K.L.E.P.T.O.S
SOFTWARE PACKAGE AND LICENCE DISCOVERY TOOL

DAP605 – SYSTEM DEVELOPMENT PROJECT

STUDENT NUMBER: 1906235

Word Count: 3,775

Document Summary

Project: **K.L.E.P.T.O.S**

Title: **SOFTWARE PACKAGE AND LICENCE DISCOVERY TOOL**

Subject: **System Report**

Abstract

Modern software development depends heavily upon integrating packages of code that are transferable between projects and provide instant access to a library of functionality. Typically, free to use under open-source licence agreements, these packages are developed by individuals or companies, then published to Microsoft's public nuget.org repository. Incorporating a third-party package into a software project can bring a number of obstacles such as knowing what licence the package is shared under, what the terms of that licence are, and how the packages author should be credited in your project. The Keeper of Licences for Each Product's Third-party Open-source Software (K.L.E.P.T.O.S or KLEPTOS) is a project commissioned by Edwards Vacuum Ltd. as a feasibility study into the degree by which the process of complying with these licence terms can be automated by an in-house software solution.

Acknowledgements

I would like to thank Brett Lawrence, an excellent line manager and supportive colleague, both inside and outside of the workplace. Furthermore, I would like to extend my gratitude to Welbert Gomides, and James Hare, for their ongoing support and training in my journey to becoming a qualified and capable developer. Additional thanks are given to the members of legal counsel that took time out of their obligations and commitments to contribute to my requirements gathering efforts. Finally, to my late friend, David Silverson, my inspiration, and mentor in all my academic pursuits since childhood, may this work be dedicated to him.

"Luck is the idol of the idle." - David Silverson

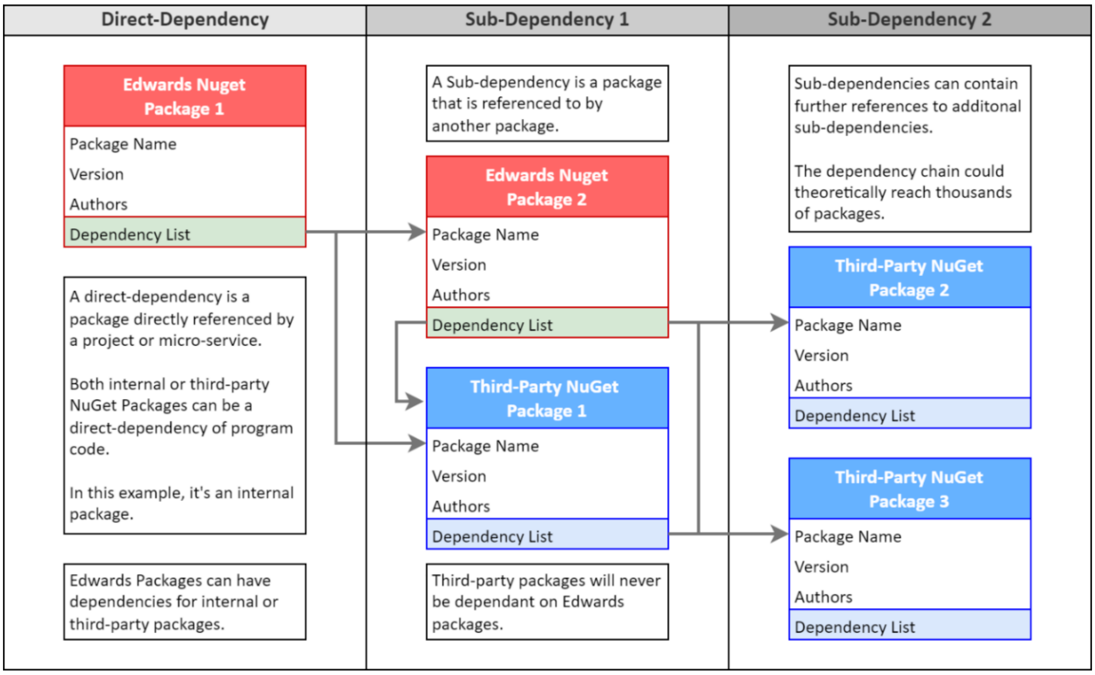
Contents

1 Introduction and Review

1.1 Organisational Context

Edwards Vacuum Ltd, a manufacturer of industrial and scientific vacuum pumps, is a subsidiary of the international Atlas Copco Group with a workforce of approximately 45,000 employees (Atlas Copco, 2023). The company's primary clientele comprises semiconductor manufacturers such as Intel, Samsung, and LG. These customers procure vacuum pump systems in large quantities from Edwards, to be utilised in specialised manufacturing facilities (predominantly making computer processors). Each Edwards pump is operated and monitored independently via a dedicated digital controller, which is interconnected to others through networked computers running Edwards' proprietary software. To support their operations, Edwards Vacuum is involved in the development and maintenance of numerous computer programs related to pump functionality. (Edwards Vacuum Ltd, 2023)

Edwards’ software development process makes use of third-party and internal packages to distribute common functionality. When a software making use of third-party packages is distributed, it’s implementation must adhere to a number of conditions, primarily, crediting the author and including the original licence texts. Furthermore, one package (third-party or internal) can depend upon multiple other packages. As a result, an Edwards software product can include upwards of one-thousand packages that must be recorded and credited. Although Edwards packages can be dependent on third-party, or internal packages, third-party packages cannot be dependent on internal packages, as they do not exist outside of the company.



A direct-dependency is a package directly referenced by a project or micro-service.

Both internal or third-party NuGet Packages can be a direct-dependency of program code.

In this example, it's an internal package.

Edwards Packages can have dependencies for internal or third-party packages.

A Sub-dependency is a package that is referenced to by another package.

Third-party packages will never be dependant on Edwards packages.

Sub-dependencies can contain further references to additonal sub-dependencies.

The dependency chain could theoretically reach thousands of packages.

Figure 1 Package Dependency Diagram

1.2 Project Aim

The project aim was to achieve a proof-of-concept program to replace the manual process of third-party package licence discovery and crediting of third-party package authors in compliance with each licence. The solution needed be able to receive the directory of C# source code that makes up a software product and extract their direct-dependencies and the direct-dependency's sub-dependencies. Then, be able to query each discovered package dependency to either factually obtain the licence information or attempt to infer a package's licence from the package's webpage. Package and licence information should be persisted for review by the legal team to indicate whether the licence information needs manual intervention. Where a package's licence information doesn't match to an example licence from the SPDX example, it should be persisted to a file that can be manually reviewed and edited.

Finally, the persisted package data must be able to be read back into the program and processed into a pdf document that credits package authors and licences.

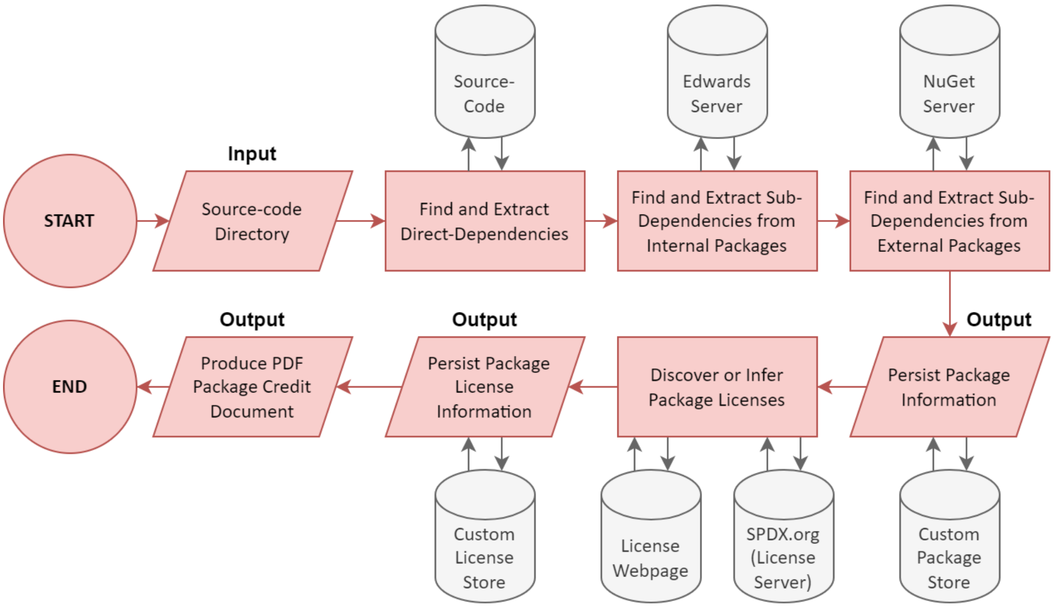

Source-Code
Edwards Server
NuGet Server
Input
START
Source-code Directory
Find and Extract Direct-Dependencies
Find and Extract Sub-Dependencies from Internal Packages
Find and Extract Sub-Dependencies from External Packages
Output
Output
Output
END
Produce PDF Package Credit Document
Persist Package License Information
Discover or Infer Package Licenses
Persist Package Information
Custom License Store
License Webpage
SPDX.org (License Server)
Custom Package Store

Figure 4 Project Solution Outcome

Figure 5 Wysocki Lifecycle Matrix (Robert Wysocki, 2012)Figure 6 Project Solution Outcome

1.3 Project Scope

As agreed with the project sponsor through a series of interviews (Appendix 3. Project Sponsor Consent Form) in the Appendix 2. Project Initiation Document (pp. 7), the solution is a graphical program that scans a C# project directory for third-party packages, extracting their licence types and texts, then compiling a document to credit package authors.

1.4 Development Lifecycle

To manage the design, development, and deployment of a software project, a Software Development Lifecycle (SDLC) was required. There are a number of varieties of SDLC, and deciding upon one for a project is entirely subjective, although can be aided by models Appendix 2. Project Initiation Document (pp. 17). To decide upon a development approach to the project, Wysocki provides a model, with axes that measure the perceived clarity of requirements & solution, and goals. This example shows the K.L.E.P.T.O.S project as being relatively well-understood in both axes. (Robert Wysocki, 2012)

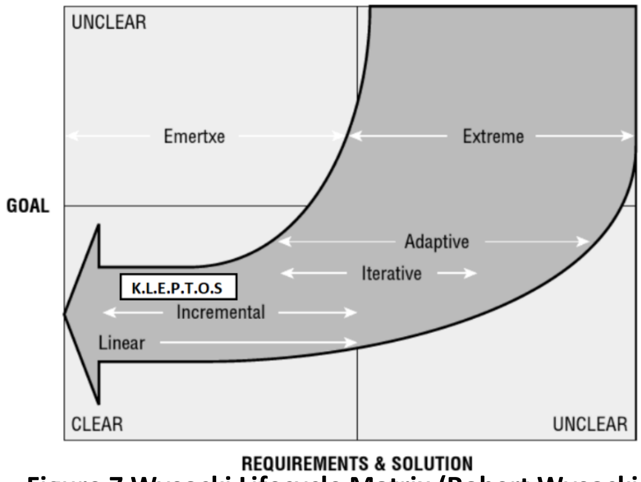


Figure 7 Wysocki Lifecycle Matrix (Robert Wysocki, 2012)

A linear (waterfall) approach is dependable for a project and helps constrain work to the pre-defined requirements, although it limits variability and must be restarted if the practicality of a solution does not meet the expected feasibility of a solution. This isn't suitable for the KLEPTOS project, as the unknown variables would result in restarting from the planning stage several times. (Adobe, 2022)

An iterative approach is more flexible, its premise is first producing a minimum-viable-product, then gradually improving it to meet additional functionality and quality standards outlined in a brief. This isn't appropriate for the KLEPTOS project, as the project is a feasibility study and isn't focused on producing a refined product for end-user consumption. (Association for Qualitative Research, 2023)

The incremental methodology was adopted for the KLEPTOS project since the high variability of functionality and the natural progression of data through the program lends itself to being developed in functional increments as visualised below (Figure 4). (Paul Ganney, 2020)

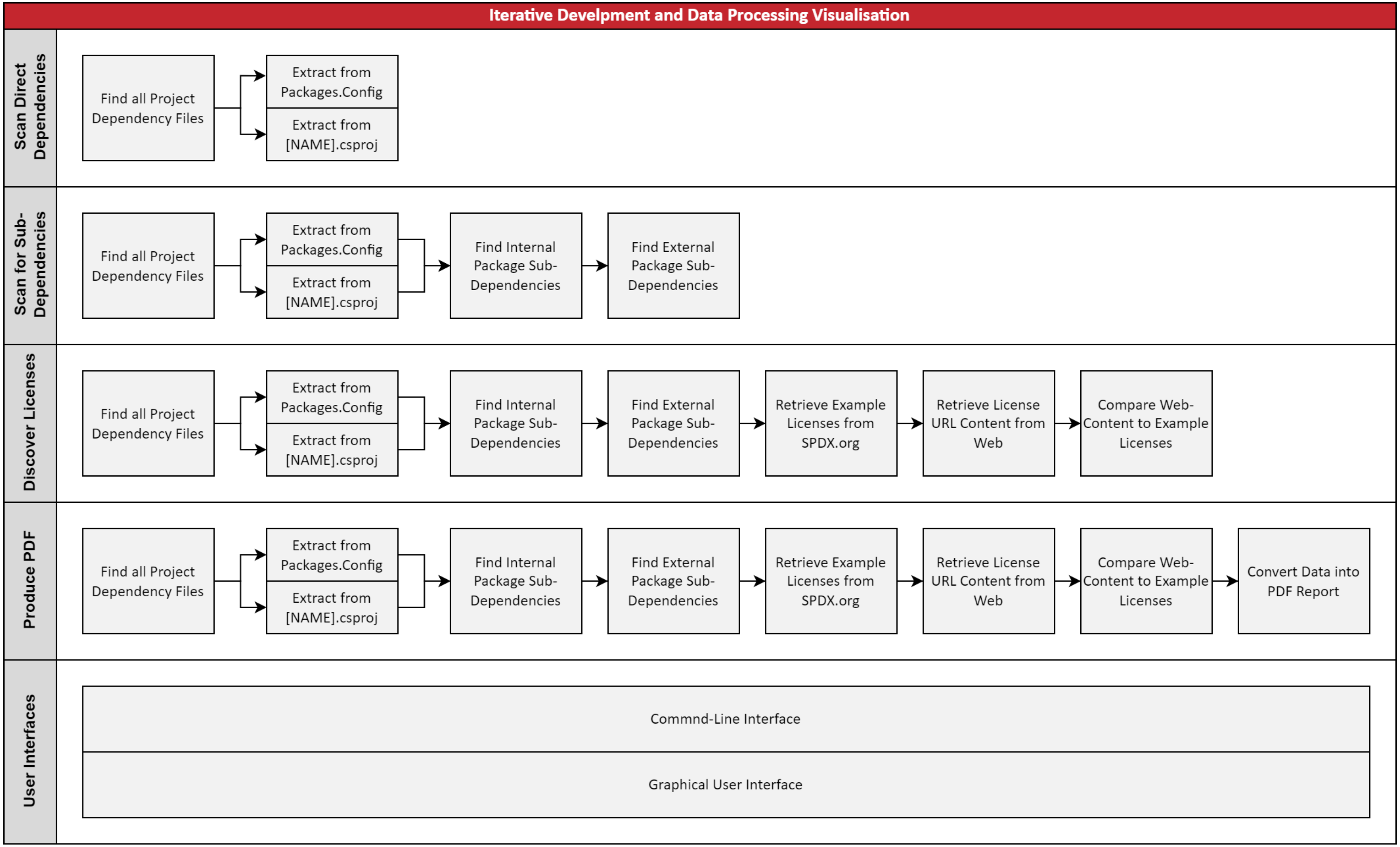

Iterative Develpment and Data Processing Visualisation
Scan Direct Dependencies
Scan for Sub-Dependencies
Discover Licenses
Produce PDF
User Interfaces
Find all Project Dependency Files
Extract from Packages.Config
Extract from [NAME].csproj
Find Internal Package Sub-Dependencies
Find External Package Sub-Dependencies
Retrieve Example Licenses from SPDX.org
Retrieve License URL Content from Web
Compare Web-Content to Example Licenses
Convert Data into PDF Report
Commnd-Line Interface
Graphical User Interface

Figure 10 Incremental Design Diagram

Figure 11 Project Deliverable VisualisationFigure 12 Incremental Design Diagram

1.5 Project Controls

The following project controls were outlined in Appendix 2. Project Controls (pp. 12) although adjustments have been made since version 1:

Project Controls	
PRINCE2® https://prince2.wiki/	Project management is controlled by the PRINCE2® methodology. PRINCE2® structures planning, execution, and project delivery. (AXELOS Ltd, 2023)
S.C.R.U.M www.scrum.org	To manage each increment, SCRUM is used to deconstruct deliverables into tasks that must meet acceptance criteria, with task points, where 8 points is ~1 weeks work. (Larman, 2004) Where each sprint is 3 weeks – start 3rd April 23.
Office 365 www.office.com	Office 365 (Word, PowerPoint, Excel, Outlook, teams, and OneDrive) is the de facto environment for Edwards project materials. It offers storage redundancy and remote access to project materials and team collaboration.
TortoiseSVN tortoisesvn.net	TortoiseSVN version control is an already available tool to this project. It enables branching, merging, and reverting changes to safely manage project development.

Table 1 Project Controls

Work was divided into "stories" meeting acceptance criteria and estimated in points aligning with SCRUM Alliances definitions (scrum.org, 2023). Changes from the initial Project Initiation Document were made after discussions with the project sponsor to improve the resources available for value-adding tasks while remaining within the constraints. The project was decided to: be managed using an Excel document instead of Axosoft, focus on incremental development and unit testing API interactions, and deliver the project as a library for interface flexibility rather than a NuGet package.

1.6 Options Analysis

To ensure that the best solution to the business was delivered, an options analysis and business case was conducted before project initiation Appendix 2. Options analysis (pp. 4). The analysis considered alternatives, such as outsourcing to a third-party service, an off-the shelf solution, simply maintaining a database of licences, and the consequences of making no adjustments. The development of an internal tool was decided by management due to zero cost, and customisability.

1.7 Deliverables

Excerpt of MoSCoW deliverables, reformatted from Appendix 2. Project Initiation Document (pp. 9)

Must Have	Functionality to scan for and build C# solutions
	Functionality to extract third-party packages and identify licences
	Functionality to export collected licence data as JSON and PDF
	A command line interface to run said library
	Documentation, appropriate commenting, and 85% +- 10%-unit test coverage
Should Have	Capability for team city integration
	Graphical user interface to operate library functionality
	Manual accommodations for unfound licences
Could Have	Flagging use of copy-left (or forbidden) packages
	Report on licence discovery success
Won't Have	Team city integration
	Analysis of Licence compliance
	Database of historically permitted licences
	Guarantee for licence compliance without oversight from intellectual property counsel

Table 2 Project Deliverables

Summarising, the KLEPTOS solution should comprise 3 significant features:

Code Library	Command Line Interface	Graphical Interface
Scanning for packages	Implements Code Library	Implements Code Library
Scan Licences from Packages	Runs without user input	Minimal Interface
Produce Credit Document	Takes executable arguments	Graphical Progress Indication

Table 3 Project Deliverable Summary

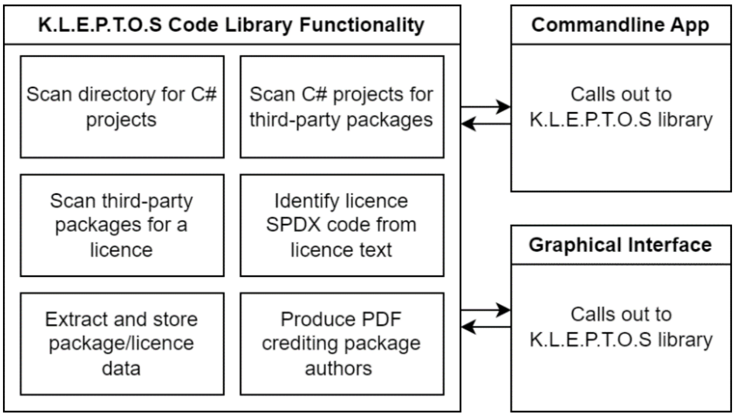

K.L.E.P.T.O.S Code Library Functionality
Scan directory for C# projects
Scan C# projects for third-party packages
Scan third-party packages for a licence
Identify licence SPDX code from licence text
Extract and store package/licence data
Produce PDF crediting package authors
Commandline App
Calls out to K.L.E.P.T.O.S library
Graphical Interface
Calls out to K.L.E.P.T.O.S library

Figure 13 Project Deliverable Visualisation

1.8 Project Constraints

As defined in Appendix 2. Project Initiation Document (pp. 10) support for the project was a low priority according to Intellectual property counsel, primarily due to ongoing business operations. Meetings to discuss best practices were expected to be brief and infrequent. Although, input from the business' legal counsel was required as an ethical consideration, as ensuring the final product includes proper credit and compliance with package licenses is imperative to protecting external developers' intellectual property.

Scope	All deliverables must be functional on a windows machine with no additional requirements
	The deliverables cannot rely on access to any third-party technologies that require paid-for services
	Deliverables cannot make use of services that share source-code for licence detection
Time	Presentation of project initiation - 28th March
	System development Project Report – 23rd June
	Demonstration of Artefact – 5th July
Cost	Budget: £0 (production to be incorporated to normal hours and personal time)
Quality	Test Driven Development (TDD) – functionality is only produced after a test is defined for the logic.
Risk	No external libraries used in the development of the tool can be used under a copy-left licence
	All external libraries must use supported libraries with public documentation
Benefits	Must reduce manual time acquiring licences
	Must produce design consistent credit document with minimal manual interaction

Table 4 Project Constraints

The project was required to be developed in C# using the Visual Studio suite of tools, adhering to Edwards standards. Only the utilisation of free open-source packages and original logic was permitted. It is important to note that artificial intelligence code generators were not allowed for support, in accordance with the company guidelines at the time of writing.

1.9 Risk Register

	Risk	Effect	Impact	Probability	Priority	Mitigation
🟦 **1**	Demanding day to day operations	work takes priority over the project	8	7	5.6	Mediate with the early careers and line manager to oversee workloads
🟩 **2**	Development requirements are underestimated	important functionality is left unfinished	8	6	4.8	Use incremental SDLC for functioning increments. Use MoSCoW to identify most critical functionality
🟩 **3**	Additional learning required for project.	Delayed and slowed rate of development.	5	6	3	Discuss progress with superiors and negotiate learning support from team members
🟥 **4**	Unanticipated complexity in discovering licenses	Extended development time.	4	6	2.4	meetings are required to assess license priorities on ad hoc basis as required by the proof of concept
🟩 **5**	Scope creep for license retrieval methods	increased development workloads	7	2	1.4	Stick to strict incremental SDLC and stories defined in PID.
🟥 **6**	Low availability of Intellectual Property Counsel	Licence compliance blocks development	4	3	1.2	liaise in advance. In absence of counsel, confer with line-manager for best guess implementation
🟩 **7**	K.L.E.P.T.O.S is ineffective at returning licenses	K.L.E.P.T.O.S is rejected by stakeholder	1	5	0.5	Marked as a proof-of-concept project, this must be anticipated, and should not be discouraging. Honesty is imperative.
X	Being banned from NuGet server	Unable to test or develop code against the server	8	-	8	Limit program concurrency to prevent reaching server limits.

Table 5 Risk Register

X Unexpected Risks 🟥 Schedule Risks 🟦 Operational Risks 🟩 Organisational Risks

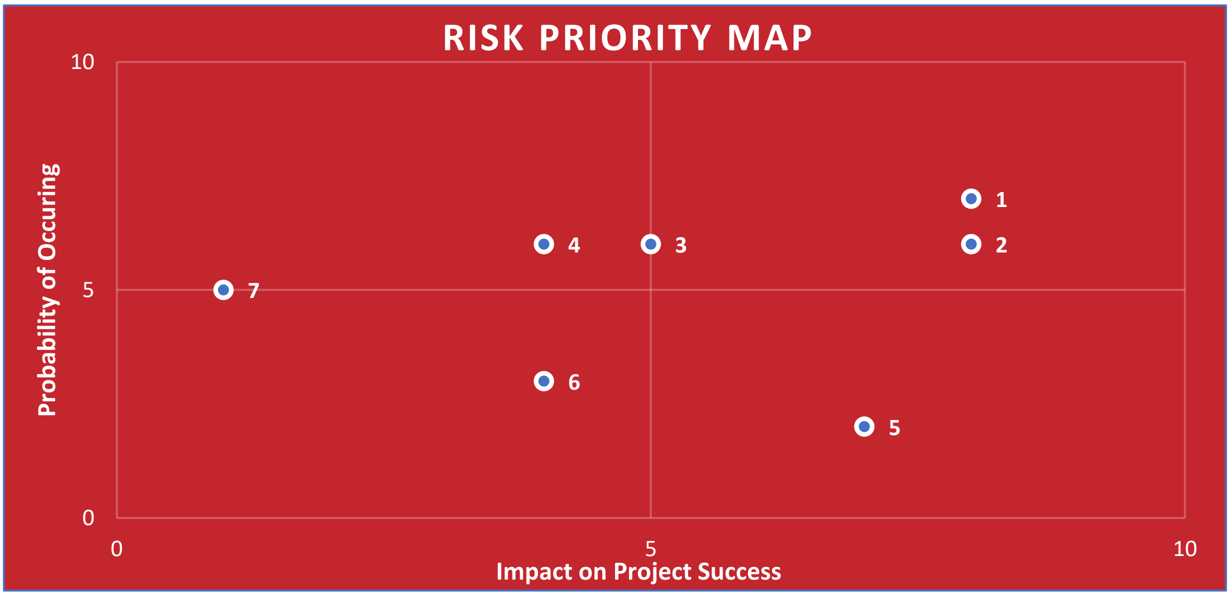


Figure 16 Risk Priority Map

St

Figure 17 Risk Priority Map

2 Specification of Requirements

To understand the variables influencing the project, a combination of primary and secondary research was conducted. Primary research comprised regular meetings with managers and team members, only after seeking ethical approval from the University of Chichester Appendix 6. Application for Ethical Approval.

2.1 Environmental Analysis - PESTLE

Understanding the environment in which Edwards Vacuum operates is crucial for an accurate assessment of requirements. Analysing the environment using the PESTLE model helps identify the variables that impact project direction and requirements. (Reding, 2021)

Political	Semi-conductor subsidies such as the CHIPs bill (Reich, 2022) in the United States has increased demand for technology driven pump-systems
	The US has banned the sale of Semi-conductor Manufacturing Equipment (SME) to China, with potential impact on Edwards Vacuum and Atlas Copco if tensions escalate. (Bureau of Industry and Security, 2022)
Economic	Supply-chain disruptions and inflated hardware prices have arisen due to being intertwined heavily with political issues. (Lu, 2022)
	Edwards Vacuum is estimated to account for ~50% of vacuums sold worldwide. Although, pumps, and pump-equipment the biggest expense for semiconductor OEMs, making them a key focus for cost savings (West, 2021).
Social	Edwards Vacuum abides by the Atlas Copco Business Code of Practice, priding itself on a socially sustainable supply chain and relationships. (Atlas Copco, 2020)
	A strict attitude to a right-by-me approach to internal and external relationships
Technological	Edwards Vacuum research cutting-edge technologies to optimize customer output in the semiconductor environment, both mechanically and digitally.
	Excluding legacy equipment, all Edwards systems are capable of digital control and management, requiring an experienced in-house software applications team.
Legal	Where dealing internationally, the company must comply with a high variety of hardware compliance, software copyright, and patent laws.
	Recent anti-trust (monopoly) and trade secret controversies resulted a confidential settlement with a competitor. (Oregon-U.S. District Court, 2022)
Ecological	Edwards produce specific pump technologies to reduce environmental impact such as, abatement systems, harsh powder capture, and acid gas scrubbing.
	All Edwards facilities are powered by renewable offset energy supplies.

