
Selection of Feature Driven Development (FDD) Model in Agile Method for Developing Information System of Mosque Management

Sasmitoh Rahmad Riady^{1)*}, Khalis Sofi²⁾, Jafar Shadiq³⁾, Rita Wahyuni Arifin⁴⁾

¹⁾³⁾⁴⁾Universitas Bina Insani, Indonesia, ²⁾Universitas Pelita Bangsa, Indonesia

¹⁾sasmitoh@binainsani.ac.id, ²⁾sofikhalis7@gmail.com, ³⁾jafarshadiq@binainsani.ac.id,

⁴⁾ritawahyuni@binainsani.ac.id

ABSTRACT

This paper discusses software development for building a project using the Feature Driven Development (FDD) model contained in the agile method. As in other agile methods, the Feature Driven Development model has additional properties to implement functions and needs in a short iteration. In terms of the characteristics contained in several other agile models, almost all of them have similarities, which makes the stakeholder confused in determining which model will be chosen as a software development method. This study focuses on the search for a number of suitable project specifications in the selection of Feature Driven Development models for software development. From the research that has been done in several papers, several aspects of design and construction in software development emphasize the quality and high level of features in developing software using the Feature Driven Development model. In facilitating the results of the Feature Driven Development model, we will provide a case study of the Mosque Management Information System which has several features, such as content management, information on prayer time, online reading Qur'an, and Petty Cash Mosque management. The Feature Driven Development model implements this case with a short iteration because this project had done in several months.

Keywords: Software Development, Agile Method, Featured Driven Development, Information System, Mosque Management Information System

INTRODUCTION

Agile models in software development have iterative characteristics and additional models through which requirements and solutions develop through consistent collaboration between customers and all other stakeholders (Saravanan, 2020). Four core features of software agile software development were stated in the agile manifesto, (1) individual and interaction over process, (2) working with software through documentation, (3) responding to changes after contact with the customer, (4) responding to changes after planning. In the agile method, there are several models, including XP, SCRUM, FDD, LSD, and AUP (Albers, Stief, Dantan, Etienne, & Siadat, 2021).

At the heart of Agile software development (Paluch et al., 2019) are the two main ideas that are very important to provide value in the form of software that works for clients and be able to respond to changes that might occur during the development in full market innovation and always changing causes must be built with various software development methods. In this case, it violates the traditional notion of the statement "planning work, and doing work that has been planned" and turns into a recurring approach to adaptive planning. The agile method has many core elements, which is a definition of value that the sponsor or client can recognize. Extreme Programming (XP) has a "client story", while FDD has a "feature" (Damodharan, 2020).

Peter Coad first introduced FDD in 1997. The Coad method focuses on object technology based on IT business. The Coad Method evolved into FDD, where many new ideas were introduced. The feature-based outline was published in 1999 in a book by Peter Coad and Jeff De Luca concerning "the java model with UML color" (Setyautami, Rubiantoro, & Azurat, 2019). FDD is an agile software development method that prioritizes two stages. The initial step of the FDD process is determining the features to implement. The next step is the most important stage of the feature-based FDD model, which is the stage of responding to changes in the features of the client, and the developers must immediately act on requests from the client, which is one of the advantages of the FDD model. In the level of accuracy for project tracking and maintenance, for the extensibility of the project code, especially in determining the quality of the work carried out in the first stage of building a piece of software (Nawaz, 2021).

In the FDD model, all development leads to repetitive development with best practices for finding effectiveness

* Corresponding author



inside software development in the industrial field. This stage emphasizes the quality aspects of the whole process and includes delivery in the actual process; along with that, the progress in the project will always be monitored. In the FDD process, there are five workflows and their method (Tashtoush et al., 2022), techniques and guidelines needed for the owner class in running a system. The next covers the purpose and role objectives needed in building a project. FDD method is very suitable for developing a system that is very critical in different features with models that are in the agile method (Šimičková, Bugarová, & Mošková, 2021).

