Al-Azhar Engineering 16th International Conference

Vol. 19, No. 72, July 2024, 15 - 30

SOFTWARE TESTING USING CUTTING-EDGE TECHNOLOGIES WITH SUPPORTING KNOWLEDGE MANAGMENT: A SURVEY

Rehab Adel AbdElrazek *, Nagdy Mahmoud Nagdy, Gamal Abdel Rahim Tharwat, Ahmed Kolkila

Computer Science Department, Faculty of Engineering, Al-Azhar University, Nasr City, 11884, Cairo, Egypt

***Correspondence:** rehabadelabdelrazek@gmail.com

Citation:

R.A. AbdElrazek, N.M. Nagdy, G.A. Tharawat and A. Kolikla," Software testing using cutting-edge technologies with supporting knowledge managment: a survey ", Journal of Al-Azhar University Engineering Sector, vol. 19, pp. 15 - 30, 2024.

Received: 14 November 2023

Revised: 11 January 2024

Accepted: 23 January 2024

DoI:10.21608/auej.2024.248722.1471

ABSTRACT

Software testing is a crucial process in the development of software applications. It involves evaluating a software system to identify any defects and ensuring that it meets the specified requirements. There are many models for applying software testing. Knowledge management (KM), which involves collecting, acquiring, modifying,and storing knowledge, The software testing phase is knowledge-intensive. Therefore, it would be useful if it impeded the knowledge management framework with software testing phases. Where the test case is used as a knowledge item. The widespread use of cutting-edge technologies such as cloud infrastructure and Raspery Pi robotic programming has introduced new possibilities for many applications. Consequently, it is useful to use cloud storage as a knowledge repository for the testing cases and establish a seamless connection between cloud storage, the software application, and Raspery Pi to store and reuse these test cases. Rapberry Pi can be used as an indicator for each testing practice's results' and knowledge item modification. This can result in numerous benefits, such as: (i) saving tester effort, testing time, and cost; and (iii) simplifying the testing process. This survey aims to gain an overview of the previous research work on the four following topics: (i) software engineering and testing; (ii) software testing with knowledge management; (iii) cloud infrastructure; and (iv) Raspberry Pi robotic programming. This will clarify the vision to use the previous cutting-edge technologies in the software testing phase's completeness with applying the knowledge management cycle. It is expected that it will reap many benefits on this topic.

KEYWORDS: Software engineering and Software testing, Knowledge management, Cloud infrastructure, Raspberry pi.

اختبار البرمجيات باستخدام أحدث التقنيات مع دعم إدارة المعرفة: دراسة استقصائية

رحاب عادل عبد الرازق*، نجدى محمود نجدى، جمال عبد الحليم ثروت، احمد عبد العزيزقلقيله

قسم النظم والحاسبات، كلية الهندسة، جامعة الأزهر، مدينة نصر، 11884، القاهرة، مصر.

***البريد الاليكتروني للباحث الرئيسي :** rehabadelabdelrazek@gmail.com

الملخص

يعد اختبار البرمجيات عملية حاسمة في تطوير تطبيقات البرمجيات. حيث أنها تختص ببناء نظام لتقييم مشاريع البرمجيات لتحديد أي عيوب مع التأكد من استيفائه للمتطلبات المحددة. وبالتالي، فهو يساعد في تحقيق النجاح الشامل لمشروع البرمجيات. هناك العديد من النماذج والأدوات لتطبيق اختبار البرمجيات. ومن الممكن إجراء عملية اختبار البرمجيات من خلال إدارة المعرفة، والتي تتضمن جمع المعرفة والحصول عليها وتعديلها وتخزينها. تتطلب مراحل اختبار البرمجيات معرفة مكثفة. لذلك، سيكون من المفيد أن يتم تنفيذها فى إطار إدارة المعرفة التى يتم تطبيقها على مراحل اختبار البرمجيات. في هذه الحالة، يتم استخدام حالة الاختبار كعنصر معرفي. لقد أدى الاستخدام الواسع النطاق للتقنيات المتطورة مثل البنية التحتية السحابية والبرمجة الروبوتية

إلى تقديم إمكانيات جديدة للعديد من التطبيقات. وبالتالي، من المفيد استخدام التخزين السحابي كمستودع معرفي Raspery Pi لبيانات حالة الاختبار. علاوة على ذلك، إنشاء اتصال سلس بين clould stoargae والتطبيقات البرمجية وRaspery Pi لتخزين وإعادة استخدام حالات الاختبار هذه. حيث يستخدم Rapberry Pi لإظهار نتائج ممارسات الاختبار مثل النجاح/الفشل أو الإشارة إلى ما إذا كانت هناك تغييرات جديدة على عنصر المعرفة.وبتطبيق ذلك سنحصل على العديد من الفوائد، مثل: (1) توفير جهد المختبر؛ (2) تقليل وقت الاختبار؛ و(3) تسهيل عملية الاختبار، خاصة أثناء مرحلتي الصيانة وتعديل البرامج. لقد أدى الاستخدام الواسع النطاق للبنية التحتية السحابية إلى ظهور إمكانيات جديدة للعديد من التطبيقات. هدف هذا الاستطلاع إلى الحصول على لمحة عامة عن الأعمال البحثية السابقة حول المواضيع الأربعة التالية: (1) هندسة البرمجيات واختبارها؛ (2) اختبار البرمجيات وإدارة المعرفة؛ (3) إنترنت الأشياء والبنية التحتية السحابية و(4) برمجة الروبوتات Raspberry Pi. يهدف هذا الاستطلاع إلى توضيح الرؤية لاستخدام التقنيات المتطورة مثل البنيه التحتيه السحابية لأداء مرحلة اختبار البرمجيات فى اطار ادارة المعرفة.ومن المتوقع ان استخدم التقنيات الحديثة فى مرحلة اختبار البرمجيات تجني فوائد كثيرة في هذا المجال. يهدف هذا الاستطلاع إلى توضيح الرؤية لاستخدام التقنيات المتطورة السابقة في استكمال مرحلة اختبار البرمجيات مع تطبيق دورة إدارة المعرفة. ومن المتوقع أن تجني فوائد كثيرة في هذا الموضوع.

