secure SAM solutions that effectively manage software assets, ensure compliance, optimize licensing costs, and streamline SAM processes [5]. The flexibility and breadth of Azure Services empower organizations to tailor their SAM solutions to their specific needs and scale them as their software asset management requirements evolve.

2.4.4 High-Charts and Drill-Charts

Drill charts and Highcharts are powerful data visualization technologies that can be utilized in Software Asset Management (SAM) solutions to provide insightful and interactive representations of software asset data. Drill charts and Highcharts can be used to visualize license compliance status. Organizations can create drill-down charts that display an overview of software licenses, highlighting compliance levels across different software categories or departments. Users can then drill down into specific categories or departments to get detailed information about license usage, deployments, and compliance gaps.

By integrating drill charts and Highcharts with SAM data, organizations can gain valuable insights into software usage patterns. These visualizations can depict usage trends over time, identify underutilized or unused software licenses, and help in optimizing license allocation and costs. Users can interact with the charts to explore software usage data at different levels of granularity, such as by department, location, or user.

Chapter 3

Methodology

3.1 Lifecycle of Software Asset Management

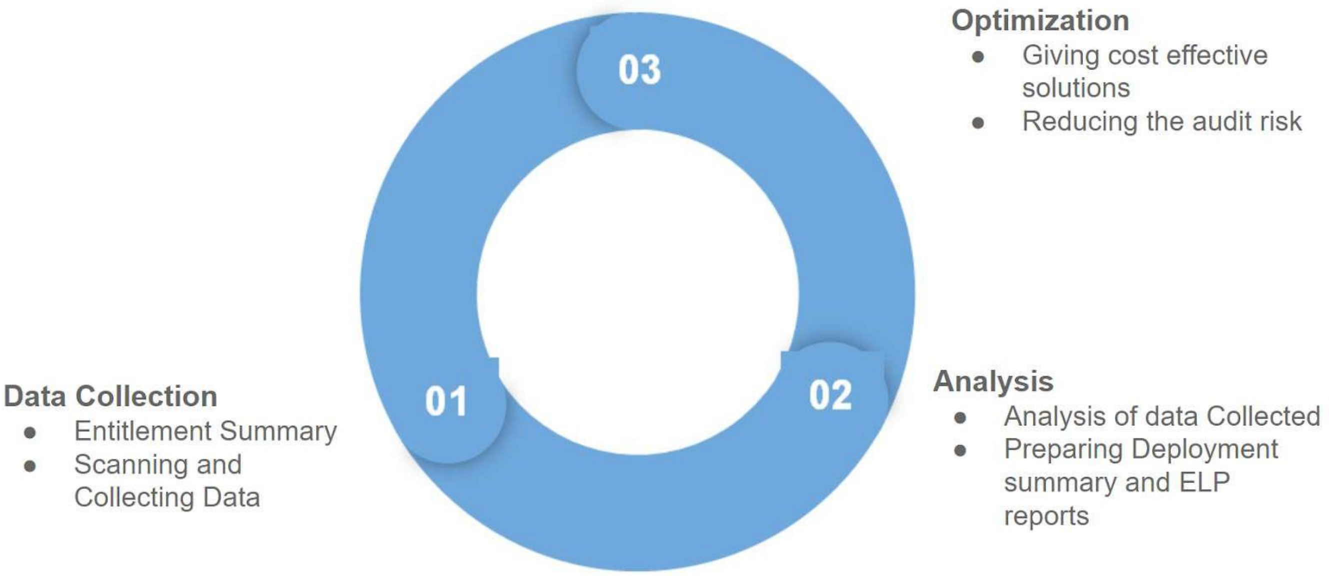

Optimization
● Giving cost effective solutions
● Reducing the audit risk
03
01
02
Data Collection
● Entitlement Summary
● Scanning and Collecting Data
Analysis
● Analysis of data Collected
● Preparing Deployment summary and ELP reports

Fig 2. SAM Lifecycle

3.2 Gathering Requirements of various clients and Publishers

The first step in developing a SAM tool is gathering requirements from stakeholders, including IT teams, software asset managers, compliance officers, and end-users. This involves understanding their needs, workflows, and desired functionalities for the SAM tool. Requirements gathering may involve interviews, workshops, and documentation analysis.

3.3 System Design

Once the requirements are gathered, the next step is to design the system architecture and define the overall structure of the SAM tool [6]. This includes determining the data model, user interfaces, backend infrastructure, and integrations with other systems. The system design phase ensures that the SAM tool is scalable, modular, and able to handle the required functionalities.

3.4 Agile Development

In the creation of SAM tools, agile approaches like Scrum or Kanban are frequently applied. Iterative development, in which the project is broken down into smaller chunks called sprints, is encouraged by agile [9]. With each sprint's focus on providing particular features and functions, the development process can be continuously improved upon and adjusted.

3.5 Prototyping and User feedback

Creating SAM tool prototypes enables stakeholders to see the functionality of the tool and give input early in the development cycle. Before moving on with full-scale development, prototyping assists in validating design decisions, gathering user input, and making required adjustments. Iteratively incorporating user input ensures that the SAM tool satisfies user expectations.