LITERATURE REVIEW

Related Work

In recent years many researchers have discussed matters relating to our research themes. Here we list several FDD-adjusted studies to reduce problems identified or integrated with other software process models to combine both models' strengths and eliminate weaknesses. The following is a summary of the related work in table 1.

Table 1. Summary Related Work

Discussion of the title	Summary
This title discusses the Hybrid agile model using SCRUM and FDD (Cherkaoui, Guerss, Marzak, & Sael, 2021) FDMD: Feature-Driven Methodology Development	CDD is an integration of XP and FDRD; CDD is a generation of requirements model that is in its form in developing software for customers and is targeted globally. The proposed method is one way to overcome the problems that often occur with changing requirements. FDM is an evaluation of the FDD method, where FDD uses the concept of object-oriented. The solution only focused on maintainability and reusability however did not address them.
This title discusses SFDD for safe software security (Cherkaoui et al., 2021)	Proposed projects that use FDD are improved for secure software development. The SFDD method is renewed by adding two phases and the role of security. The proposed model is only focused on security related.
This title discusses web-based information system (Sofi, Sunge, & Riady, 2020)	The web-based information system can use several type methods, perhaps FDD model.
This title discusses the development of a secure website using FDD using case studies (Yakovyna & Symets, 2020)	This research, researchers conducted in this study examines the suitability of developing software tools for a website. No proper and clear guidance is provided to customize the FDD for secure development.

Difference Between Agile and Traditional Models

The most noticeable difference between the agile model and the conventional method is the method that previously could produce results quickly and cheaply for a complex project with unclear requirements. The agile method emphasizes teamwork, building software that works, collaborating with customers, and responding to changes requested by customers, while the conventional method emphasizes contracts, plans, processes, documents and tools. The following is a table that illustrates the differences between agile models and traditional models .

Table 2. Deference Between Agile and Traditional

Method/Criteria	Model Traditional	Model Agile
requirements for users	user requirements must be detailed before implementation	Iterative
Development direction	Fixed	Readily changeable

* Corresponding author



Method/Criteria	Model Traditional	Model Agile
Rework cost	High	Low
Customer involvement	Low	High
checking in testing	done after coding is complete	done every iteration
scale in the selection of suitable projects	Large scaled	Low to medium scale
high-quality skills required for members of the developer	the selection of workers is not required with special expertise	need to have special expertise and knowledge in doing business

Source: SDLC Cycle AGILE vs Traditional Approaches (Leau, Loo, Tham, & Tan, 2014)

Comparative Study

The following are the results of a comparative study of a literature review that answers some questions from which model will be chosen to build a project? This question will raise another question: What is the agile method? There are several methods that differ from one to the other (Alsaqqa, Sawalha, & Abdel-nabi, 2020), so the development team must learn and compare all the methods that are in the agile method (Edison, Wang, & Conboy, 2021).

Table 3. Comparative Study

Criteria	Scrum	XP	DSDM	FDD	CRYSTAL	ASD
Time Size (Month)	5-9	2-10	2-10	4-20	All	Small to Large
Project Size	All	Small	All	Large	All	Small
Iteration Length (Week)	4	2	-	2	Project specific	4-8
Roles and Responsibilities	v	v	v	v	x	x
Process centric	x	x	v	v	x	x
People Centric	v	v	x	x	x	x
Virtual Team Support	v	x	x	v	v	v
High Risk	v	x	v	x	v	v
Documentation	basic	basic	exist	important	important	Basic
Daily Meeting	v	v	x	x	x	v
Information Sharing	x	x	v	v	x	x
Meeting Face-to-Face	x	x	x	x	v	x
Model Based	x	x	x	x	x	x

x: Not Defined

v: Defined

-: Not Specified

Source: A Comparative Study of Agile Methods (Merzouk, Elhadi, Ennaji, Marzak, & Sael, 2017)

METHOD

In determining a method in the project that is suitable for what we will use, in this case, study how we get an FDD method for the project; we will work on it as follows.