Table 6 Business PESTLE analysis (Reding, 2021)

2.2 Environmental Analysis – SWOT

The variables observed in the PESTLE analysis can be augmented by performing a SWOT analysis on the internal and external influences on Edwards Vacuum. (Kenton, 2022)

Strengths	Weaknesses
• High market share of Vacuum Systems (low customer mobility) • American policies to in-shore semiconductor manufacture • Economies of scale & low product variety • Positive social image as an ethical business • Experienced software development teams	• Saturated market share • Exposed to political tensions and sanctions beyond business influence • Legacy equipment in the field • Inflated component and labour prices • Legal complexity with international business, operating across 5 continents
Opportunities	**Threats**
• Greater investment into digital pump management technologies, benefiting from economies of scale and low customer mobility permitting less competitive pricing • Increased demand for vacuum equipment from policies promoting in-shoring of semiconductor manufacturing • Combining legal counsel with in-house software teams to optimise legal processes	• Policies preventing sale and service of vacuum equipment to China and Taiwan could cut revenue streams • Legacy equipment returning for service requires disproportionate resources due to obsolete components • Saturated market share limits growth in a stable market, and innovation is limited in a well-developed technology

Table 7 Business SWOT Analysis (Kenton, 2022)

With recent lawsuits against the business, it proves that despite its size and concealed nature, the business is not above legal scrutiny, and increased demand from new fabs in the U.S. comes with it the requirement of rapid development of modern pump management software. Without compromise to the integrity of the legal processes that verify our applications, it is critical to business success that these processes are optimised.

An automated software solution to resolve open-source licence discovery will have to comply with these requirements and navigate the issue delicately.

2.3 What is an Open-Source Licence?

When sharing a code library, it is important to clarify the terms and conditions for its use. Authors often specify whether their code is free for personal or commercial use and may require subsequent implementations to be open source. Licenses can also offer legal protection if the code malfunctions. Here are simplified explanations of different license types. (OpenSource.org, 2023)

	Open-Source Licences		
	Public Domain	Permissive License	Copy-Left
Generalised Description	Grants all rights. Freely accessible, usable, and modifiable.	A program that uses a package under a permissive license must mention the author and a copy of the original license text.	Same requirements as permissive licenses, although the program that implements a package must also be public under the same licence.
Example Licence Expressions	**PD** (Public Domain), **CC0** (Creative Commons).	**MIT** (Massachusetts Institute of Technology), **Apache-2.0** (Apache Software Foundation).	**GPL** (GNU General Public License), **MPL** (Mozilla Public License).

Least Restrictive — Most Restrictive

Table 8 Open-Source licence type summary (OpenSource.org, 2023)

With open-source software becoming more prevalent in personal and commercial use, the Software Package Data Exchange (SPDX) is a project to standardise package information such as licenses, with the aim:

> *"…to reduce redundant work by providing common formats for organizations and communities to share important data, thereby streamlining and improving compliance, security, and dependability."*
>
> (SPDX, 2021)

The SPDX project (https://spdx.dev) records and shares all ISO standardised open-source licences from A-Z on their website – of which there are over 500. Individuals are free to apply one or more of these licenses to their work, when sharing to a public repository (e.g., NuGet.org). All licenses recorded by the SPDX project have a unique case-sensitive, whitespace-sensitive, short-hand identifier, known as a "license expression" for example: MIT, Apache-2.0, and GPL. (SPDX, 2021)

As Edwards is a developer of proprietary software, the business' legal counsel advise that projects cannot make use of packages that are licensed under copy-left, as doing so would cascade the package's open-source requirement to our software, making Edwards products publicly available. Fortunately, packages licensed under permissive licenses are mostly compliant with the business' requirements and are acceptable to be used. [appendix?]

2.4 Open-source Licence Precedent

The Free Software Foundation (FSF) filed a lawsuit against Cisco Systems in 2008, alleging violations of licensing terms for FSF-copyrighted programs in Cisco's Linksys products. The Software Freedom Law Center represented FSF, seeking an injunction and profits from Cisco's alleged unlawful distribution. The case resulted in a settlement in 2009, with Cisco appointing a director for license compliance and making a financial contribution to FSF, proving that failing to adhere to license terms can have non-trivial financial consequences. (Smith, 2009)

2.5 Existing Licence Approval Process

Prior to 2013, Edwards' software products had minimal usage of open-source libraries. Since then, they have only incorporated a small number of acquired packages. Currently, as discussed with the legal counsel, there is no formal process for license approval, however approximately it involved developers documenting package dependencies and license details. This information is compiled into a spreadsheet of less than 50 packages, which is then shared with the SCRUM Master. As part of the product release, the SCRUM Master creates a README file to comply with individual license requirements. This solution is not scalable to projects that now involve upwards of one thousand packages.

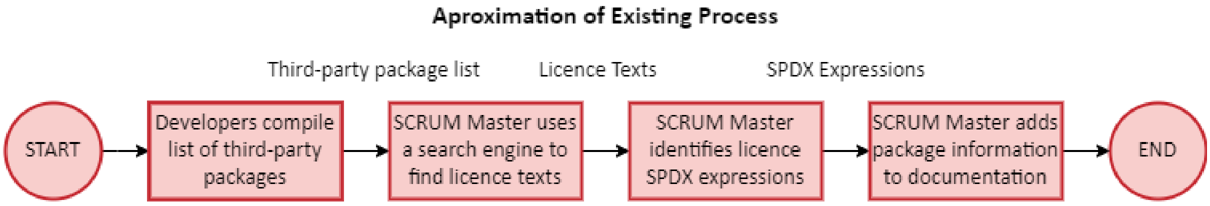

Aproximation of Existing Process
Third-party package list
Licence Texts
SPDX Expressions
START
Developers compile list of third-party packages
SCRUM Master uses a search engine to find licence texts
SCRUM Master identifies licence SPDX expressions
SCRUM Master adds package information to documentation
END

Figure 19 Existing licence discovery and attribution process

2.6 User Interface Requirements

As the project deliverable contains a graphical interface requirement, it is critical to consider the user's experience. Three important teachings from Schneiderman's 8 golden principles of design are consistency, user feedback, and reduce short-term memory load. As proof-of-concept internal tool, implantation of Schneiderman's remaining rules, such as universal usability and permitting reversal of actions are not feasible within the project constraints. Although, the interface will still be required to be accessible to a non-technical user, and intuitive to use. Also employing Gutenberg's areas of strength to give the interface flow to mitigate cognitive strain. Features are demonstrated in Appendix 5. User Interface and Design.

2.7 Package License Text Identification – Distance Algorithms

Where a package's metadata includes a URL to a license text, but does not include a license expression, the program must retrieve the license text from the licence's webpage. The text content of the licence webpage invariably has subtle changes to the official licence text such as author names, dates, and webpage content, this makes identifying a license from a literal string comparison against the SPDX database ineffective. The program must be able to identify the license in a more human way.

A number of algorithms implement what's known as 'fuzzy' or 'approximate' string matching, typically based off of a concept known as 'edit-distance'. These algorithms compare two strings and output a score relative to the number of hypothetical edits that would be required to move from one string to another. (Navarro, 2023)

The consideration of varying algorithms is expanded upon below.

2.7.1 Word Mover's Distance (WMD)

The Word Mover's Distance Algorithm capitalises upon the word2vec and GloVe techniques of vectorising the semantic relationship between words valued on their proximity in training data. Once trained, the semantic relationship vectors can be used to compare two strings.

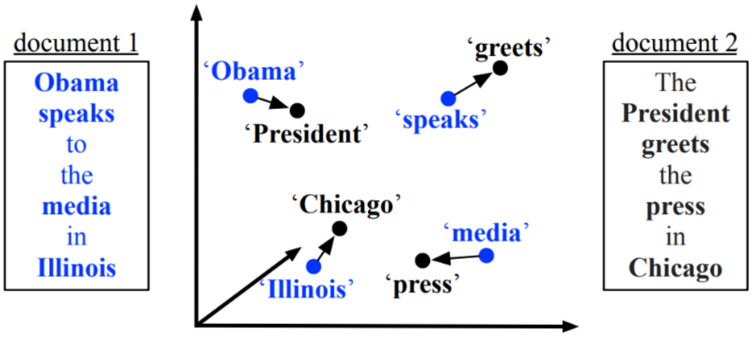


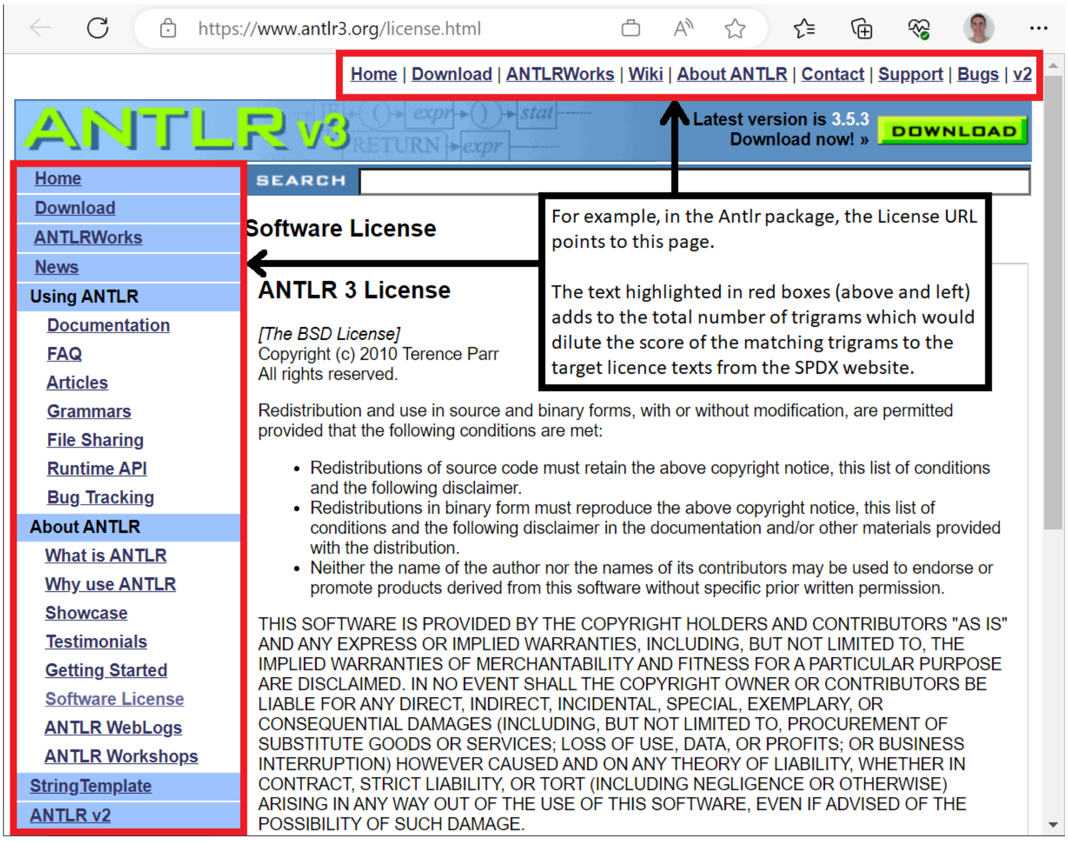
Figure 20 Word2Vec embedding visualisation (Matt J Kusner,

WMD encompasses both semantic and syntactic aspects in assessing the similarity between text documents. This metric quantifies the dissimilarity between two texts by determining the minimum distance required for the embedded words of one document to traverse in order to reach the embedded words of the other document. (Matt J Kusner, 2015)

License texts are sensitive to more than just semantics of substitute words, but also the specific order and position of the words in license texts, which the WMD formula does not consider effectively.

2.7.2 Trigram Distance Formula (TDF)

The Trigram Distance Formula operates by stripping whitespace and dividing into overlapping sequences of three characters, known as trigrams. For example, the string "example" would generate the trigrams "exa", "xam", "amp", "mpl", "ple". The trigrams from both strings are compared to identify the trigrams they share. The distance is computed by subtracting the count of common trigrams from the total number of trigrams in both strings. This calculation provides a measure of dissimilarity between the strings. (National Library of Medicine, 2023)



Software License

ANTLR 3 License

[The BSD License]
Copyright (c) 2010 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For example, in the Antlr package, the License URL points to this page.

The text highlighted in red boxes (above and left) adds to the total number of trigrams which would dilute the score of the matching trigrams to the target licence texts from the SPDX website.

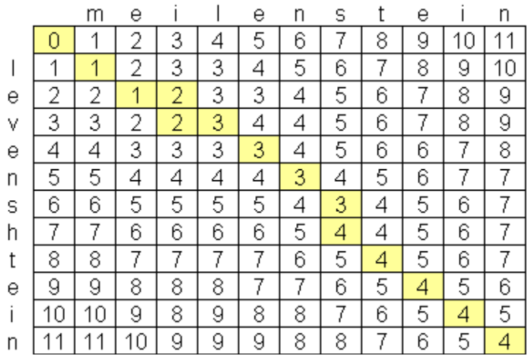
Figure 21 Example licence webpage, annotated

The TDF approach would again be unsuitable, as data retrieved from a package’s license page can include far more text than the license text alone, such as headers, footers, and asides. This would increase the number of non-matching Trigrams and dilute the score of the match making the result inaccurate.

2.7.3 Levenshtein Distance Formula (LDF)

The Levenshtein distance formula calculates the minimum number of single-character or whole word edits required to transform one string into another. It works by initializing a matrix with dimensions based on the lengths of the input strings.

Through iteration and comparison of words, costs are calculated for each position in the matrix, representing the transformation required to reach that point. The matrix is then updated by assigning the minimum cost among insertion, deletion, or substitution operations. The Levenshtein distance is obtained from the bottom-right element of the matrix and provides a measure of dissimilarity between the strings. (Levenshtein.net, 2023)



		m	e	i	l	e	n	s	t	e	i	n
	0	1	2	3	4	5	6	7	8	9	10	11
l	1	1	2	3	3	4	5	6	7	8	9	10
e	2	2	1	2	3	3	4	5	6	7	8	9
v	3	3	2	2	3	4	4	5	6	7	8	9
e	4	4	3	3	3	3	4	5	6	6	7	8
n	5	5	4	4	4	4	3	4	5	6	7	7
s	6	6	5	5	5	5	4	3	4	5	6	7
h	7	7	6	6	6	6	5	4	4	5	6	7
t	8	8	7	7	7	7	6	5	4	5	6	7
e	9	9	8	8	8	7	7	6	5	4	5	6
i	10	10	9	8	9	8	8	7	6	5	4	5
n	11	11	10	9	9	9	8	8	7	6	5	4

Figure 22 Levenshtein Matrix, path of fewest edits (Levenshtein.net, 2023)

The formula is mathematical proof of the fewest number of edits required to move from one string to another. The example given compares individual characters but can be implemented where each column/row represents a word. The benefit of Levenshtein Distance is that it preserves word order and is not skewed if the string on one axis is a substring of the other.

2.7.4 Comparing Algorithm Time Complexities

Big O notation is a mathematical notation used to describe the efficiency or scalability of an algorithm by analysing its worst-case behaviour as the input size grows. It provides a simplified way to express how the runtime or resource usage of an algorithm increases relative to the input size. (Massachusetts Institutes of Technology, 2023)

	Trigram Distance	**Levenshtein Distance**	**Word Mover's Distance**
Big O Notation	O(n + m)	O(n * m)	O(n^3 log(n))
Values	*n* is the number of **letters** in string 1, *m* is the number of **letters** in string 2.	*n* is the number of **words** in string 1, *m* is the number of **words** in string 2.	*n* is the number of **words** between all input strings.
Complexity	Linear increase against *n*.	Approximately squared increase against *n*.	Approximately cubic increase against *n*.

Highly Scalable

Less Scalable

Table 9 Big O notation algorithm scalability comparison

2.7.5 Research Conclusion

The Levenshtein formula was selected as the primary choice for license identification due to its enhanced accuracy in recognizing word order, ability to accept substring and literal word comparisons, and improved scalability compared to the Word Mover's Distance algorithm.

2.8 Use Case Diagrams

Before undertaking research design, it's important to understand the use cases, and under what circumstances a user will interact with the program. Use cases are linked to the projects acceptance criteria Appendix 2. Project Initiation Document (pp.11)

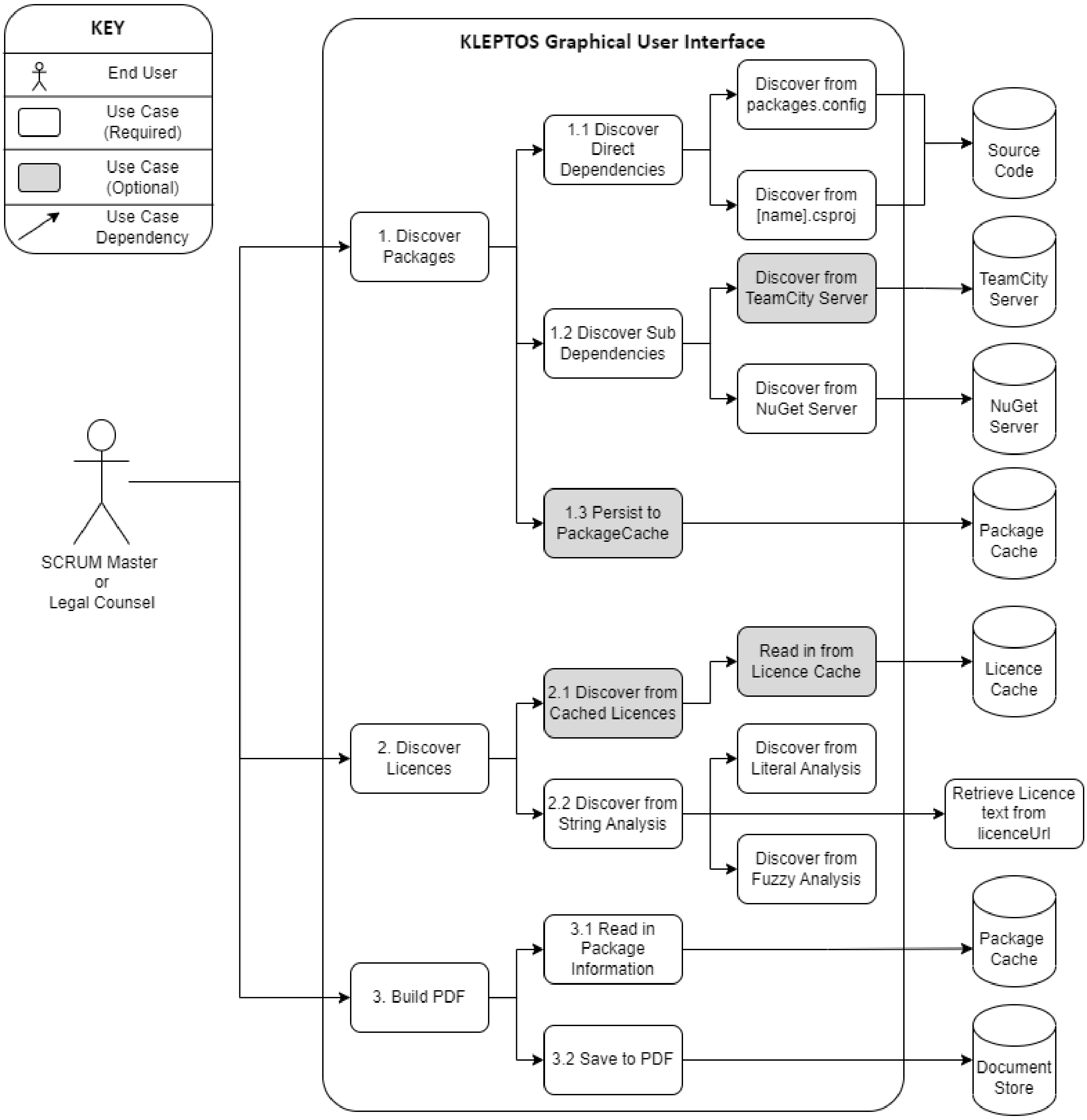

KEY
End User
Use Case (Required)
Use Case (Optional)
Use Case Dependency
KLEPTOS Graphical User Interface
SCRUM Master or Legal Counsel
1. Discover Packages
1.1 Discover Direct Dependencies
Discover from packages.config
Discover from [name].csproj
Source Code
1.2 Discover Sub Dependencies
Discover from TeamCity Server
TeamCity Server
Discover from NuGet Server
NuGet Server
1.3 Persist to PackageCache
Package Cache
2. Discover Licences
2.1 Discover from Cached Licences
Read in from Licence Cache
Licence Cache
2.2 Discover from String Analysis
Discover from Literal Analysis
Discover from Fuzzy Analysis
Retrieve Licence text from licenceUrl
3. Build PDF
3.1 Read in Package Information
Package Cache
3.2 Save to PDF
Document Store

Figure 23 KLEPTOS use-case diagram

2.9 Solution Entity Relationships

The entity relationship diagram is a high-level UML diagram of the components that were required to build the project. Where an interface is used, the underlying logic varies, but the functions remain the same.

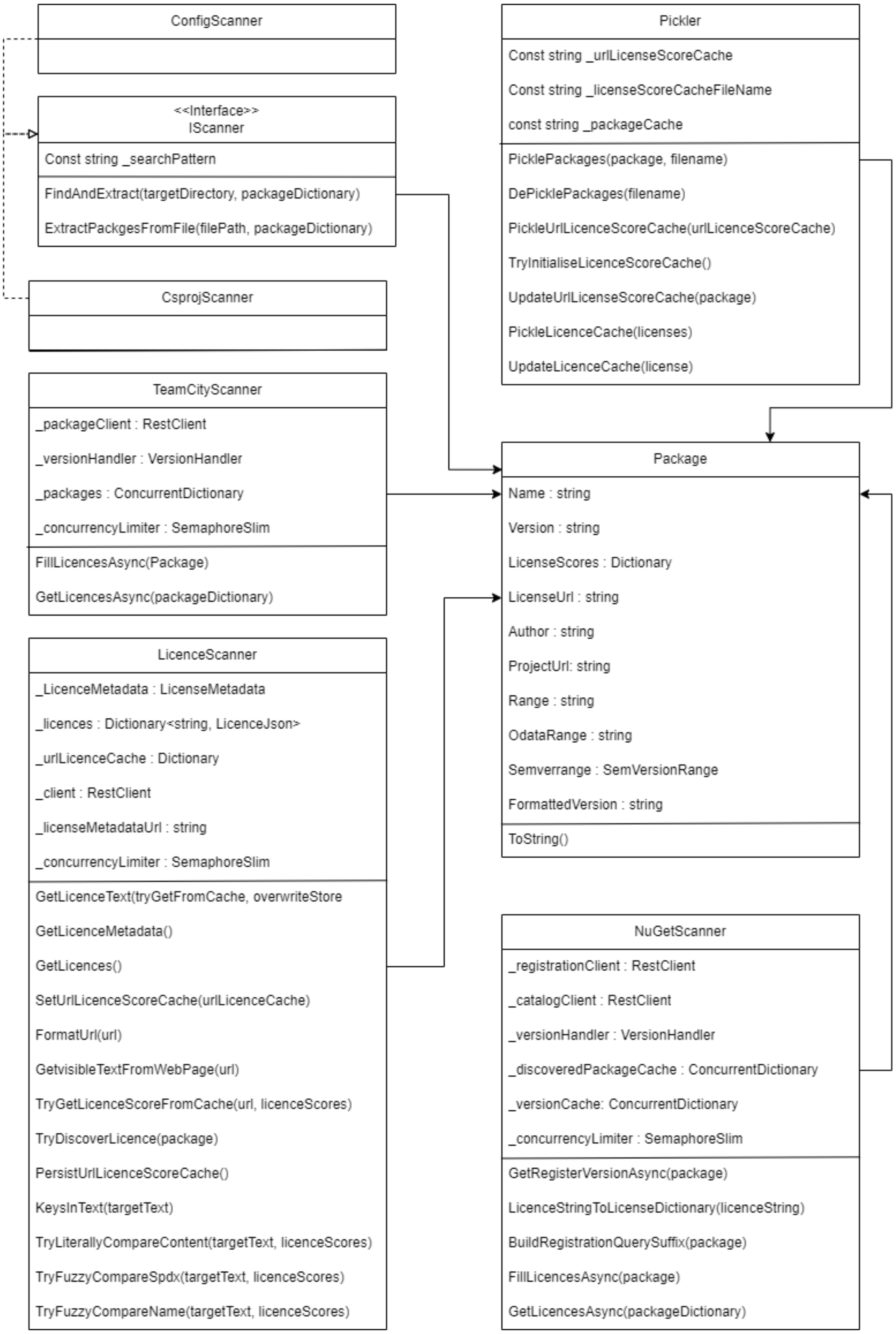

ConfigScanner
<<Interface>>
IScanner
Const string _searchPattern
FindAndExtract(targetDirectory, packageDictionary)
ExtractPackgesFromFile(filePath, packageDictionary)
CsprojScanner
TeamCityScanner
_packageClient : RestClient
_versionHandler : VersionHandler
_packages : ConcurrentDictionary
_concurrencyLimiter : SemaphoreSlim
FillLicencesAsync(Package)
GetLicencesAsync(packageDictionary)
LicenceScanner
_LicenceMetadata : LicenseMetadata
_licences : Dictionary<string, LicenceJson>
_urlLicenceCache : Dictionary
_client : RestClient
_licenseMetadataUrl : string
_concurrencyLimiter : SemaphoreSlim
GetLicenceText(tryGetFromCache, overwriteStore
GetLicenceMetadata()
GetLicences()
SetUrlLicenceScoreCache(urlLicenceCache)
FormatUrl(url)
GetvisibleTextFromWebPage(url)
TryGetLicenceScoreFromCache(url, licenceScores)
TryDiscoverLicence(package)
PersistUrlLicenceScoreCache()
KeysInText(targetText)
TryLiterallyCompareContent(targetText, licenceScores)
TryFuzzyCompareSpdx(targetText, licenceScores)
TryFuzzyCompareName(targetText, licenceScores)
Pickler
Const string _urlLicenseScoreCache
Const string _licenseScoreCacheFileName
const string _packageCache
PicklePackages(package, filename)
DePicklePackages(filename)
PickleUrlLicenceScoreCache(urlLicenceScoreCache)
TryInitialiseLicenceScoreCache()
UpdateUrlLicenseScoreCache(package)
PickleLicenceCache(licenses)
UpdateLicenceCache(license)
Package
Name : string
Version : string
LicenseScores : Dictionary
LicenseUrl : string
Author : string
ProjectUrl: string
Range : string
OdataRange : string
Semverrange : SemVersionRange
FormattedVersion : string
ToString()
NuGetScanner
_registrationClient : RestClient
_catalogClient : RestClient
_versionHandler : VersionHandler
_discoveredPackageCache : ConcurrentDictionary
_versionCache: ConcurrentDictionary
_concurrencyLimiter : SemaphoreSlim
GetRegisterVersionAsync(package)
LicenceStringToLicenseDictionary(licenceString)
BuildRegistrationQuerySuffix(package)
FillLicencesAsync(package)
GetLicencesAsync(packageDictionary)