الكلمات المفتاحية : هندسة واختبار البرمجيات، ادارة المعرفة، البنيه التحتيه السحابية، Raspberry Pi

1. Introduction

Software engineering is a constantly evolving field that encompasses various areas, including problem modelling, software design, validation and verification, process performance, and management [1].

Experts believe software productivity is critical, as it connects technical and economic factors in software engineering [2]. Initially, software productivity was emphasized, but later the focus shifted to quality [3]. Various technologies have been created to aid developers and testers and increase productivity [4].Utilizing cloud services and shared services can increase productivity[5].

In software engineering, quality refers to the software's ability to perform tasks accurately [6].Software testing is crucial for high-quality solutions, especially compared to other development phases[7].Various testing standards and models have emerged to meet organizational needs [8,9,10]. These models enable organizations to choose the best fit.Continuous development for SW projects increases teams’ capabilities to solve problems [11].

Relying on knowledge encourages production, exchange, and storage to improve accessibility and usability within an organization [12]. KM in software testing is gaining the interest of various researchers [13]. Human testers' knowledge and skills are crucial in quickly detecting faults and adapting to new situations during software testing [14]. Organizations use KM activities during software testing to cut costs and improve software quality [15]. Collaboration and knowledge exchange aid management, but ensuring system quality remains a major IT challenge [16]. The creation and analysis of functional requirements, project planning and estimation, programming, testing, etc. are examples of tasks that involve knowledge. Developing methods to integrate the knowledge involved in a project is one of the challenges in software engineering [17].

The IoT movement has been developing and accelerating at a pace never before seen in every sector of the world. IoT architecture refers to a network of physical items that exchange data over the Internet. Device, edge, fog, and cloud are the four distinct levels that make up this architecture in sophisticated IoT deployments [18]. The IoT is improving the quality of life and productivity by integrating everyday objects and changing how we interact with our surroundings [19].

There are many applications for IoT, such as (i) "smart education," which includes IoT and AI in the educational system, which seems attractive since it directly affects students' motivation, engagement, and attendance [20], and (ii) healthcare testing. IoT applications, which are integrated with various third-party healthcare applications and medical devices through REST APIs [21].

IoT is a rapidly developing technology that allows millions of different devices to communicate at high speeds and provide widespread services without the need for human involvement. IoT can be used with apps that rely on sensors to interact with the environment, which is the result of the potential rise of electronic devices in sensing systems [15].

Academics and research organizations aim to conduct cutting-edge studies. Low-cost robot development has benefited from the use of embedded systems, single-board computers, and microcomputers. Various systems and approaches are used with the Raspberry Pi [22]. This will establish machine-to-machine (M2M) communication where Raspberry Pi communicates with cloud infrastructure [23].

To perform software testing phases, researchers and organisations have developed many models.There are a few research papers that have presented software testing practices with knowledge management. So it creates a loss in storing, retrieval, and improvement of the generated knowledge items(Test cases), which enable the development team to reuse them[70]. This survey presents some works related to software testing with knowledge management and their gaps. In addition, it suggests the idea of applying knowledge management to software testing practices, especially in planning and design, since these stages are knowledge-intensive and include all testing types such as unit,integration, system, and acceptance tests. Also, highlight features in other topics that are related to cutting-edge technology, such as cloud infrastructure to store testing knowledge items in cloud storage and using Rasperypi to convert testing phase results to something tangible, such as showing an alarm for an important event in these stages. This idea is suitable for IoT software applications that include enhanced connectivity, real-time monitoring, automation, data-driven insights, and improved application data.This survey presents several research work. That can be defined by a set of features: (i) reference number; (ii) references to the associated publications; (iii) year of publication; (iv) the publisher's goals; (v) findings and the research results; and (vi) repository sources.

The reporting survey data helps open minds to applying cutting-edge technologies in the software testing with knowledge management. This can be achieved by establishing a seamless connection between clould storage , software application and a Raspberry Pi board through cloud networks. Concequently,This new SW testing strategy can produce many benefits, such as: (i) test environment scalability; (ii) test data generation; and (iii) guarantee testing data-driven insights for quick decision-making. (iv) Supporting SW programs maintenance and updates.Based on existing literature, this survey hypothesises that the software testing phase is for an IoT application for a software project whose tools are fit for developing IoT apps since its environment is being built utilizing a cloud infrastructure environment.

2. Research process and papers scope work

This section presents the whole research process and the main scope of this article and other related papers' scope. The survey was conducted over various years, with the lowest year being 1981 and the highest year being 2023, related to software testing using cutting-edge technologies to support and manage knowledge.Fig. 1 represents the papers selected according to the publication's year.

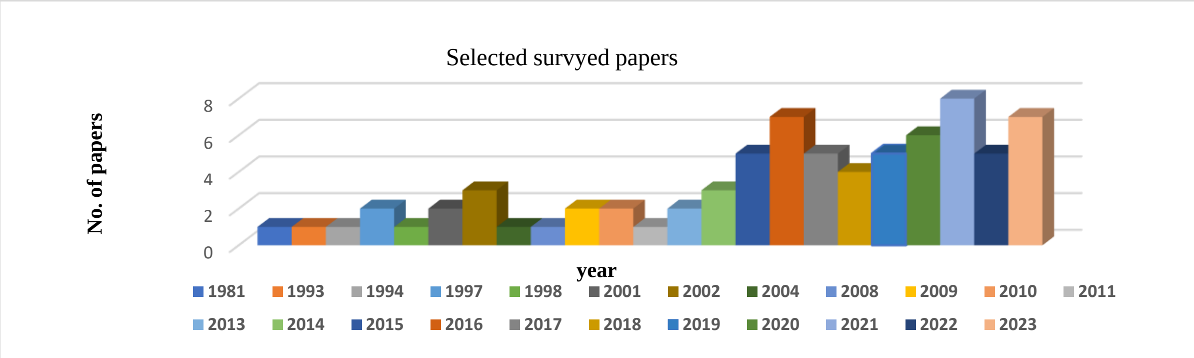

Fig. 1. Selected surveyed papers in different publication years.