* Corresponding author



Specification of FDD

From the results of several literature reviews, there are several specifications for developing software that is suitable in the selection of FDD models for a project, including them: (1) A project that has many features and requires a high level, (2) need work that is quickly completed even though the project is high level, (3) has a high level of change from client requests due to the many features in a project, (4) requires several teams to work on the project to be divided into several features, (5) the many features that will be developed, it takes several teams to work on a project., (6) the need for documentation from the client and the developer, (7) in the development of software that is high profile level it takes workers who have high skills, in the work of developers, not daily meeting is needed, (8) requires iteration lengths range once every week in working on features, (9) requires sharing information from each team working on features that are being built.

FDD Model

In software development at this time, the FDD model is very often used due to a large number of software features; the cluster method provides problem-solving for software development, one of which is the FDD model that focuses on development projects with many features. FDD also has five processes in process, including the one in Figure.

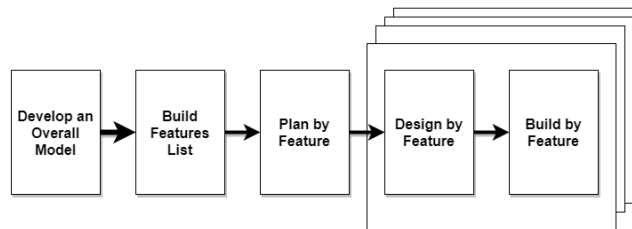


Figure 1: Process FDD

Source: a tool for supporters in feature-based development (Rychlý, 2007).

The explanation of the Figure above is (1) Develop an Overall Model, determine the model for developing a system, (2) Build Features List, create a feature list which has in the menu, module, etc., (3) Plan by Feature, create planning for the future Feature when it has a developing, (4) Design by feature, create a design and that Feature, (5) Build by Feature, release the feature based of the design.

Case Study

Hidayatutholibin Mosque is one of mosque places in Cijambe, Sukadami Village, South Cikarang, and Bekasi. As a regional mosque that uses around society, it can accommodate jama'ah around more or less 500 jama'ah. To improve the mosque's services, the building is under construction to expand the mosque's capacity. Currently, it has a problem in management, such as data of officers, real-time prayer time, online reading Qur'an and the petty cash has not managed transparently and sometimes it has an error calculating. So, we proposed a system to help the mosque management manage the petty cash. We build the project using the FDD model's terms and conditions as described in Table 3. We illustrate the scope of work for role and responsibility in Table 4 below and develop the project for six months.

Table 4
Development Team and Client

The Development Team	The Client Domain Expert
SAS (PM). A project manager who is quite experienced	TH, the Head of Mosque Management. He is responsible for managing all of the activities of Mosque, such as Petty Cash, the schedule of the prayer leader, and many more.
JFR. He is a professional and experienced senior developer	ADE the Finance Officer. He is responsible of petty cash management.
KHS as a Business Analyst who analyzes	BYG as a General Affair who is responsible for

* Corresponding author



The Development Team	The Client Domain Expert
the project rule process	managing all of the Mosque Management processes
RTA has a role as a General Administration, who is responsible for managing all documentation	RHM as a secretary of Mosque Management who responds to all documentation.

Why Use FDD Models?

1. Communication is very important for developers and clients because it has the potential for the success of the project connected that has a communication channel; communication also has the potential to increase dramatically so that more and more will be added.
2. Complexity in FDD explains all the problematic domains, including small problems that can be solved quickly, usually, if this problem reaches two weeks, the problem will get worse, and if not resolved, then this problem will damage communication between teams. Even within this complexity, the FDD model divides the project into several iterations so that the time gap between program analysis and testing of errors can be reduced, including reducing the cost of improvements in software development.
3. In the concept of FDD, the quality is expanded so that it is not just to test the code but also includes things that encode coding standards, measurements for audits and others.

RESULT

The information system of mosque management for Hidayatutholibin Mosque is built web-based with various menus to support the management to control the activities. The menus are the main page for managing the updated content, prayer time, online reading Qur'an, petty cash management, and the profile of all officers.