Figure 24 UML diagram of proposed solution

3 Analysis and Design

Following an options analysis performed in Appendix 2. Project Initiation Document (pp. 6), the solution decided upon was a proof-of-concept program as opposed to an off-the-shelf solution, that would attempt to replicate each of the stages of the license approval process that would be performed manually. The program would need to:

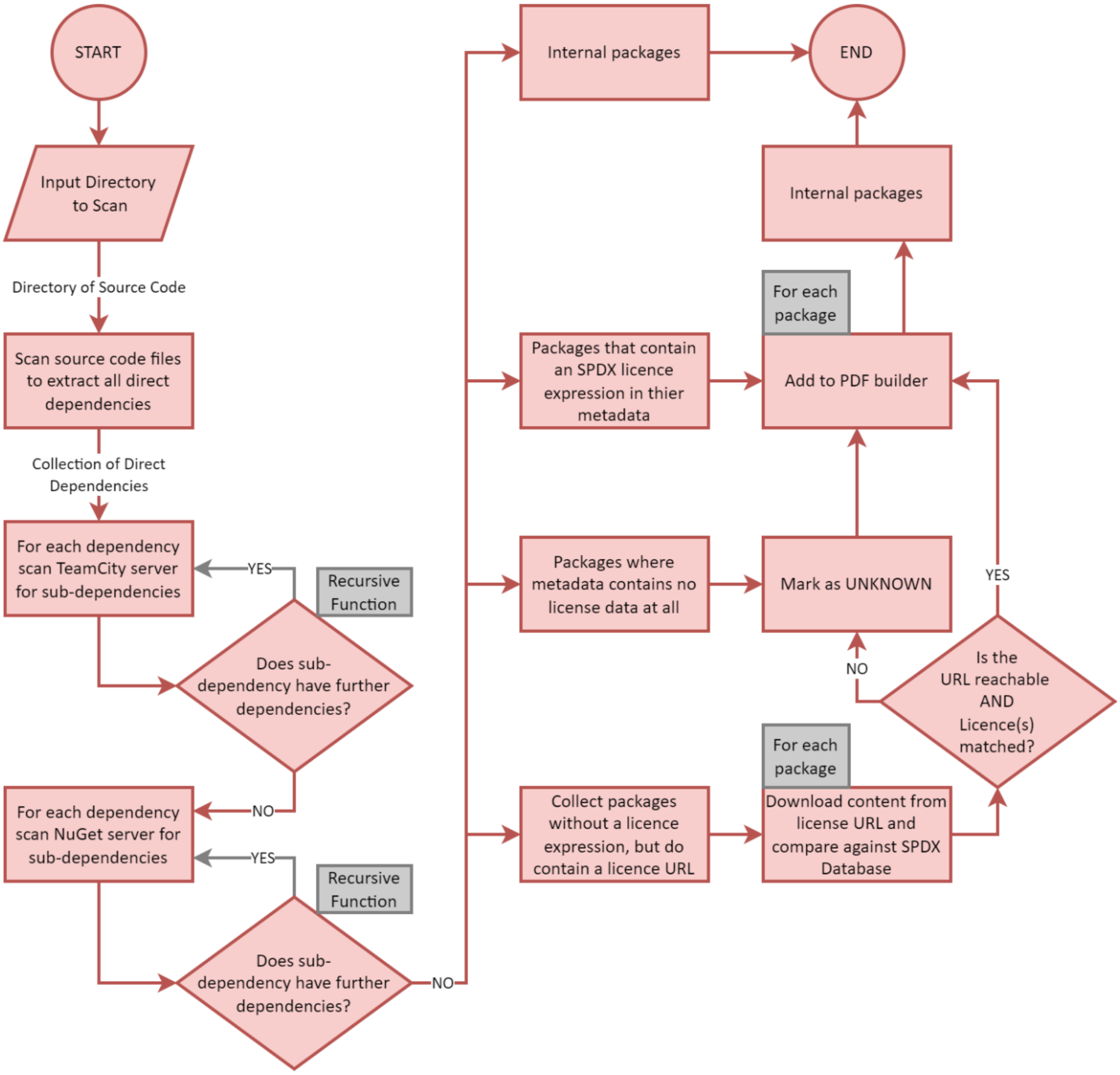

START
Input Directory to Scan
Directory of Source Code
Scan source code files to extract all direct dependencies
Collection of Direct Dependencies
For each dependency scan TeamCity server for sub-dependencies
YES
Recursive Function
Does sub-dependency have further dependencies?
NO
For each dependency scan NuGet server for sub-dependencies
YES
Recursive Function
Does sub-dependency have further dependencies?
NO
Internal packages
END
Internal packages
For each package
Packages that contain an SPDX licence expression in thier metadata
Add to PDF builder
Packages where metadata contains no license data at all
Mark as UNKNOWN
YES
NO
Is the URL reachable AND Licence(s) matched?
For each package
Collect packages without a licence expression, but do contain a licence URL
Download content from license URL and compare against SPDX Database

Figure 25 Flow diagram of proposed solution

As disclosed in the constraints, access to legal counsel is limited, furthermore with limited experience with the C# programming language (a requirement of the project), and little-to-no existing documentation for the internal and external NuGet APIs. Pairing these uncertainties, with no existing formal process, means that designing such a system to atomic granularity will be built upon an unsteady foundation. For this reason, I have elected to focus design to the most critical, mathematical, and replicable component of the code: Identifying package license texts from the SPDX database.

3.1 Design Approach

As the design requirements had a distinct level of uncertainty as a proof-of-concept program, much of the analysis and design of project specifics depended on external variables that were unknowable prior to engaging with undocumented Microsoft and TeamCity systems.

3.2 Package Dependency Gathering

The first increment of delivering a solution that could gather direct dependencies, posed no unexpected issues. Standard .Net functions were used to discover and deserialise all “packages.config” and “.csproj” files in a project directory. One optimisation was discovered, by restricting the directory search to only include those on the main trunk branch of the project. Eliminating the need to search redundant projects that were no longer in-use.

Issues did however arise, upon the following iteration when developing the recursive algorithm to query API endpoints with direct dependency information. Initially, three hurdles had to be overcome:

1. There is little public documentation that exists for accessing the NuGet.org server API. This was mostly approach with a brute-force approach to discovering query parameters and response types.
2. There is even less documentation for the internal TeamCity server where Edwards packages are stored. The TeamCity API uses a legacy equivalent of the public NuGet server, but queries and responses unexpectedly take an entirely different format.
3. The dependency information retrieved from both internal and external servers does not provide a license version, rather a range of versions that are accepted for the dependency. A custom version formatting algorithm was designed to extract the most recent dependency version from another API endpoint within the range syntax provided.

Once resolved and ahead of schedule, time was allowed to implement optimisations, including:

1. Asynchronous and threading applied to TeamCity and NuGet recursive package scanners. Although too successful, as the rate of requests was so great that the machine was suspended from the NuGet.org server, suspecting a DDoS attack. This delayed development for two days while a concurrency limiter was implemented.
2. Caching of discovered licences in a thread-safe concurrent dictionary. This reduced redundant requests, by checking the cache for discovered licences.

Once optimised, the package serialisation class was developed to enable the persisting of package data to JSON files locally.

The next increment was license discovery. From an example project, of the 827 third-party dependencies found, only 307 of them included the licence expression in their metadata. The remaining 520 packages would have to be identified from the webpage content linked by the licence URL. To achieve this, the following research had to be conducted.

3.3 PDF builder

The PDF builder class leveraged the PDFsharp library, although the library is comprehensive in its design functionality – allowing point control of graphics positioning – it is notably lacking in advanced features. Unlike comparative libraries in python such as FPDF, there is no text-wrap or tabulating function requiring the custom design of both.

3.4 User Interface

The command line interface was designed in tandem with much of the other functionality for manual testing, in the final iteration it was refined. The graphical interface made use of Visual Studio’s in-built WinForms designer and was far quicker to develop than anticipated. The addition of progress bars required some imaginative design and is estimate-based as it’s not possible to know how many dependencies will be discovered by the tool, it’s not possible to represent accurately what percentage of progress has been made. The final implementation is in Appendix 5 User Interface Research and Design.

Wireframe:

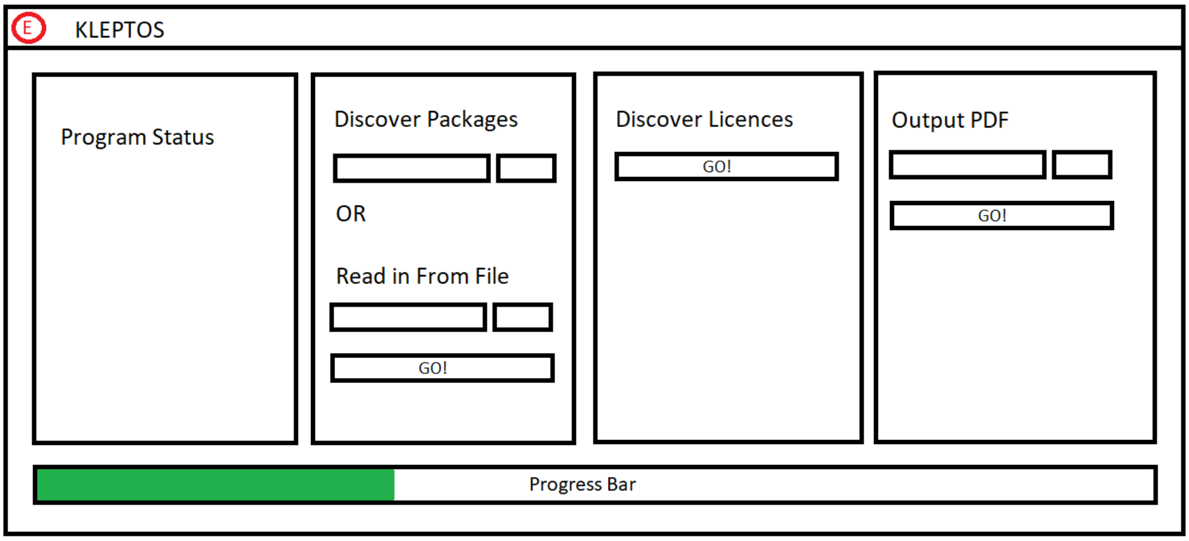
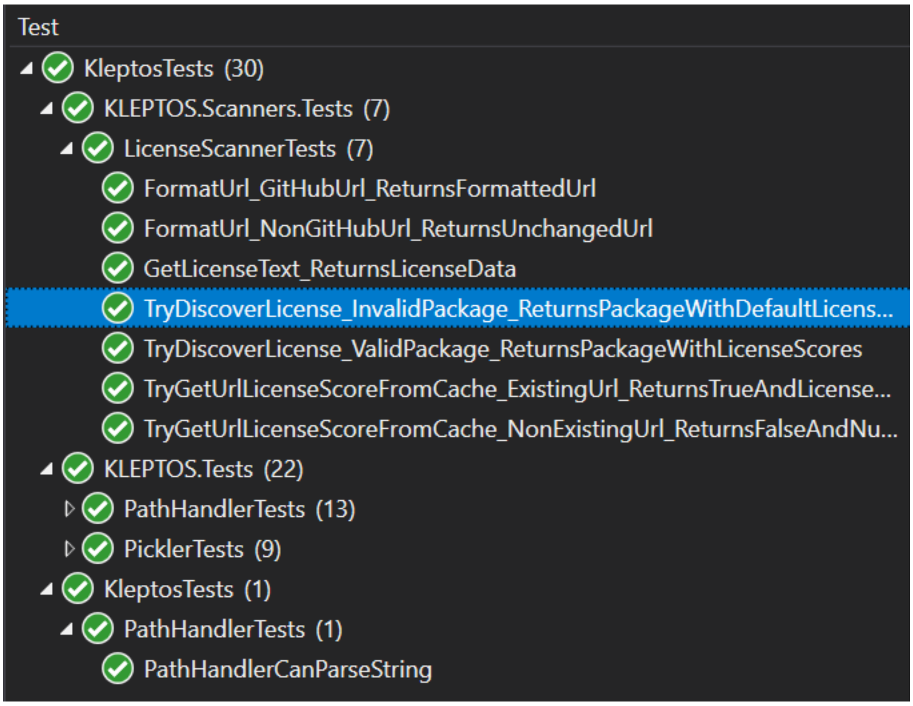


Figure 26 Wireframe of proposed graphical interface

3.5 Testing and Test Plan

Due to the proof-of concept delivery of the project, the priority to the project sponsor was a functional program, at their request unit testing was postponed to a later iteration of the project. To facilitate a quality requirement manual testing was conducted to ensure functionality was present Appendix 1. Manual Testing.

Some unit testing was conducted on critical functionality to assess functionality regression, evidenced below:



```
Test
◢ ✓ KleptosTests (30)
  ◢ ✓ KLEPTOS.Scanners.Tests (7)
    ◢ ✓ LicenseScannerTests (7)
        ✓ FormatUrl_GitHubUrl_ReturnsFormattedUrl
        ✓ FormatUrl_NonGitHubUrl_ReturnsUnchangedUrl
        ✓ GetLicenseText_ReturnsLicenseData
        ✓ TryDiscoverLicense_InvalidPackage_ReturnsPackageWithDefaultLicens...
        ✓ TryDiscoverLicense_ValidPackage_ReturnsPackageWithLicenseScores
        ✓ TryGetUrlLicenseScoreFromCache_ExistingUrl_ReturnsTrueAndLicense...
        ✓ TryGetUrlLicenseScoreFromCache_NonExistingUrl_ReturnsFalseAndNu...
  ◢ ✓ KLEPTOS.Tests (22)
    ▷ ✓ PathHandlerTests (13)
    ▷ ✓ PicklerTests (9)
  ◢ ✓ KleptosTests (1)
    ◢ ✓ PathHandlerTests (1)
        ✓ PathHandlerCanParseString
```

Figure 27 Unit testing

Chapter 4: Conclusion and Recommendations

Upon delivery and demonstration of the KLEPTOS solution to management, a project closure review was conducted. In cooperation with the Project Sponsor, the initial acceptance criteria from Appendix 2. Project Initiation Document (pp. 11) was compared against the delivered solution in a project review document Appendix 7. Project Review. From this review 14 out of 17 criteria were met. Two of the three un-achieved acceptance criteria were non-functional requirements that were not value adding to achieving the proof-of-concept solution. The criteria – after discussion with the manager – were forgone in favour of development of functional requirements, due to the resources available and time constraints. Despite this, a primitive user manual was produced to assist with project handover Appendix 8. KLEPTOS User Manual.

From the MoSCow deliverables, five out of five must haves were met, three out of three should haves were met, and one out of two could haves were met. The criterion that was not delivered, was the flagging of copy-left licences. The development of this functionality was attempted, although, was difficult to test, due to the absence of any copy-left licences being present in any Edwards products, this proved difficult to test or assure in quality, and therefore were omitted.

The one functional criterion recorded in the project review as N/a, was developed but not included in the project due to a radical change in design during the early stages of development, from installing every package and sub-dependency, to using the server APIs to retrieve metadata. No longer installing entire packages, and instead retrieving only the metadata was a major optimisation, reducing peak RAM consumption from 32GB, down to 256MB, and the impact on storage requirements from 60GB to 1GB. These optimisations were the recommendation of a senior engineer, as part of the formal code-review process, and were reviewed and approved by the project sponsor. The redesign of the functionality proved to be a major setback in the development timeline, but worthwhile in the end usability and accessibility to the program.

Despite the successes of achieving the deliverables and acceptance criteria, one criticism to be noted is that the initial project requirements did not consider the outcome that a significant number of third-party packages (developed by Microsoft) are not recorded in the SPDX database. Of 800 third-party packages discovered in one solution, 302 of them were Microsoft packages, of which only 27 were licensed under an SPDX license. Fortunately, the design choice to permit manual review licences prior to producing the PDF document enables these packages to be credited manually, although reduces the scalability of the solution due to manual input and reveals an oversight in the choice of SDLC. Possibly, an iterative approach would've supported an adjustment of approach to accommodate these licence cases.

Although the project underwent a significant change from Test-Driven Development under the instruction of the Project Sponsor, a unit test plan was devised and included Appendix 9. Test Plan to ensure future development and hand over of the tool conforms with best quality practice and discipline.

Upon project closure, the Project Sponsor, Brett Lawrence, provided this statement:

"The end-product was a success in delivering 14 of the 17 acceptance criteria. The decision to use the Levenshtein distance formula to identify licence expressions from license texts proves to be an out-of-the box, though particularly effective approach, with room to be researched and optimised further. Proving that licence information can be reliably and accurately extracted and credited via an automated solution paves the way for development into a fully-fledged internal tool to optimise and support the businesses needs in crediting third-party packages and authors."

- Brett Lawrence, Project Sponsor and Line Manager, June 2023

My personal learning from the project management experience is that communication and research are critical to success, especially when knowledge and resources from others are at your disposal. Had other engineers been consulted sooner in the planning stage, radical changes to the approach of functionality would not have been required.

As the project progresses into becoming a qualified internal tool, the approach to development will need to be corrected to an iterative approach, where additional functionality is no longer a priority, but instead, making major improvements and optimisations to existing logic. Furthermore, support from legal council must be obtained, and professional quality controls, put in place.

References

Adobe. (2022, March 18). *Waterfall Methodology: A Complete Guide*. Retrieved from Adobe: https://business.adobe.com/blog/basics/waterfall

Association for Qualitative Research. (2023, April 20th). *Iterative Approach*. Retrieved from Association for Qualitative Research: https://www.aqr.org.uk/glossary/iterative-approach#:~:text=An%20iterative%20approach%20is%20one,the%20inputs%20for%20subsequent%20interviews.

Atlas Copco. (2020). *Business Code of Practise.* Retrieved from Atlas Copco: https://viewer.atlascopco.com/bcop-ed11-2020-english-en/#page/1

Atlas Copco. (2023, May 23). *About Us*. Retrieved from Atlas Copco: https://www.atlascopcogroup.com/en/about-us#:~:text=Where%20industrial%20ideas%20come%20to,grow%20and%20drive%20society%20forward.

Bureau of Industry and Security. (2022, October 7). *Commerce Implements New Export Controls on Advanced Computing and Semiconductor.* Retrieved from Bureau of Industry and Security: https://www.bis.doc.gov/index.php/documents/about-bis/newsroom/press-releases/3158-2022-10-07-bis-press-release-advanced-computing-and-semiconductor-manufacturing-controls-final/file

Edwards Vacuum Ltd. (2023, May 23). *About us*. Retrieved from Edwards: https://www.edwardsvacuum.com/en-uk/about-us

Gates, W. H. (1976, February 3). *Open Letter to Hobbyists.* Retrieved from DigiBarn: https://digibarn.com/collections/newsletters/homebrew/V2_01/gatesletter.html

Haack, P. (2010, October 21). *Changing the NuPack Project Name*. Retrieved from Outercurve Foundation Blog: https://web.archive.org/web/20101025072857/http://www.outercurve.org/Blogs/EntryId/22/Changing-the-NuPack-Project-Name

Just a Drop. (2019). *Atlas Copco*. Retrieved from Just a Drop: https://www.justadrop.org/atlas-copco

Kenton, W. (2022, August 10). *SWOT Analysis: How To With Table and Example*. Retrieved from Investopedia: https://www.investopedia.com/terms/s/swot.asp

Levenshtein.net. (2023, May 5). *The Levenshtein Algorithm*. Retrieved from Levenshtein: Levenshtein.net

Lu, G. A. (2022, March 22). *Taiwan's Semiconductor Dominance: Implications for Cross-Strait Relations and the Prospect of Forceful Unification*. Retrieved from Center for Strategic & International Studies: https://www.csis.org/blogs/perspectives-innovation/taiwans-semiconductor-dominance-implications-cross-strait-relations

Massachusetts Institutes of Technology. (2023, may 6). *Big O.* Retrieved from Massachusetts Institute of Technology: https://web.mit.edu/16.070/www/lecture/big_o.pdf

Matt J Kusner, Y. S. (2015). *http://jmlr.org/proceedings/papers/v37/kusnerb15.pdf.* St. Louis, MO: Washington University.

Matt Kusner, Y. S. (2015, June 21). *From Word Embeddings To Document Distances.* Retrieved from jmlr.org: https://web.archive.org/web/20150621055801/http://jmlr.org/proceedings/papers/v37/kusnerb15.pdf

National Library of Medicine. (2023, May 4). *Trigram Algorithm*. Retrieved from National Library of Medicine: https://lhncbc.nlm.nih.gov/ii/tools/MTI/trigram.html

Navarro, G. (2023, May 4). *A Guided Tour to Approximate String Matching.* Retrieved from University of Chile: https://repositorio.uchile.cl/bitstream/handle/2250/126168/Navarro_Gonzalo_Guided_tour.pdf

OpenSource.org. (2023, April 20th). *OSI Approved Licenses*. Retrieved from open source initiative: https://opensource.org/licenses/

Oregon-U.S. District Court. (2022, September 7). *CM/ECF - USDC Oregon-Confirm Request.* Retrieved from Oregon-U.S. District Court: https://ecf.ord.uscourts.gov/doc1/15118467227

Paul Ganney, E. C. (2020). *Incremental Model*. Retrieved from Science Direct: https://www.sciencedirect.com/topics/computer-science/incremental-model

Reding, M. (2021, October 6). *What is a PESTLE Analysis?* Retrieved from CPD Online College: https://cpdonline.co.uk/knowledge-base/business/pestle-analysis/

Reich, R. (2022, June). *Why is the US about to give away $52bn to corporations like Intel?* Retrieved from The Guardian: https://www.theguardian.com/commentisfree/2022/jun/26/us-chips-act-intel-robert-reich

Robert Wysocki, R. S. (2012). *Effective project management: traditional, agile, extreme.* Indianapolis, indiana: Wiley.

scrum.org. (2023, May 17). *Welome to the Home of SCRUM!* Retrieved from scrum.org: scrum.org

Smith, B. (2009, May 20). *FSF Settles Suit Against Cisco*. Retrieved from Free Software Foundation: https://www.fsf.org/news/2009-05-cisco-settlement.html

SPDX. (2021). *Our Vision*. Retrieved from SPDX: https://spdx.dev/about/

West, J. (2021, August 10). *European Suppliers Dominate Vacuum Subsystems Market*. Retrieved from SEMI: https://www.semi.org/en/blogs/business-markets/european-suppliers-dominate-vacuum-subsystems-market

Table of Figures

Table of Tables

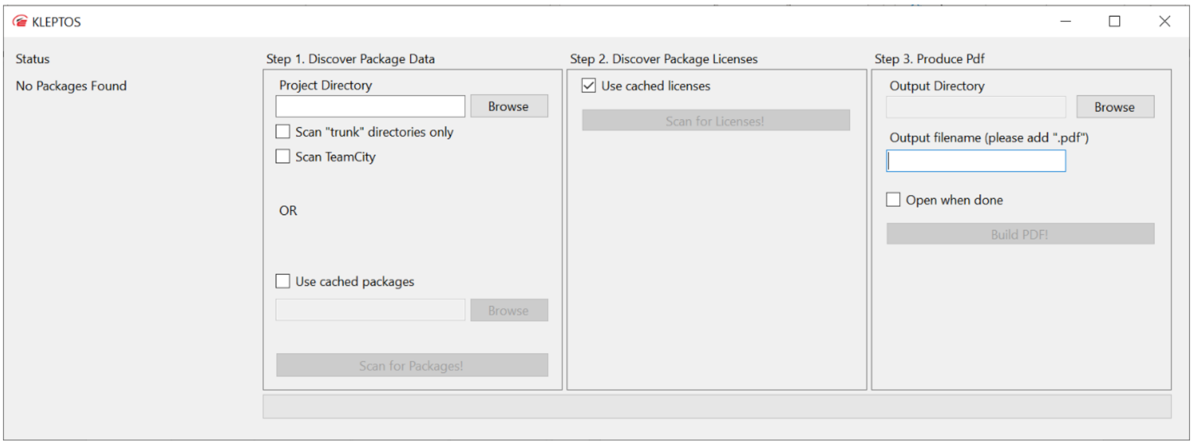
Glossary

Dependency	*A reference to a NuGet package that is required by either another NuGet package, executable program, or code library.*
Direct-Dependency	*A reference to a NuGet package that is directly required by an executable program.*
Sub-Dependency	*A NuGet package that is required by another NuGet package before it is installed in a project.*
API	*A set of functions or procedures that allow access to data or features of a service. Services are typically hosted online.*
API Request	*A command (sometimes containing a number of parameters) from one service to another API Service.*
Open-Source	*Software that has been made freely available to be redistributed and sometimes modified.*
Package	*A library of code that is able to be imported to other programs to reproduce functionality that is likely to be of use in more than scenario.*
NuGet Package	*A compressed library of code that is contains the necessary instructions to be imported to .Net programs to reproduce functionality that is likely to be of use in more than scenario.*

Appendix

1 Manual Testing

1.1 Initiation screen



No Packages Found

Project Directory:

□ Scan "trunk" directories only

□ Scan TeamCity

□ Use cached packages

☑ Use cached licenses

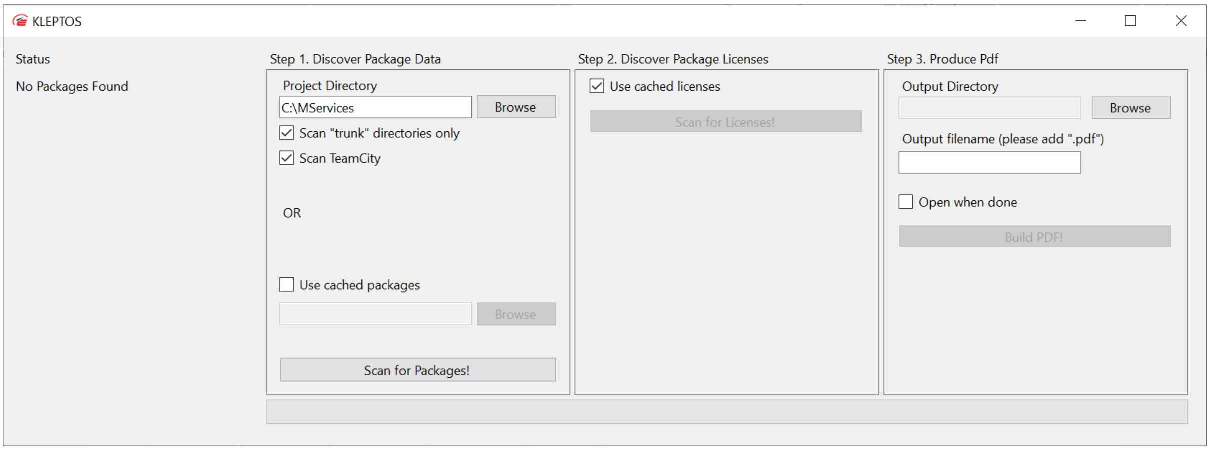
Output Directory:

Output filename (please add ".pdf"):

□ Open when done

1.2 Scan new Directory OR Read in persisted file

Field disabled “Use cached packages”:



No Packages Found

Project Directory: C:\MServices

☑ Scan "trunk" directories only

☑ Scan TeamCity

□ Use cached packages

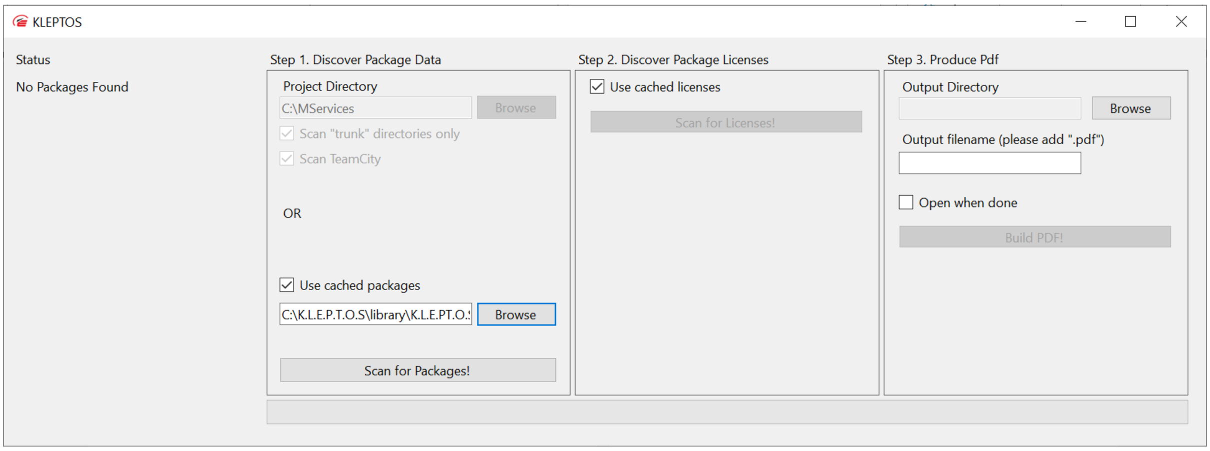
☑ Use cached licenses

Output Directory:

Output filename (please add ".pdf"):

□ Open when done

Field enabled “Use cached packages



No Packages Found

Project Directory: C:\MServices

☑ Scan "trunk" directories only

☑ Scan TeamCity

☑ Use cached packages

C:\K.L.E.P.T.O.S\library\K.L.E.PT.O.:

☑ Use cached licenses

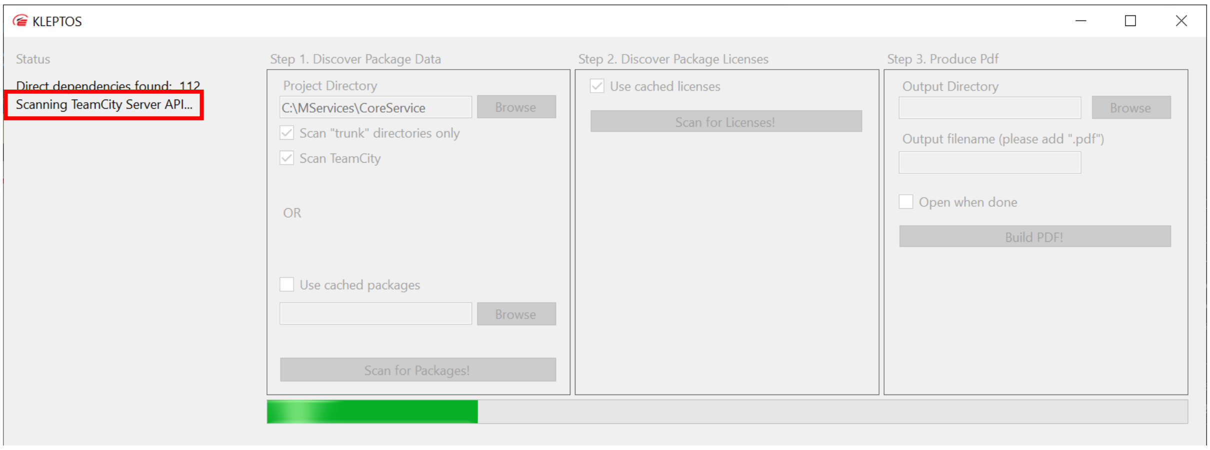
Output Directory:

Output filename (please add ".pdf"):

□ Open when done

1.3 Scanning new directory

Scanning Team City, all inputs disabled, 112 direct dependencies found:



KLEPTOS

Status

Direct dependencies found: 112
Scanning TeamCity Server API...

Step 1. Discover Package Data

Project Directory: C:\MServices\CoreService

☑ Scan "trunk" directories only

☑ Scan TeamCity

OR

□ Use cached packages

Step 2. Discover Package Licenses

☑ Use cached licenses

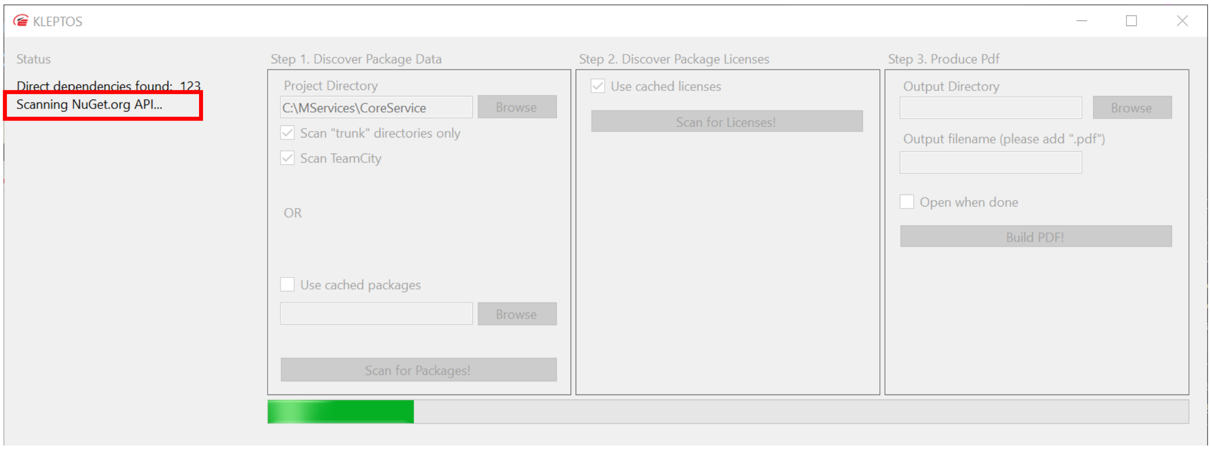
Step 3. Produce Pdf

Output Directory

Output filename (please add ".pdf")

□ Open when done

Scanning NuGet server:



KLEPTOS

Status

Direct dependencies found: 123
Scanning NuGet.org API...

Step 1. Discover Package Data

Project Directory: C:\MServices\CoreService

☑ Scan "trunk" directories only

☑ Scan TeamCity

OR

□ Use cached packages

Step 2. Discover Package Licenses

☑ Use cached licenses

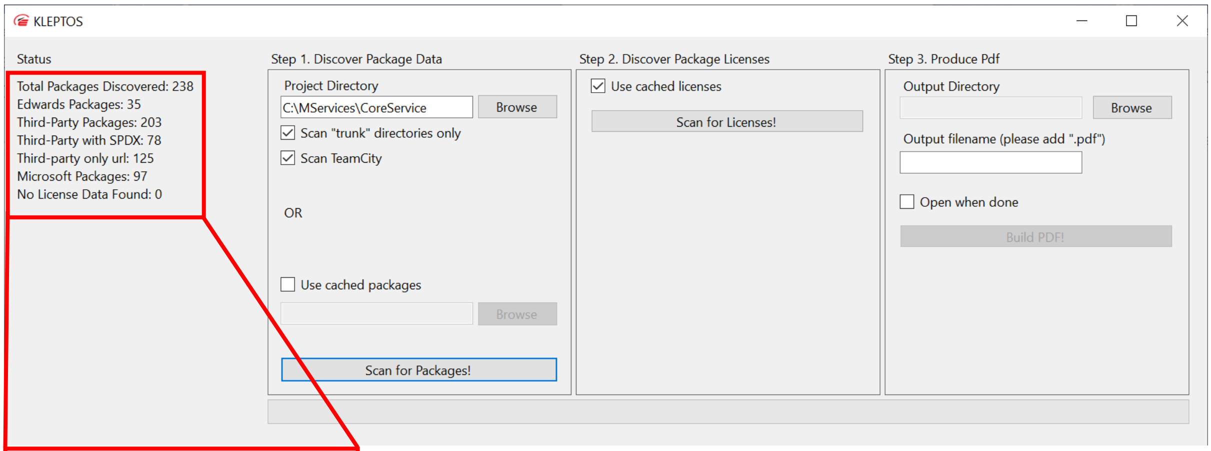
Step 3. Produce Pdf

Output Directory

Output filename (please add ".pdf")

□ Open when done

Discovered packages:



KLEPTOS

Status

Total Packages Discovered: 238
Edwards Packages: 35
Third-Party Packages: 203
Third-Party with SPDX: 78
Third-party only url: 125
Microsoft Packages: 97
No License Data Found: 0

Step 1. Discover Package Data

Project Directory: C:\MServices\CoreService

☑ Scan "trunk" directories only

☑ Scan TeamCity

OR

□ Use cached packages

Step 2. Discover Package Licenses

☑ Use cached licenses

Step 3. Produce Pdf

Output Directory

Output filename (please add ".pdf")

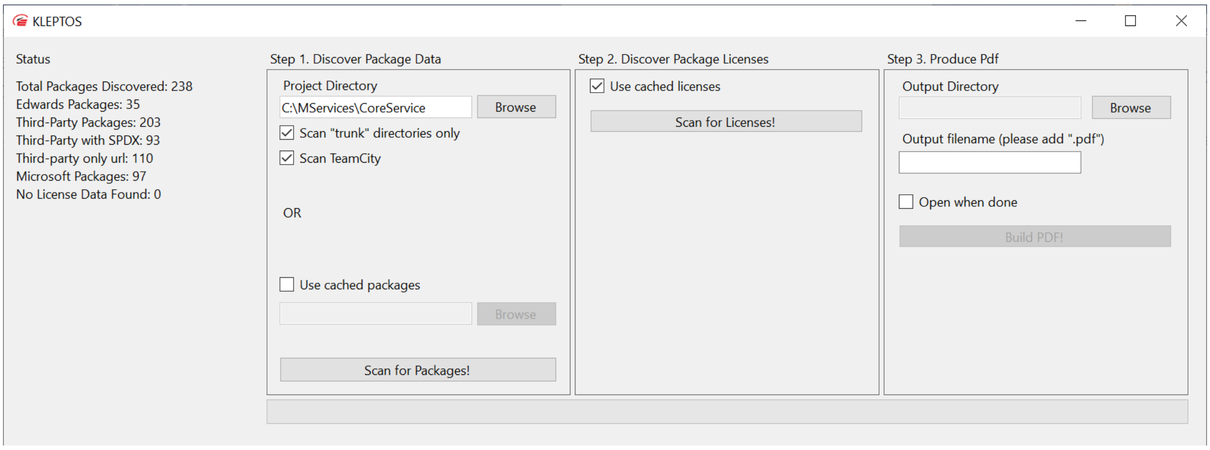
□ Open when done

Total Packages Discovered: 238
Edwards Packages: 35
Third-Party Packages: 203
Third-Party with SPDX: 78
Third-party only url: 125
Microsoft Packages: 97
No License Data Found: 0

< Total licences found.

1.4 Licence Scanner

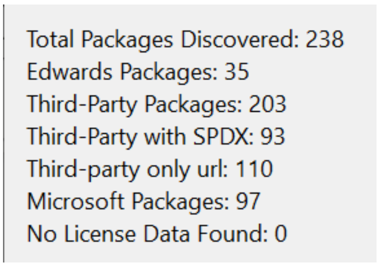
Previously only 78 licences contained SPDX license expressions. After running “Step 2. Discover package Licenses”, 93 package license expressions are found:



Status

Total Packages Discovered: 238
Edwards Packages: 35
Third-Party Packages: 203
Third-Party with SPDX: 93
Third-party only url: 110
Microsoft Packages: 97
No License Data Found: 0

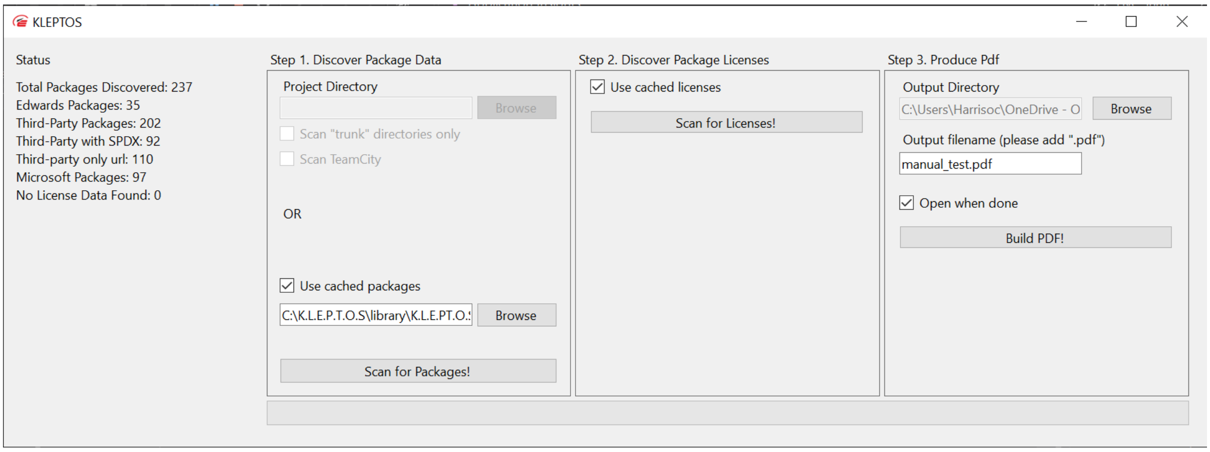
Project Directory: C:\MServices\CoreService
☑ Scan "trunk" directories only
☑ Scan TeamCity
☐ Use cached packages
☑ Use cached licenses
Output Directory:
Output filename (please add ".pdf"):
☐ Open when done



Total Packages Discovered: 238
Edwards Packages: 35
Third-Party Packages: 203
Third-Party with SPDX: 93
Third-party only url: 110
Microsoft Packages: 97
No License Data Found: 0

1.5 PDF Builder

Produce PDF input data filled out:



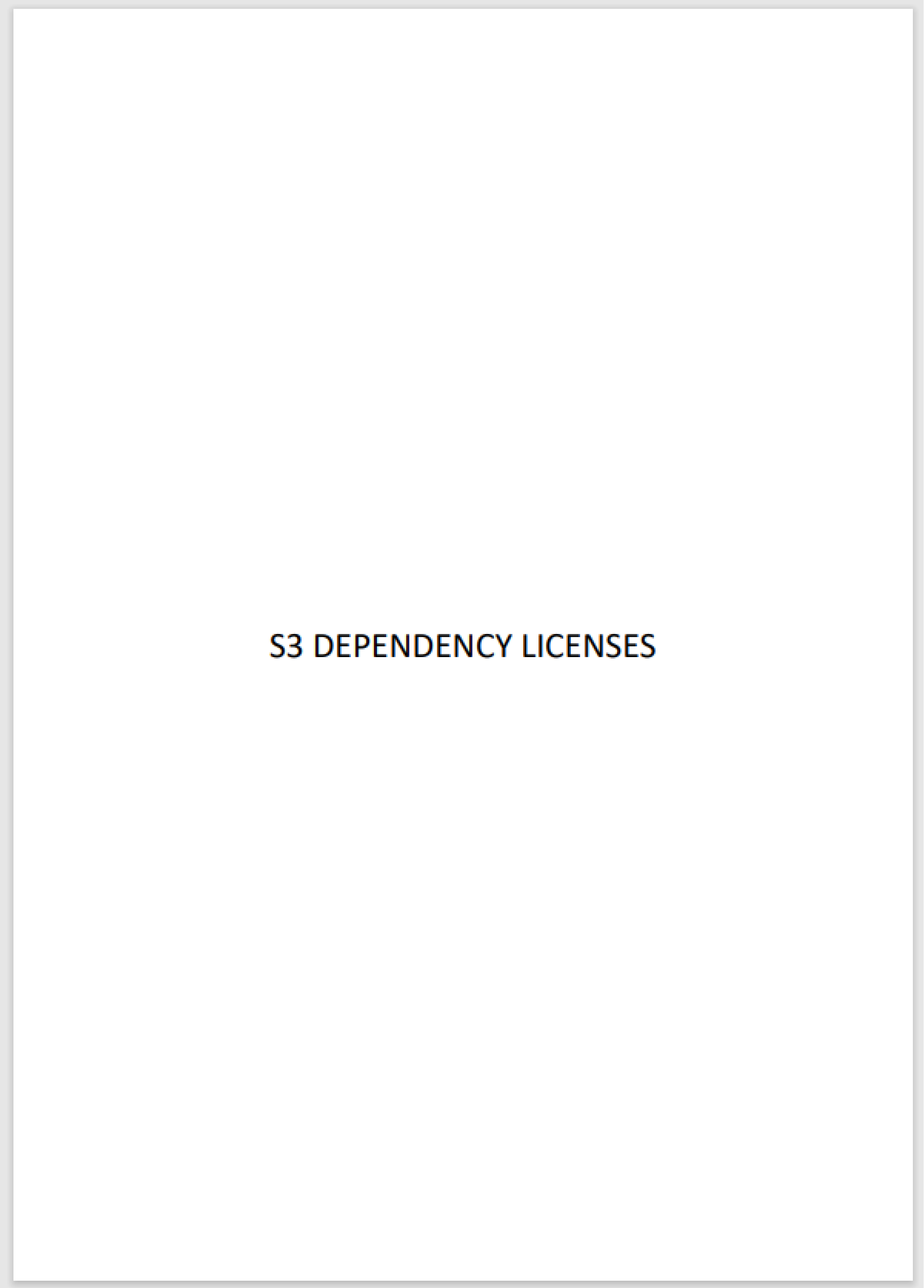
Status

Total Packages Discovered: 237
Edwards Packages: 35
Third-Party Packages: 202
Third-Party with SPDX: 92
Third-party only url: 110
Microsoft Packages: 97
No License Data Found: 0

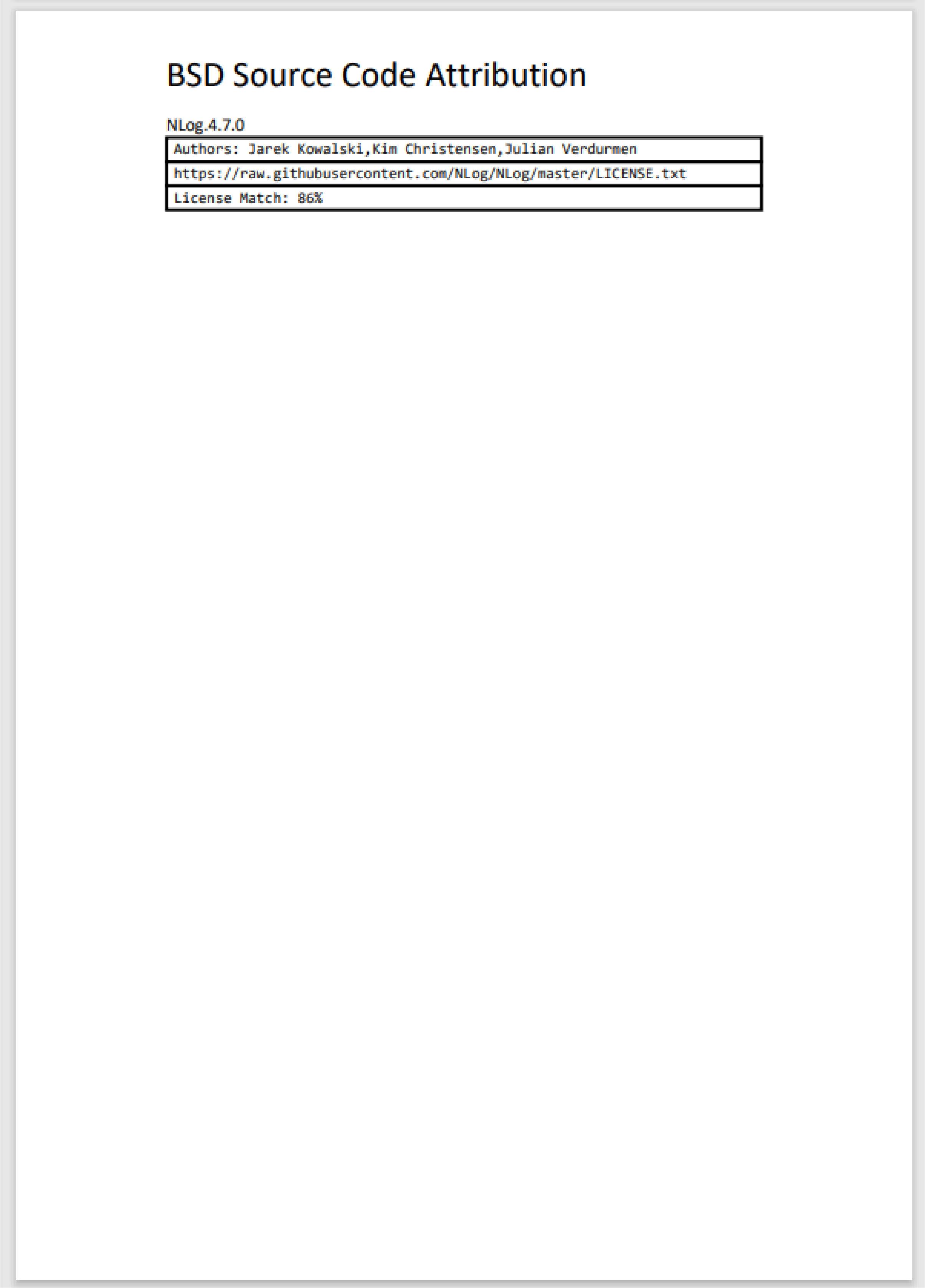
Project Directory:
☐ Scan "trunk" directories only
☐ Scan TeamCity
☑ Use cached packages: C:\K.L.E.P.T.O.S\library\K.L.E.P.T.O.:
☑ Use cached licenses
Output Directory: C:\Users\Harrisoc\OneDrive - O
Output filename (please add ".pdf"): manual_test.pdf
☑ Open when done

NOTE: upon reviewing previous screenshots, notice how the user is protected against advancing to the next step of licence discovery by disabling buttons ahead, for example, when packages are not discovered, or fields are not filled out.

PDF output filename manual_test.pdf:



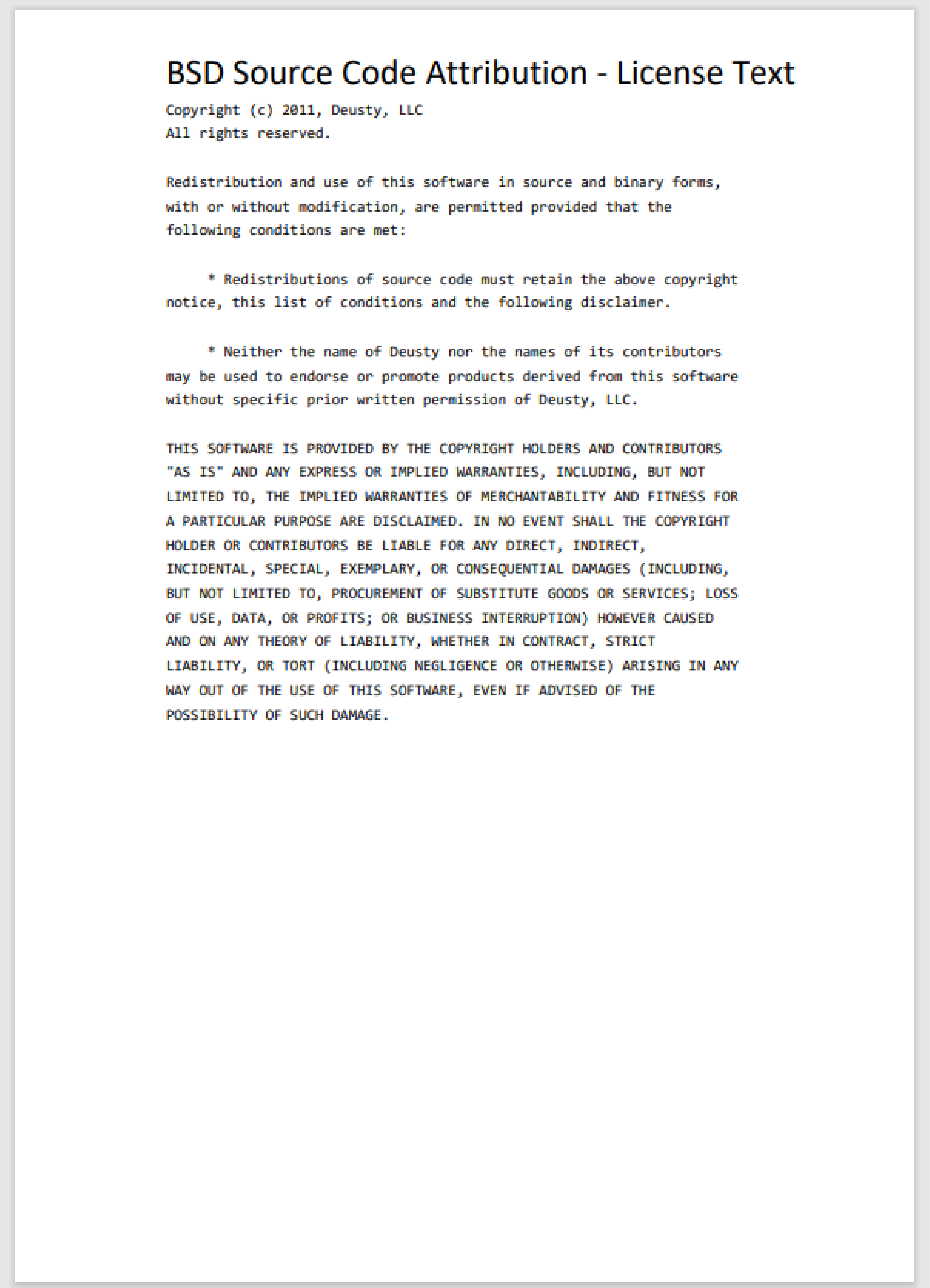
S3 DEPENDENCY LICENSES



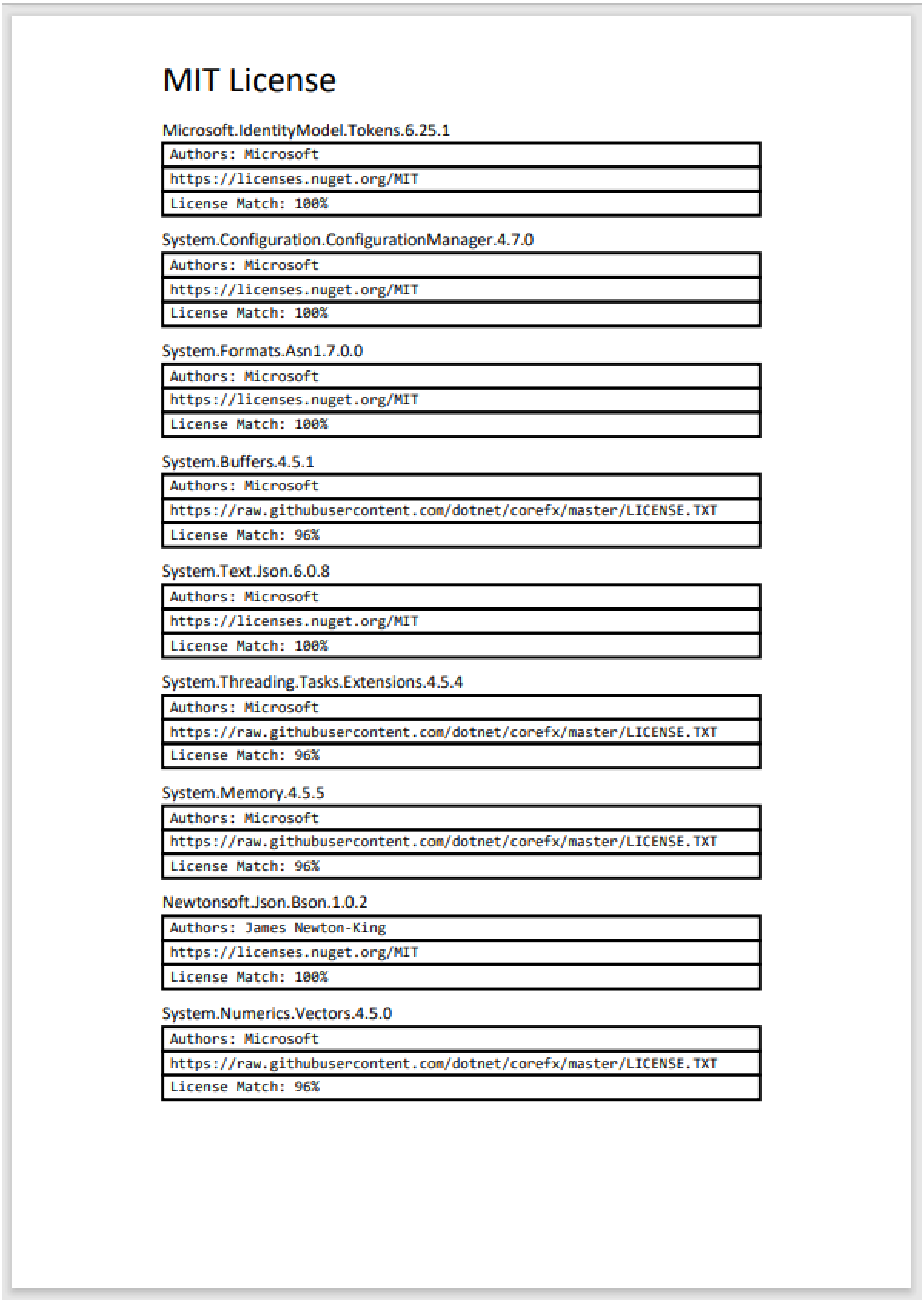
BSD Source Code Attribution

NLog.4.7.0

Authors: Jarek Kowalski,Kim Christensen,Julian Verdurmen
https://raw.githubusercontent.com/NLog/NLog/master/LICENSE.txt
License Match: 86%