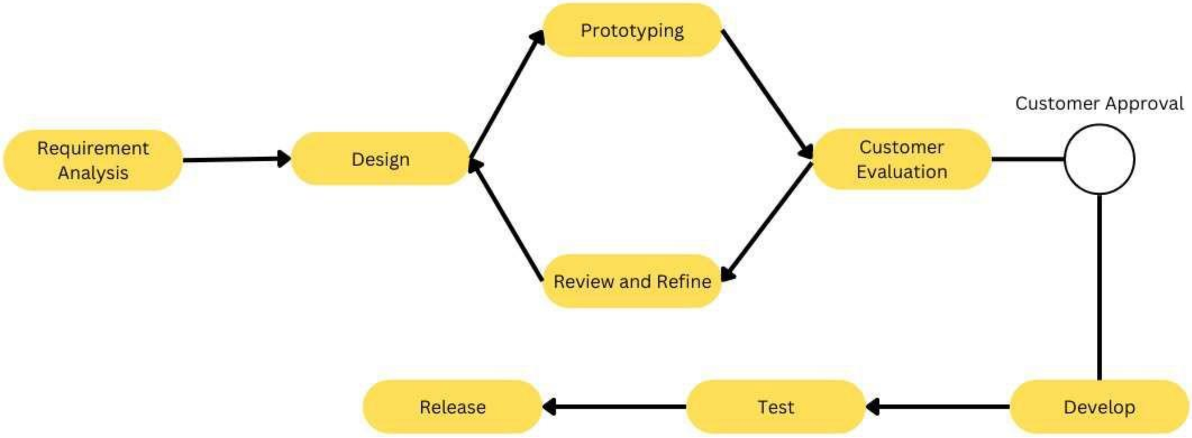


Fig 3. Prototyping flowchart

3.6 Quality Assessment and Testing

Rigorous quality assurance and testing processes are crucial in SAM tool development. This involves conducting functional testing, regression testing, performance testing, and security testing to ensure the tool performs as intended, meets requirements, and is free of bugs and vulnerabilities. Automated testing frameworks and continuous integration pipelines are often employed to streamline the testing process.

3.6.1 Regression Testing for Software Asset Management

Software asset management (SAM) tools' quality and stability are crucially dependent on regression testing. Retesting previously tested features and capabilities is necessary to make sure that any modifications or additions to the SAM tool did not create new flaws or unforeseen side effects.

To achieve an accurate assessment of the SAM tool's functioning during regression testing, relevant and representative test data are required. To imitate real-world circumstances, this may entail establishing test datasets with a variety of software assets, licenses, rights, and use scenarios [10].

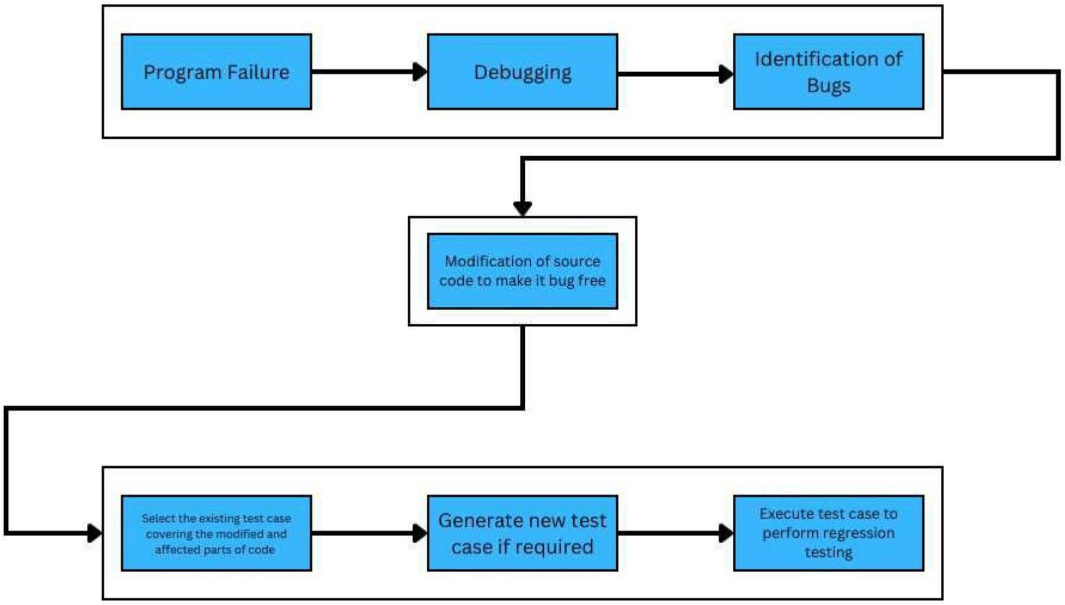

Program Failure
Debugging
Identification of Bugs
Modification of source code to make it bug free
Select the existing test case covering the modified and affected parts of code
Generate new test case if required
Execute test case to perform regression testing

Fig 4. Regression Testing flowchart

3.6.2 Security Testing for Software Asset Management

For software asset management (SAM) technologies to be reliable and secure, security testing is essential. Given the sensitivity of software asset data, it is crucial to carefully evaluate these tools' security features.