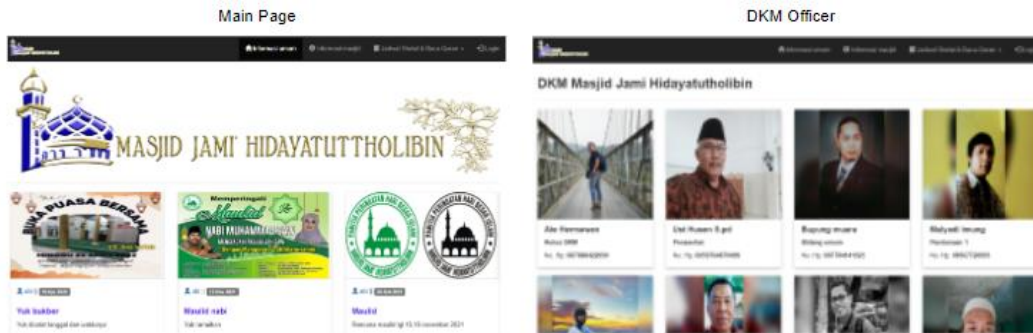


Figure 2: Main Page and DKM Officer

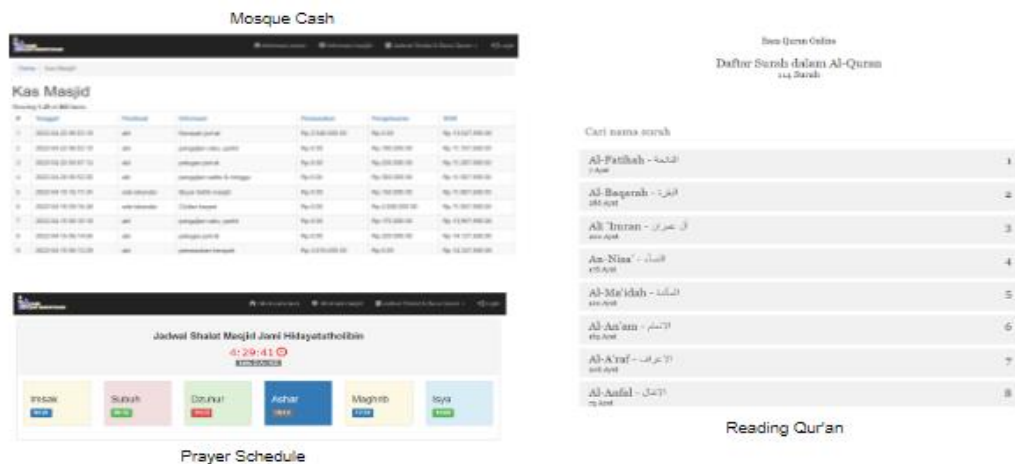


Figure 3: Petty Cash, Prayer Time, Reading Qur'an

* Corresponding author



The figure below 4 is a view of the backend for the officer who manages the activities. This menu contains Kas Masjid, managed by the DKM masjid, such as debit, credit, and how much the balance exists.

#	Tanggal	Pembuat	Informasi	Pemasukan	Pengeluaran	SISA
1	2022-04-22 06:03:18	abi	Keropak jum'at	Rp 2,540,000.00	Rp 0.00	Rp 13,647,590.00
2	2022-04-22 06:02:10	abi	pengajian rabu, parkir	Rp 0.00	Rp 180,000.00	Rp 11,107,590.00
3	2022-04-22 05:57:14	abi	petugas jum'at	Rp 0.00	Rp 220,000.00	Rp 11,287,590.00

Figure 4: Management Mosque Cash

DISCUSSIONS

From the explanation above, some things that should be discussed in the project are implementing the FDD model, utilizing the online source of Qur'an and prayer time for mosque management, and implementing petty cash management.

FDD Process

this discussion explains how the process of FDD contained in Figure 1. follows the steps

1. Develop the Overall Model in the development team, all members must work together under the command of the chief architect, and the other members must start the work that starts according to the scope that has been divided.