BSD Source Code Attribution - License Text



MIT License

Microsoft.IdentityModel.Tokens.6.25.1

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Configuration.ConfigurationManager.4.7.0

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Formats.Asn1.7.0.0

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Buffers.4.5.1

Authors: Microsoft
https://raw.githubusercontent.com/dotnet/corefx/master/LICENSE.TXT
License Match: 96%

System.Text.Json.6.0.8

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Threading.Tasks.Extensions.4.5.4

Authors: Microsoft
https://raw.githubusercontent.com/dotnet/corefx/master/LICENSE.TXT
License Match: 96%

System.Memory.4.5.5

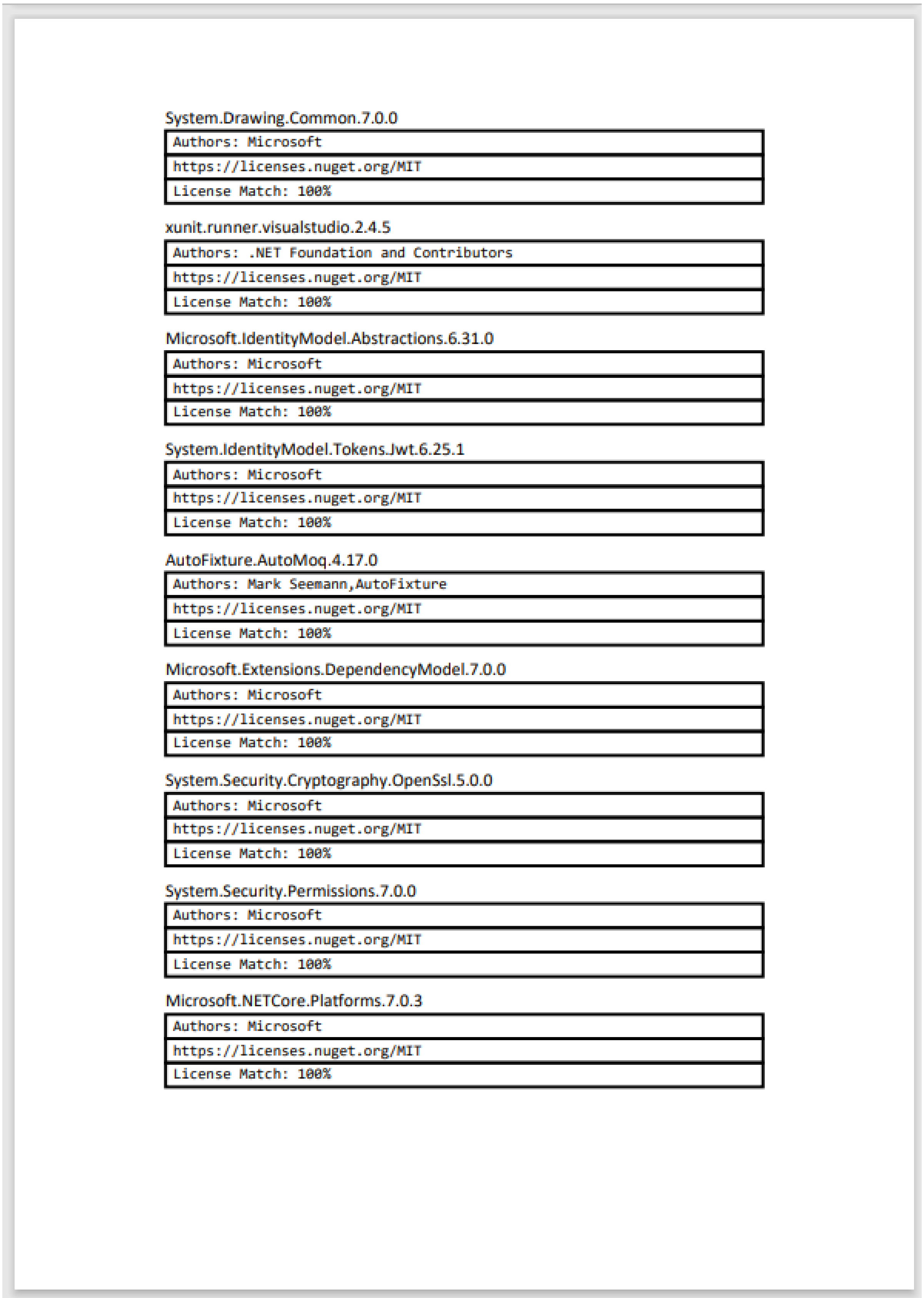
Authors: Microsoft
https://raw.githubusercontent.com/dotnet/corefx/master/LICENSE.TXT
License Match: 96%

Newtonsoft.Json.Bson.1.0.2

Authors: James Newton-King
https://licenses.nuget.org/MIT
License Match: 100%

System.Numerics.Vectors.4.5.0

Authors: Microsoft
https://raw.githubusercontent.com/dotnet/corefx/master/LICENSE.TXT
License Match: 96%



System.Drawing.Common.7.0.0

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

xunit.runner.visualstudio.2.4.5

Authors: .NET Foundation and Contributors
https://licenses.nuget.org/MIT
License Match: 100%

Microsoft.IdentityModel.Abstractions.6.31.0

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.IdentityModel.Tokens.Jwt.6.25.1

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

AutoFixture.AutoMoq.4.17.0

Authors: Mark Seemann,AutoFixture
https://licenses.nuget.org/MIT
License Match: 100%

Microsoft.Extensions.DependencyModel.7.0.0

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Security.Cryptography.OpenSsl.5.0.0

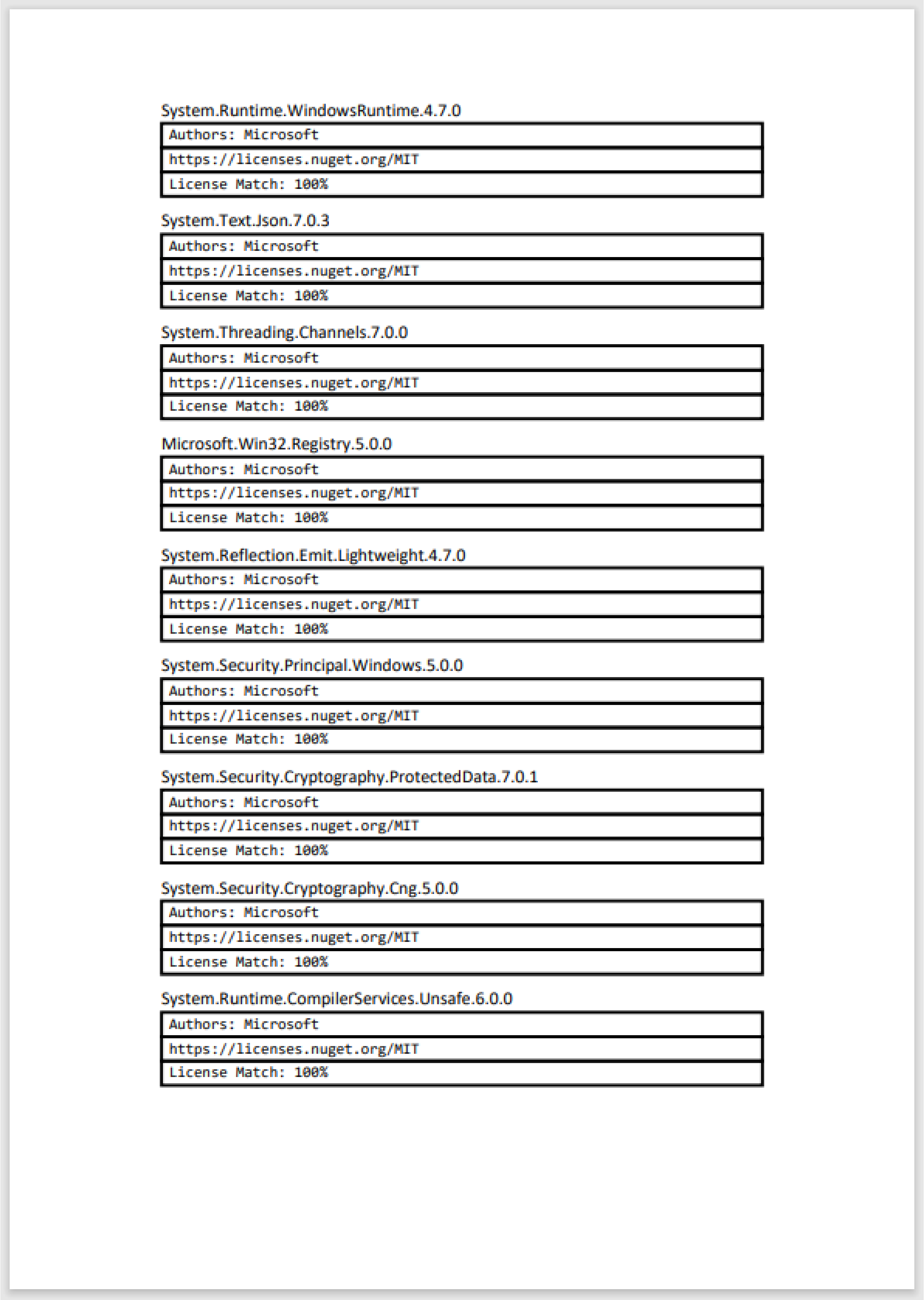
Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Security.Permissions.7.0.0

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

Microsoft.NETCore.Platforms.7.0.3

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%



System.Runtime.WindowsRuntime.4.7.0

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Text.Json.7.0.3

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Threading.Channels.7.0.0

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

Microsoft.Win32.Registry.5.0.0

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Reflection.Emit.Lightweight.4.7.0

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Security.Principal.Windows.5.0.0

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Security.Cryptography.ProtectedData.7.0.1

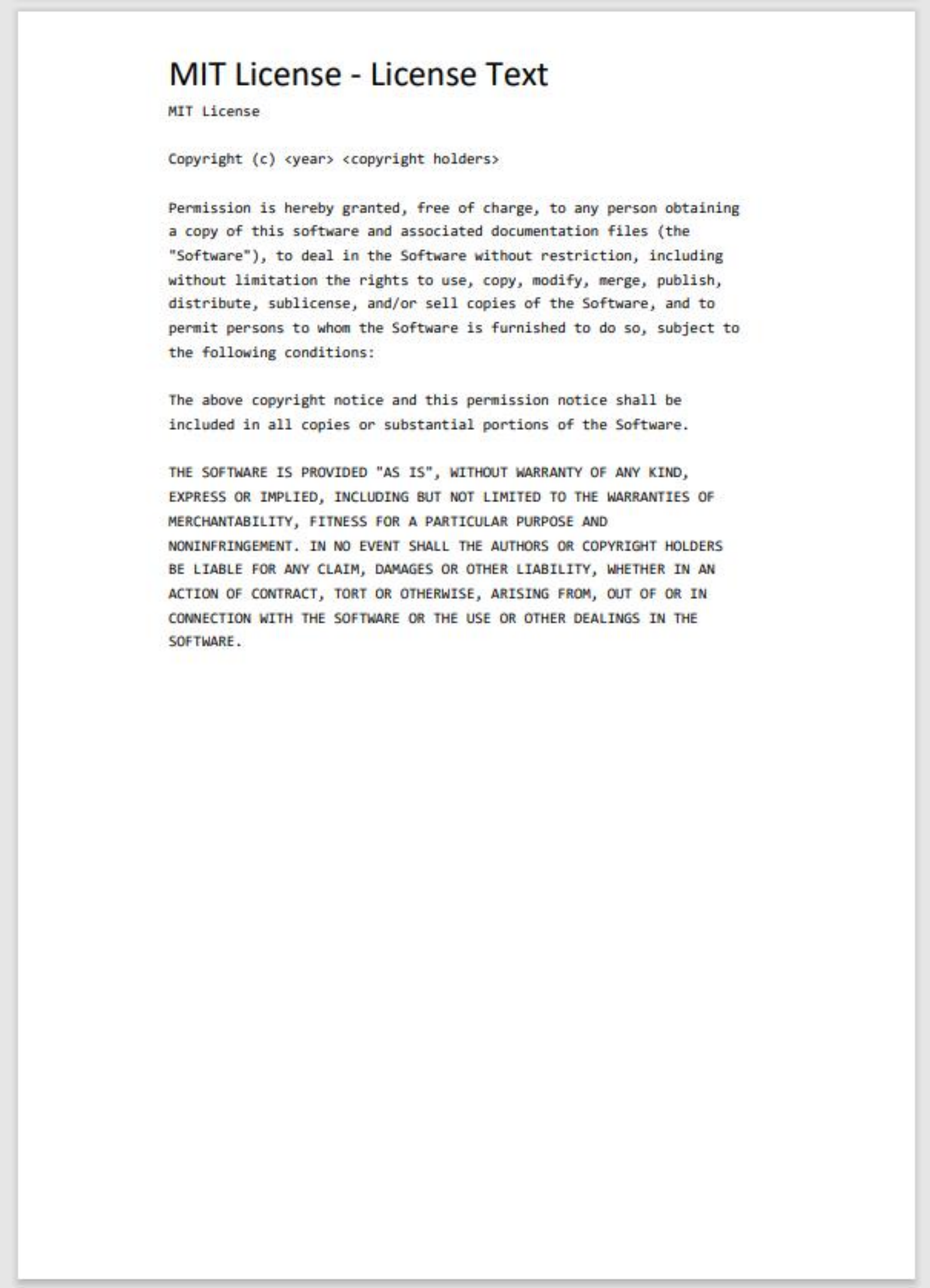
Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Security.Cryptography.Cng.5.0.0

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Runtime.CompilerServices.Unsafe.6.0.0

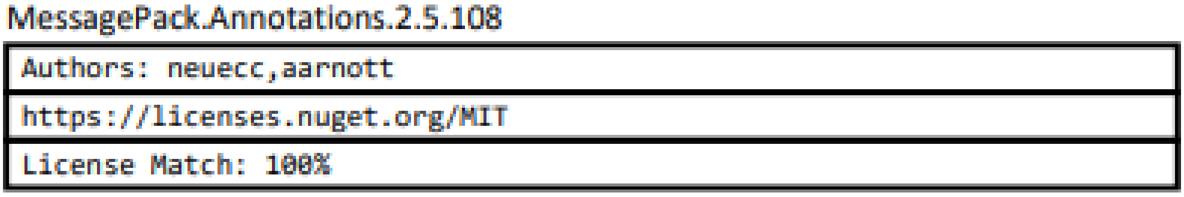
Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%



MIT License - License Text

```
MIT License

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

Example package licence data:



MessagePack.Annotations.2.5.108