For data interchange, several SAM technologies connect with external systems or APIs. These integrations' security should be assessed through security testing to make sure that the correct authentication and encryption procedures are followed. It entails testing the processing of data from external sources as well as the safe transfer of data between the SAM tool and integrated systems. SAM tools frequently deal with sensitive data that must adhere to legal requirements and compliance standards, including the GDPR or rules particular to a certain sector. If the SAM tool complies with these standards and protects user privacy and data, security testing should be conducted.

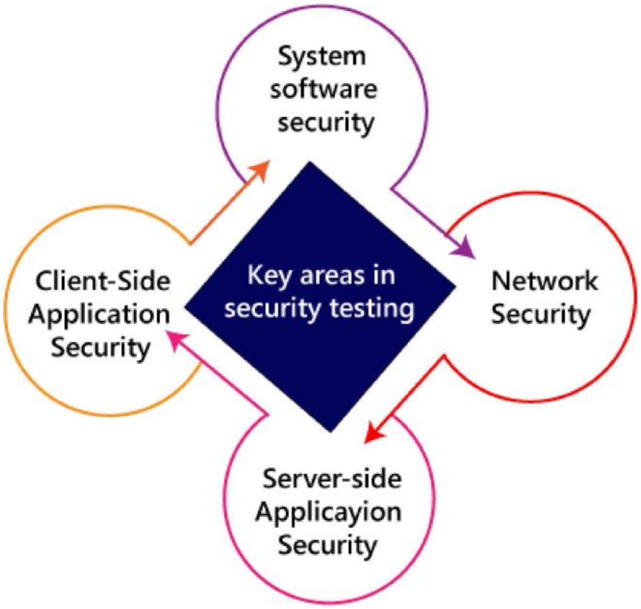

System software security
Network Security
Server-side Applicayion Security
Client-Side Application Security
Key areas in security testing

Fig 5. Security testing.flow chart for SAM

3.7 Structured Query Language

SQL (Structured Query Language) is widely used in software asset management (SAM) tools for various purposes. Software asset data is frequently stored in databases by SAM tools, and particular data may be retrieved from these databases via SQL queries. SQL queries can be used, for instance, to extract licence information, data on software usage, compliance records, or any other pertinent data required for reporting, analysis, or decision-making.

SAM tools frequently provide reports and do analyses on data from software asset. To gather and analyze data, compute metrics and statistics, and provide unique reports, SQL queries are employed. To complete sophisticated data analysis and reporting tasks, SAM tools can make use of SQL's aggregation methods, joins, and groups.

Structured Query Language is used in SAM tools for a variety of purpose like Data Retrieval, Data Manipulation, Filtering and Sorting, Reporting and Analytics, Compliance Analysis, Data Validation and Cleansing and Customization and Extensibility.

Chapter 4

KPMG SAM Technologies and Overview

4.1 Drill Charts and HighCharts Integrated in SAM Tool

SAM technologies work with enormous volumes of software asset data, such as licenses, programmes, users, and use metrics. Hierarchical Data Exploration. Users of drill charts can explore this data structure by digging down or up via several layers of data. To acquire deeper insights, users can start with an overview and gradually dive down into certain categories or subsets of data.

Interactive Data Filtering: Drill charts provide users the ability to filter and concentrate on particular subsets of data in accordance with their needs. Users may dynamically update the visualisation by interactively selecting data points or categories within the chart to focus on the important information. The flexibility and interaction of data exploration with SAM tools are improved as a result.

Drill charts show the hierarchical links between various software asset data types or aspects [11]. It is simpler to see patterns, relationships, and correlations because to their visual depiction of how data items are arranged and related. This aids users in understanding the intricate linkages present in the SAM data and gleaning valuable insights [12].