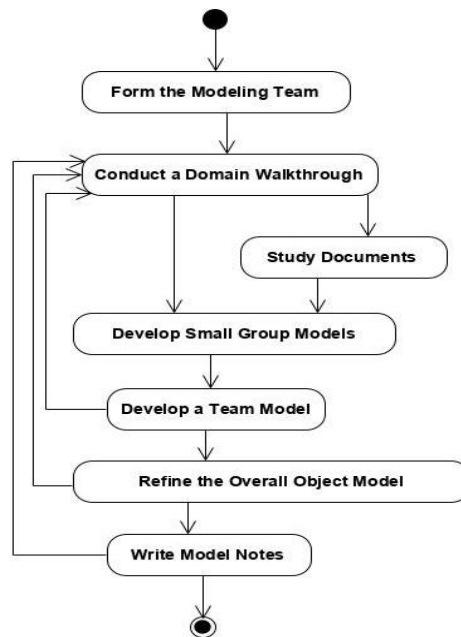


Figure 5: Diagram devolving overall model

* Corresponding author



2. Build a Feature List A team which usually consists of only the Programmer Chair and in the process of one is formed to describe each of their functions. Based on the domain partition by Domain Expert in process 1, the team split the domain into several areas (Set of main features). Each area is further divided into several activities (feature sets). steps in several processes to change to a feature list. how to make a list of features so that each team is related to one another?

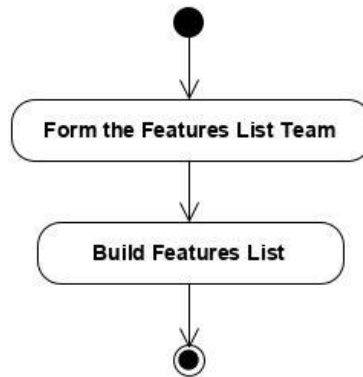


Figure 6: Flow Diagram Building

3. Design by Feature Some features have been scheduled for development by assigning the head programmers. Chief Programmer chooses features to develop from a list that has been assigned.

