e-ISSN: 2583-3472

Technoarete Transactions on Advances in Computer Applications
Vol-1, Issue-2, June 2022

A review on the various tools and methods used in the UK to analyse software requirements

Dr. Rajendra Prasad P [1*], Satya Prakash Yadav [2]

[1] Assistant Professor, Department of Electronics and Communication Engineering, M S Ramaiah Institute of Technology, India
[2] Gl Bajaj Institute of Technology and Management, Greater Noida, India
*Corresponding Author Email: [1]rajisvec@gmail.com

Abstract

This paper reviews the various tools and methods used in the UK to analyse software requirements. The review considers the various models and techniques used to analyse software requirements and presents a summary of the benefits, challenges and limitations of each. The review also presents some of the best practices for software requirement analysis, as well as identifying common pitfalls and potential improvement areas. The paper concludes by providing an overview of the current state of software requirement analysis in the UK and provides recommendations for future research and development.

Keywords

Agile methodology, Engineering tools, Software Requirements Analysis (SRA), UML diagrams.

INTRODUCTION

Software requirements analysis is a vital part of the software development process. It is the process of gathering, documenting and analysing the required elements of a software system. In the UK, there are methods that can be used to analyse software requirements. In this review, we will explore some of these tools and methods and discuss their pros and cons.

Agile Methodology

It is based on the principles of collaboration, customer satisfaction, and continual improvement. Agile encourages developers to work closely with customers to ensure that their needs are taken into account throughout the development process. This makes it well-suited to analysing software requirements, as it allows developers to quickly understand user needs and adjust the scope of the project as needed.

Pros

- Flexible: Agile allows developers to quickly adjust the scope of the project, allowing them to respond to changes in user needs.
- Collaborative: Agile encourages collaboration between the customer and the development team, ensuring that user needs are taken into account.
- Iterative: Agile allows developers to quickly develop and improve software, helping to ensure that the end product meets user needs.

Cons

- Time-consuming: Agile can be time-consuming, as it requires a lot of collaboration and iteration.
- Difficult to manage: Agile can be difficult to manage, as it requires close supervision from the project manager.
- Costly: Agile can be costly, as it requires developers to spend more time on the project.

Case Analysis

It is a useful tool for understanding how users will interact with the system and what the system needs to do in order to meet those needs. Use Case Analysis is well-suited to analysing software requirements, as it allows developers to gain a better understanding of the user's needs and develop a system that meets those needs.

Pros

- Understandable: Use Case Analysis can be easily understood by developers, as it is a graphical representation of the user's needs.
- Flexible: Use Case Analysis allows developers to quickly adjust the scope of the project in response to changes in user needs.
- Iterative: Use Case Analysis allows developers to quickly develop and improve software, helping to ensure that the end product meets user needs.

Cons

- Time-consuming: Use Case Analysis can be time-consuming, as it requires a lot of detail work.
- Difficult to manage: Use Case Analysis can be difficult to manage, as it requires close supervision from the project manager.
- Costly: Use Case Analysis can be costly, as it requires developers to spend more time on the project.

DEFINITION OF SOFTWARE REQUIREMENTS ANALYSIS

Software Requirements Analysis (SRA) is developed to determine what the system needs to do and how it should be designed to achieve the desired outcomes. It involves examining existing systems, processes, data, and equipment, as well as researching customer needs and expectations [30]. The SRA process is used to identify, document, and validate the requirements of a system. It is also used to develop a plan for the development of the system. SRA is a critical part of the software development process and is essential to the success of a software system.

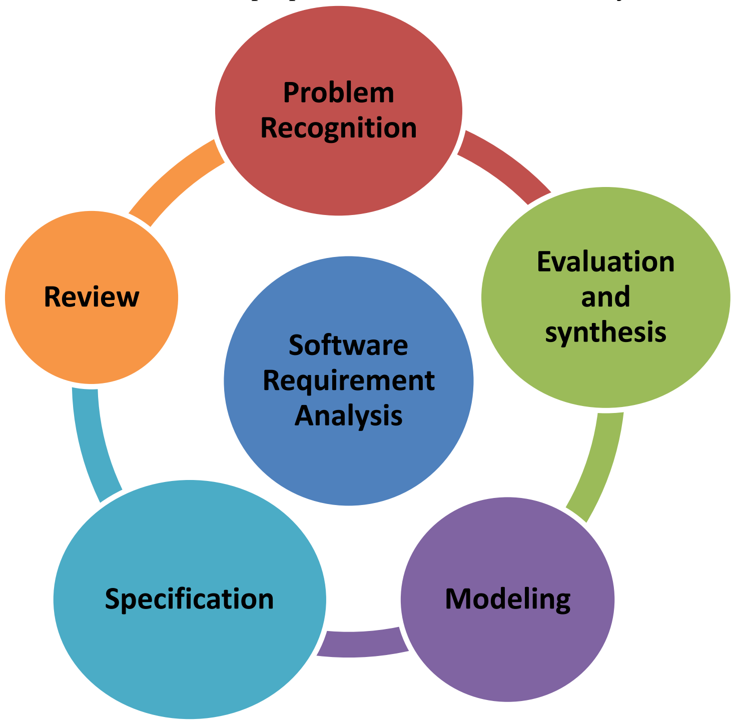


Figure 1: Software Requirement Analysis
(Source: made by the author)