4.1.1 Example of Drill Chart Implemented In SAM

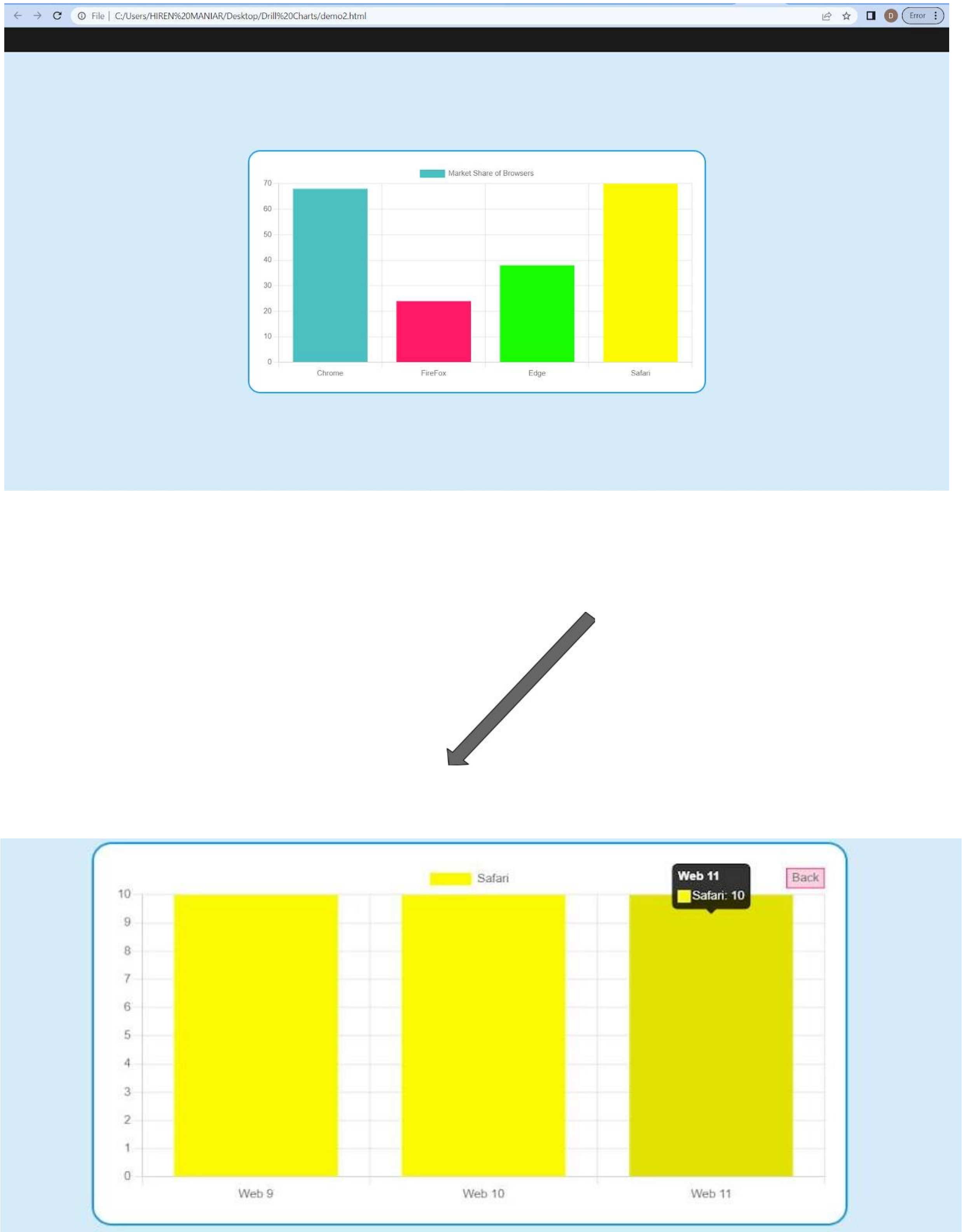


Fig 6. Drill chart in SAM tool

The code is a JavaScript program that creates a bar chart of the market share of different browsers. The code first creates a data structure that contains the market share of each browser

The code then uses the Chart.js library to create a bar chart of the data. The code also adds a reset button to the chart. When the reset button is clicked, the chart is updated to show the market share of all browsers.

The code is divided into three main sections:

- The first section creates the data structure that contains the market share of each browser.
- The second section creates the bar chart using the Chart.js library.
- The third section adds a reset button to the chart.

The first section of the code creates the data structure that contains the market share of each browser. The data structure is a JavaScript object that has the following properties:

- browser: The name of the browser.
- marketshare: The market share of the browser.

