Proceedings of SEML 2025 Symposium: Machine Learning Theory and Applications
DOI: 10.54254/2755-2721/2025.TJ22234

Applications of Modern Software Engineering Methods in Cloud Service Projects - An Example of Charging Piles

Yuyang Xu

Northwest Normal University, School of Computer Science and Engineering, Lanzhou, China
xyy88988@126.com

Abstract: In the environment of the continuous development of the new energy industry, the traditional software engineering methods have been gradually backward in the face of the ever-complex demands, so this paper analyses the successful implementation of the charging pile cloud service project of Company T as an example, and explores how the modern software engineering methods can promote the development efficiency and the development quality of the cloud service project. This paper adopts the case study method and comparative analysis method. It is found that after applying the new software engineering methodology, the development quality and development efficiency of the project are significantly improved, and the final product has high reliability and stability, which meets the current market demand. Compared with the products developed using the traditional methodology, the products developed using the software engineering methodology have gone through the development process of requirements definition and system design, agile development and continuous integration, and have an advantage in the development cycle and development quality. In the subsequent operation and maintenance, the new software engineering approach uses DevOps thinking to incorporate post-operation at the time of development, which allows for continuous updating of the product operation and maintenance, and reduces costs through automated operation and maintenance. This paper supplements the theory of software engineering methodology for cloud service project development and further illustrates the significance of applying the new methodology to software development projects, which can provide reference suggestions for the development of the industry.

Keywords: Software Engineering Methods, Cloud Services, Charging Pile, Project Development

1. Introduction

With the transformation of the global energy structure and the enhancement of environmental protection awareness, the new energy automobile industry is developing rapidly, and from the data, by 2023, the penetration rate of new energy automobiles in Europe had reached 25%, and the penetration rate within China had even reached 31.6%, which proves that the development prospects of the new energy automobile industry in the world are still very broad [1]. At the same time, the development of the digital economy also makes the development of new energy vehicle industry have a new power and new direction [2]. And the construction of charging piles as its important infrastructure is also getting more and more attention. Now many charging pile construction projects use cloud service technology, but in the development and project operation and maintenance,

DOI: 10.54254/2755-2721/2025.TJ22234

problems will continue to appear. The current means of solving the problem are mostly the traditional development methods, which are more cumbersome. The application of software engineering methods in software development projects in the field of people's livelihoods is less researched, and this part of the theory is relatively vacant, resulting in most of the industry still being in the traditional development method mode.

Therefore, this paper discusses the role of the application of modern software engineering methods for the development of cloud service projects through case studies and comparative analyses of Company T's charging pile cloud service project. In addition, comparing with the traditional development mode, it illustrates the practical benefits of applying modern software engineering methods. Therefore, the research in this paper can supplement the theory of the application of software engineering methods and provide reference opinions for the development mode of software development projects for related enterprises.

2. Software Engineering Methods and Current Issues in the Charging Pile Cloud Service Project

2.1. Overview of the Core of the Software Engineering Methodology

The software engineering approach is a theoretical and practical framework that applies a set of theories and practices to guide the development, maintenance and management of software from a systematic, standardized and quantifiable perspective [3]. Its goal is to improve software quality, reduce software development costs, shorten the software development cycle, and be able to meet the user's needs of the software delivered to the user on schedule. Among them, the software development life cycle is the foundational framework of software engineering methodology, which includes the phases from requirements extraction, requirements analysis, design and testing of software until product retirement. Common SDLCs include the waterfall model, iterative model, agile model, spiral model, etc., each of which has a corresponding applicable development environment. For example, the waterfall model based on a linear sequence is suitable for the development of relatively simple projects with well-defined requirements, while the agile model emphasizes the qualities of rapid iteration, feedback from stakeholders and users, and the ability of the development team to work together, and is suitable for the development of projects with frequent changes and complexities.

The requirements part of the development environment mentioned above all comes from requirements engineering in software engineering, which is the basic work of software engineering used to determine the user and system requirements, using a systematic approach and engineering management tools to efficiently develop software requirement specifications that can accurately express the user's needs, and its main phases include requirements acquisition, requirements modelling, formal description of requirements, requirements validation and requirements management [4]. These task segments are interrelated with each other and have feedback, so that when there is a problem in one segment, it can be rolled back to the previous stage and redo it to ensure the solidity of the final requirements.

The software design phase converts requirements into system architecture and module design after clarifying requirements to ensure system scalability and maintainability. The coding and implementation phase turns the design into executable code, which requires focusing on code specification, modular programming, code reuse, code testing, and using iterative development to improve code quality and readability.