In the UK, there are several tools and methods used to carry out software requirements analysis. These include requirements engineering tools, UML diagrams, use cases, and storyboards. Requirements engineering tools are used to capture, organize, and track the requirements of a software system [25]. UML diagrams are used to visualize the relationships between system components and to define the architecture of a system. Use cases are used to describe the interactions between the system and its users. Finally, storyboards are used to create visual representations of user interface designs. These tools and methods are used to ensure that the software system meets the requirements of its users and that it is designed to meet the desired objectives.

Software Requirements Analysis includes identifying and understanding the needs of the stakeholders, eliciting and refining requirements, and determining how to satisfy those needs. In the United Kingdom, there are several methods and tools used to analyse software requirements. These include use case modelling, user stories, storyboarding, prototyping, and acceptance testing.

Use case modelling is a technique for capturing and documenting the functional requirements of a software system [26]. This method involves describing the system in terms of scenarios, or use cases, which are descriptions of how the system will be used by its stakeholders. This helps to ensure that all stakeholders have a clear understanding of how the system will work and how it will meet their needs.

User stories are a type of requirement that focuses on using the system. They are written in the form of a narration, making them easier to understand and to relate to. User stories help to capture the user's experience of using the system and can be used to drive the development process.

Storyboarding is a visualisation technique used to capture the user's experience of using the system. It involves creating a storyboard, or a series of images, to represent a user's experience of using the system. This helps to ensure that all stakeholders have a clear understanding of how the system will work and how it will meet the user's needs.

Prototyping is a technique for exploring the design of a system before it is built. This helps to ensure that the system will meet the user's needs and is an important part of the requirements analysis process.

Acceptance testing is a process used to ensure that the system meets the user's requirements. It involves testing the system against a set of criteria to ensure that it meets the user's needs.

BENEFITS OF SOFTWARE REQUIREMENTS ANALYSIS

Software Requirements Analysis is an important process that helps to identify and define the functional requirements of a software system [29]. It helps to ensure that the software system meets user needs and is designed with the correct functionality.

In the UK, a number of different tools and methods are available for software requirements analysis. These include:

1. Feasibility Analysis: This is used to determine whether the software project is feasible and whether it is worth

pursuing. It involves investigating the technical, economic, and legal aspects of the project.

2. Requirements Gathering: This is used to identify and document the user requirements and preferences for the software system. It involves interviewing stakeholders, conducting surveys, and analysing existing systems.

3. Use Case Modelling: This is used to identify the different scenarios that the software system must support [28]. It involves creating use cases that describe the different steps of a process and the different actors involved.

4. Prototyping: This is used to build a basic version of the software system to test ideas and assumptions. It helps to identify any gaps or inconsistencies in the requirements.

5. Quality Assurance: This is used to ensure that the software system meets the specified requirements. It involves testing the system and verifying that it meets the user needs.

6. Documentation: This is used to document the software requirements in detail so that they can be shared with developers and other stakeholders. It helps to ensure that everyone is on the same page and that the requirements are clear.

These tools and methods can be used to ensure that the software system meets the user requirements and is built with the correct functionality [21]. By using them, organizations can ensure that their software projects are successful and deliver the desired results.

TOOLS AND METHODS USED IN THE UK

Requirements Engineering

Requirements engineering is the process of gathering and analysing customer requirements for a software project. This process helps to identify the purpose of the software and determine what features it should have. It also involves creating a set of requirements that the software must meet [26]. The requirements engineering process typically involves brainstorming, interviews, workshops and prototyping.