The selection of articles of interest for this survey was based on the search terms that are related to the targeted keywords, publications' dates, and publications' sources. The five full-text databases (sources) are as follows:(i) ACM Digital Library; (ii) IEEE Xplore Digital Library; (iii) Arxiv Digital Library; (iv) Springer Link; and (v) Others. The number of papers selected from particular sources is presented in Table 1.

Table 1. Sources of papers

Source	**Number of papers(N)**	**% (Sum of N = 76)**
ACM Digital Library	11	15
IEEE Xplore Digital Library	14	18
ArXiv Digital Library	3	4
Springer Link.	4	5
Others	44	58
Sum	76	100%

The first major decision was to cover the scope of our interest so it covers not only agile software self-testing problem and knowledge management, but also includes other related topics The selected papers were classified into four topics of scope, which is equivalent to four classification types (topic categories). In context, these categories have been matched to the main interested topic and other related topics; Table 2 lists the suggested topic categories. This table identifies topic categories by using the following columns: (i) category id; and (ii) category description. Accordingly, these categories were expressed as follows: "C1": software engineering and testing phase; "C2": software testing with knowledge management; "C3": cloud infrastructure; C4": raspberry Pi robotic programming.

Table 2. Topics scope categories

Category id	**Category description**
C1	Software engineering and testing phase
C2	Software testing with knowledge management
C3	Cloud infrastructure
C4	Raspberry Pi robotic programming

The survey results are presented in the sections below. The article is structured as follows: Section 3 provides an overview of software engineering and testing phase. Section 4 presents the relationship between software testing and knowledge management,moreover presents how software testing can be affected by KM scenarios and an overview of the academic discipline. Section 5 explains cloud infrastructure and presents their related previous research work. Section 6 provides a discussion on the Raspberry Pi and its benefits. Section 7 introduced discussion and comprehensive analyzes of what was shown in the survey. Section 8 concludes the article and provides directions for future work.

3. Software engineering and Testing phase

Software engineering applies engineering principles to software development [24,25]. It has broadened with easier development tools [22]. Software development stages: analysis, design, programming, documentation, testing, maintenance, and error resolution provide superior solutions[26,27,28]. Software projects often have conflicting effort-spending views among project managers, test managers, and business managers, who prioritize timelines, quality assurance, and budgets, respectively [7, 29,30]. The development phase of the SDLC is more error-prone and vulnerable to attacks due to unintentional programmer mistakes [31].

Software's importance has grown, yet many projects fail due to factors like cost overruns and poor quality[32]. Software testing is essential in the software development process, as it validates and verifies the quality of a software product[22]. Most software is created and used without taking security needs into account[33].Therefore, software has gotten more complicated, mission-, safety-, and activity-critical, requiring an improvement in quality [33,34,35].

Testing is crucial but resource-intensive, accounting for over half of development costs. Reviewing software during development has gained interest for cost and complexity reduction [36]. Software testing costs aim to be roughly 50% of the total cost of software development [37]. There is evidence that problem identification and fault correction are among the elements that most significantly cause budget overruns[32].

Functional and structural testing are common software testing methods. Functional testing derives test cases from software specs, while structural testing uses implementation criteria to ensure all structures are exercised during execution[37]. Testers should concentrate on data value flows[38]. Software releases modify or introduce functionality, and each release undergoes quality review to meet requirements and address critical bugs[39].

Testing software is assessing the system being tested (SUT) as it is being executed with the objective of identifying any defects [40,29]. Acceptance, system, integration, module, and unit testing are common testing phases, also known as test levels. Acceptance testing evaluates the software under test (SUT) based on requirements and business processes[29]. A vital part of user acceptance testing is accessibility testing[40]. Mobile apps highlight the importance of accessibility for impaired users, but testing for accessibility remains challenging and requires improved automated tools to detect specific issues[41].

Testing (integration, module, unit) and Continuous Integration ensure software quality. Continuous Integration (CI) prevents "Integration Hell.”. The use of platforms for Continuous Integration (CI) and Continuous Delivery (CD) to speed up accessibility and adoption testing[41]. Continuous integration and automated testing projects have reported adding more test cases[42].

Regression testing checks for unintended consequences throughout software development as a subphase of testing[29,43,44]. Regression test case generation produces suitable test cases, and early prioritization reduces effort for managers and the testing team[45,46].

Test case methods include prioritization and reduction, which sort test cases based on priority and remove duplicates to speed up regression testing. Unit testing automates test case execution [47,69]. Software testing priorities can be set by grouping related user stories into modules[48]. As in agile environments, continuous integration and unit tests provide ongoing feedback to developers on code quality[49,62] and increase the quality of the code by reducing the number of issues caused by delayed code verification[50].

4. Software testing with Knowledge Managment

This section illustrates how software testing and knowledge management are mutually influential. Software testing involves managing large amounts of information, highlighting the importance of understanding business practices in software engineering. Knowledge combines past experiences, personal values, understanding, and expert insights, fostering the creation and evaluation of new information[51].

On some software project sites, there is a lack of highly competent human resources for software development. It is costly and difficult to move qualified teams from one location to another[52,53]. Organizational management must establish appropriate channels for effective knowledge exchange to achieve expected objectives[54]. Thus, Effective KM is necessary for organizations to attain maximum organizational excellence[55,56]. Organizations will need a lot of work and effort to implement KM[57,58].

knowledge management (KM) cycle, which involves three activities: knowledge capture and/or creation, knowledge sharing and dissemination, and knowledge utilization and/or application [54,59,60]. Knowledge capture identifies existing knowledge, while knowledge creation generates new knowledge by organizing explicit and tacit knowledge[59]. Ontology, a knowledge modelling technique, enables shared understanding of information structures across diverse application reas[61].Software development is a collaborative and knowledge-intensive process[17,43]. Furthermore,software testing is a knowledge-based stage of software development[51]. Informal knowledge enhances software processes, while formalized knowledge simplifies information retrieval and project identification[17]. Table 3 lists the retained proposals and shows some research work that are related to software testing and knowledge management.