```
Authors: neuecc,aarnott
https://licenses.nuget.org/MIT
License Match: 100%
```

2 Project Initiation Document

PROJECT INITIATION DOCUMENT

(Version created to support DAP605)

Project name *Keeper of Licenses for Each Project's Open-Source Software*

Release Final
Date: 27th April 2023

PRINCE2 Based on a reduced version of the PRINCE2 PID documentary requirements

Author:	Charlie Harrison
Owner:	Charlie Harrison
Client:	Applications Team
Document Number:	45807

Document History

Document Location	This document is only valid on the day it was printed. The source of the document will be found in the Control section of the Project File.

Revision date	Summary of Changes	Changes marked
28th March	First revision PID, with constraints, scope, and business case	
27th April	Revision with line manager, re-evaluating and removing need for Test-driven development, Axosoft hosting, and packaging of final project with NuGet	

Purpose

To define the project, to form the basis for its management and the assessment of overall success.

Contents This publication contains the following topics:

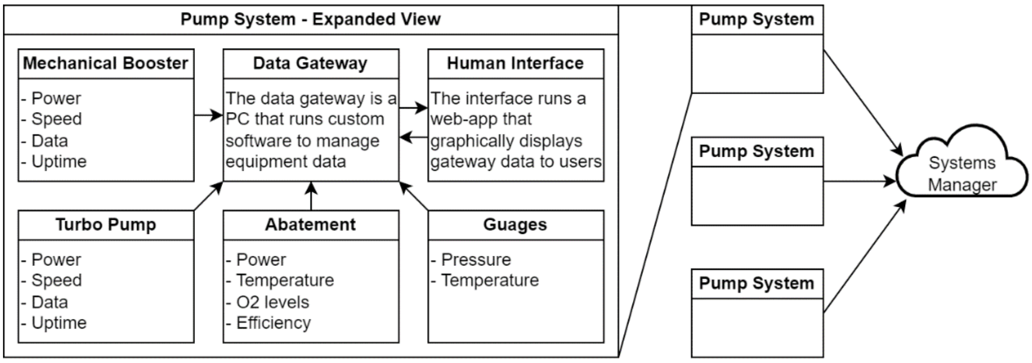
Background

Edwards Vacuum, a producer of industrial and scientific vacuum pumps, belongs to the Atlas Copco Group – employing ~45,000 people. The business' largest customers are semi-conductor manufacturers such as Intel, Samsung, and LG. The systems are purchased in batches of hundreds, deployed in factories known as "*fabs*". Each system is driven and monitored separately by sophisticated controllers, then connected into a networked computer that runs proprietary Edwards software. The company is in development or maintenance of hundreds of pump related computer programs, at any time.

Modern software development involves the practice of implementing third-party libraries that provide common functionality. The libraries are widely available through public repositories and are often free to use in personal & commercial applications (open source), providing the original author is visibly credited. Although, different licenses can require additional clauses.

Identifying and collating all the third-party libraries and their respective licenses manually is an important but enduring task, especially on projects using hundreds of libraries. License information can be stored in several ways within each library, or sometimes entirely separately.

The aim is to build a proof-of-concept system to automate the identification of license types used in Edwards' program's and to produce a document crediting authors in compliance with their license's requirements.



Project Definition

Project objectives

The project should achieve a proof-of-concept program for the elimination of the process of manual licence discovery, and the automation of crediting package authors in compliance with respective licence's requirements.
The project should produce a solution that can assess an Edwards C# project.

Business Case

The current process of manually identifying licenses of third-party packages introduces redundant cost and vulnerability to the development process in the form of developer time used on non-value adding tasks, and the omission of a license in contravention of the license requirements.

Introducing an automated program will mitigate (if not eliminate) human intervention in the discovery and publication of hundreds of license documents and authors per program, expediating the process, and adopting a more dependable, and consistent result.

The potential cost of failing to credit an author could amount to a lawsuit and potentially fines or reputational damage.

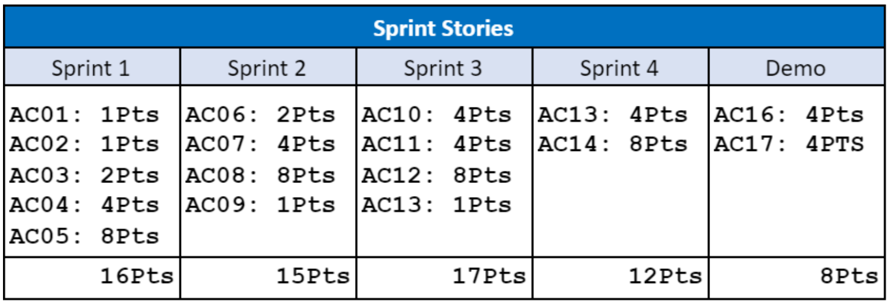
Options Analysis

Option	Positives	Negatives	Conclusion
Do Nothing	No additional risk No developer time spent on non-value adding task	Time consuming for legal team. Workload may increase as open source becomes more popular	This option becomes expensive as legal counsel reviews all the projects.
Manually maintain a record of packages	In-expensive Low development overhead Dependable	Will not update if licenses change. Requires discipline from developers	Due to data half-life, license records become inaccurate.
Off-the-Shelf Solution	Most reliable automation solution Multi-language compatible	Won't integrate into Edwards environment. Could cost £10's of thousands/year for a team of 12.	At $3,100 per developer annually this solution is not cost-effective https://snyk.io/
Develop Custom package detection software.	Only expense is development and management time Can integrate into team city Expandable/ customisable	Adopts development risks Not immediately available Limited subject knowledge & experience.	This solution is inexpensive and better integrates with existing Edwards systems. Furthermore, it's extendable.

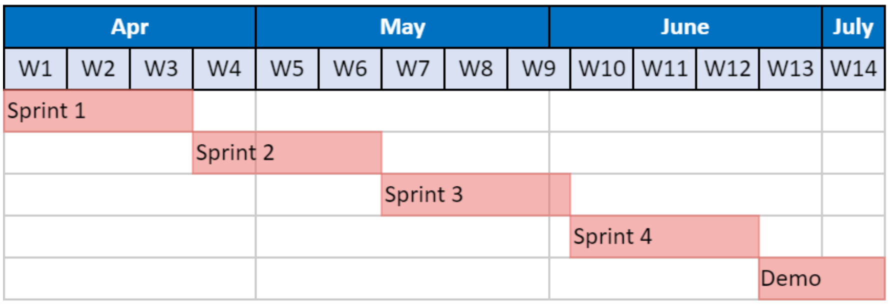
Defined method of approach

The chosen development lifecycle is SCRUM incremental. Based on Wysocki’s PMLC approaches matrix (Robert Wysocki, 2012) – See attachments. Using the matrix’s axis for clarity of goal, requirements & solution, two options are suggested: incremental, and linear (waterfall).

The project deliverables lend themselves to an incremental progression of code e.g., library > console app > graphical interface. These increments will be delivered in sprints that span 3-weeks. Acceptance criteria are targets for completion. For reference 8 points is ~1 weeks work.



Sprint Stories				
Sprint 1	Sprint 2	Sprint 3	Sprint 4	Demo
AC01: 1Pts AC02: 1Pts AC03: 2Pts AC04: 4Pts AC05: 8Pts	AC06: 2Pts AC07: 4Pts AC08: 8Pts AC09: 1Pts	AC10: 4Pts AC11: 4Pts AC12: 8Pts AC13: 1Pts	AC13: 4Pts AC14: 8Pts	AC16: 4Pts AC17: 4PTS
16Pts	15Pts	17Pts	12Pts	8Pts



Apr				May					June				July
W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
Sprint 1	▨	▨											
			Sprint 2	▨	▨								
						Sprint 3	▨	▨					
									Sprint 4	▨	▨		
												Demo	▨

, Test Driven Development (TDD) will be no longer be employed as a measure, as agreed with line manager Brett Lawrence due to requirements not beig sufficienly defineable. TDD is a method of development that requires unit tests to be written in accordance with acceptance criteria prior to application code being written; the benefit is that it prioritises quality over quantity, and serves as a guide for keeping development within scope (Ron Jeffries, 2007)

Project scope

The solution – in its completed state – will be a graphical program that scans a C# project directory for third-party packages, it must be able to unpack them and extract what license type the package uses. The program should be able to identify and retrieve any license texts defined in the package.

Where licence texts aren't declared within a package, the program should attempt to pull the license text from the repository, if not defined there, the program should attempt to identify the license from other reasonably expected sources. Any packages with no detectable licenses should be compiled to a list with relevant information to assist with manually investigating an open-source license.

Once a complete mapping of packages, licenses, and authors are available in memory, the program should produce an intelligible document crediting the authors, that can be included in Edwards programs. A report should also be available to present a confidence score in the license found by the program.

Project deliverables

The final project artefact:

Must Have	Functionality to scan for and build C# solutions
	Functionality to extract third-party packages and identify licences
	Functionality to export collected licence data as JSON and PDF
	A command line interface to run said library
	Documentation, appropriate commenting, and 85% +- 10% unit test coverage
Should Have	Capability for team city integration
	Graphical user interface to operate library functionality
	Manual accommodations for unfound licences
Could Have	Flagging use of copy-left (or forbidden) packages
	Report on licence discovery success
Won't Have	Team city integration
	Analysis of Licence compliance
	Database of historically permitted licences
	Guarantee for licence compliance without oversight from intellectual property counsel

Non-negotiable Documents	Description
Project Brief	A brief, outlining the project.
Project Initiation Document	This document, detailing the foreseeable project variables.
Project Initiation Presentation	A presentation summarising the PID.
System Development Report	A report reflecting on project successes and failures, offering improvements
Demonstration of Artefact Presentation	An in-person delivery of the outcomes of the project's successes & failures.

Exclusions

The project is exclusively limited to facilitating:

A proof of concept that an automated tool can *suggest* what license might be used in a project. Not a replacement for legal counsel.
Edwards projects, where project structures are available entirely locally (no server functionality).
C# projects, where third-party libraries are installed as NuGet packages.
Microsoft licenses and those defined at (https://spdx.org/licenses).
Custom licences should be accommodated for manually still, for their high variability.

Constraints

As outlined by the PRINCE2 framework (Siegelaub, 2007):

Scope	All deliverables must be functional on a windows machine with no additional requirements. The deliverables cannot rely on access to any third-party technologies that require paid-for services. Deliverables cannot make use of services that share source-code for licence detection.
Time	Presentation of project initiation - 28th March System development Project Report – 23rd June Demonstration of Artefact – 3rd July
Cost	Budget: £0 (production to be incorporated to normal hours and personal time)
Quality	Manual testing and unit test plan to be conducted.
Risk	No external libraries used in the development of the tool can be used under a copy-left licence. All external libraries must use supported libraries with public documentation.
Benefits	Must reduce manual time acquiring licenses Must produce design consistent credit document with minimal human interaction

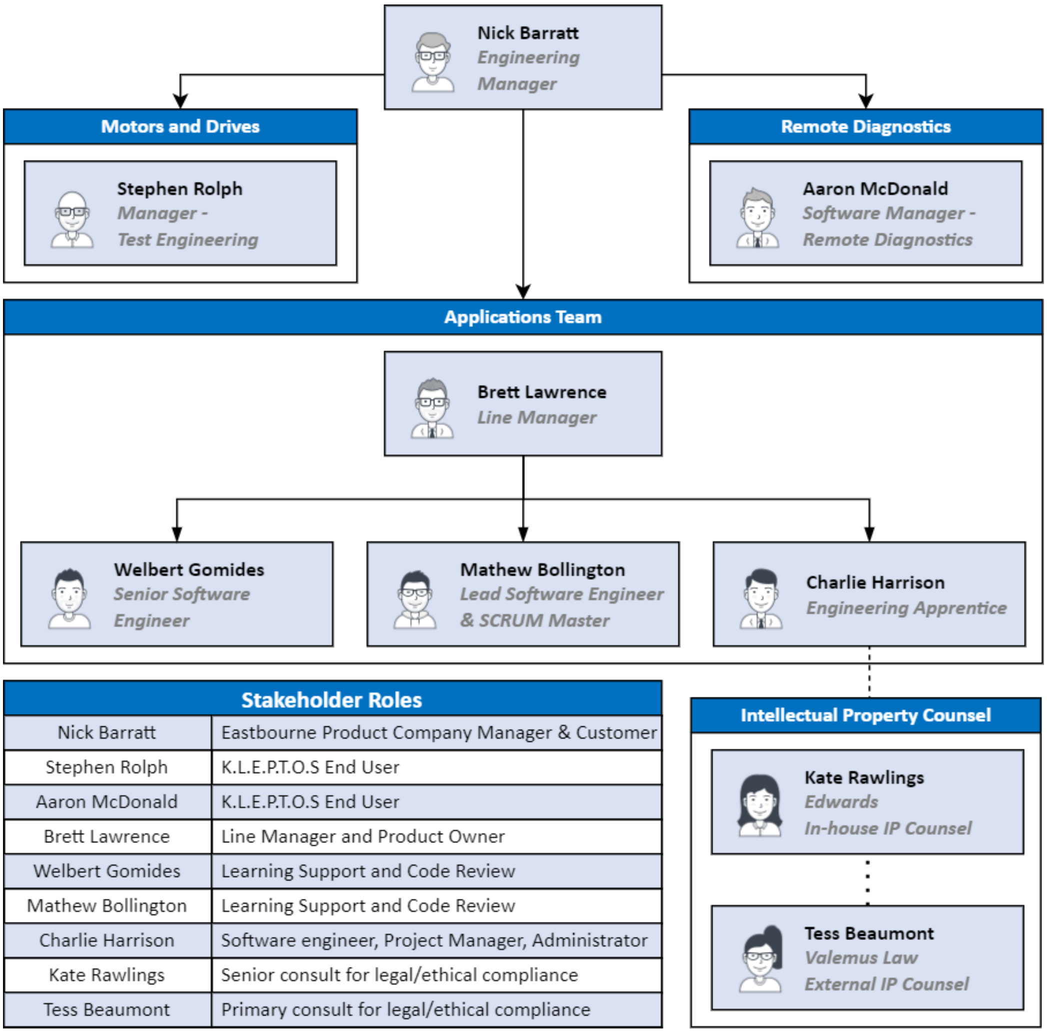
Support for the project is a low priority (due to business operations) from Intellectual property counsel. Meetings for best practice are expected to be brief and infrequent.

Project must be developed in C# using the Visual Studio suite of tools, compliant with Edwards standards, and only the use of free open-source packages is allowed, in addition to original logic.

Artificial Intelligence code generators cannot be used for support, as per company guidelines at the time of writing.

Assumptions	Assumed that the line-manager and learning support will be available within a day's notice at most. It's also assumed that a stakeholder review will occur at the closing of each increment to review and ensure acceptance criteria are met. It's assumed that access to the company's SVN server is always available, and that all C# projects relevant to the K.L.E.P.T.O.S tool are stored on this server It's assumed that contractors and employees of external services are equivalent to full-time permanent employees of teams, as the duration of the project should not exceed their contracts.

Project Organisation Structure



Stakeholder Roles	
Nick Barratt	Eastbourne Product Company Manager & Customer
Stephen Rolph	K.L.E.P.T.O.S End User
Aaron McDonald	K.L.E.P.T.O.S End User
Brett Lawrence	Line Manager and Product Owner
Welbert Gomides	Learning Support and Code Review
Mathew Bollington	Learning Support and Code Review
Charlie Harrison	Software engineer, Project Manager, Administrator
Kate Rawlings	Senior consult for legal/ethical compliance
Tess Beaumont	Primary consult for legal/ethical compliance

The diagram is not comprehensive of the entire organisation, only stakeholders involved in production and a sample of end users. This sample is descriptive – not prescriptive – as end user availability may vary with higher priority projects. Project roles are not PRINCE2 standard.

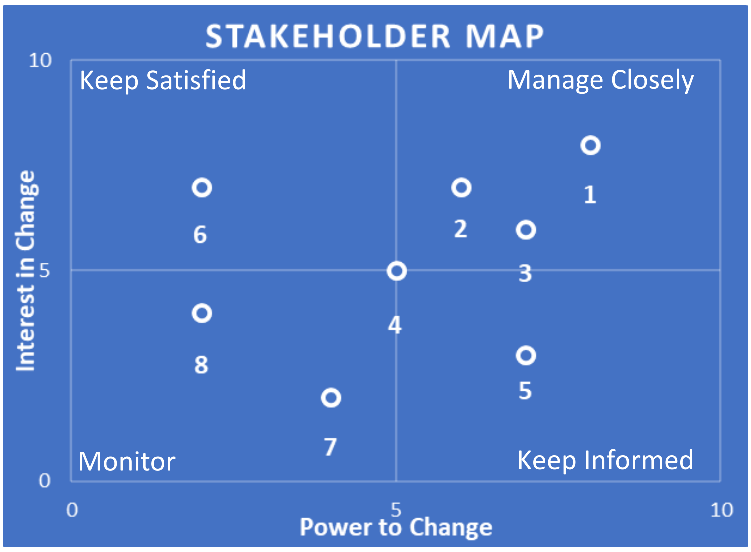
Furthermore, Welbert Gomides has been declared as providing support for learning unfamiliar technologies, although this could be another software engineer, respective to availability or technology.

The arrowed lines represent how instructions are passed.
Dotted lines represent project party relationships

Communication Plan

Stakeholders will be invited to discussions as their knowledge or investment in the project is required. It is imperative that no stakeholder is overlooked upon arranging a meeting. Therefore, this document demands that upon arranging a significant meeting in the development of this project, a table must be produced with a justification either way for the involvement or absence of s stakeholder. (Stakeholder Register.xlsx)

It is assumed all stakeholders are invited to the final demonstration of the artefact.



Priority	Stakeholder	Power	Interest
1	Brett Lawrence	8	8
2	Mathew Bollington	6	7
3	Tess Beaumont	7	6
4	Welbert Gomides	5	5
5	Kate Rawlings	7	3
6	Aaron McDonald	2	7
7	Nick Barratt	4	2
8	Stephen Rolph	2	4

Details	Description
Name: Brett Lawrence	To communicate once or twice per week to monitor progress and best practice. Discussions will predominantly be informal and brief, otherwise recorded.
Email: *brett.lawrence@edwardsvacuum.com*	
Comms: *Weekly*	
Name: Mathew Bollington	Discussions will have note taking encouraged. Progress will be conveyed in sprint-ly increments, with demonstratable increments. Guidance offered, if necessary.
Email: *mathew.bollington@edwardsvacuum.com*	
Comms: *Sprint-ly (3-weeks)*	
Name: Welbert Gomides	With the implementation of unfamiliar technologies, support from experienced colleagues is imperative to project success. Another colleague may be in place of Welbert.
Email: *welbert.pires-gomides@capgemini.com*	
Comms: *Ad-hoc*	
Name: Tess Beaumont	As legal counsel, conversations will be necessary even in the absence of any queries. Without experience, it may not be known what is unknown, until revealed by an expert.
Email: t.beaumont@valemuslaw.com	
Comms: *Bi-Sprint-ly (6-weeks estimated)*	

	I would like to be able to	So that I can	Acceptance Criteria
As a developer	call a code library that scans, builds, and extracts packages in a project directory	have access to all the extracted packages belonging to a project.	AC1: Library trawls directory, and correctly identifies C# solutions AC2: The library can build solutions. AC3: Library can identify third-party licences in a project
	call a library that un-zips, analyses, identifies the type of third-party licence from a package	output a mapping of license types to their respective packages and package authors.	AC4: The library can take any package and unzip it AC5: The library Identifies a licence used by a package in the: metadata, license text, or url AC6: The library can persist licence and package data to a file
	call a library that can interpret license/ package/author mappings and build an intelligible PDF crediting author	add crediting pdf to software to comply with licence demands, and easily review licenses used in a project.	AC7: Library can open, read, and de-serialise extracted mappings into C# objects. AC8: C# objects can dynamically convert into PDF generation instructions AC9: Output PDF conforms to Edwards design specifications
	have access to comprehensive documentation	augment the functionality of the program, such as integration with build servers	AC10: HTML documentation produced by tool such as Doxygen AC11: Code thoroughly commented
As a manager	use an intuitive command-line interface to interact with all the K.L.E.P.T.O.S libraries	automate much of the license extraction, reviewing, and crediting process.	AC12: All library functions: scan, build, extract, and credit are implemented through the command-line AC13: Command-line app is protected against improper inputs
As a user	use a graphical interface that implements the functionality of all the K.L.E.P.T.O.S libraries	more easily distribute the tool so other teams can apply to their projects.	AC14: All library functions: scan, build, extract, and credit are implemented through a graphical interface AC15: User interface implements good user experience design
	follow a user guide for each user interface	understand fully, the capabilities, and limitations of the tools.	AC16: Manual covers all: library, command-line, and graphical use cases. AC17: Comprehensive documentation of object, methods, and data structures

The project source code for each increment will be produced under Test Driven Development. The target unit test requirement is 85% +- 10%, to ensure scope is strictly adhered to and that the software is resistant to functionality regression from future increments.

The delivered source code must:

- Conform to Edwards best practice
- Consistently detect at least 75% of licences in a C# projects, and correctly and consistently flag ALL libraries that require manual intervention
- The project should be able to properly format the output document that credits library authors to consider the high variability of input string lengths such as author and package names
- Output interfaces and documents must conform to Edwards official branding and style guides
- Reliably operate without any other pre-requisite installations of third-party software

These will be manually tested sprint-ly and at the final demonstration of the artefact to Edwards employees.

Project Tolerances

General tolerances:

Scope	The functionality may extend to include scanning packages from source code in languages other than C# 75% licence automatically discovered +-10% PDF credit document may not need to comply with all style guidelines due to any constraints (as it's proof of concept) Graphical Interface is accepted even without optimal user experience design or branding (as it's proof of concept) All C# solutions must be buildable +-0%
Time	0% tolerance in timelines – except in extenuating circumstances due to university requirements
Cost	0% tolerance - no budget available
Quality	+-10% in code coverage 100% of undiscovered licences flagged +-0% Must be stable and function without pre-requisite installs.
Risk	Third-party packages are permitted in the development of K.L.E.P.T.O.S libraries, although must be manually recorded, and not use any copy-left or paid for licencing Intellectual property counsel is optional for refining licence adoption approach, dependant on availability.
Benefits	the time saving is difficult to measure, as existing processes vary wildly based on project complexity, although K..L.E.P.T.O.S should be un-questionably more efficient.

Project Controls

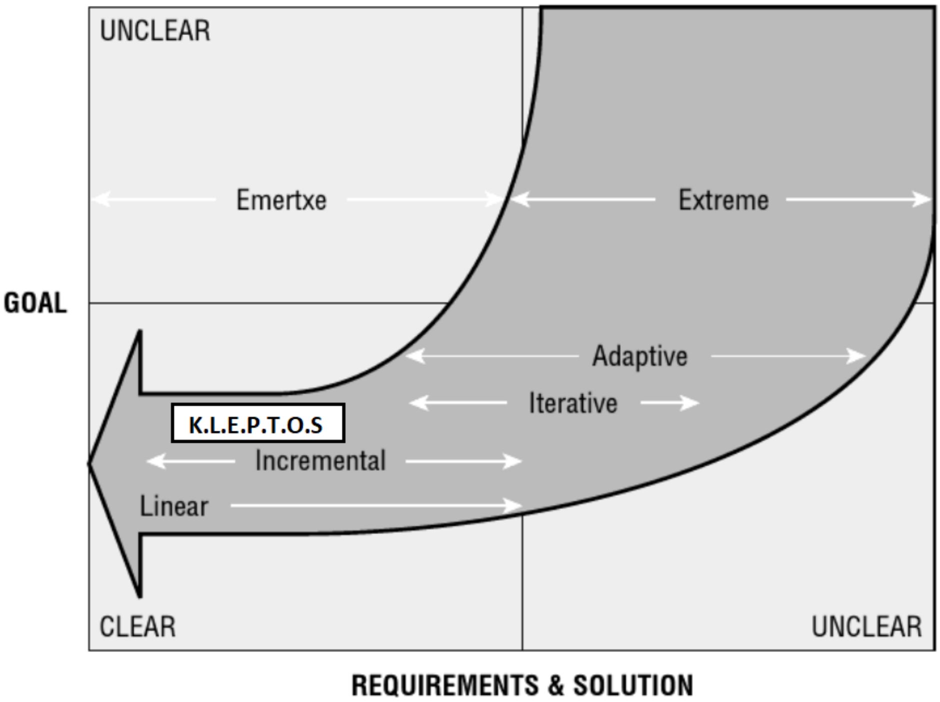
Project Controls	
PRINCE2® https://prince2.wiki/	Project management is controlled by the PRINCE2® methodology. PRINCE2® structures planning, execution, and project delivery. (AXELOS Ltd, 2023)
S.C.R.U.M www.scrum.org	To manage each increment, SCRUM is used to deconstruct deliverables into tasks that must meet acceptance criteria, with task points, where 8 points is ~1 weeks work. (Larman, 2004) Where each sprint is 3 weeks – start 3rd April 23.
Office 365 www.office.com	Office 365 (Word, PowerPoint, Excel, Outlook, teams, and OneDrive) is the de facto environment for Edwards project materials. It offers storage redundancy and remote access to project materials and team collaboration.
TortoiseSVN tortoisesvn.net	TortoiseSVN version control is an already available tool to this project. It enables branching, merging, and reverting changes to safely manage project development.

[Changes 27th April, Removed Axosoft, and Test-Driven Development]

Requests for change are not within the permit of this project, although in any anomalous cases the request for any change can only be approved by the product owner Brett Lawrence. Any requests will be referred through him.

Sprints are 3-week long periods, where code is planned, developed, and reviewed. Work is broken into *"stories"* of work that are fulfilled by meeting acceptance criteria. Each is estimated by a number of points, and allocated, this process is agile to the previous sprint's performance, as such, the provided Gantt chart permits space at the end of the project for overrun tasks. At the end of each sprint the successfully developed increment should be demonstrated and reviewed by the team.

Attachments

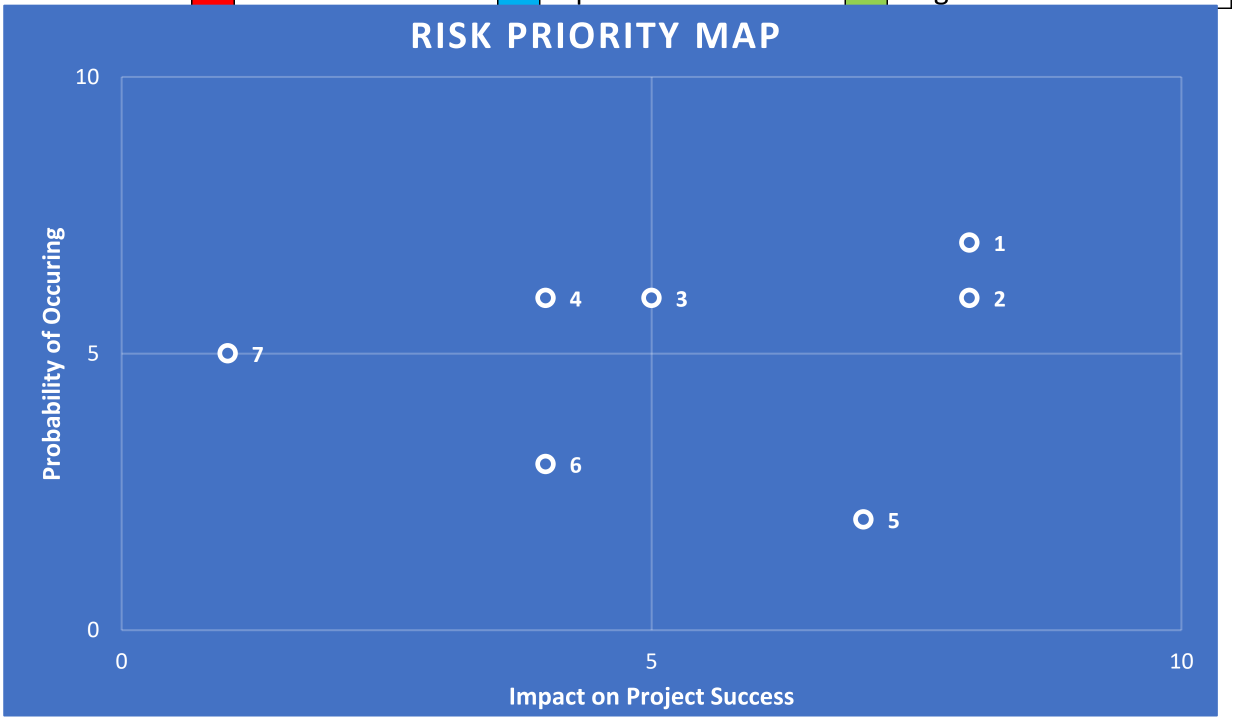


To decide upon a development approach to the project, Wysocki provides a model, with axes that measure the perceived clarity of requirements & solution, and goals. This example shows the K.L.E.P.T.O.S project as being relatively well-understood in both axes.

The limitation of this model is that the values are entirely subjective and variable to the understanding of the project prior to initiation. Despite this is supportive of allowing variables such as confidence and familiarity with a particular SDLC to influence the outcome, which is appropriate for smaller, solo projects.

	Risk	Effect	Impact	Probability	Priority	Mitigation
🟦 1	Demanding day to day operations	work takes priority over the project	8	7	5.6	Mediate with the early careers and line manager to oversee workloads
🟩 2	Development requirements are underestimated	important functionality is left unfinished	8	6	4.8	Use incremental SDLC for functioning increments.` Use MoSCoW to identify most critical functionality
🟩 3	Additional learning required for project.	Delayed and slowed rate of development.	5	6	3	Discuss progress with superiors and negotiate learning support from team members
🟥 4	Unanticipated complexity in discovering licenses	Extended development time.	4	6	2.4	meetings are required to assess license priorities on ad hoc basis as required by the proof of concept
🟩 5	Scope creep for license retrieval methods	increased development workloads	7	2	1.4	Stick to strict incremental SDLC and stories defined in PID.
🟥 6	Low availability of Intellectual Property Counsel	Licence compliance blocks development	4	3	1.2	liaise in advance. In absence of counsel, confer with line-manager for best guess implementation
🟩 7	K.L.E.P.T.O.S is ineffective at returning licenses	K.L.E.P.T.O.S is rejected by stakeholder	1	5	0.5	Marked as a proof-of-concept project, this must be anticipated, and should not be discouraging. Honesty is imperative.

🟥 Schedule Risks	🟦 Operational Risks	🟩 Organisational Risks



Bibliography

B Randell, F. Z. (1968). Iterative Multi-Level Modelling: a methodology for Computer System Design. IEEE CS Press.

Larman, C. (2004). Agile & Iterative Development. Boston MA: Pearson Education. Ltd, A. (2023, March 20). PRINCE2® Project Management Certifications. Retrieved from AXELOS: https://www.axelos.com/certifications/propath/prince2-projectmanagement/

Mendelow. (1991). Stakeholder mapping. Proceedings of the 2nd International Conference on In- formation Systems. Cambridge MA: Zagreb International Review of Economics & Business. Open Source Guide. (2023, March 20).

Open Source Guides. Retrieved from Open Source Guide: https://opensource.guide/

Robert Wysocki, R. S. (2012). Effective project management: traditional, agile, extreme. Indianapolis, indiana: Wiley.

Ron Jeffries, G. M. (2007, May). Guest Editors' Introduction: TDD--The Art of Fearless Programming. IEEE Software.

Siegelaub, J. M. (2007). Six (yes six!) constraints: an enhanced model for project control. PMI® Global Congress. Atlanta, GA: Project Management Institute

3 Project Sponsor Consent Form

University of Chichester	**CONSENT FORM FOR UNIVERSITY OF CHICHESTER RESEARCH PROJECT** **To be used for Interview**
SOFTWARE PACKAGE AND LICENCE DISCOVERY TOOL	

Contact details of researcher

Brett Lawrence brett.lawrence@edwardsvacuum.com

Statement of consent

By signing below, you are indicating that you:

- Have read and understood the information document regarding this research project.
- Have had any questions answered to your satisfaction.
- Understand that if you have any additional questions you can contact the research team.
- Understand that participation is entirely voluntary and that you are free to withdraw without comment or penalty.
- That you are aware of the timescales and that if you wish to exercise your right to request erasure of your personal data following collection and analysis 6th July 2023 this may not be possible having regard to permitted exemptions for research under data protection legislation i.e. where it would seriously impair the achievement of the research objectives and that you have the right to object (as indicated on the Information Sheet)