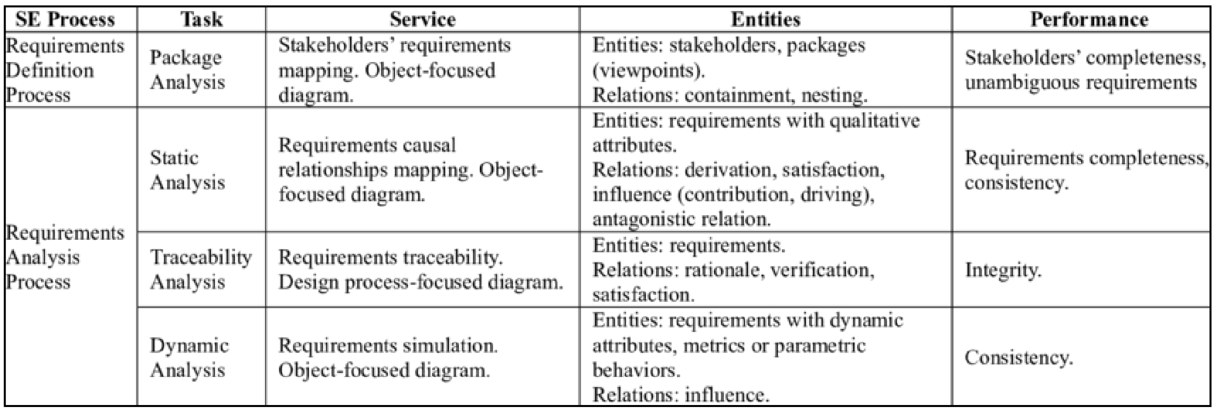
In the UK, there are a number of tools and methods used to analyse software requirements. These include agile software development, user stories, prototyping, use cases and requirement management tools.

Agile Software making

This approach allows teams to quickly adapt to changing customer needs and requirements. Agile software development is based on the principles of collaboration, customer focus and iterative development.

User Stories

User stories are short descriptions of a feature that a user would like the software to have. These stories are used to capture customer requirements and are written in a user-friendly language that the customer can easily understand.



SE Process	Task	Service	Entities	Performance
Requirements Definition Process	Package Analysis	Stakeholders' requirements mapping. Object-focused diagram.	Entities: stakeholders, packages (viewpoints). Relations: containment, nesting.	Stakeholders' completeness, unambiguous requirements
Requirements Analysis Process	Static Analysis	Requirements causal relationships mapping. Object-focused diagram.	Entities: requirements with qualitative attributes. Relations: derivation, satisfaction, influence (contribution, driving), antagonistic relation.	Requirements completeness, consistency.
	Traceability Analysis	Requirements traceability. Design process-focused diagram.	Entities: requirements. Relations: rationale, verification, satisfaction.	Integrity.
	Dynamic Analysis	Requirements simulation. Object-focused diagram.	Entities: requirements with dynamic attributes, metrics or parametric behaviors. Relations: influence.	Consistency.

Table 1: Different descriptions of the SE processes
(Source: Schwarze et al. 2020 [22])

Prototyping

Prototyping is a technique used to create a visual representation of a software application. This allows customers to get an idea of how the software will look and feel before it is developed [22]. Prototypes can also be used to test out different features and user interfaces.

Use Cases

Use cases are tools used to describe the interactions between a user and a system. They provide a detailed description of how a user would use the system to complete a task. Use cases are typically written in a narrative format and can include diagrams and flowcharts.

Requirement Management Tools

Examples of requirement management tools include JIRA, IBM Rational DOORS and IBM Rational Requirements Composer [24].Overall, requirements engineering is a critical process in software development. The tools and methods mentioned above are just some of the ways that teams in the UK can analyse customer requirements. By using these tools and methods, teams can ensure that they deliver a product that meets the customer's needs.

Requirements engineering is the process of defining and managing the requirements of a product or system. It is typically divided into five main activities: elicitation, analysis, specification, validation and management. Requirements engineering is used in many different industries and organisations, including software, construction, engineering, manufacturing, and healthcare.

In the UK, there are several different tools and methods used to analyse software requirements. These include:

1. Requirements Analysis Workshops: Requirements analysis workshops are used to identify and document the requirements of a system. The process involves stakeholders such as users, analysts, and developers, who are invited to participate in a facilitated session [23]. During the workshop, the participants discuss and agree upon the requirements for the system and document them in a requirements document.

2. Use Cases: Use cases are used to describe how a system will be used in different scenarios. Use cases are written from the perspective of the user, and describe the steps that the user must take in order to achieve a certain goal [11]. They are used to help developers and stakeholders understand the system requirements and design a solution that meets those needs.

3. Prototyping: Prototyping is a method used to create a working version of a system. This allows developers to quickly and easily test the system before it is developed in full. It also allows users to provide feedback on the system before it is released.

4. Software Requirements Specification (SRS): An SRS is a document that describes the functional, non-functional, and interface requirements for a software system. The SRS should be detailed enough to allow developers to create a solution that meets the system's requirements.

5. Use Case Diagrams: Use case diagrams are used to graphically represent the interactions between a user and a system. They are used to show how the user interacts with the system and how the system responds to those interactions.

6. System Flow Diagrams: System flow diagrams are used to visually represent the flow of data through a system [19]. They are used to identify potential areas of improvement, identify potential problems, and provide insight into how the system works.