Table 3. Software Testing and Knowledge Management proposals Surveyed.

No.	Ref	Year	Title	Goal/Focus	Main finding/Results
7	[70]	2019	Software testing process models benefits & drawbacks: a systematic literature review	Determine the characteristics of existing testing models, such as the domain, environment of application, improved aspects, benefits and drawbacks	Some models are not suitable for all organizations and domains, while most existing models are only suitable for generic conditions.
8	[74]	2020	Testing as a service (taas) a systematic literature map	1. The analysis of 76 TaaS cloud platforms identifies new characteristics of TaaS as a well-established research field 2. Analysis outcomes of existing works in test development practice.	Discusses gaps in the literature on Testing as a Service (TaaS), including the lack of standardization in TaaS frameworks and the need for further security studies.
9	[63]	2022	Ontology-Based Knowledge Management Tools for knowledge Sharing in Organization- A Review	assesses 13 ontology-based KM solutions for promoting information sharing using 10 relevant comparison criteria and a platform diagram.	Tools have limitations in extracting knowledge from implicit sources, making people the primary knowledge source.

Effective knowledge management achieves organizational goals by leveraging the knowledge of both team members and outsiders through gathering, storing, and utilizing knowledge[62]. Knowledge is valued across various fields, but managing it is challenging for software development organizations, and project success depends on effective knowledge management[43]. In context, skill management is crucial for software development organizations, and testing employees require expertise in this area, which may be difficult to articulate explicitly[63].

KM research is relevant to the government and other industries. The SECI model aids in exchanging tacit knowledge in software testing, which is difficult to convert into explicit forms [14,72]. Software practitioners use agile practices to facilitate knowledge exchange within organizations [17,49,64].Software practices struggle to convert tacit knowledge into explicit knowledge, but presenting it in the right form can aid better management [65].

5. Cloud Infrastructure

This section illustrates some applications and benefits for applying cloud infrastructure in modern several systems either in software or educational or healthcare etc. Internet of Things (IoT) is a key enabling technology for building smart spaces, which can support efficient in-person and virtual learning environments [20].

As highlighted in the preceding research, the study delves into the utilization of cutting-edge cloud-based modelling techniques and advanced tools to design and develop robust IoT systems.Therefore, cloud-based computing opens up a lot of possibilities for software development [15]. Another research proposes a layered model to cover traditional and emerging cloud paradigms. They categorize existing works based on the resilient model and analyze the commonly used techniques at each layer. Globally, cloud infrastructures have emerged as the predominant and favoured approach for delivering cloud services, facilitating seamless data exchange between cloud data storage and the client side, which has led to a considerable social dependence on them. Improve cloud infrastructure resilience and guarantee continuous service delivery under diverse circumstances [67]. Table 4 lists the retained proposals and shows some research work that are related to IoT and cloud infrastructure.

Table 4. IoT and Cloud infrastructure proposals surveyed

No.	Ref.	Year	Title	Goal/Focus	Main finding/Results
14	[21]	2023	Testing real-world healthcare IoT application and medical devices through REST APIs: Experiences and lessons learned	Provides insights and lessons learned from testing a real-world healthcare IoT application.	RES Test results reveal 56% coverage for REST API failures and nine potential faults in five APIs, with constraint-based testing (CBT), adaptive random testing(ART), and random testing (RT) testing techniques being more effective than fuzz testing(FT).
15	[18]	2021	Modeling self-adaptative IoT architecture	Presents a framework that allows for the modeling and analysis of self-adaptive behaviors in IoT systems. The suggested Domain-Specific Language for IoT (DSL) covers both static and dynamic features of IoT deployments.	Provides special research, methods, and theories in the field of IoT-native software engineering. Additionally, it illustrates the challenges brought on by the most widespread characteristics of IoT contexts.
16	[19]	2022	Just enough, just in time, just for "me": fundamental principles for engineering IoT-native software systems	Proposes three fundamental principles for engineering IoT-native software systems. These principles, namely "just enough," "just in time," and "just for 'me', address the challenges of IoT environments, such as resource limits and technology heterogeneity of IoT devices.	There is an IoT-native trend in designing, developing, deploying, and maintaining software systems. The authors distilled these three principles to guide the engineering of IoT-native software systems.
17	[20]		Towards Smart Education through Internet of Things: A Survey	Discusses the potential of IoT (Internet of Things) technology in creating smart spaces that can enhance both face-to-face and online education systems that known as smart education.	Shows the problems with the traditional educational system, and an exploratory study creates novel paths for academics and industry players to incorporate IoT and AI into intelligent education.

6. Raspberry Pi Robotics Programming

This section presents overview for some Raspberry Pi robotics applications. Raspberry Pi is a small and affordable computer that can connect to screens and peripherals like a regular desktop computer. Raspberry Pi enables people of all ages to learn about computer devices and programming languages like Python and Scratch. It offers features such as internet browsing, video viewing, document writing, and gaming[66].

The majority of Internet of Things (IoT) applications developed today come in the form of hardware and related software modules. Future cross-manufacturer IoT app shops will need to loosen the tight link between hardware and software[68]. The Raspberry Pi is used in IoT as an edge device that links different sensors to show data processing results.

Table 5 lists the recognized applications, displays some research on Raspberry Pi robotics, and explains how it may be used with cloud infrastructure and the Internet of Things.