- Understand that all information will be stored securely and used in line with data protection legislation and no personal information will be shared with third parties.
- Understand that if you have concerns about the ethical conduct of the research project you can contact the Head of Research, Research Office on 01243 816000 or email research@chi.ac.uk.
- Agree to participate in the research project.

Please tick the relevant box below:

☒ I **agree** for the interview to be video recorded.

□ I **do not agree** for the interview to be video recorded.

Name Brett Lawrence

Signature

Date 24/4/2023

PLEASE RETURN THE SIGNED CONSENT FORM TO THE RESEARCHER.

4 Supplementary Research

4.1 Attitudes to Open-Source Software

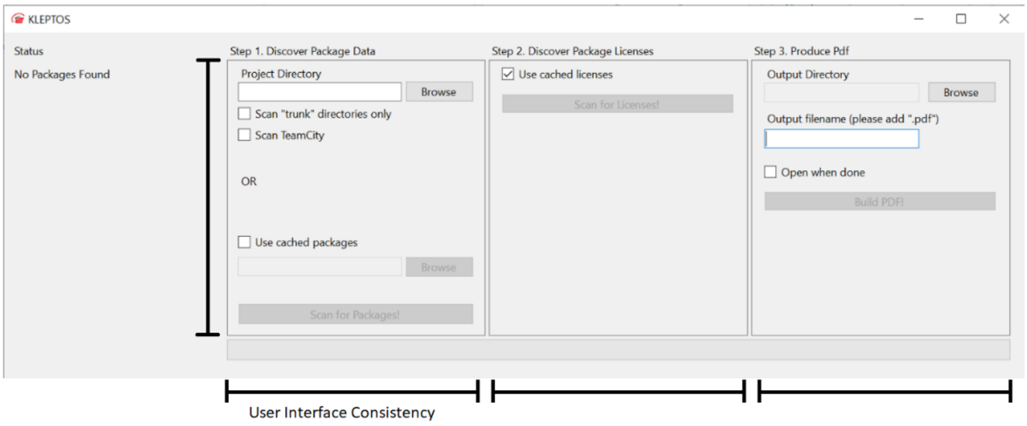
Open-source NuGet packages are a recent development, having only existed in some form since 2010, although more niche open-source projects have been in existence longer (Haack, 2010). Famously, Microsoft's founder spoke against sharing software for free, with the earliest example being Bill Gates' *"Open Letter to Hobbyists"* in 1976 stating:

> *"What hobbyist can put 3-man years into programming, finding all bugs, documenting his product and distribute it for free?"*
>
> (Gates, 1976)

Somewhat Ironically, the open-source NuGet repository is now one of the greatest selling points of the Microsoft development environment, where hobbyists and organisations freely share functional packages of code that mitigate the redundant development of pre-existing solutions to a problem.

With current software projects at Edwards approaching their seventh year of development, the concept of having thousands of open-source licences to manage in a project is novel and requires a scalable innovative solution.

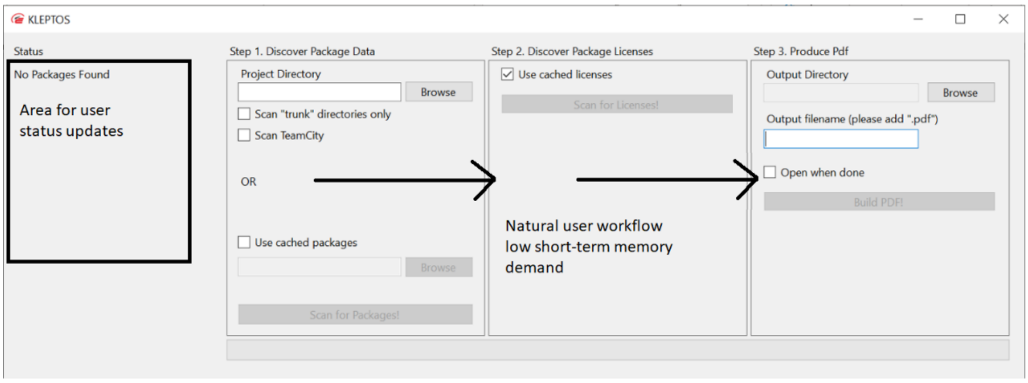
5 User Interface Research and Design



No Packages Found

□ Scan "trunk" directories only
□ Scan TeamCity
□ Use cached packages
☑ Use cached licenses
□ Open when done

User Interface Consistency

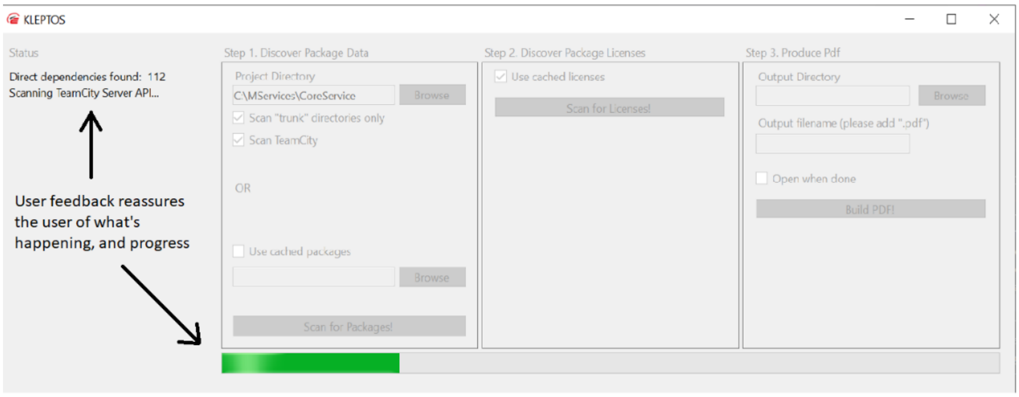


No Packages Found

□ Scan "trunk" directories only
□ Scan TeamCity
□ Use cached packages
☑ Use cached licenses
□ Open when done

Area for user status updates

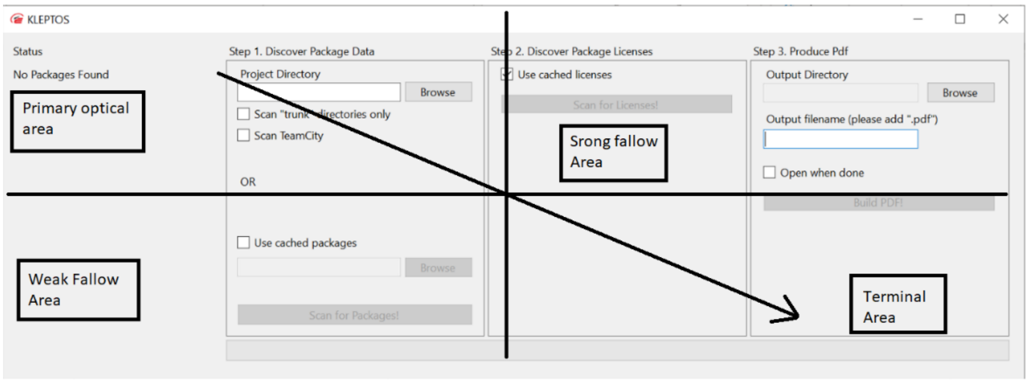
Natural user workflow low short-term memory demand



Direct dependencies found: 112
Scanning TeamCity Server API...

C:\MServices\CoreService
☑ Scan "trunk" directories only
☑ Scan TeamCity
□ Use cached packages
☑ Use cached licenses
□ Open when done

User feedback reassures the user of what's happening, and progress



No Packages Found

□ Scan "trunk" directories only
□ Scan TeamCity
□ Use cached packages
☑ Use cached licenses
□ Open when done

Primary optical area

Srong fallow Area

Weak Fallow Area

Terminal Area

As prescribed by Gutenberg's areas of optical strength, the primary optical area is of the greatest natural attraction to the eye, hence the critical status informations location. As the user progresses through the steps of the program's menus they will naturally end at the terminal area, which is an inuitive and unstraining location to terminate functionality.

6 Application for Ethical Approval

Application for Ethical Approval: For all applications for ethical approval (staff/PGR/Masters/UG)

This form should be used by ALL members of the University including undergraduate students, postgraduate research and postgraduate taught students, staff and those in visiting or emeritus roles who wish to undertake research involving human participants under the name of the University of Chichester. You do not need to complete this form if your research does not involve human participants directly or indirectly (e.g. observation studies) (see section 4.1 of the Research Ethics Policy (REP) for more information). However, you are expected to work within the Research Ethics Policy and Researcher Code of Conduct. The University does not conduct research on animals. If your proposed project involves animals in any way please seek advice from the Research Office before proceeding. Researchers wishing to use tissue cultures in their research should contact the Research Office in the first instance. Researchers should consider the provenance of tissue samples/cultures/cell-lines and associated growth media (or similar) and whether immortalised and/or animal-free alternatives are available.

THIS FORM MUST BE COMPLETED AND APPROVED by the relevant person(s) and if categorised as Category B it must be approved by the Research Ethics Committee (REC) prior to commencement of research. Full guidance on the Application process can be found in the body and appendices of the Research Ethics Policy.

REQUIRED DOCUMENTATION Each Application must be submitted alongside relevant consent forms, information letters/sheets, and debriefing sheets. This documentation should be version numbered and dated.

Categorisation of applications for ethical approval

Category A projects are less likely to involve participants from vulnerable groups (e.g. children, or persons with disabilities) and/or involve sensitive issues or areas/activities that entail a level of risk of distress or harm to participants or researchers. They only need to be approved by your supervisor and do not need to be considered by the Research Ethics Committee. The Research Ethics Policy provides further guidance on categorisation and areas of risk.

Category A+ for specific cases of withholding information / intentional deceit as occurs in single blind or double blind trials (as described above), where the only reason for identifying the project as a Category B is the withholding of information / intentional deceit. If there is any other aspect of the study that would lead to a Category B categorisation (e.g. the study involves a vulnerable group such as children, people with a disability, or those with a mental health problem, who are not persons with whom the applicant normally works: see clause 10.1.5 of Research Ethics Policy) then the exception does not apply and the application for ethical approval is classified as Category B and treated accordingly. The application would be approved by the line manager/supervisor (as with Category A applications) and also by an independent scrutiniser drawn from a pool of experienced researchers within the Institute/Department approved by its Head/Director. They do not need to be considered by the Research Ethics Committee. This would apply to category A+ applications from undergraduate students as well as staff and postgraduates.

Category B projects need to be considered by the Research Ethics Committee. The process of approval can take several weeks or longer depending on the number of applications being considered at any one time and the resolution of any issues that are raised by the Committee. It is fairly common for applications to be returned for further amendments prior to approval. The Committee expects applications from students to be of the same quality as those from staff. A helpful way to consider this position is to consider the research project from the point of view of the research participant.

Undergraduate or taught postgraduate student applicants: Your tutors and programme team will be able to advise you on how and when to complete this form. Your project supervisor is responsible for categorising your application as Category A, A+ or Category B and for authorising it.
Communications relating to Category B applications should be between the supervisor and the clerk to the Research Ethics Committee. The student should not contact the clerk directly.

The completed form will be kept for a period of five years after approval.

Postgraduate research students: Your PhD supervisor is responsible for categorising your application as Category A, A+ or Category B and for authorising it.
Academic Staff: Your line manager is responsible for categorising your application as Category A, A+ or Category B and for authorising it.
Emeritus or Visiting roles: The Head of Department / Director of Institute of the area to which you are linked is responsible for categorising your application as Category A, A+ or Category B and for authorising it.

[*this is a detachable front sheet, the form begins on the next page*]

Section A: Basic Information

A1: Title of study:	**Open-Source License Reporting Tool**
A2: Name of Applicant: (in collaborative projects, just name the lead applicant)	Charlie Thomas Harrison
A3: Position of Applicant (e.g. UG/Masters/PGR student, academic) If you hold multiple roles within the University, please write in the role which is pertinent to this specific study.	UG student
A4: Programme of study: (for UG or taught Masters students only)	Bachelor of Science (Hons)
A5: Department of Applicant:	Business School

A6: Checklist to ensure application is complete. Have you prepared the following documents to accompany your application for ethical approval, please tick the appropriate column for each of the following:

Documents / Addenda	**Yes**	**No**	**N/A**
Confirmation of Ethical Approval of any other organisation (e.g. NHS, MoD, National Offender Management Service)			X
Recruitment information / advertisement (e.g. draft text for email/ poster/social media/letter)			X
Information sheet for participants			X
Information sheet for carers/guardians			X
Information sheet/letter for gatekeepers e.g. Head teacher, teacher, coach			X
Consent form for participants			X
Assent form for younger children			X
Documentation relating to the permission of third parties other than the participant, guardian, carer or gatekeeper (e.g. external body whose permission is required)			X
Medical questionnaire / Health screening questionnaire			X
Secondary information sheet for projects involving intentional deceit/withholding information			X
Secondary consent form for projects involving intentional deceit/withholding information			X
Debrief sheet to give to participants after they have participated			X
Statements about completeness of the application	**Yes**	**No**	**N/A**
For research involving under 18s or vulnerable groups, where necessary, a statement has been included on all information sheets that the investigators have passed appropriate ***Disclosure and Barring Service***[1] checks			
I can confirm that the relevant documents listed above make use of document references including date and version number			

Declaration of the applicant:

I confirm my responsibility to deliver the research project in accordance with the University of Chichester's policies and procedures, which include the University's '*Financial Regulations*', '*Research Ethics Policy', 'Electronic Information Security Policy'* and *'Privacy Standard'* and, where externally funded, with the terms and conditions of the research funder.

In signing this research ethics application form I am also confirming that:

- The research study must not begin until ethical approval has been granted.

[1] *Working with under 18's or other vulnerable groups may require a Disclosure and Barring Service Check. Contact HR@chi.ac.uk if you are not sure whether you have an up to date and relevant DBS check or if you require more information. Do note that a DBS check may take several weeks to obtain.*

- The form is accurate to the best of my knowledge and belief.
- There is no potential material interest that may, or may appear to, impair the independence and objectivity of researchers conducting this project.
- Subject to the research being approved, I undertake to adhere to the project protocol without deviation (unless by specific and prior agreement) and to comply with any conditions set out in the letter from the University ethics reviewers notifying me of this.
- I undertake to inform the ethics reviewers of significant changes to the protocol (by contacting the clerk to the Research Ethics Committee (research@chi.ac.uk) in the first instance).
- I understand that the project, including research records and data, may be subject to inspection for audit purposes, if required in future, in keeping with the University's Privacy Standard.
- I understand that all processing of personal data in relation to the proposed project must comply with data protection legislation.
- I understand that personal data about me as a researcher in this form will be held by those involved in the ethics review procedure (e.g. the Research Ethics Committee and its officers and/or ethics reviewers) for five years after the research has ended, after which time the data will be securely destroyed/deleted.
- I understand that all conditions apply to any co-applicants and researchers involved in the study, and that it is my responsibility to ensure that they abide by them.
- For the Student Investigator: I understand my responsibilities to work within a set of safety, ethical and other guidelines as agreed in advance with my supervisor and understand that I must comply with the University's regulations and any other applicable code of ethics at all times.

Applicant's signature:

Date:

Section B: Authoriser assessment and approval

Where Applicants are students (undergraduate or postgraduate) supervisors should authorise this form; where applicants are staff members their line manager (or nominated signatory) should authorise this form.

B1: Name of Authoriser:	Paul Kooner-Evans
B2: Position of Authoriser: (e.g. supervisor, line manager)	Supervisor

AUTHORISER:
Please categorise the application (A, A+ or B) ensure that the application form and all of the required documentation are complete before signing this application.
Authoriser assessment: (tick as appropriate – *see Section 10 of the Research Ethics Policy*)

Category A: Proceed with the research project. ***Undergraduate and Postgraduate Taught Masters applications***: Form and documentation retained at Department level. ***Research Masters, PhD and staff applications***: Form and documentation forwarded to the Research Office *research@chi.ac.uk*	
Category A+: for studies where information is withheld/there is an element of deceit or similar (see Appendix 13) Proceed with the research project. ***Undergraduate and Postgraduate Taught Masters applications***: Form and documentation retained at Department level. ***Research Masters, PhD and staff applications***: Form and documentation forwarded to the Research Office *research@chi.ac.uk*	
Category B: Submit to the Ethical Approval Sub-group for consideration. *research@chi.ac.uk* Proceed only when approval granted by the Chair of the Research Ethics Committee	

Authoriser, please provide a comment on your assessment of the research project and for those projects involving vulnerable groups that you are authorising as Category A please justify this classification in the box below. As a further point, do make appropriate reference to any other codes of practice in your discipline particularly if you think that the proposed research may be in tension with those codes.
For Category A+: the application would be approved by the line manager/supervisor (as with Category A applications) and also by an independent scrutiniser drawn from a pool of experienced researchers within the Institute/Department approved by its Head/Director

Comment:

Authoriser's declaration:

- I have read the Research Ethics Policy and this has informed my judgement as to the category of assessment of this application.
- I understand that the applicant has taken account of the Research Ethics Policy and other relevant University policies in preparing this application.
- For Supervisors: I understand my responsibilities as supervisor, and will ensure, to the best of my abilities, that the student investigator abides by the University's Research Ethics Policy at all times.

Authoriser, please complete this table making it clear which version of the application form you are approving:

Version of the form (e.g. original version/ amended version following REC sub-group comments)	**Signature of authoriser**	**Date**

For Category A+ independent scrutiniser must also sign as authoriser.

For RO use: IF CATEGORY B: Signature of the Chair of the Research Ethics Committee.

Signature:

Date:

Please note that the Research Office will retain all applications for ethical approval for 5 years after the research project has ended as stated in the University's Privacy Standard. It is the researcher's responsibility to let the Research Office know when the project has ended.

.

SECTION C: Ethical Review Questions

C1. Does the study involve human participants?

Yes

Participants in research are taken to include all those involved in the research activity either directly or indirectly and either passively, such as when being observed part of an educational context, or actively, such as when taking part in an interview procedure.

NB: the University does not conduct research on animals. If your proposed project involves animals in any way (including animal tissue) please seek advice from the Research Office before proceeding.

C2a. Might the research entail a higher than normal risk of damage to the reputation of the University, since it will be undertaken under its auspices? *(e.g. research with a country with questionable human rights, research with a tobacco company. See section 9.3 of the REP). If a research partnership has been established with an industry partner please ensure that the University is not linked to claims made by that company regarding benefits of their products unless substantiated evidence of beneficial effects is available.*

Answer: No

C2b. If your answer to 2a was yes, please describe the potential risk to the University's reputation and how this risk will be mitigated. If no, please jump to C2c.

N/a

C2c. Does the research concern groups or materials that might be construed as extremist, security sensitive or terrorist?

Answer: No

If 'Yes' please describe how you will manage the research so that it is not in breach of the Terrorism Act (2006) which outlaws the dissemination of records, statements and other documents that can be interpreted as promoting or endorsing terrorist acts. For example, relevant documents, records, information and data pertaining to the research can be stored on a secure University server. The research should also not be in breach of the Counter-Terrorism and Sentencing Act (2021) and the Revised Prevent Duty Guidance (2015). Contact the Chair of the Research Ethics Committee in the first instance if you are unsure as to how to proceed.

*If you answered **Yes** to question C2c then please complete the additional pro-forma available from the Research Ethics Moodle: **Approval to undertake research concerning groups or materials that might be construed as extremist, security sensitive or terrorist**. Please append the completed form to this application.*

C2d. Does your research fit into any of the following security-sensitive categories? If so, please indicate which:

i.	Commissioned by the military:	No
ii.	Commissioned under an EU security call:	No
iii.	Involve the acquisition of security clearances:	No

If you answered yes to any of the above please provide further information

N/a

C3. Why should this research study be undertaken?

Brief description of purpose of study/rationale (up to 500 words)

To obtain requirements and a business-case to produce a tool that ensures software produced by the company (Edwards Vacuum Ltd) – that employs third-party open-source libraries – does so while displaying appropriate credit to permit the right to reproduce the libraries in the company's products, while mitigating necessary human involvement.

C4a. What are you planning to do? (up to 500 words)

Provide a description of the methodology for the proposed research, including proposed method and duration (start and end date) of data collection, recruitment information (including exclusion/inclusion criteria, recruitment methods etc.), tasks assigned to participants of the research and the proposed method and duration of data analysis. Please include information about location, including details of any special facilities to be used and any factors relating to the study site/location that might give rise to additional risk of harm or distress to participants or members of the research team together with measures taken to minimise and manage such risks

If the proposed research makes use of pre-established and generally accepted techniques, e.g. established laboratory protocols, validated questionnaires, please refer to this in your answer to this question. If it is helpful for the panel to receive further documentation describing the methodology then please append this to your application and make specific reference to it in box 3a below. For Category B applications please include the data collection sheet as an appendix.

Consult the company’s Intellectual Property department, line managers, and software engineers on the requirements and business case of the program, in addition to current best practice in manually obtaining licences and crediting their authors.

This may consist of email, VOIP call, or face-to-face discussions.

C4b. Is this research externally funded?

No

If, the answer yes, please name the research funder(s) here:

N/a

C5a. Who are you recruiting and how?

Please answer the questions in the table below. If you are using posters/flyers, you may not know the exact number of people you will be contacting for recruitment purposes. If this is the case, please indicate this in the first two questions.

How many people will you contact for recruitment purposes?	*~5*
How will you contact them?	*Company Email*
How many participants are you hoping to recruit in total?	*~5*
What will they be asked to do? (e.g. x1 hour long interview, answer a questionnaire, etc.)	*Mostly interviews of approximately 1 hour to gather requirements. Additionally, gathering resources and documentation.*

C5b. Who are the participants?

Please indicate the number of participants in each of the groups in the table below. If the precise number of participants is not known then please make an estimate. Please enter '0' in the 'Numbers in study' column for those groups that are not included in your study. Please note that the examples provided of different sorts of vulnerability are not an exhaustive list.

Participant	Numbers in study
Adults with no health or social problems known to the researcher, i.e. not in a vulnerable group:	**~5**
Children aged 16-17 with no known health or social problems:	**0**
Children under 16 years of age with no known health or social problems:	**0**
Adults who would be considered as vulnerable e.g. those in care, with learning difficulties, a disability, homeless, English as a second language, service users of mental health services, with reduced mental capacity Identify reason for being classed as vulnerable group and indicate 'numbers in study' in next column adjacent to each reason (expand the form as necessary): **……………………………………………..** **……………………………………………..**	**0**
Children (aged <18) who would be considered as particularly vulnerable e.g. those in care, with learning difficulties, disability, English as a second language Identify reason for being classed as vulnerable group and indicate 'numbers in study' in next column adjacent to each reason (expand the form as necessary): …………………………………………….. ……………………………………………..	**0**
Other participants not covered by the categories listed above (please list): ***List other categories here:*** ……………………………………………..	**0**

C6a. Is there something about the context and/or setting which means that the potential risk of harm/distress to participants or research is lower than might be expected normally (see examples below)?

No

Consider if the study is part of routine activity which involves persons with whom you normally work in a typical work context e.g. Teachers working with children in a classroom setting, researchers in the performing arts working with performers, sports coaches working with athletes/players or research involving students in an academic setting.

If yes, please elaborate here:

N/a

C6b. Is the process of the study and/or its results likely to produce distress, anxiety or harm in the participants even if this would be what they would normally experience in your work with them?