7. It involves testing the system to ensure it meets the requirements and is free from errors. SQA is usually performed by a dedicated SQA team.

8. Decision Analysis: Decision analysis is used to analyse the cost and benefit of different options. It is used to identify which option is best for the organisation and which option is most cost-effective.

These are just some of the tools and methods used to analyse software requirements in the UK. Each organisation may have its own preferred tools and methods, depending on its size and the type of project [12]. It is important to understand the tools and methods used in your organisation, so that you can ensure that the requirements are properly analysed and documented.

Use Case Analysis

Use case analysis is a common tool used in the UK to analyse software requirements. This method involves looking at how a user would interact with the system, and understanding the steps they would take to complete a task. This helps to identify the functionality that should be built into the software, as well as any potential problems that may arise. By mapping out the user's interactions with the system, it is possible to create a detailed picture of what the user needs, and how the system should be designed to meet those needs.

Data Flow Diagrams

Data flow diagrams (DFDs) are another tool used to analyse software requirements in the UK. This method involves creating a diagram which shows the flow of data through the system, from the user input to the output. The diagram is used to identify areas of the system which may require additional functionality or attention [18].

Functional Decomposition

Functional decomposition is a further method used to analyse software requirements in the UK. This involves breaking down the system into its component parts, and understanding how each part interacts with the others [13]. This helps to identify any areas which may need additional functionality, or which may need to be redesigned in order to meet the user's needs. By understanding the system at a fundamental level, it is possible to ensure that the system meets the user's requirements.

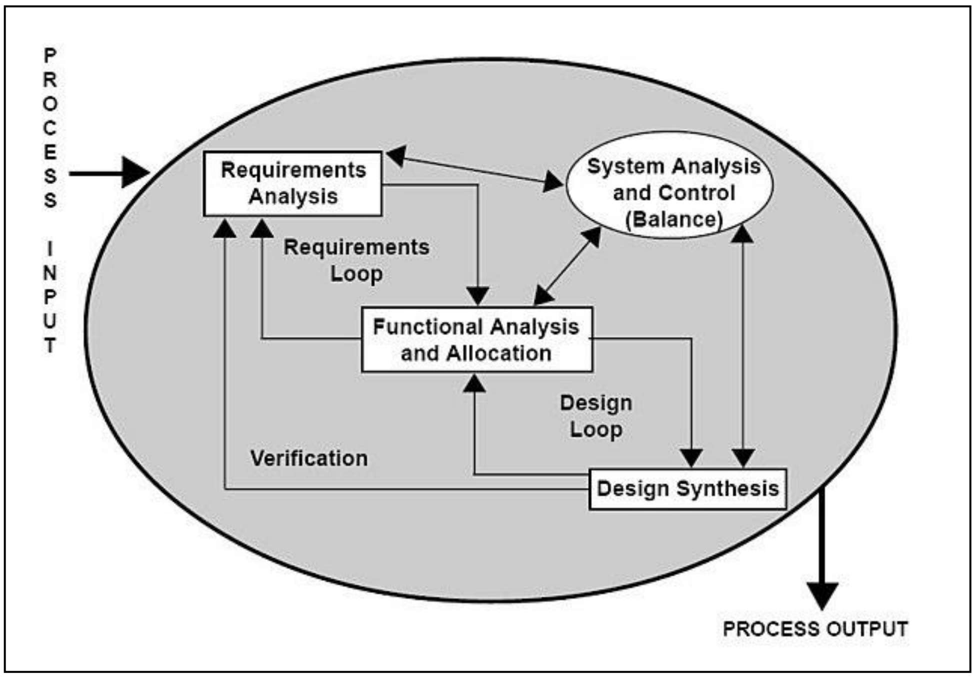


Figure 2: Process of input to Output via SE processes
(Source: Leij-Halfwerk et al. 2019 [17])

Prototyping

Prototyping is a tool used in the UK to analyse software requirements which involves creating a working model of the system. This model can be used to test the functionality of the system, as well as to demonstrate the user experience [17]. By testing the system in this way, it is possible to identify any potential problems or areas of improvement before they become an issue. Prototyping can also help to identify areas of the system which need additional functionality, or which may need to be redesigned in order to meet the user's requirements.

Use case analysis is a method of identifying, clarifying, and organizing system requirements. It is used to define the scope of a system and the functionality it will provide [14]. In the United Kingdom, use case analysis is often used to analyse software requirements. It is a method of breaking down the functionality of a system into individual scenarios and use cases. Each use case is then evaluated to determine its requirements and to identify any potential problems or opportunities. By modelling the system at a high level of abstraction, it is possible to identify areas where more detailed analysis may be required.