The next software testing is a key part of software quality assurance, including unit testing, integration testing, system testing, and acceptance testing, which verifies the correctness of the software functions and the stability of the performance with multi-layer testing. Combining script testing methods and exploratory testing ideas in testing can improve software testing effectiveness

DOI: 10.54254/2755-2721/2025.TJ22234

and efficiency [5]. Software maintenance is the longest process in the software development cycle, including error-correcting maintenance, adaptive maintenance, perfect maintenance and preventive maintenance, to ensure that the software has been running to meet user needs. Project management is an important element of software engineering, which ensures that projects are executed on schedule, on budget, and with quality through project planning, risk management, teamwork, and quality management.

Driven by technology, modern software engineering methods such as Agile development, DevOps, CI/CD are gradually gaining popularity. Agile development uses team-centred development, iterative development, and user feedback; DevOps uses tools to reach integration of development and operation and maintenance; and CI/CD enables rapid release of code to packages, while software engineering methods use a range of tools as a way of increasing the software development efficiency and improving the quality of code development.

2.2. Current Challenges of Charging Pile Cloud Services

At present, with the rapid development of electric vehicles, the charging pile project and the application of cloud computing technology have increasingly become the direction of development. Through the application of cloud computing technology, the charging pile cloud platform system has the roles of remote monitoring, intelligent scheduling, user management and settlement and payment, which greatly enhances the use of the charging pile cloud platform, but there are more problems and challenges in the use of the charging pile cloud platform system, which need to be effectively dealt with through software engineering technology means and methods.

First of all, in charging management, despite the rapid development of the new energy vehicle industry in recent years, the management of electric vehicle charging piles still exists more problems, mainly including imperfect planning layout, low coverage of charging pile facilities in villages, immature management services, etc. In addition to the lack of effective integrated management of the large influx of private charging piles, which increases the management cost at the same time and leads to a high proportion of charging pile idle rate [6], resulting in the difficulties of vehicle owners charging and operators' profitability, which causes the dilemma of difficult charging for vehicle owners and difficult profitability for operators [6]. Secondly, in terms of charging, the charging system of the charging pile cloud service platform does not have uniform and clear regulations, which is easy to cause user dissatisfaction. For example, some platforms have the situation of "unclear price" and "unbalanced charges", which leads to high charges when users actually use them. Unreasonable and transparent charges will cause unnecessary trouble for users. In the face of a continuous demand, many project operators due to the use of traditional development model makes the demand can’t be met in a timely manner, the loss of market competitiveness, how to quickly solve the ever-changing and updated market needs, and to make the corresponding updates and improvements to obtain the trust of users, is the majority of charging pile cloud services project developers face problems. Therefore, it is extremely important to use modern software engineering methods to improve the development efficiency and product quality of the project.

3. Application of Modern Software Engineering Methods in Charging Pile Cloud Services

3.1. Requirements Acquisition and System Design

Requirements engineering provides the basis for the research and development of the charging pile cloud service project, and the process of obtaining the final requirements is essentially the process of determining user requirements and business objectives. The requirements in the charging pile cloud service project mainly include user requirements, business requirements and technical requirements. User requirements include functional requirements such as user charging reservation, timely feedback

DOI: 10.54254/2755-2721/2025.TJ22234

of settlement bill and fault repair, and should also include the needs of differences between elderly users and ordinary users. Business needs include charging pile operation and management needs, data analysis needs, dynamic pricing strategy needs, etc. The determination of these needs determines the business value of the charging pile cloud service project. Technical requirements include high concurrency processing, data storage and visualization, system scalability, etc. These requirements help to improve the stability and scalability of the system. A clear and traceable specification of software requirements can be obtained using the engineering requirements acquisition approach in requirements engineering. At the same time, combined with the agile requirements engineering approach, so that the requirements acquisition is not bound to the beginning, but in the whole development process of continuous iteration [7], which helps to achieve a dynamic response to the demand, the final charging pile cloud service products for the market demand for the response to the guarantee.

In order to handle high concurrency periods and for modular deployment, microservices architecture can be used. The use of this architecture can decouple the functional modules and achieve system scalability and continuous updating [8]. The system data structure design can also help the project team to accurately and accurately realize the specific functional design, database design for the current urgent need of information, in order to effectively solve the efficient storage of data, querying and other design requirements, and further improve the security of the database design.

3.2. Agile Development and Continuous Integration

Agile development applied to software projects is a software project development mode in the form of team collaboration to rapidly iterate, respond to user feedback, and continuously accumulate and improve system functions to obtain rapid delivery. The specific agile development practices in the charging pile cloud service project are user stories, iterative development, and team collaboration. Among them, user stories are concise descriptions of requirements and representations from the user's perspective, which mainly contain descriptions, values, and acceptance criteria [9]. The combination of using user stories can make the requirements clearer. Iterative development is the process of slicing the development cycle into shorter iteration cycles, implementing a small portion of functionality per iteration cycle, making adjustments through user feedback, and delivering a runnable incremental product at the end of the iteration cycle [10]. Team collaboration allows the development team to communicate and coordinate efficiently through daily station meetings and iteration review meetings.