Table 5. Raspberry Pi application proposals

No.	Ref.	Year	Title	Goal/Focus	Main finding/Results
18	[39]	2023	Robotics Solution for Agriculture: Automated and IoT-enabled Tomato ,Picking and Packing	Introduces a robotics solution for the food industry, specifically focusing on automated tomato picking and packing.	SMEs face food safety and labor shortage issues, while automated tomato processing ensures high-quality fresh, perishable tomatoes.
19	[71]	2022	An Effective Forest Fire Detection Framework Using Heterogeneous Wireless Multimedia Sensor Networks	Presents of a novel hierarchical approach for forest fire detection in IoT-based surveillance systems.	The suggested system closely matches the top-performing architectures from previous studies and lightweight architectures, with 98.28% validation accuracy and 29.94% energy savings.
20	[73]	2021	Integration of a Presence Robot in a Smart Home	Presents new solution for configuring a Raspberry Pi as a home robot, using shared data to adapt its behaviors. The system uses a smartphone app for remote configuration, following IoT standards. The usability of the system is validated, and it can be extended to other Linux devices.	Robots offer a practical, accessible solution for the growing smart home market, offering ease of use and the ability to expand to additional devices.
21	[74]	2023	A Modular Platform for Collaborative, Distributed Sensor Fusion	Designs modular sensing platform for autonomous vehicles (AV) called AVstack. It designed to overcome the challenges presented by closed-source AV platforms and testing infrastructures. It emphasizes sensor fusion to enhance the decision-making capabilities of AV to enhance the decision-making capabilities of AV.	With AVstack and ZMQ, software ecosystem implementations for modular testing can be tested with little software overhead on datasets, simulators, and real-world sensing settings.
22	[75]	2023	Software Reliability Analysis with Various Metrics using Ensembling Machine Learning Approach	Proposes a surveillance robot for monitoring and identifying intrusions near international boundaries. The robot is a Raspberry Pi 3 Model B as its computer, activates the camera upon detecting motion and streams live footage through the Internet of Things.	The surveillance robot uses Internet of Things video streaming to detect border intruders and alert security personnel, controlled by a Raspberry Pi 3 Model B, and an infrared sensor.
23	[76]		Fault diagnosis of computer numerical control (CNC) machine using cloud knowledge base and monitoring system	Create a CNC defect detection system by utilising a webcam, a Raspberry Pi 3, optical character recognition (OCR), and a cloud knowledge base to monitor the alarm panel of CNC drivers.	Remotely operated a CNC machine, receiving alarm diagnostic messages from a local host. A firebase cloud alert was registered, notifying maintenance officials. The solution improved maintenance speed, accuracy, and efficiency, is inexpensive, and easy to upgrade.

7. Discussion and Comperhensive analysis

According to this survey findings. Table 6 shows the article frequencies that can be classified within the four interested topic categories, such as follows (i) Software engineering and testing phase, (ii) Software testing with knowledge management, (iii) Cloud infrastructure, and (iv) Raspberry Pi robotic programming. Different presented research features helping in achieving many goals that are illustrated as follows:

(i) Providing an overview of the principles linked to these topics from the viewpoint of previous researchers.
(ii) Exploring earlier research work in these topics.

In table 6, out of 76 article references and included the repeated references in more category. In this survey "Software engineering and testing phase" is the most cited with a frequency 41 that represents (47%) category. The category " Software testing with knowledge management" next highest with frequency 24 represents (27%) category. Another important category in this study is Raspberry Pi robotic programming with a frequency 14 that represents (16%) category, finally the least cited " IoT and Cloud infrastructure" with frequency 9 represents (10%) category.

Table 6. Papers Classified in each topic scop

Category id	Topics scope	Freq.	% (N= 88)	References No.
C1	Software engineering and testing phase	41	47	[1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[38],[62],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[69]
C2	Software testing with knowledge management	24	27	[12],[13],[14],[15],[16],[17],[43],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62],[63],[64],[65],[70],[72]
C3	IoT and Cloud infrastructure	9	10	[15],[18],[19],[20],[21],[23],[34],[44],[67]
C4	Raspberry Pi robotic programming	14	16	[22],[23],[25],[32],[35],[36],[39],[66],[68],[71],[73],[74],[75],[76]
Sum		88		

The survey aims to present research on various topics, hoping to find a new trend in applying cutting-edge technologies for software testing to accommodate all software testing types such as unit, integration, system, and regression. Thereby, there is a transition from traditional work to modern technologies. Furthermore, a new software testing environment will be created. The main idea for this proposal is that the Raspberry Pi will exchange data with a cloud server machine. Cloud databases will serve as repositories for the data. In addition, the cloud acts as a central hub for collecting and analyzing data. Moreover, that will leverage many advantages in the software testing phase, such as the following:

1. Building a knowledge repository for all test case scenarios and storing data related to each software project component. This will contribute to overcoming the obstacles associated with preparing the test case data, such as redundancy and priority.
2. This will be useful in information technology, either in distributed teams or distributed projects.
3. Facilitating the testing of program modifications during project updates and maintenance.
4. Reducing the testers efforts and the overload in preparing test cases, which helps speed up the test implementation.
5. It stimulates and improves the software marketing process by providing prepared samples to the customer, even if he does not have sufficient experience in the field of the project, and it is considered part of the project requirements.
6. Converting the test events' result into something tangible by displaying an alert such as a light or sound from the Raspberry Pi device.

8. Conclusion

Software engineering practitioners can effectively become aware of knowledge. especially the software testing phase. As it serves to guarantee that the customer receives an effective and error-free product. The essential tasks for the software testing phase are (i) selecting suitable test case scenarios, predicting test data with no redundancy, and (ii) executing a testing programme in an easier manner. The two previous steps should be equivalent to the project's requirements. Testing program results are the actual data. Then a comparison will be made between predicted and actual data; if they are equal, then the test will pass; otherwise, it will fail.

In conclusion, this study provides regular field surveys and data collection over an extended period, enabling a more robust understanding of software testing with knowledge management. And introduce features and benefits for cutting-edge technology such as cloud infrastructure and the Raspberry Pi. By pursuing the avenues mentioned above, we can further refine our understanding of the presented research topics. thereby opening up an effective strategy for using cutting-edge technologies in the software testing process with knowledge management.