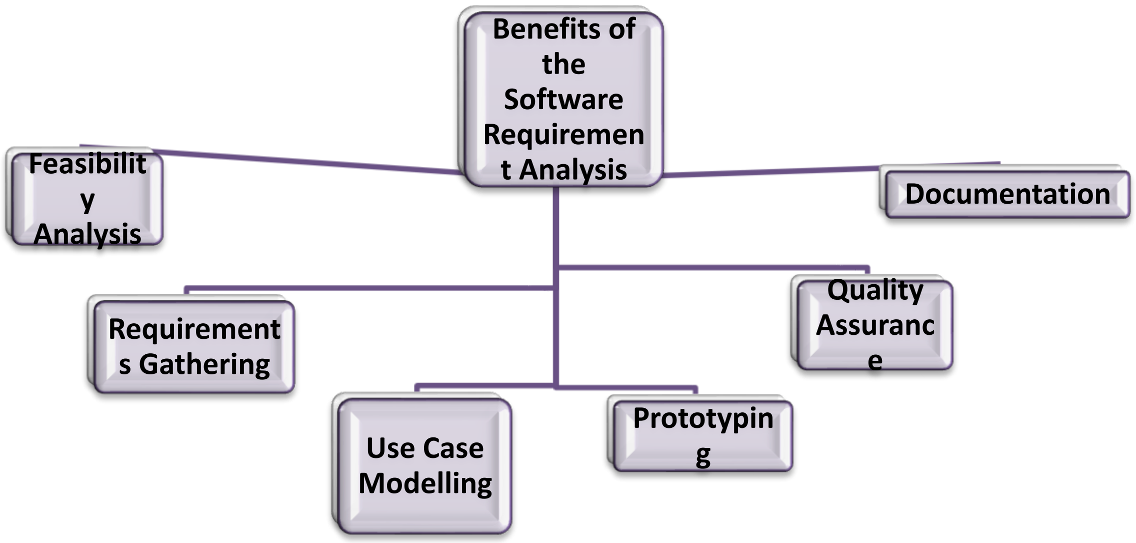


Figure 3: Benefits of the Software Requirement Analysis
(Source: Made by the author)

UML Diagrams

UML diagrams are another common technique used in the UK to analyse software requirements. UML diagrams provide a visual representation of the relationships between different elements of the system. The diagrams are used to identify how the components of the system interact and how data flows between them [16]. UML diagrams are useful for understanding complex systems, as they provide a high-level overview of the system. UML diagrams can be used to identify any potential problems or issues that may arise during the development of the system.

Functional Decomposition

Functional decomposition is a technique used to break down the functionality of a system into smaller, more manageable components. It is a useful tool for analysing complex systems, as it helps to identify the various components of the system and how they interact with each other. By breaking down the functionality of the system into smaller components, it is easier to identify any potential problems or areas for improvement. Functional decomposition is often used in the UK to analyse software requirements.

Requirements Traceability

Requirements traceability is a technique used to track changes to requirements throughout the development process. It is used to ensure that the final system meets all of the requirements that were specified at the beginning of the project. Requirements traceability also helps to ensure that the system is developed according to the timeline and budget. In the UK, requirements traceability is often used to analyse software requirements.

These are some of the tools and methods used in the UK to analyse software requirements [15]. However, all of these methods can be used to ensure that the system is developed according to the specified requirements.

Storyboarding

Storyboarding is a widely used tool in the UK to analyse software requirements. It is a visualisation technique used to capture the main ideas, steps, and flow of a software system or process. It is a popular and effective tool used to create a visual representation of a system in order to help stakeholders better understand the requirements of a system.

Storyboarding can be used to define the overall structure of the system, including the user interface, data flow, and processes [1]. It is also useful for exploring and visualising the relationships between different components of the system. The storyboarding process also helps stakeholders to identify potential risks, errors, and usability issues, as well as to identify potential opportunities for improvement.

In the UK, storyboarding is often used in conjunction with other methods and tools, such as requirements elicitation, user story mapping, use cases, user stories, and user journey mapping. These methods can be used to refine and clarify the storyboarded system, and to identify potential areas of improvement. Storyboarding is also often used as a way to communicate requirements to developers and stakeholders, and to ensure that all stakeholders are on the same page with regards to the desired outcomes of the system.

Storyboarding is a method used to analyse software requirements in the UK. It is a visual approach to the process of requirement gathering and analysis. The main goal of storyboarding is to capture the user’s requirements and create a visual representation of the proposed software [9]. This type of analysis allows product teams to quickly identify the scope of a project and identify potential issues that may arise in the development process.

The storyboarding process involves creating a series of diagrams that describe the flow of the software from the user’s perspective. This can be done by creating a visual storyboard of user interactions with the software, or by creating a series of use cases that illustrate how a user would interact with the software. The diagrams can then be analysed to identify any gaps in the requirements or to identify areas that require further clarification.