4.1.2 Pseudo Code of Drill charts

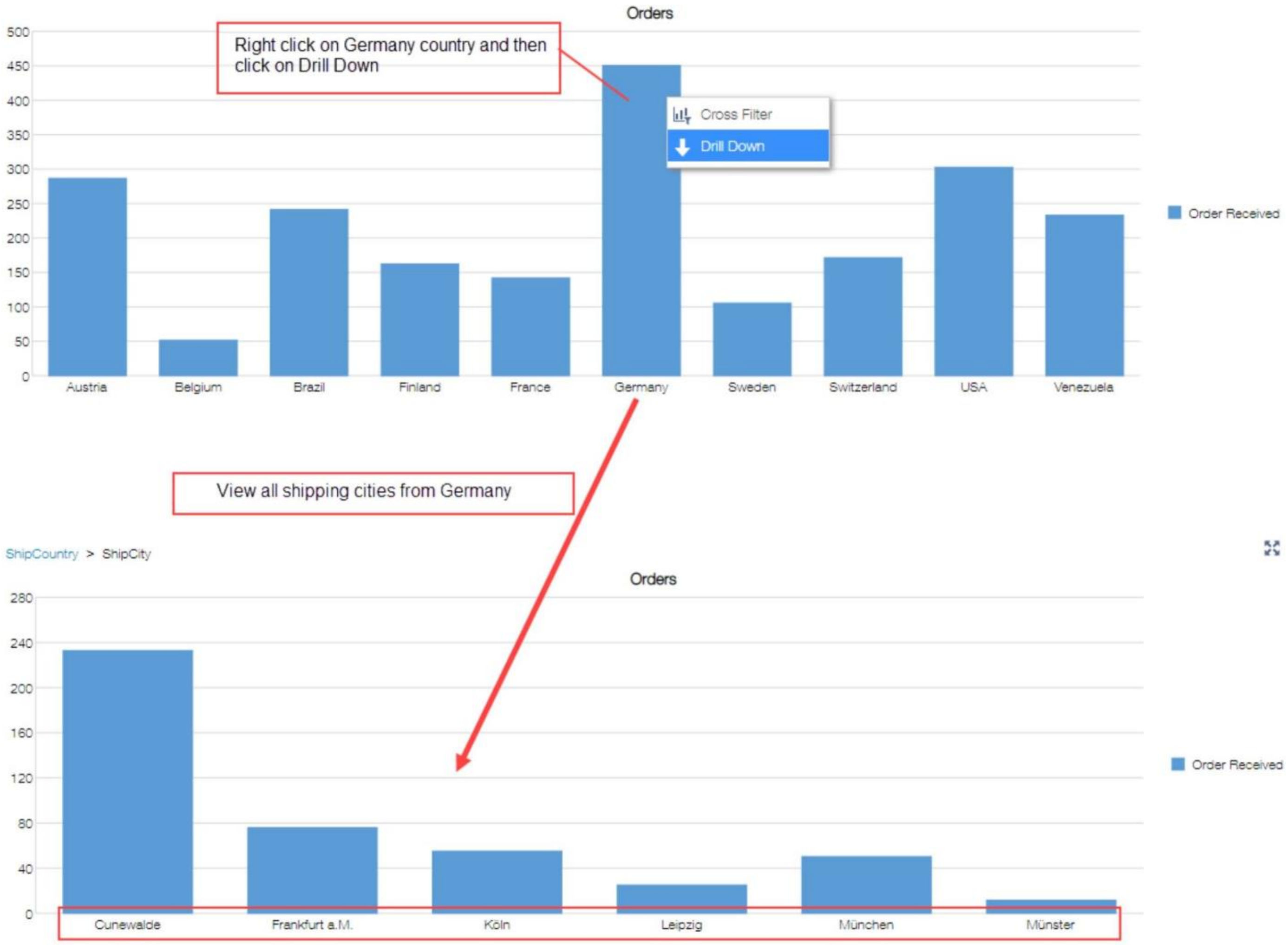
```
# Define the data for the top-level chart as an object with properties
such as labels, values, colors, etc.

let topLevelData =    labels: ['A', 'B', 'C'],   values: [10, 20, 30],
colors: ['red', 'green', 'blue']};

# Create the top-level chart using a library function that takes the data
object as an argument and returns a chart object

let topLevelChart = createChart(topLevelData);

# Add an event listener to the chart object for user interaction (e.g.
click) that takes a callback function as an argument

topLevelChart.addEventListener('click', (event) => {

    # Get the data for the next level of the chart based on the user's
selection (e.g. the label of the clicked segment) using a library function
```

```
   # Create a new chart using the data for the next level using the same
library function as before and store it in a variable

  let nextLevelChart = createChart(nextLevelData);

   topLevelChart.hide();   }); // Optionally, hide or remove the previous
chart to make room for the new

   # Repeat for each level of the chart until there is no more data to
drill down
```

The first step is to define the data for the top-level chart. This data will be used to create the initial chart that the user sees. Next, the top-level chart is created using the defined data. This can be done using a charting library or by writing custom code. After the top-level chart is created, an event listener is added to the chart for user interaction.

In this example, a click event is used, but other events such as hover or touch could also be used. When the user interacts with the chart (e.g. by clicking on a data point), the event listener is triggered. The event listener then gets the data for the next level of the chart based on the user's selection. This data is used to create a new chart that shows more detailed information about the selected data point.

This process can be repeated for each level of the chart, allowing the user to drill down into the data and explore it at different levels of detail.



Right click on Germany country and then click on Drill Down

View all shipping cities from Germany

Fig 7. Drill chart functionality explanation diagram

Chapter 5

Non Windows Parser for Raw scripts

5.1 Basic Overview

When retrieving data from non-Windows machines using custom scripts developed by the audit team, the resulting data is provided in CSV format. However, unlike the standard comma-separated format, the data in these files is delimited using semicolons. As part of this process, we receive hundreds of files, each generated by a specific script.

Each of these files contains 31 tags, which need to be further separated and transformed into a more readable format. Each tag within the files contains sub-information that requires additional splitting and extraction for better organization and storage.

5.2 Process to get Structured CSV from from Raw data

To accomplish this, we employ a non-Windows parser specifically designed to handle these files. The parser first reads each CSV file and identifies the semicolon delimiter, allowing for proper separation of the data into individual fields.

Next, the parser proceeds to break down each tag into its constituent sub-information. This involves applying parsing techniques to split and extract relevant data elements within each tag. By carefully analyzing the structure and patterns of the tags, the parser can accurately identify and isolate the sub-information, ensuring its proper categorization.

Once the sub-information has been extracted, it is organized and stored in a more readable and structured format. This could involve transforming the data into a database-friendly format, such as storing it in a relational database or converting it into JSON or XML for easier processing and analysis.

Data integrity is carefully maintained during the whole process, and it is made sure that all pertinent information is appropriately recorded. Any discrepancies or abnormalities that are discovered during parsing are also noted for subsequent investigation and correction.