References

[1] Glass, R., Vessey, I., & Ramesh, V. (2002). Research in software engineering: an analysis of the literature. Information And Software Technology,44(8), 491-506. doi: 10.1016/s0950-5849(02)00049-6.

[2] B. W. Boehm, Software Engineering Economics. Prentice Hall, 1981.

[3] Leveson, N. G., & Turner, C. S. (1993). An investigation of the Therac-25 accidents. Computer, 26(7), 18-41.

[4] L. Yu, L. Zhang, H. Xiang, Y. Su, W. Zhao, and J. Zhu. 2009. A Framework of Testing as a Service. In International Conference on Management and Service Science. 1–4.

[5] Qi Zhang, Lu Cheng, and Raouf Boutaba. 2010. Cloud computing: state-of-the-art and research challenges. Journal of Internet Services and Applications 1, 1 (01 May 2010), 7–18.

[6] P. Bourque and Richard E. Fairley (Eds.). 2014. Guide to the Software Engineering Body of Knowledge (SWEBOK(R)):Version 3.0 (3rd ed.). IEEE Computer Society Press.

[7] Hrabovská K, Rossi B, Pitner T. Software testing process models benefits & drawbacks: a systematic literature review. arXiv preprint arXiv:1901.01450. 2019 Jan 5.

[8] Wasif Afzal, Snehal Alone, Kerstin Glocksien, and Richard Torkar. Software test process improvement approaches: A systematic literature review and an industrialcase study. Journal of Systems and Software,111(Supplement C):1–33, January 2016. ISSN 0164-1212. doi: 10.1016/j.jss.2015.08.048.

[9] Cecilia Garcia, Abraham D´avila, and Marcelo Pessoa.Test Process Models: Systematic Literature Review.Communications in Computer and Information Science, pages 84–93. Springer, Cham, November 2014. ISBN 9783319130354 9783319130361. doi:10.1007/978-3-319-13036-1 8.

[10] Vahid Garousi, Michael Felderer, and Tuna Hacalo˘glu.Software test maturity assessment and test process improvement:A multivocal literature review. Information and Software Technology, 85(Supplement C):16–42, May 2017. ISSN 0950-5849. doi: 10.1016/j.infsof.2017.01.001.

[11] Jussi Kasurinen, Ossi Taipale, and Kari Smolander.How test organizations adopt new testing practices and methods? In Software Testing, Verification and Validation Workshops (ICSTW), 2011 IEEE Fourth International Conference on, pages 553–558. IEEE, 2011.

[12] Selic, B. Personal reflections on automation, programming culture, and model-based software engineering. Autom Softw Eng 15, 379–391 (2008). https://doi.org/10.1007/s10515-008-0035-7.

[13] O'Leary, D., Studer, R., 2001. Knowledge management: an interdisciplinary approach. IEEE Intell. Syst. 16.

[14] C. P. Maciel, E. Souza, N. L. Vijaykumar, R. Falbo, G. V. Meinerz,and K. R. Felizardo, "An Empirical Study on the Knowledge Management Practice in Software Testing," in Experimental Software Engineering Latin American Workshop (ESELAW'18). XXI Ibero-American Conference on Software Engineering (CIBSE),2018.

[15] F. Ihirwe, A. Indamutsa, D. D. Ruscio, S. Mazzini and A. Pierantonio, "Cloud-based modeling in IoT domain: a survey, open challenges and opportunities," 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C),Fukuoka, Japan, 2021, pp. 73-82, doi: 10.1109/MODELS-C53483.2021.00018.

[16] Karvonen T, Sharp H, Barroca L (2018) Enterprise agility: why is transformation so hard? In: Garbajosa J, Wang X, Aguiar A (eds) International conference on Agile software development, Porto, Portugal.

[17] J. Itkonen, M. V. Mäntylä, and C. Lassenius, "Test better by exploring: Harnessing human skills and knowledge," IEEE Software,vol. 33, pp. 90-96, 2015.

[18] I. Alfonso, K. Garcés, H. Castro and J. Cabot, "Modeling self-adaptative IoT architectures," 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), Fukuoka, Japan, 2021, pp. 761-766, doi: 10.1109/MODELS-C53483.2021.00122.

[19] Zheng Li and Rajiv Ranjan. 2022. Just enough, just in time, just for "me": fundamental principles for engineering IoT-native software systems. In Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER '22). Association for Computing Machinery, New York, NY, USA, 56–60. https://doi.org/10.1145/3510455.3512785.

[20] Afzal Badshah, Anwar Ghani, Ali Daud, Ateeqa Jalal, Muhammad Bilal, and Jon Crowcroft. 2023. Towards Smart Education through Internet of Things: A Survey. ACM Comput. Surv. 56, 2, Article 26 (February 2024), 33 pages. https://doi.org/10.1145/3610401.

[21] Sartaj, Hassan, et al. "Testing real-world healthcare iot application: Experiences and lessons learned." arXiv preprint arXiv:2309.04230 (2023).

[22] S. E. Mathe, A. C. Pamarthy, H. K. Kondaveeti and S. Vappangi, "A Review on Raspberry Pi and its Robotic Applications," 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP), Vijayawada, India, 2022, pp. 1-6, doi: 10.1109/AISP53593.2022.9760590.

[23] Bruel JM, Ebersold S, Galinier F, Mazzara M, Naumchev A, Meyer B. The role of formalism in system requirements. ACM Computing Surveys (CSUR). 2021 May 23;54(5):1-36.

[24] Mohammad Esteki, Taghi Javdani Gandomani, Hadi Khosravi Farsani, "A risk management framework for distributed scrum using PRINCE2 methodology", Bulletin of Electrical Engineering and Informatics, Vol.9, No. 3, pp. 1299~1310, 2020, Doi: 10.1109/ICISE.2019.00014.