Another tool used in the UK to analyse software requirements is prototyping. Prototyping is the process of creating a model of the proposed software solution. This can be done using wireframes, mock-ups, or interactive prototypes [8]. This approach allows product teams to quickly evaluate the proposed solution and identify potential design issues before implementing the software. Finally, the UK also utilises informal requirements gathering techniques such as interviews, focus groups, and surveys.

User Story Mapping

User story mapping is a popular and widely used tool for software requirements analysis in the UK. It is a technique that helps teams focus on user goals and objectives when designing, developing, and launching a product or service. User story mapping involves creating a visual representation of a user’s journey through a product or service, from the user’s perspective. It is used to better understand user needs and help prioritize features [3]. It helps teams to align their product vision, identify missing pieces and potential risks, and ensure that the product is being built in the right order.

Name	Description	Kind
System Description Specification	Describes the technical requirements for particular software.	Structural
Goal Model	Captures the purpose of the organization as a goal tree.	Behavioral
Domain Model	Focuses in the language of the organization and its system.	Structural

Organization	Documents of the interaction between different actors and the organization.	Structural
Role Model	Finds the organization roles and the goals they achieve in the different roles and external actors.	Structural

Table 2: Description of different models of the software analysis
(Source: Made by the author)

The user story map is a visual tool that outlines a user's journey through a product or service. It starts with the user's initial goal and works its way through the different stages, tasks, and actions they must take to complete the desired outcome [2]. The user story map helps teams identify potential areas of improvement, prioritize features, and keep track of progress.

Other tools and methods used to analyse software requirements in the UK include:

Prototyping: Prototyping is a great tool for gathering user feedback and validating design decisions. Prototypes can be used to test the usability and functionality of a product or service before it is released. Prototypes can help teams identify potential risks, iterate quickly, and get a better understanding of user needs.

Requirements Workshops: Requirements workshops are a great way for teams to collaborate and communicate on the development of a product or service. They help to define the scope of a project, identify stakeholders, and understand user needs. Requirements workshops also allow teams to brainstorm ideas, collaborate on design decisions, and document decisions.

Business Process Modelling: Business process modelling is a tool used to understand organizational goals and objectives [7]. It is used to identify, analyze, and document the steps and processes required to meet the objectives. Business process modelling helps teams identify potential risks, identify ways to improve efficiency, and ensure that stakeholders are on the same page.These are just some of the tools and methods used to analyse software requirements in the UK.

Agile Methodology

Agile methodology is a popular software development method used in the UK. It is a collaborative approach to problem solving that focuses on customer satisfaction, collaboration, and rapid delivery. It is used to efficiently deliver high-quality software and products in shorter development cycles. Agile methodology involves breaking down large projects into smaller components and iteratively working on them. The iterative process allows for quick changes and feedback loops. It also allows teams to be more creative and flexible when developing solutions.

Waterfall Methodology

Waterfall methodology is another popular software development method used in the UK. It follows a linear progression from planning to development to testing [2]. It is a traditional approach to software development that does not allow for much flexibility or rapid iteration. Waterfall methodology is often used in larger projects where the requirements are well-defined from the start and there is no need for changes.

Prototyping

Prototyping is a great way to quickly develop a working model of a software system. It is used to identify and refine user requirements, evaluate design alternatives, and test system functionality. This process allows for the rapid iteration of ideas and makes it easier to identify potential problems before the development phase.

Usability Testing

Usability testing is a technique used to evaluate the usability of a system, product, or service. It is used to identify usability problems and user experience issues [4]. Usability testing is typically conducted with a representative sample of users who are asked to complete tasks using the system. This allows developers to identify any potential issues before the system is released to the public.

Systems Analysis

Systems analysis is a process used to analyse the requirements of a software system. It involves studying the system’s design, architecture, hardware and software components, and user interface. It is used to identify potential problems and ensure that the system meets the user’s needs. Systems analysis is used to ensure that the system is developed to meet its goals and objectives.

These are some of the most common methods and tools used in the UK to analyse software requirements. Each method has its own advantages and disadvantages, so it is important to choose the right one for your project [7]. When it comes to software development, it is important to remember that the best results are achieved when the right methodology is combined with the right team and the right tools.

CONCLUSION

The UK has an array of tools and methods that can be used to analyse software requirements. These include traditional methods such as interviews, document analysis and surveys as well as more modern techniques such as agile methodologies and user stories. There is no one size fits all approach to analysing software requirements, and organisations should consider the best approach for their particular needs. Ultimately, it is important to understand the users' needs and goals in order to ensure the requirements are properly captured and addressed.

Software requirements analysis is a critical activity for successful software development. Additionally, the UK also makes use of other approaches, such as use cases, user stories and agile methods, to ensure that software requirements are properly understood and documented. Ultimately, the selection of the right tools and methods for software requirements analysis will depend on the specific project and the needs of the stakeholders.