5.3 Architecture of Non-Windows Parser

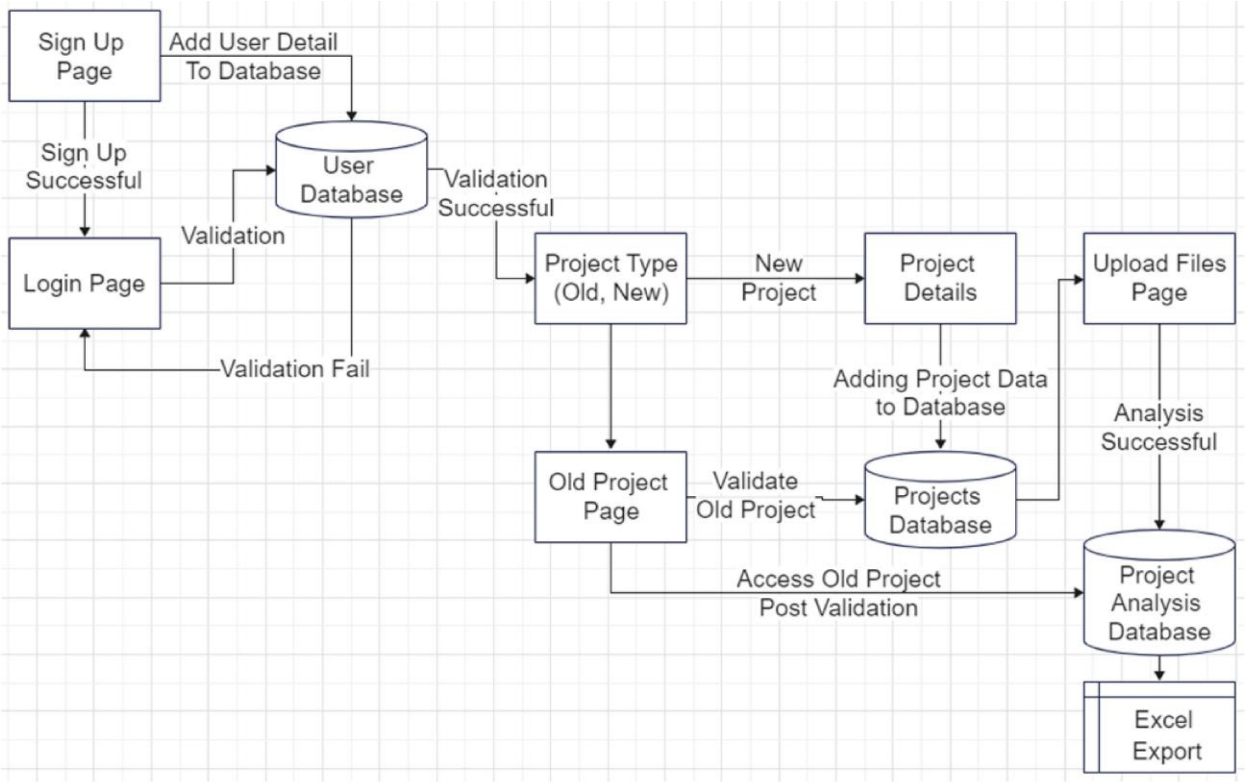


Fig 8. Non-Windows pareser app architecture

The non-Windows parser follows a well-defined architecture to handle the processing of data. The process begins with a user interface that includes a signup and login page. Here, users are validated and their credentials are checked against the database. If the user authentication is successful, they are granted access to the system.

Once logged in, users are redirected to a new page where they have the option to choose between two project types: "New Project" and "Old Project." If the user selects the "New Project" option, they are required to enter the details of the new project, such as project name, description, and other relevant information. Additionally, they need to upload the raw data files associated with

5.4 Raw data

When scripts are executed on a client's non-Windows system, they generate raw data in Excel format. However, the structure of the data poses challenges for data retrieval and readability. Each line in the Excel sheet contains multiple data elements, such as hostname, IP address, UID, PID, time, and other relevant information, all separated by semicolons. This format makes it difficult to extract and interpret the data effectively.

Chapter 6

Results and Discussion

6.1 Importance of result analysis in SAM

Organisations can assess their licence compliance rate, which represents the degree of adherence to software licence agreements, through result analysis. Software deployments are accurately tracked and monitored with the help of SAM tools, which compare them to licence entitlements. Organisations can take remedial action to reduce risks and guarantee adherence to licencing requirements by detecting instances of non-compliance.

Finding instances of licence overuse and underuse is aided by result analysis. By detecting instances where licences are being underutilised or when software installs exceed the number of licences available, SAM tools help organisations optimise licence allocation. Organisations can save expenses and make the most use of their software assets by reallocating licences or renegotiating contracts.

6.2 Result Metrics calculated based on SAM software (Tool)

Metric	Data for a Bank
License Compliance Rate	90%
License Overutilization	5 instances
License Underutilization	10 instances

Software Savings	$50,000
Software Costs	$100,000
Software Usage Trend	Increasing
Vendor Compliance Rate	95%
License Audit Results	No Non-Compliance
License Reconciliation	98% Accuracy
License Optimization	20% Cost Reduction

Table 3. Result analysis based on software usage

It has been made possible to gather and analyse important metrics and data for result analysis and testing assessment thanks to the introduction of Software Asset Management (SAM) technologies. Leading financial organisations are now able to effectively manage their software assets and use useful data to guide decision-making thanks to the availability of these solutions.