See section 5 of the REP.

Answer: No

If you answered Yes to 6b, please answer 6c below:

C6c. Is the process of the study and/or its results likely to produce distress or anxiety in the participants *beyond* what they would normally experience in your work with them?

Answer: No

If yes this Application must be categorised as 'B'

Please provide details:

N/a

C6d. What steps will you take to deal with any distress or anxiety produced?

E.g. have a relevant professional on-hand to support distressed/anxious participants. Careful signposting to counselling or other relevant professional services. Other follow-up support.

N/a

C6e. What is the potential for benefit to research participants, if any?
E.g. Participants may gain an increased awareness of some issue or some aspect of themselves.

The participants stand to gain an improvement in their third-party license approval process through full automation and detection of open-source license types. This can be a costly manual process, otherwise. Those being interviewed for requirements, processes, and legal dependencies are typically responsible for carrying out this process.

C7. Are there any conflicts of interests which need to be considered and addressed?
(For example, does the research involve students whom you teach, colleagues, fellow students, family members? Do the funders, researchers, participants or others involved in the research have any vested interest in achieving a particular outcome? *See section 9 of the Research Ethics Policy (REP)*)

Answer: No

If conflicts of interest are envisaged, indicate how they have been addressed:

N/a

C8. Will any payment, gifts, rewards or inducements be offered to participants to take part in the study? *See section 11 of the REP.*

Answer: No

Please provide brief details and a justification:

N/a

C9a. Will the study involve withholding information or misleading participants as part of its methodology? *(Please refer to sections 6.2 and 10 of the REP for further guidance)*

Answer: No

Please provide details if this has not already been explained in section 3a:

N/a

C9b. Do you envisage that withholding information or misleading participants in this way will lead to any anxiety, distress or harm?

Answer: No

Please justify your answer to 9b:

N/a

It is the University Research Ethics Policy that all projects with the exception of double-blind placebo trials (or similar) will be categorise as Category B. Double-blind placebo trials (or similar) may be categorised as Category A+.

C10a. Does your proposal raise other ethical issues apart from the potential for distress, anxiety, or harm?

Answer: Yes

C10b. If your answer to C10a. was ‘yes’, please briefly describe those ethical issues and how you intend to mitigate them and/or manage them in the proposed study, otherwise jump to C10c.

The chance of licenses of third-party libraries going un-detected poses the ethical issue of illegally using the work without the proper and relevant credit. For each license used by a program, the program will have to provide a score of its certainty that not only what it found was a license, but also how certain it was that the license is the correct SPDX code.

C10c Does your proposed study give rise to any potential risk of harm or distress to yourself or other members of the research team? OR is there any risk that you could find yourself in a vulnerable position as you carry out your study.

Answer: No

If you answer ‘yes’ to either of these points please explain briefly what the risks are and what steps you are taking in order to minimise and manage those risks.

For example does your study involve you in 1-1 interviews in a private setting that might suggest precautions need to be taken relating to lone-working (See section 9 of the REP), Have you considered the likelihood of a participant(s) disclosing sensitive information to you about illegal or harmful behaviour and what actions you would take in such circumstances?

N/a

C11a. Will informed consent of the participants be obtained and if so, how?

Answer: Yes

See section 6 of the REP to help you answer this question. Section 6.3.1 covers research that involves observing behaviour in a public place where gaining informed consent may not be practical or feasible.

When and how will informed consent be obtained? Will it be written or oral consent bearing mind that oral consent will not be considered adequate other than in exceptional circumstances and must be appropriately justified in your application?

NB: Ethical approval should, as a principle, be sought before research participants are approached.

Using the informed consent standard document to be read and signed prior to interviews.

C11b. Is there anyone whose permission should be sought in order to conduct your study? E.g. Head teacher of a school, parents/guardians of child participants.

Answer: No

When and how will informed consent be obtained and from whom? Will it be written or oral consent bearing mind that oral consent will not be considered adequate other than in exceptional circumstances and must be appropriately justified in your application? If you are seeking to gain 'loco parentis' consent from a school rather than seeking individual parental consent please describe your reasoning.

N/a

C11c. Do you need to seek the permission of any other organisations, individuals or groups other than outlined in 11b? E.g. the Research Ethics Committee of partner or participating organisations. Organisations like the NHS and the Prison Service have specific systems for granting ethical approval for research.

Answer: No

Please note that all applications must go through the University of Chichester Application for Ethical Approval process and that they must meet the Research Ethics Policy (REP) requirements. Other prior approval will be taken into account but will not in itself be sufficient to gain University Research Ethics Approval. Each application must normally be accompanied by evidence (e.g. formal statement from the appropriate Ethics Committee) confirming approval by the external body (and any concerns/issues identified). In cases where an external body requires prior approval from the University Research Ethics Policy (such as some NHS work) the Research Ethics Committee (REC) may grant in principle approval pending written confirmation of ethical approval by the external body.

Please describe the permission that is required and how you will be seeking that permission: Please attach any relevant documentation e.g. letter, that relates to the seeking of the relevant permissions.

N/a

C12a. It is normally required that a participant's data is treated confidentiality and stored securely at the outset of, during and after the research study. Will this be the case?
How long will data be stored before being destroyed?

Answer: No

If the answer is 'yes' please describe how you will be maintaining the confidentiality of participants' data. If the answer is 'no' please justify the exceptional circumstances that mean that confidentiality will not be guaranteed. See section 7 of the REP.

Please make reference to measures you are taking to ensure security of data from the point of data collection, transfer from notebooks/voice recorders etc., onto secure devices, to the point of analysis, sharing and final storage. If you are planning to store sensitive data on portable devices or media, you should only store such data if there is an immediate need and should remove these data when this immediate need no longer exists. All sensitive data stored on portable devices or media must be strongly encrypted greatly reducing the risk of the data falling into the wrong hands if the device or media is stolen.

Research projects should be undertaken in accordance with the University's Electronic Information Security Policy and Privacy Standard. Staff should also refer to the Data Protection Guidance for Staff (Section 9 on Research). Completed consent forms should be stored securely and the agreed retention period for these is 5 years, after which they should be securely destroyed/deleted.

Please provide details:

N/a

C12b. It is normally required that the anonymity of participants is maintained and/or that an individual's responses are not linked with their identity. Will this be the case?

Answer: No

If the answer is 'yes' please describe how you will be maintaining the anonymity of participants. If the answer is 'no' please justify the circumstances that mean that anonymity will not be guaranteed. See section 7 of the REP. NB: in group studies it is likely that each individual in the group will be aware that others in the group are participating in the study – they are therefore not anonymous to each other. However, their identity should not normally be associated with their individual responses. In some studies individual participants may not want their identify known to other participants and the study must be designed and undertaken accordingly.

Please provide details:

N/a

C13. Will participants have a right to comment or veto material you produce about them?

Answer: Yes

Please give details and if your answer is 'no' then please provide a justification.

Company material shared in interviews has the right to be vetoed by the individual themselves, or line managers in contact with the project.

C14. Does the project involve the use of or generation/creation of audio, audio visual or electronic material or recordings directly relating to the participants?

Answer: Yes

If yes, please describe how the collection and storage of this will be managed bearing in mind data protection, confidentiality and anonymity issues (see section 7 of the REP). If you are planning to store sensitive data on portable devices or media, you should only store such data if there is an immediate need and should remove these data when this immediate need no longer exists. All sensitive data stored on portable devices or media must be strongly encrypted greatly reducing the risk of the data falling into the wrong hands if the device or media is stolen

All data (e.g., transcripts, audio/video recordings, emails, and documents) will be stored on the companies SharePoint only accessible locally via the company network or encrypted VPN, protected by a two-factor authenticated Microsoft single sign-on portal.

C15. How will the participants be debriefed?

It is expected that wherever possible all participants will receive some form of debriefing. This might be a verbal debriefing or a written debriefing depending on the context of the study. Debriefing provides an opportunity to remind participants of the procedures and outcomes of the research, and to provide further assurances on areas such as confidentiality, anonymity, and retention of data. Projects that intentionally withhold information or deceive as part of their methodology must include a written debrief sheet. (Please refer to sections 6.1 and 6.2 of the REP for further guidance)

Post-interview minutes will be provided, and a brief period at the end of interviews to allow participants to review information they've shared in case redaction is desired.

C16a. Will your results be available in the public arena? (e.g. publication in journals, books, shown or performed in a public space, presented at a conference, internet publication and placing a dissertation in the library) *see section 8 of the REP.*

Answer: Yes

If yes, please provide brief details:

NB: Please note that if participants wish to exercise their right to withdraw or request erasure of their personal data following collection and analysis this may not be possible having regard to permitted exemptions for research under data protection legislation i.e. where it would seriously impair the achievement of the research objectives. Notwithstanding the above, data subjects must still be advised of their rights to object in the information sheet, which can only be overridden if the "research is necessary for a task carried out for reasons of public interest".

Final dissertation and supplementary documentation may be available digitally via the University of Chichester Library.

C16b. Will your research data be made available in the public arena?

Certain research funding bodies require that research data is made Open Access i.e. freely available to the public. The University has an Open Access Policy that outlines the expectations and requirements for researchers at the University. Contact the Chair of the Research Ethics Committee in the first instance if you are unsure as to how to proceed.

Answer: No

If yes, please provide brief details as to how the data will be prepared for public access including an overview of the meta-data that will accompany published data sets. Please also confirm that your intentions with respect to making data open access are clearly communicated to participants so that they can provide informed consent:

N/a

C17. Are there any additional comments or information you consider relevant, or any additional information that you require from the Committee?

None at time of writing.

[end of form]

7 Project Review

#	Acceptance Criteria	Achieved?	Comments
AC01	Library trawls directory, and correctly identifies C# solutions	Yes	CsprojScanner and ConfigScanner classes manually tested on a subset of files retrieves all direct-dependencies
AC02	The library can build solutions	N/a	As discussed with a senior engineer a more efficient approach using API requests was implemented
AC03	The library can identify third-party licenses in a project	Yes	Using package name formatting, Edwards packages are identified seperately from external packages
AC04	The library can take any package and unzip it	Yes	Yes, although, due using API requests, this functionality is not accessible from either interface due to redundancy to new processes
AC05	The library identifies a licence used by a package in the: metadata, licence text, or url	Yes	Yes, LicenceScanner class identifies licences uses literal and fuzzy string comparisions against SPDX database
AC06	the library can persist licence and package data to a file	Yes	Pickler class can pickle (persist/serialise) package, licence, and licenceScore c# objects to permanent system files. Additional CRUD functionality was implemented too.
AC07	Library can open, read, and de-serialise extracted mappings into C# objects.	Yes	Pickler class also deserialises back into C# objects for automatic manipulation
AC08	C# object can dynamically convert into PDF generation instructions	Yes	PdfBuilder class implements text wrapping and tabulation of licence data for credit document
AC09	Output PDF to Edwards design specifications	No	Due to time constraints, and proof-of-concept design, priority was given to functional requirements
AC10	HTML documentation produced by tool such as Doxygen	No	Due to time constraints, and proof-of-concept design, priority was given to functional requirements
AC11	Code thoroughly commented	Yes	Comments implemented using XML comments that are interpretable by Visual Studio
AC12	All library functions: scan, build, extract, and credit are implemented through the command-line	Yes	Yes, main function in the Program project implements all necessary functionality to discover and credit licences
AC13	Command-line app is protected against improper inputs.	Yes	Input handling checks against file-system, and input characters to protect against un-handleable inputs

AC14	All library functions: scan, build, extract, and credit are implemented through a graphical interface	Yes	Winforms was used to design a simple, progressive, graphical interface that implements library functions
AC15	User interface implements good user experience design	Yes	Gutenberg areas researched and considered in addition to Schneiderman's eight golden rules of design, implemented in interface.
AC16	Manual covers all: library, command-line, and graphical use cases.	No	User manual not produced yet
AC17	Comprehensive documentation of objects, methods, and data structures.	No	Aside from comments, documentation was forgone in favour of improving functional requirements

8 User Manual

KLEPTOS PROGRAM USER MANUAL

KLEPTOS PROGRAM USER MANUAL

siranjeevi.gopi@external.atlascopco.com
EDWARDS VACUUM LTD

siranjeevi.gopi@external.atlascopco.com
EDWARDS VACUUM LTD

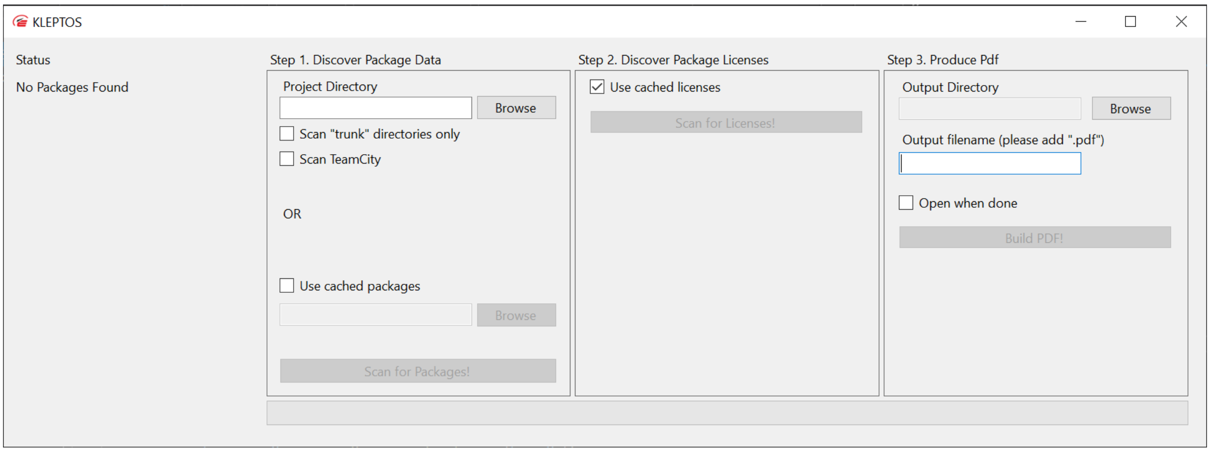
Contents

1 Introduction

The KLEPTOS program is an internal tool designed for the use of discovering, identifying, and crediting third-party packages and package licences that are used in Edwards Vacuum software products. The program is not a panacea to replace good practises in ensuring licence compliance, and as of version 0.1.3 should be cautioned that it offers only a proof-of-concept solution to the licence discovery process.

2 Step 1. Discover Package Data

Upon executing the program, the user is greeted with the following screen:



Status

No Packages Found

☐ Scan "trunk" directories only

☐ Scan TeamCity

☐ Use cached packages

☑ Use cached licenses

☐ Open when done

Before any further functionality is enabled to the user, they must discover a project's packages. The user can elect to either scan new packages from source code (which will require an internet connection) or to read-in previously discovered and persisted packages from a JSON file.

2.1 Discovering New Packages

To discover new packages from source-code, the source code must be available locally on the machine. Typically, this would be retrieved from the SVN server, although may vary depending on the development teams processes and tools. If uncertain how to obtain source code for an Edwards product, consult team management for support.

1. Click the "browse" button to open the explorer window.
2. Navigate to the directory where the source-code is stored on your machine.
3. To optimise the discovery of packages, select the "Scan 'trunk' directories only" checkbox to constrain file search to sub-directories containing 'trunk' paths. Directories in the trunk path are the most developed and intended to be distributed with the final product.
4. Select the "Scan TeamCity" checkbox if network access to the Internal TeamCity server is available. This server holds all internal Edwards packages and ensures a more comprehensive scanning for third-party dependencies.

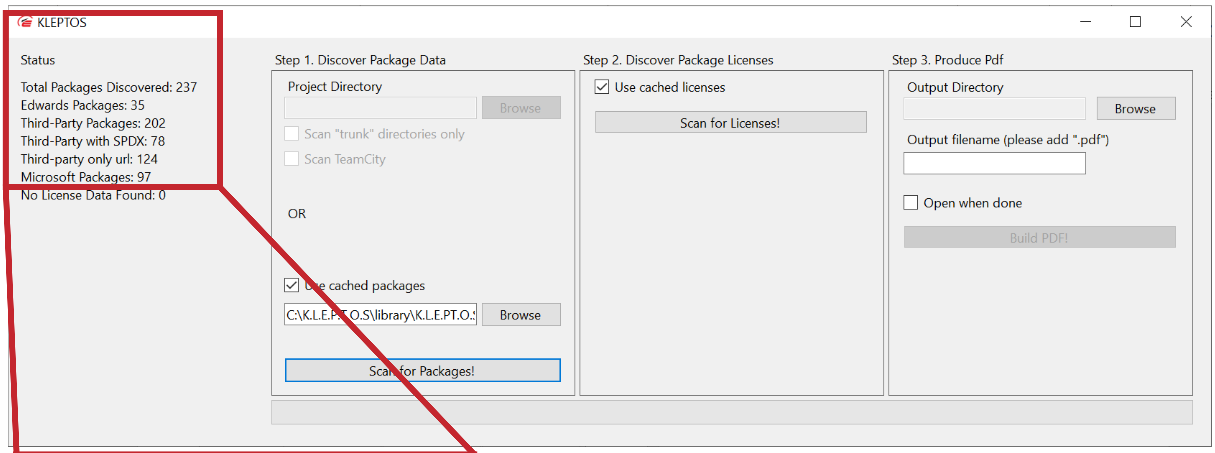
5. Once fields are populated, the “Scan for Packages!” button should be enabled and the program ready to begin the discovery process. Click this to proceed.

2.2 Read in previously discovered packages

This does not require any network connections, although does require that packages have previously been discovered by the machine. To proceed click the “browse” button and navigate to the location of the JSON file that contains persisted package data.

2.3 Post-licence Discovery Screen

After scanning source code, or importing from a persisted file, the screen should resemble the following:



KLEPTOS

Status

Total Packages Discovered: 237
Edwards Packages: 35
Third-Party Packages: 202
Third-Party with SPDX: 78
Third-party only url: 124
Microsoft Packages: 97
No License Data Found: 0

Step 1. Discover Package Data

Project Directory

□ Scan "trunk" directories only
□ Scan TeamCity

OR

☑ Use cached packages
C:\K.L.E.P.T.O.S\library\K.L.E.PT.O.:

Step 2. Discover Package Licenses

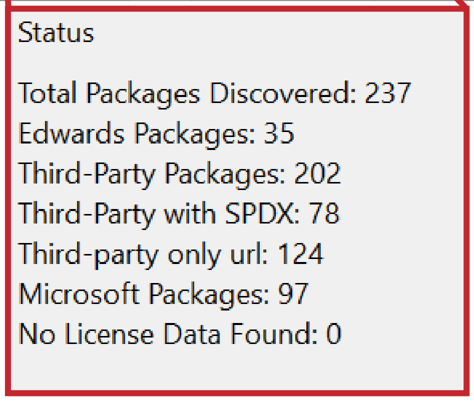
☑ Use cached licenses

Step 3. Produce Pdf

Output Directory

Output filename (please add ".pdf")

□ Open when done



Status

Total Packages Discovered: 237
Edwards Packages: 35
Third-Party Packages: 202
Third-Party with SPDX: 78
Third-party only url: 124
Microsoft Packages: 97
No License Data Found: 0

The status field displays a breakdown of the most pertinent categorisations of packages.

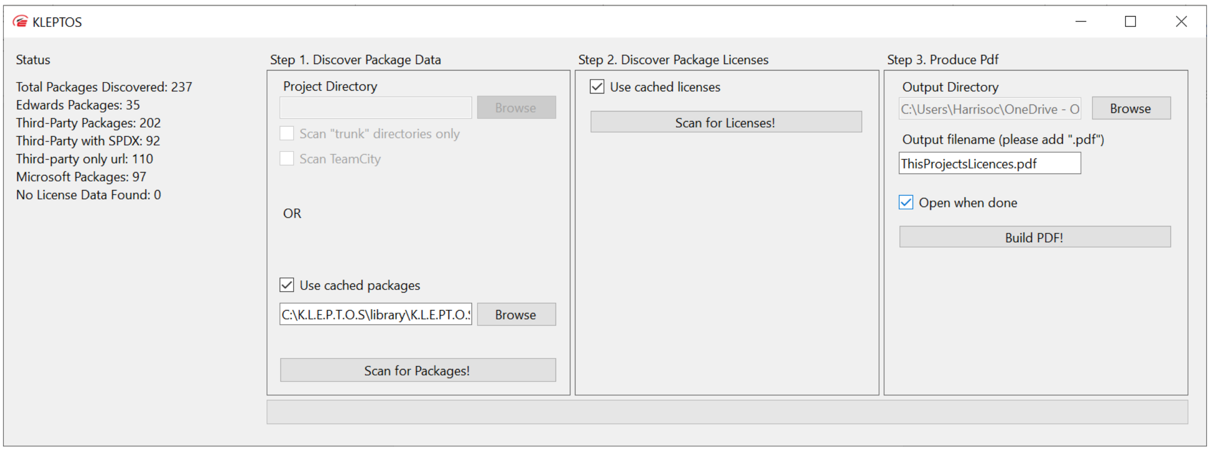
3 Step 2. Discover Package Licenses

Once packages have been discovered and are visible in the status field, the “Step 2.” Pane should be enabled.

The checkbox “use cached licenses” can be selected to attempt to read in previously discovered package licences from memory, to optimise the licence discovery process. If this is the first time the program is run, the discovered license cache will be empty, for a project with over 500 licences to be discovered, expect the process to run in excess of six-hours, to provide an accurate analysis of licence texts against the official SPDX database. If the cached licenses read-in from memory encompass all packages discovered, the process should take no more than one minute.

4 Step 3. Produce PDF

Once packages and their respective licences have been discovered, it is possible to collate the data into a document to credit package authors and licences.



KLEPTOS

Status

Total Packages Discovered: 237
Edwards Packages: 35
Third-Party Packages: 202
Third-Party with SPDX: 92
Third-party only url: 110
Microsoft Packages: 97
No License Data Found: 0

Step 1. Discover Package Data

Project Directory

□ Scan "trunk" directories only

□ Scan TeamCity

OR

☑ Use cached packages

C:\K.L.E.P.T.O.S\library\K.L.E.PT.O.:

Step 2. Discover Package Licenses

☑ Use cached licenses

Step 3. Produce Pdf

Output Directory: C:\Users\Harrisoc\OneDrive - O

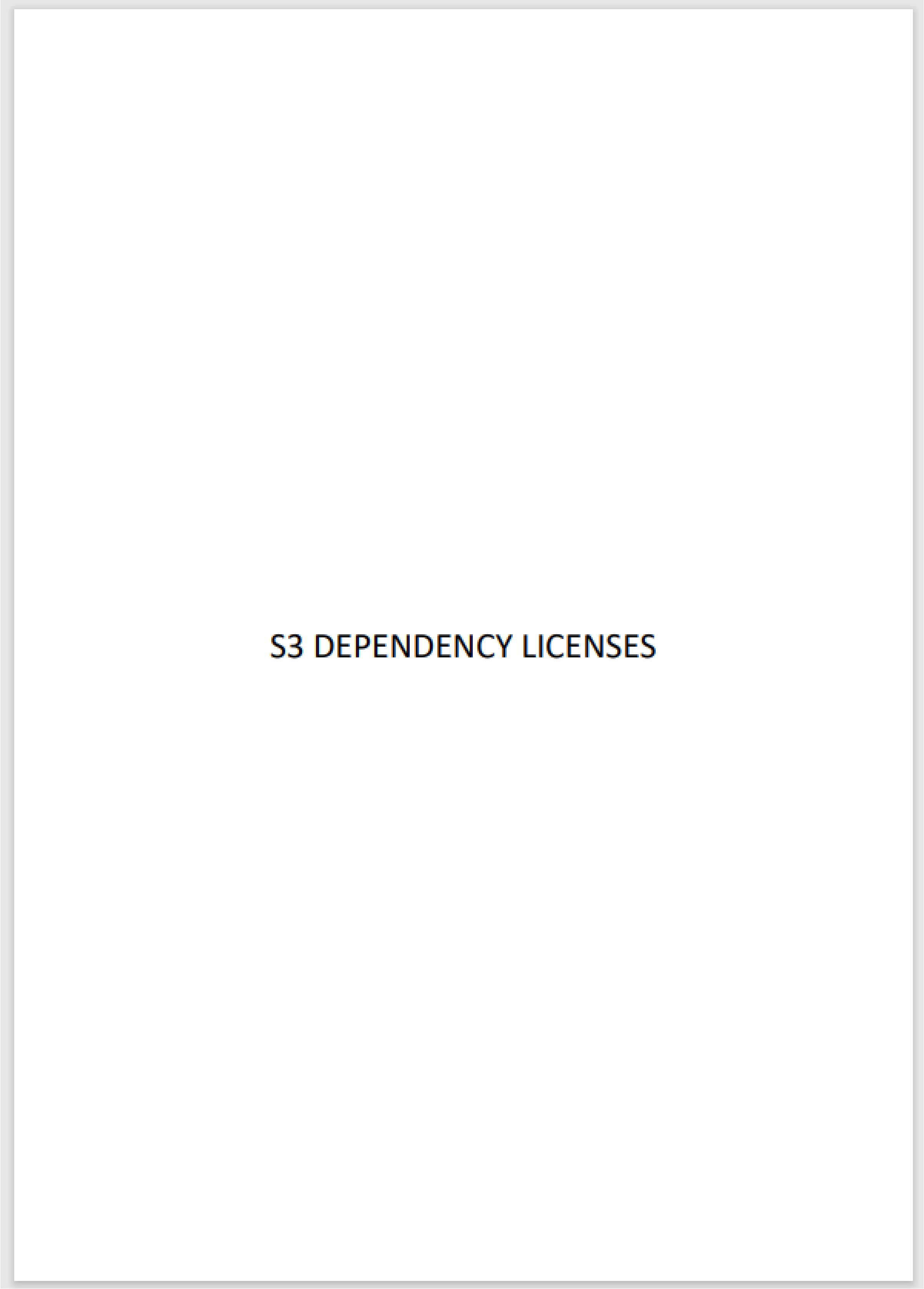
Output filename (please add ".pdf"): ThisProjectsLicences.pdf

☑ Open when done

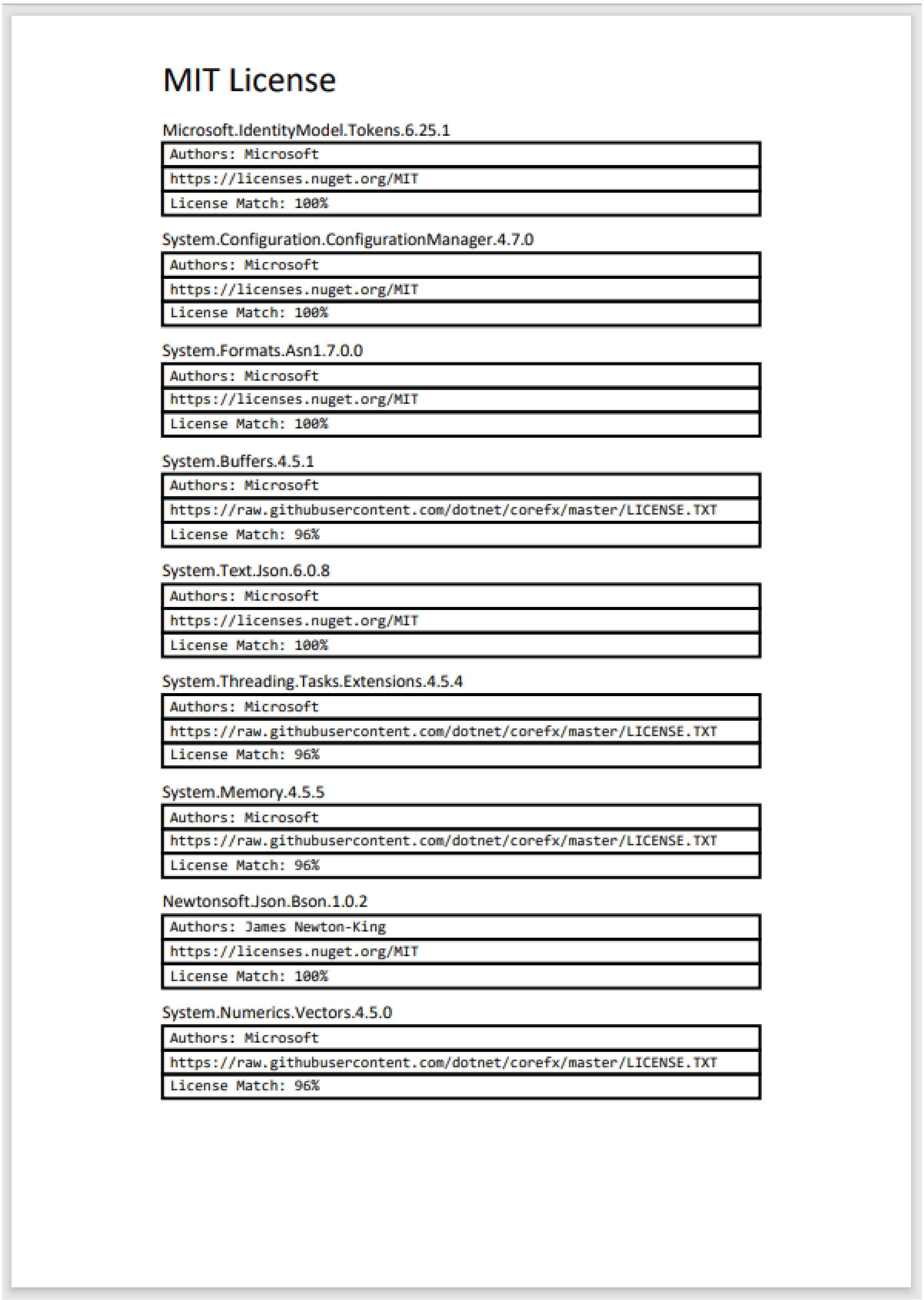
1. Using the “browse” button, select a directory to which you wish to output the PDF document.
2. Next, enter a name for the PDF document (include the .pdf file extension in the naming”
3. If you wish to open the document immediately after creation, selet the “open when done” checkbox. Regardless, the document will be saved locally to the location provided.
4. Review the document.

4.1 Example Subset of the Licence Document

PDF output filename manual_test.pdf:



S3 DEPENDENCY LICENSES



MIT License

Microsoft.IdentityModel.Tokens.6.25.1

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Configuration.ConfigurationManager.4.7.0

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Formats.Asn1.7.0.0

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Buffers.4.5.1

Authors: Microsoft
https://raw.githubusercontent.com/dotnet/corefx/master/LICENSE.TXT
License Match: 96%

System.Text.Json.6.0.8

Authors: Microsoft
https://licenses.nuget.org/MIT
License Match: 100%

System.Threading.Tasks.Extensions.4.5.4

Authors: Microsoft
https://raw.githubusercontent.com/dotnet/corefx/master/LICENSE.TXT
License Match: 96%

System.Memory.4.5.5

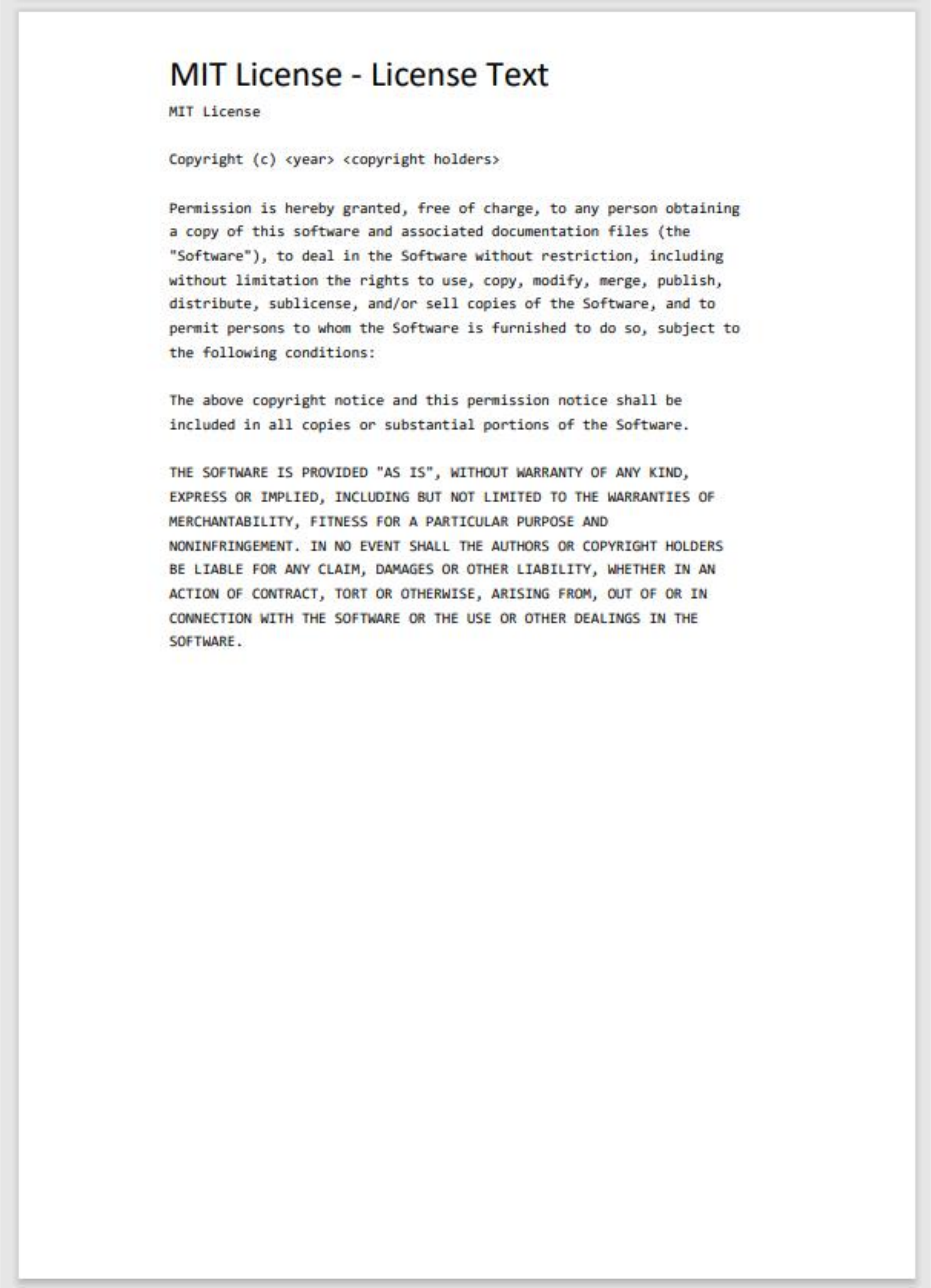
Authors: Microsoft
https://raw.githubusercontent.com/dotnet/corefx/master/LICENSE.TXT
License Match: 96%

Newtonsoft.Json.Bson.1.0.2

Authors: James Newton-King
https://licenses.nuget.org/MIT
License Match: 100%

System.Numerics.Vectors.4.5.0

Authors: Microsoft
https://raw.githubusercontent.com/dotnet/corefx/master/LICENSE.TXT
License Match: 96%



MIT License - License Text

MIT License

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

[END OF USER MANUAL]

9 Test Plan

Classes	Description
IScanner Classes	Tests must use mocking of the local filesystem and edge-case and regression testing of functionality, to ensure quality is maintained in the process of retrieving package dependencies.
API Scanners	Unit Testing and Manual tests, should consider the high variety of package return formats, both XML and JSON formats should be deserialiseable in addition to being able to retrieve packages with specific versions, or when provided with a version range.
LicenseScanner	Tests, both manual unit testing, will need to be run on the live system to: 1. Retrieve official licence texts from the SPDX database. 2. Deserialise licenceScoreCahe files into C# objects 3. Edge-case scenarios where licence URLs return error 400, 404, 500 and alternative status codes. 4. The Comparison of licence webpage content against the database with literal string comparison 5. The Comparison of licence webpage content against the database with fuzzy string comparision
PDFBuilder	Test manually that the output pdf includes: 1. Proper formatting of tabularised data and table cells 2. Where data such as authors, URL, and licence texts extends greater than the width of the page, proper text-wrapping is applied. 3. LicenceURL cells are hyperlinked to the correct licence text for the package 4. Licences match the licence text from the licence URL