Summary of Tools and Methods

In the UK, the most common tools and methods used to analyse software requirements are those from the Unified Modelling Language (UML). It provides a visual representation of the software requirements and helps to simplify the process of analysing them [6]. Other popular tools used in the UK are tools based on the Business Process Model and Notation (BPMN) standard, as well as tools such as Use Cases, Activity Diagrams, and Storyboards.

When working with UML, the most important part is the use of diagrams. Diagrams help to illustrate the structure of the software system and provide a visual representation of the relationships between the components. Diagrams are used to identify the functional, non-functional, and technical requirements of the system, as well as to identify the relationships between them. Other popular UML-based tools include class diagrams, sequence diagrams, and state diagrams.

Use Cases are another popular tool used to analyse software requirements. Use Cases are documents that describe the interactions between the user and the system. They provide a detailed description of the system's functionality and are used to identify the functionality the user needs from the system.

Activity Diagrams are similar to Use Cases in that they provide a visual representation of the system's functionality [5]. They are used to identify the tasks and activities that the user needs to perform in order to interact with the system.

Storyboards are used to create a visual representation of the system's user interface. They provide a way for the user to interact with the system, and are used to identify the user's needs and preferences.

Finally, there are tools such as BPMN and process modelling tools. These tools are used to create business process diagrams that provide a visual representation of the system's functionality. They are used to identify the workflow of the system, as well as to identify any potential problems or inefficiencies.

Overall, the tools and methods used in the UK to analyse software requirements are varied and depend on the needs and preferences of the user [3]. UML is the most popular tool, but tools such as Use Cases, Activity Diagrams, and Storyboards are also commonly used. BPMN and process modelling tools are also useful for creating business process diagrams.

Advantages and Disadvantages of Software Requirements Analysis

Software Requirements Analysis (SRA) is the process of evaluating and analysing software requirements. It helps to ensure that the software meets all the requirements of the user and that the software development team understands what the user needs and wants from the software. SRA can be used to identify any potential problems in the software requirements and help to create a plan for the development of the software.

Advantages

1. Improved Quality: Software Requirements Analysis helps to ensure that the software is of high quality and meets the user's needs [19]. It can help to identify potential problems in the software requirements and help to create a plan for the development of the software.

2. Cost Savings: SRA can help to reduce costs associated with developing software. By identifying potential problems and making plans to address them during the development process, developers can avoid costly and time-consuming changes later on.

3. Improved Efficiency: SRA can help to identify areas where changes in the software requirements would result in improved efficiency. This can help to reduce the time and cost associated with development.

4. Improved Team Communication: SRA helps to ensure that all members of the software development team are on the same page and working together towards a common goal.

Disadvantages

1. Time Consumption: SRA can be time consuming and can add additional time to the development process.

2. Resource Intensive: SRA can be resource intensive and can require additional personnel and resources to complete.

3. Technical Limitations: SRA can be limited by the technical knowledge of the team and can be difficult to complete for complex software projects.

4. Subjective Analysis: SRA can be a subjective process and can be biased depending on the experience and point of view of the analyst.

REFERENCES

[1] Muka, Taulant, et al. "A 24-step guide on how to design, conduct, and successfully publish a systematic review and meta-analysis in medical research." *European journal of epidemiology* 35.1 (2020): 49-60.

[2] O’Connor, Cliodhna, and Helene Joffe. "Intercoder reliability in qualitative research: debates and practical guidelines." *International journal of qualitative methods* 19 (2020): 1609406919899220.

[3] Longhurst, Georga J., et al. "Strength, weakness, opportunity, threat (SWOT) analysis of the adaptations to anatomical education in the United Kingdom and Republic of Ireland in response to the Covid-19 pandemic." *Anatomical sciences education* 13.3 (2020): 301-311.

[4] Longhurst, Georga J., et al. "Strength, weakness, opportunity, threat (SWOT) analysis of the adaptations to anatomical education in the United Kingdom and Republic of Ireland in response to the Covid-19 pandemic." *Anatomical sciences education* 13.3 (2020): 301-311.

[5] Braun, Virginia, and Victoria Clarke. "One size fits all? What counts as quality practice in (reflexive) thematic analysis?." *Qualitative research in psychology* 18.3 (2021): 328-352.

[6] Pappas, Ilias O., and Arch G. Woodside. "Fuzzy-set Qualitative Comparative Analysis (fsQCA): Guidelines for research practice in Information Systems and marketing." *International Journal of Information Management* 58 (2021): 102310.

[7] Jiang, Longda, et al. "A resource-efficient tool for mixed model association analysis of large-scale data." *Nature genetics* 51.12 (2019): 1749-1755.