Continuous integration is a development model that supports agile development and uses automated tools to complete automatic construction, automated testing and deployment. The use of continuous integration in the development of the charging pile cloud service project can avoid the trouble caused by the integration of code spacing and improve the efficiency of the project development work and code quality, for example, through the submission of the code when the automated testing tools for unit testing, and integration testing are used, to ensure that the project does not cause errors when the code is submitted. The use of continuous integration to support agile projects can ensure real-time response to demand and provide high-quality software.

3.3. DevOps in Operations and Optimization

The idea of DevOps ties development and operations together, which can lead to automation and high efficiency in software delivery and operations. In the charging pile cloud service project, a DevOps platform based on container technology and microservice architecture can be used, which enables automated testing, monitoring and log management, and continuous optimization [11]. The charging pile cloud service uses microservice architecture, through the method of splitting modules. While solving the problem of high concurrency, the scalability of the system is preserved; monitoring and

DOI: 10.54254/2755-2721/2025.TJ22234

log management are done through the use of monitoring tools and log analysis tools to quickly locate and solve the problem, and then through the monitoring information and the user's use of the feedback, the performance of the system is constantly optimized and the use of the experience is then followed by the automation of deployment testing to achieve continuous optimization.

4. Charging Pile Cloud Service Project Case Study

The successful implementation of the charging pile cloud service project of Company T is actually one of the values of the application of modern software engineering methodology practice in the charging pile project. By decomposing the project in accordance with the software engineering methodology, sorting out and integrating the key functional systems of the charging pile itself, the charging pile cloud platform, the charging pile APP, the charging pile and the power grid, and utilizing the systematic and synergistic way of working with the software, the project has carried out a successful practical exploration of software engineering construction. Next, the key links and steps in the software engineering life cycle are taken as the entry point for specific analyses.

For this project, the precise definition of requirements was the basis for success. When defining the requirements, the overall requirements of improving the utilization of charging piles, achieving load balancing on the grid and enhancing user experience were identified at the requirements engineering level through requirements acquisition and analysis. From the top down, a hierarchical requirements model was constructed, with the ultimate goal to be achieved as the clue. For example, the user study shows that "displaying real-time tariff information" is the most important concern of users, and this requirement can be decomposed into "real-time update of tariff information" and "timely delivery", which can then be taken as the most important requirement of the app. In this way, the software can be designed to meet the customer's needs. In addition, in requirements engineering, we verify the credibility of the requirements through backward-looking methods such as prototype review, user acceptance, and iteration to further improve the fit between the requirements and the product. We also pay constant attention to key requirements during the development process and adopt iterative thinking to respond to requirements dynamically. In the design, the system is designed using micro-service architecture and modular design thinking, and the equipment access module, data processing module, intelligent scheduling module, fault diagnosis module, etc., are designed separately, and each module is loosely coupled with a standard interface. For example, the charging pile equipment access module adopts the Internet of Things protocol to access the cloud platform, and future network upgrades will not have an impact on the current charging pile equipment access module.

During the development of the project, the agile development model and iteration cycle were determined, and the core functions were quickly implemented through extreme programming and rapid iteration. The TDD development model was applied to describe the test scenarios before coding, and compared with the traditional testing methods, TDD can ensure higher test coverage as a way to ensure the correctness of the algorithms, and to reduce the procedural errors due to the addition of new functions. The use of automated testing tools to cover most interface tests reduces the defect rate even further, while eliminating human error in manual testing and improving the reliability and consistency of test results. Continuously responding to user needs through continuous integration as a way to gain greater market influence.

DevOps ideas into it, that is, in the development and operation of the line of work together with the consideration of automation and predictive maintenance, such as the use of Jenkins software to do continuous integration, each submission of the code through Jenkins to complete the cloud platform deployment; the use of Prometheus monitoring software, the charging pile of the on-line situation in a timely manner to detect the discovery; based on the history of the maintenance work orders information for the maintenance time prediction, do a good job of predictive maintenance,

DOI: 10.54254/2755-2721/2025.TJ22234

replace parts in advance and other measures. In the decision-making process, the construction of a data warehouse integrates charging record data, grid load data, user behavior and other multi-source data, and uses Tableau and other carriers to form visual reports. For example, through the regional charging peak data and charging flow data, etc. to carry out the tariff separation strategy, scientific charging of electricity, and reduce the modification of the car caused by the problem of electricity prices.

5. Comparative Analysis of Modern Engineering Methods and Traditional Methods

The traditional approach development model has progressed with the times, and due to its inflexibility towards software development, it has become difficult to face the future trend of changing requirements. In contrast, modern software engineering methods can optimize the development process to get a product that is more in line with the actual requirements. The life cycle of software developed using traditional development methods, i.e. using the most basic waterfall model, is collected here for comparison with the expected results if modern software engineering development methods are used. The details are shown in Table 1.