[25] Howison, J., & Bullard, J. (2016). Software in the scientific literature:Problems with seeing, finding, and using software mentioned in the biol-ogy literature. Journal of the Association for Information Science and Technology, 67(9), 2137-2155.

[26] Manhas, J. Initial framework for website design and development. Int. j. inf. tecnol. 9, 363–375 (2017). https://doi.org/10.1007/s41870-017-0045-4.

[27] Al-dweeri, R.M., Obeidat, Z.M., Al-dwiry, M.A., Alshurideh, M.T., Alhorani, A.M.: The impact of e-service quality and e-loyalty on online shopping: moderating effect of e-satisfactionand e-trust. Int. J. Mark. Stud. 9(2), 92 (2017).

[28] Dikert K, Paasivaara M, Lassenius C (2016) Challenges and success factors for large-scale agile transformations: a systematic literature review. J Syst Softw 119(September 2016):87–108.

[29] J. C. Westland, “The cost of errors in software development: Evidence from industry,” J. Syst. Softw., vol. 62, no. 1, pp. 1–9, 2002.

[30] Paul Ammann and Jeff Offutt. 2016. Introduction to software testing. Cambridge University Press.

[31] L. Bracciale, P. Loreti, A. Detti, R. Paolillo, and N. B. Melazzi,``Lightweight named object: An ICN-based abstraction for IoT device programming and management," IEEE Internet Things J., vol. 6, no. 3,pp. 50295039, Jun. 2019.

[32] Z. A. Maher, H. Shaikh, M. S. Khan, A. Arbaaeen, and A. Shah, “Factors affecting secure software development practices among developersAn investigation”, in Proc. ICETAS, 2018, pp. 16.

[33] A. K. Srivastava and S.Kumar, “An effective computational technique for taxonomic position of security vulnerability in software development,"J. Comput. Sci., vol. 25, pp. 388396, Mar. 2018.

[34] Elfriede Dustin. Effective software testing: 50 specific ways to improve your testing. Addison-Wesley, 2002.

[35] Bruno Rossi, Barbara Russo, and Giancarlo Succi.Modelling failures occurrences of open-source software with reliability growth. In IFIP International Conference on Open-Source Systems, pages 268–280.Springer, 2010.

[36] Goyal, S., Bhatia, P.K. Software fault prediction using lion optimization algorithm. Int. j. inf. tecnol. 13, 2185–2190 (2021). https://doi.org/10.1007/s41870-021-00804-w.

[37] Sumit Gupta, Sunil Indori, and Atishey Bansal. 2015. Automation beyond application test scripting. Proceedings of the 15th Annual International Software Testing Conference.

[38] H. Zhu, P. A. V. Hall, and J. H. R. May, “Software unit test coverage and adequacy,” ACM Comput. Surveys, vol. 29, no. 4, pp. 366–427, 1997.

[39] A. Stepanova, H. Pham and A. Zourmand, "Robotics Solution for Agriculture: Automated and IoT-enabled Tomato Picking and Packing," 2023 IEEE International Conference on Agrosystem Engineering, Technology & Applications (AGRETA), Shah Alam, Malaysia, 2023, pp. 108-112, doi: 10.1109/AGRETA57740.2023.10262632.

[40] G Iqbal, J., Firdous, T., Shrivastava, A.K. et al. Modelling and predicting software vulnerabilities using a sigmoid function. Int. j. inf. tecnol. 14, 649–655 (2022). https://doi.org/10.1007/s41870-021-00844-2.

[41] Integration Hell. https://www.solutionsiq.com/agile-glossary/integration-hell/.

[42] S. Vasanthapriyan, J. Tian, and J. Xiang, "A survey on knowledge management in software engineering," in 2015 IEEE International Conference on Software Quality, Reliability and Security-Companion, 2015, pp. 237-244.

[43] Filip I. Vokolos and Phyllis. G. Frankl. 1997. Pythia: A regression test selection tool based on textual differencing. In Proceedings of the 3rd International Conference on Reliability, Quality and Safety of Software-Intensive Systems. Chapman & Hall, Ltd. London, UK, 3–21.

[44] Nguyen-Duc A, Abrahamsson P. Continuous experimentation on artificial intelligence software: a research agenda. InProceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering 2020 Nov 8 (pp. 1513-1516).

[45] A. Schwartz and H. Do. 2016. Cost-effective regression testing through adaptive test prioritization strategies. J. Syst. Softw.115, 61–81. DOI:10.1016/j.jss.2016.01.018.

[46] Monalisha Khandai, Arup A. Acharya, and Durga P. Mohapatra. 2011. A survey on test case generation from UML model.Int. J. Comput. Sci. Info. Technol. 2, 3, 1164–1171.

[47] Girardon G, Costa V, Machado R, Bernardino M, Legramante G, Basso FP, de Macedo Rodrigues E, Neto A. Testing as a service (taas) a systematic literature map. InProceedings of the 35th Annual ACM Symposium on Applied Computing 2020 Mar 30 (pp. 1989-1996)

[48] A. Orso and G. Rothermel, "Software testing: A research travelogue (2000–2014)," in Proc. Future Softw. Eng., 2014, pp. 117–132.

[49] P. Kandil, S. Moussa, and N. Badr. 2017. Cluster based test cases prioritization and selection technique for agile regression testing. J. Softw.: Evol. Process. 29, 6.

[50] https://doi.org/10.1007/s10515-007-0015-3.

[51] Yang, T.-H., Ku, C.-Y., Liu, M.-N.: Case study: Application of enhanced Delphi method for software development and evaluation in medical institutes. Kybernetes 45(4), 637–649 (2016).

[52] E. R. Moreno, V. M. Cabello, F. López-Cortés, and E. A. Cedeño, `The reective practice as a tool for making tacit knowledge explicit,"Reective Pract., vol. 22, no. 4, pp. 116, May 2021.