[8] Abels, Esther, et al. "Computational pathology definitions, best practices, and recommendations for regulatory guidance: a white paper from the Digital Pathology Association." *The Journal of pathology* 249.3 (2019): 286-294.

[9] Bond, Melissa, Olaf Zawacki-Richter, and Mark Nichols. "Revisiting five decades of educational technology research: A content and authorship analysis of the British Journal of Educational Technology." *British journal of educational technology* 50.1 (2019): 12-63.

[10] Bitkina, Olga Vl, Hyun K. Kim, and Jaehyun Park. "Usability and user experience of medical devices: An overview of the current state, analysis methodologies, and future challenges." *International Journal of Industrial Ergonomics* 76 (2020): 102932.

[11] Vindrola-Padros, Cecilia, et al. "Perceptions and experiences of healthcare workers during the COVID-19 pandemic in the UK." *BMJ open* 10.11 (2020): e040503.

[12] Yaacoub, Jean-Paul, et al. "Security analysis of drones systems: Attacks, limitations, and recommendations." *Internet of Things* 11 (2020): 100218.

[13] Dagoumas, Athanasios S., and Nikolaos E. Koltsaklis. "Review of models for integrating renewable energy in the generation expansion planning." *Applied Energy* 242 (2019): 1573-1587.

[14] Ngiam, Kee Yuan, and Wei Khor. "Big data and machine learning algorithms for health-care delivery." *The Lancet Oncology* 20.5 (2019): e262-e273.

[15] Li, Jie, Floris Goerlandt, and Genserik Reniers. "An overview of scientometric mapping for the safety science community: Methods, tools, and framework." *Safety Science* 134 (2021): 105093.

[16] Li, Tianjing, Julian PT Higgins, and Jonathan J. Deeks. "Collecting data." *Cochrane handbook for systematic reviews of interventions* (2019): 109-141.

[17] Leij-Halfwerk, Susanne, et al. "Prevalence of protein-energy malnutrition risk in European older adults in community, residential and hospital settings, according to 22 malnutrition screening tools validated for use in adults≥ 65 years: a systematic review and meta-analysis." *Maturitas* 126 (2019): 80-89.

[18] Teasdale, Scott B., et al. "Dietary intake of people with severe mental illness: systematic review and meta-analysis." *The British Journal of Psychiatry* 214.5 (2019): 251-259.

[19] Wolff, Robert F., et al. "PROBAST: a tool to assess the risk of bias and applicability of prediction model studies." *Annals of internal medicine* 170.1 (2019): 51-58.

[20] Alqudah, Ahmad M., et al. "GWAS: fast-forwarding gene identification and characterization in temperate cereals: lessons from barley–a review." *Journal of advanced research* 22 (2020): 119-135.

[21] Lamprecht, Anna-Lena, et al. "Towards FAIR principles for research software." *Data Science* 3.1 (2020): 37-59.

[22] Schwarze, Katharina, et al. "The complete costs of genome sequencing: a microcosting study in cancer and rare diseases from a single center in the United Kingdom." *Genetics in Medicine* 22.1 (2020): 85-94.

[23] Albahri, O. S., et al. "Systematic review of artificial intelligence techniques in the detection and classification of COVID-19 medical images in terms of evaluation and benchmarking: Taxonomy analysis, challenges, future solutions and methodological aspects." *Journal of infection and public health* 13.10 (2020): 1381-1396.

[24] Lai, Meng-Chuan, et al. "Prevalence of co-occurring mental health diagnoses in the autism population: a systematic review and meta-analysis." *The Lancet Psychiatry* 6.10 (2019): 819-829.

[25] Abdi, Sarah, et al. "Understanding the care and support needs of older people: a scoping review and categorisation using the WHO international classification of functioning, disability and health framework (ICF)." *BMC geriatrics* 19.1 (2019): 1-15.

[26] Darko, Amos, et al. "A scientometric analysis and visualization of global green building research." *Building and Environment* 149 (2019): 501-511.

[27] Elliott, Maxwell L., et al. "What is the test-retest reliability of common task-functional MRI measures? New empirical evidence and a meta-analysis." *Psychological Science* 31.7 (2020): 792-806.

[28] Bond, Melissa, et al. "Mapping research in student engagement and educational technology in higher education: A systematic evidence map." *International journal of educational technology in higher education* 17.1 (2020): 1-30.

[29] Butler, Claire, et al. "Methodology." *Optimum models of hospice at home services for end-of-life care in England: a realist-informed mixed-methods evaluation*. National Institute for Health and Care Research, 2022.

[30] Lim, Eun-Jin, et al. "Systematic review and meta-analysis of the prevalence of chronic fatigue syndrome/myalgic encephalomyelitis (CFS/ME)." *Journal of translational medicine* 18.1 (2020): 1-15.