Table 1: Comparison of the benefits of the two software engineering approach models

dimension (math.)	Expected effectiveness of the application of software engineering methods	Shortcomings of the traditional model
demand response	Adopting iterative development thinking, dynamically adjusting key requirements so that the product meets the user's needs	Demand only comes from the demand acquisition stage at the beginning, leading to rigid demand that is easily eliminated from the market.
development capability	Agile short iteration cycles, ability to respond quickly to the market, high software flexibility and fewer defects	Waterfall models have long delivery cycles and missed market opportunities.
O&M costs	Adopting the DevOps mindset combines future operations at the development stage, making development and operations no longer two separate parts of the process, and subsequently reducing labor costs through automated operations and maintenance.	Development and operation are less related to each other, and operation mainly relies on manual on-site inspection, with high operation and maintenance costs and time-consuming and labor-intensive.
System Reliability	Using TDD development model, high test coverage, small failure rate.	The traditional testing methodology has less test coverage, more hidden problems, and frequent downtime affecting services.
Data Decision Making	The database is included in the development and the data is integrated into visual reports, which can make decision-making more scientific.	(a) Data are homogeneous and do not lead to statements that can support decision-making, and empirical decisions lead to unrealistic program implementation.

DOI: 10.54254/2755-2721/2025.TJ22234

6. Conclusion

As intelligence and complexity become the new normal of project development, traditional development methods struggle to cope with projects that require rapid iteration and complex requirements. In this context, modern software engineering methodology builds an upgrade solution for charging pile cloud service projects throughout the software lifecycle through the deep integration of systematic development processes and intelligent tools.

The implementation of this methodology covers the core stages of requirements analysis, system design, programming development, and later operation and maintenance, and overall achieves the purpose of improving project benefits and reducing costs. Through the introduction of the TDD development model, automated testing tools, etc., the project team not only significantly improved the development efficiency, but also realized the real-time dynamic control of system stability indicators. In addition, the continuous development of cloud computing technology enables software development to break through the performance bottleneck of traditional methods, so that modern software engineering methods are more suitable for the ever-increasing needs of cloud service projects.

According to the current status of the charging pile cloud service application, although this kind of emerging methodology has not yet been popularized on a large scale in cloud service projects in the field of people's livelihood, its technological advantages have already begun to emerge. Through comparative analysis, it can be found that projects adopting modern engineering methodologies have gained considerable improvement in demand response speed, product fault tolerance and other key indicators compared with traditional models. And after combining DevOps ideas, the later operation also achieves the purpose of cost reduction and efficiency.

With the continuous development of emerging technologies and the continuous practice of software engineering methods in the future, software engineering methods will intersect with more different fields, and the various models that are currently more separate will gradually be integrated to form a set or several sets of more mature methodologies. The cross-domain synergy of software engineering methods will provide strong support for future smart city infrastructure construction and become a better choice for complex system development.

References

[1] *Zilong Qin. (2024) Analysis of the current situation of the development of new energy automobile industry in China and its countermeasures. Modern Industrial Economy and Informationization,14:131-143.*

[2] *Ze Tian, Zixian Liu, Yangjun Ren. (2025) Mechanism research on digital economy empowering new energy automobile industry high-quality development. Journal of Industrial Technological Economics, 2:67-77.*

[3] *Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: a Practitioner's Approach (9th ed.). McGraw-Hill Education.*

[4] *Huaxiao Liu. (2014) Research on Some Lssues in Requirements Engineering.4. (Dissertation).*

[5] *Jingshan Kang, Yong Han, Yingbo Duan. (2024) Research on Software Testing Methods. Electronic Product Reliability and Environmental Testing, 5:76 -82.*

[6] *Pingping Fu, Fuyong Zheng, Hua Wang, Min Li, Shi Yu. (2022) Development of Intelligent Charging Pile Management System and Mobile Application Based on Cloud Platform. Microcomputer Applications, 6:180-183.*

[7] *Haoyu Wang. (2022) Review on Technologies of Requirement Engineering of Software. Computer Science, S2:766-779.*

[8] *Miao Guo. (2024) DevOps platform design based on containers and microservices [J]. Computer Knowledge and Technology, 2024, 20(27):52-55. DOI:10.14004/j.cnki.ckt.2024.1393.*

[9] *Ru An. (2024). Agile Development Methodology and Project Management Practice in Software Technology. Changjiang Information & Communications, 10:156-158.*

[10] *Xiumin Hua. (2022). Research on quality management of internet projects based on agile development model. Electronic Communication & Computer Science, 4(1).*

[11] *Zhi Jin, Lin Liu, Xiaohong Chen, Tong Li. (2023) Software Requirements Engineering Methodology and Practice, Tsinghua University Press, Beijing.*