[53] Hui Yi Chiang, Bertrand M. T. Lin, "A Decision Model for Human Resource Allocation in Project Management of Software Development," in IEEE Access, vol. 8, pp. 38073-38081, 2020, Doi: 10.1109/ACCESS.2020.2975829.

[54] Fernandes, C. I., Ferreira, J. J., & Raposo, M. L. (2013). Drivers to firm innovation and their effects on performance: an international comparison. International Entrepreneurship and Management Journal,9(4), 557–580.

[55] T. H. Davenport and L. Prusak, Working knowledge: How organizations manage what they know: Harvard Business Press,1998.

[56] Alavi, M., & Leidner, D. E. (2001). Knowledge management and knowledge management systems: conceptual foundations and research issues. MIS Quarterly, 25(1), 107–136.

[57] H. A. van den Berg, ``Three shapes of organisational knowledge,"J. Knowl. Manage., vol. 17, no. 2, pp. 159174, Mar. 2013.

[58] N. Shaw and P. Liu, ``A knowledge management system (KMS) using a storytelling-based approach to collect tacit knowledge," in Proc.SoutheastCon, Mar. 2016, pp. 16.

[59] F. Rezaei, M. Khalilzadeh, and P. Soleimani, ``Factors affecting knowledge management and its effect on organizational performance: Mediating the role of human capital," Adv. Hum.-Comput. Interact., vol. 2021, Feb. 2021,Art. no. 8857572.

[60] Ciancarini, P., Presutti, V. (2004). Towards Ontology Driven Software Design. In: Wirsing, M., Knapp, A., Balsamo, S. (eds) Radical Innovations of Software and Systems Engineering in the Future. RISSEF 2002. Lecture Notes in Computer Science, vol 2941. Springer, Berlin, Heidelberg.

[61] M. Z. Islam, S. M. Jasimuddin, and I. Hasan, ``Organizational culture,structure, technology infrastructure and knowledge sharing: Empirical evidence from MNCs based in Malaysia," VINE, vol. 45, no. 1, pp. 6788,Feb. 2015.

[62] Khalil C, Khalil S. Exploring knowledge management in agile software development organizations. International Entrepreneurship and Management Journal. 2020 Jun;16(2):555-69.

[63] Osman MA, Noah SA, Saad S. Ontology-Based Knowledge Management Tools for Knowledge Sharing in organization—A Review. IEEE Access. 2022 Mar 31;10:43267-83 OR Mohamad Amin Osman,Shahrul Azman Mohd noah,Saidah Saad, "Ontology-Based Knowledge Management Tools for Knowledge Sharing in Organization—A Review," in IEEE Access, Vol. 10, pp. 43267-43283, 2022, Doi: 10.1109/ACCESS.2022.3163758.

[64] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods: Review and analysis." 2002.

[65] Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. Organization Science, 5(1), 14–37.

[66] Thomas Welsh and Elhadj Benkhelifa. 2020. On Resilience in Cloud Computing: A Survey of Techniques across the Cloud Domain. ACM Comput. Surv. 53, 3, Article 59 (May 2021), 36 pages. https://doi.org/10.1145/3388922.

[67] S. Karthikeyan, R. A. Raj, M. V. Cruz, L. Chen, J. L. A. Vishal and V. S. Rohith, "A Systematic Analysis on Raspberry Pi Prototyping: Uses, Challenges, Benefits, and Drawbacks," in IEEE Internet of Things Journal, vol. 10, no. 16, pp. 14397-14417, 15 Aug.15, 2023, doi: 10.1109/JIOT.2023.3262942.

[68] Jörg Christian Kirchhof, Anno Kleiss, Bernhard Rumpe, David Schmalzing, Philipp Schneider, and Andreas Wortmann. 2022. Model-driven Self-adaptive Deployment of Internet of Things Applications with Automated Modification Proposals. ACM Trans. Internet Things 3, 4, Article 30 (November 2022), 30 pages. https://doi.org/10.1145/3549553.

[69] M. Khatibsyarbini, M. A. Isa, D. N. Jawawi, and R. Tumeng. 2017. Test case prioritization approaches in regression testing:A systematic literature review. Info. Softw. Technol. 93, 1, 74–93. DOI:10.1016/j.infsof.2017.08.014.

[70] Hrabovská K, Rossi B, Pitner T. Software testing process models benefits & drawbacks: a systematic literature review. arXiv preprint arXiv:1901.01450. 2019 Jan 5.

[71] Kizilkaya, Burak, et al. "An effective forest fire detection framework using heterogeneous wireless multimedia sensor networks." ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 18.2 (2022): 1-21.

[72] Idrus, Hariaty Mohd. "Tacit knowledge in software testing: A systematic review." In 2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS), pp. 1-6. IEEE, 2019.

[73] S. Capy, E. Coronado, P. Osorio, S. Hagane, D. Deuff and G. Venture, "Integration of a Presence Robot in a Smart Home," 2023 3rd International Conference on Computer, Control and

Robotics (ICCCR), Shanghai, China, 2023, pp. 192-197, doi: 10.1109/ICCCR56747.2023.10194211.

[74] R. Spencer Hallyburton, Nate Zelter, David Hunt, Kristen Angell, and Miroslav Pajic. 2023. A Modular Platform For Collaborative, Distributed Sensor Fusion. In Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023) (ICCPS '23). Association for Computing Machinery, New York, NY, USA, 268–269. https://doi.org/10.1145/3576841.3589630.

[75] P. William, S. Singh, Y. Kumar, A. K. Rao, A. K. Khan and D. Singh, "Software Reliability Analysis with Various Metrics using Ensembling Machine Learning Approach," 2023 World Conference on Communication & Computing (WCONF), RAIPUR, India, 2023, pp. 1-6, doi:10.1109/WCONF58270.2023.10235066.

[76] Mosaad, Nashwa, Khaled A. Elshafey, and Ahmed N. El-Mahdy. "Fault diagnosis of computer numerical control (CNC) machine using cloud knowledge base and monitoring system." Journal of Al-Azhar University Engineering Sector 16.60 (2021): 757-775.