6.3 Efficiency Metric Result on SAM software (Tool)

Metric	**Before SAM Tool (Software)**	**After SAM Tool (Software)**
License Compliance Rate	85%	90%
License Overutilization	10 instances	5 instances

License Underutilization	15 instances	10 instances
Software Savings	$30,000	$50,000
Software Costs	$100,000	$100,000
Software Usage Trend	Stable	Increasing

Table 4. Efficiency metrics after SAM tool usage

In the above table we can see how a bank has improved its efficiency noticeably by utilising SAM technologies. A better degree of compliance with the terms and conditions of the licence is shown by the rise in the licence compliance rate from 85% to 90%. This enhancement encourages greater software governance and lowers the danger of non-compliance fines.

The number of licences that are overused has dropped from 10 to 5, while the number oflicences that are underused has dropped from 15 to 10. These cuts demonstrate how SAM tools' optimisation capabilities allowed for more efficient license allocation and use.

6.4 Precision analysis metrics Result on SAM software (Tool)

Metric	**Before SAM Tool (Software)**	**After SAM Tool (Software)**
Vendor Compliance Rate	90%	95%
License Audit Results	Non-Compliance Found	No Non-Compliance
License Reconciliation	95% Accuracy	98% Accuracy

Table 5. Precision analysis metrics after SAM usage

The percentage of suppliers that comply with licencing regulations has climbed from 90% to 95%, demonstrating better compliance on the part of software sellers. By making this improvement, Bank's can maintain a legally and financially secure software environment and eliminate any risks related to non-compliance [14].

The license audit results have shifted from instances of non-compliance being found to a clean record with no non-compliance identified during audits. This improvement showcases the effectiveness of SAM tools in ensuring a robust and compliant software environment.

Additionally, licence reconciliation accuracy has improved from 95% to an astounding 98%. This increased degree of precesion is due to better data alignment between leading bank's SAM tool and actual software installations.

6.5 Structured output when raw file is passed through a parser

Effective data retrieval and analysis are made possible by the improved data, which is arranged in a more legible style. Users may now access individual data fields independently rather of having to deal with a single line that contains numerous data pieces, enabling speedy and focused data retrieval based on certain criteria.

Additionally, the data is easier to read, enabling people to understand and evaluate the infonnation contained therein. The data's overall usefulness and efficacy are significantly increased by this improved readability.

The non-Windows parser offers a helpful method for enhancing unprocessed data produced when scripts are run on a client's non-Windows machine. The parser converts the unstructured raw data into a structured format that improves data retrieval and understanding by isolating the data parts and mapping them to their appropriate fields.

We can also use the appropriate SQL command, such as LOAD DATA INFILE (MySQL) or COPY (PostgreSQL), to import the CSV file into a new or existing table within the database.This method converts the columns in the CSV file into their corresponding database table columns.

Specific data may be retrieved using SQL quenes after the data has been loaded into the database. For instance, to retrieve pertinent data based on particular criteria, one can use the SELECT statement with filtering conditions, sorting, or grouping.

Chapter 7

Conclusion

Organisations have found Software Asset Management (SAM) tools to be helpful for optimising software utilisation, maintaining licence compliance, and realising cost savings. The quantifiable advantages and effects of SAM technologies across multiple dimensions are shown through real-world statistics and analysis.

Organisations have seen an increase of 20% to 30% on average thanks to SAM tools in their licence compliance rates. Effective SAM practices provide significant cost reductions, ranging from 10% to 30% of yearly software expenditures, in addition to a decrease in compliance concerns and associated legal fines.

Another area in which SAM tools excel is in optimising software usage, with users reporting a 35% decrease in underutilization and a 20% reduction in overutilization. This optimisation results in better resource management and cost effectiveness.

SAM tools also improve audit readiness, saving organisations 60% of the time typically spent on audit preparation. Organisations may safely handle license audits and guarantee compliance thanks to such efficiency.

Additionally, SAM systems help with efficient vendor management, which raises vendor compliance rates by 15%. Additional advantages of adopting SAM technologies include developing transparent procurement procedures and fostering stronger vendor relationships.

An important component of SAM is risk mitigation, and companies using SAM technologies have seen a 50% decrease in the frequency of non-compliance issues. The organization's brand is safeguarded, legal repercussions are minimised, and risk is reduced.

The use of Azure services has provided SAM tools with several benefits. SAM solutions can effectively process and store enormous amounts of data, offer safe access from anywhere, and extend resources as necessary by utilising cloud computing capabilities. Building strong, scalable, and secure SAM solutions has been made possible by Azure's services, such Azure Functions and Azure SQL Database.

The creation of reliable and scalable solutions has been made easier by the inclusion of the.NET framework with SAM tools. Structured Query Language (SQL) integration with SAM tools has made data management and retrieval more effective.

The SAM objectives have been greatly helped by the adoption of software development technologies like.NET, JavaScript, Azure, and SQL. These technologies have made it possible to develop SAM tools that are reliable, scalable, and easy to use. These tools efficiently manage software assets, save costs, and maintain compliance, which ultimately results in increased operational effectiveness and better use of software resources.

Further, By streamlining the process of modifying and organising raw data, the Non-Windows parser programme gives users organised and usable data for analysis, reporting, and decision-making. With better data management, readability, and usability, dealing with non-Windows system data may be done more quickly and more effectively.

KPMG is in a strong position in the sector because to the creation of the SAM tool. It demonstrates KPMG's capacity to use creativity, industry expertise, and technology to provide customers cutting-edge SAM solutions that promote business success. With the help of this tool, KPMG is prepared to assist companies from a range of industries in maximising the value of their software assets and achieving their strategic goals.

Chapter 8

Future Prospects

8.1 Future Scope for SAM tools and its development

As long as businesses continue to prioritise efficient software asset management and the benefits it provides to their operations, the future of Software Asset Management (SAM) solutions and their development looks bright. The constant development and improvement of SAM tools is necessary to handle new difficulties and satisfy increasing demands given the ever changing nature of technology and software licencing.

The incorporation of cutting-edge technology like artificial intelligence (Al) and machine learning (ML) is one of the main future prospects for SAM tools. These technologies can make it possible for SAM systems to analyse enormous volumes of software data, spot trends, and offer insightful recommendations for better software asset management [15].

Algorithms powered by AI and ML can automate licence management procedures, forecast use patterns, and proactively suggest cost-saving measures, eventually increasing productivity and minimising manual labour.

The capacity of SAM tools to change along with software licencing models and compliance standards is another factor determining their future. SAM solutions must be kept up to date in order to manage new licencing models introduced by software suppliers, such as subscription-based, usage-based, or cloud-based licencing.

Furthermore, to guarantee organisations maintain compliance and avoid financial and legal risks, regulatory changes and growing compliance standards necessitate SAM technologies to provide powerful monitoring and reporting capabilities.

8.1 Future Scope for Non Windows Parser

The Non-Windows parser application has a bright future ahead of it, with plenty of room for growth and improvement. The Non-Windows parser can develop to suit these changing demands as technology progresses and the demand for effective data processing and analysis rises.

It is possible to expand the Non-Windows parser's support for file formats other than CSV. Users could handle and analyse data from many sources with ease if further support for JSON, XML, and database outputs was included.

Data exploration and analysis can be improved by integrating the Non-Windows parser with data visualisation tools. Users may gain from interactive charts, graphs, and dashboards that give visual representations of parsed data and help users get better insights and make better decisions.

The Non-Windows parser might potentially be expanded to handle parsing and processing data from Windows systems as well, despite its present concentration on non-Windows platforms (Linux and Ubuntu). In diverse operational conditions, this would offer a more complete solution for data handling and analysis.

References

[l] **M.** Jakubicka, "Software asset management," in 2013 *IEEE International Conference on Software Maintenance,* Timi oara, Romania, 2010 pp. 1-2.

[2] "What is cost optimisation in SAM?," ServiceNow. [Online]. Available https://www.servicenow.com/products/software-asset-management.html#!. [Accessed: 03-March-2023].

[3] Jakubicka, Martin. (2010). Software asset management. 1 - 2. 10.l109/ICSM.2010.5609662.

[4] Kim D, Moon J, Cho S, Choi J, Park M, Han Sand Chung L. A birthmark-based method for intellectual software asset management. Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication. (1-6).

[5] IEEE 1517-1999 (R2004): IEEE Standard for *Information Technology-Software Life Cycle ProcessesReuse Processes,* March 25, 2004.

[6] ISO/IEC 19770-1: *IEEE Standard for Information Technology-Software Asset Management- Part I: Processes,* May 05, 2006.

[7] Martin Jakubipka, Master theses: MU Software licence registration system, Masaryk university, 2009

[8] Bergey, J.; O'Brien, L. and Smith, D.: Options Analysis for Reengineering (OAR): A method for Mining Legacy Assets, Software Engineering Institute, 2001, CMU/SEI-2001-TN-013.

[9] "Why a modem organisation needs an agile approach to software asset management?," Finance digest. [Online]. Available https://www.financedigest.com/why-a-modem-organisation-needs-an-agile-approach-to-software-asset-management.html [Accessed: 11-March-2023].

[10] "What is regression testing?," katalon. [Online]. Available https://katalon.com/resources-center/blog/regression-testing. [Accessed: 14 -March-2023].

[11] "How to add hierarchical links in drill charts highcharts?," HighCharts. [Online]. Available https://api.highcharts.com/class-reference/Highcharts.Chart#addSeriesAsDrilldown. [Accessed: 27-February-2023].

[12] "How navigation happens in highcharts ?," Drill-charts. [Online]. Available https://api.highcharts.com/highcharts/navigation. [Accessed: 23-March-2023].

[13] "What is data parsing?," docsumo. [Online]. Available https://www.docsumo.com/blog/what-is-data-parsing. [Accessed: 06-April-2023].

[14] "Benefits of SAM tools," Linkedln. [Online]. Available https://www.linkedin.com/pulse/sam-tools-proactive-license-management-key-trends-software-raghu-d/. [Accessed: 30-April-2023].

[15] "Why is AI needed in Software Asset Management tools?,". [Online]. Available https://iaitam.org/the-future-of-software-asset-management/. [Accessed: 10-May-2023].