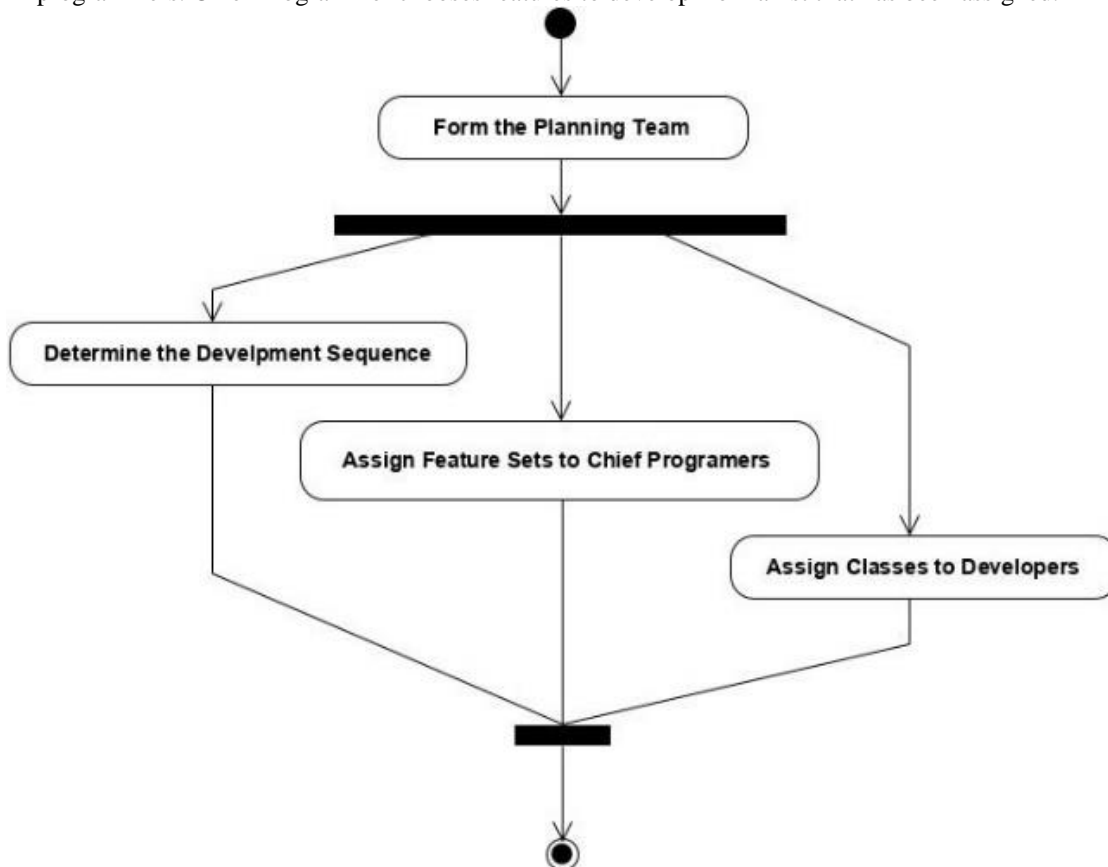


Figure 7: Diagram Plan by Feature

* Corresponding author



4. Design by Feature Some features have been scheduled for development by assigning the head programmers. Chief Programmer chooses features to develop from a list that has been assigned.

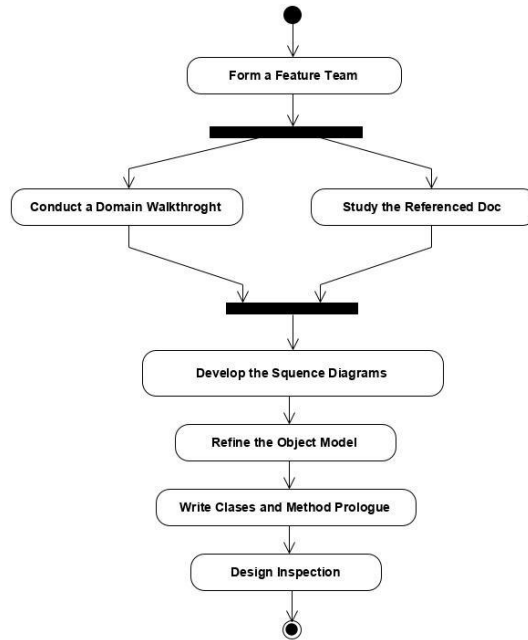


Figure 8: Diagram Design by Feature

5. Build by Feature A design released in the design by feature process will then be called a package. In this process, even the class owner participates in the formation of packages which will be released into a feature; after the process is finished, there will be testing and checking for the program code, and then it will be given to the head of the programmer to be determined.

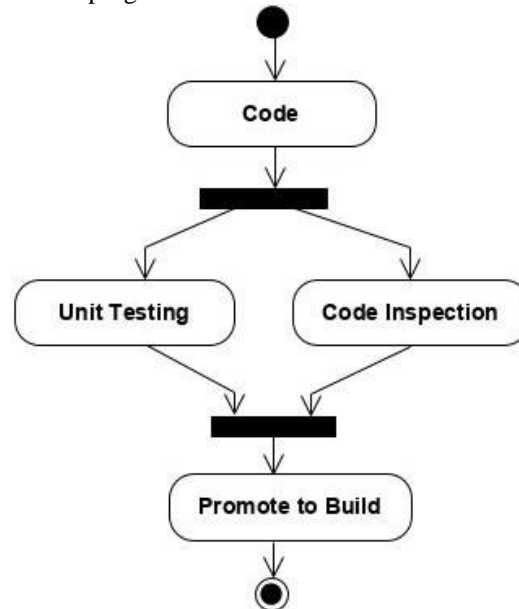


Figure 9: Flow Diagram Build by Feature

* Corresponding author



GET API

For getting API data of qur'an and real time prayer time using API technique, where the source of Fathimah Bot API for prayer time and the quran source of quran.kemenag.go.id. Figure 10 below describes the architecture of getting data on prayer time and the Qur'an.

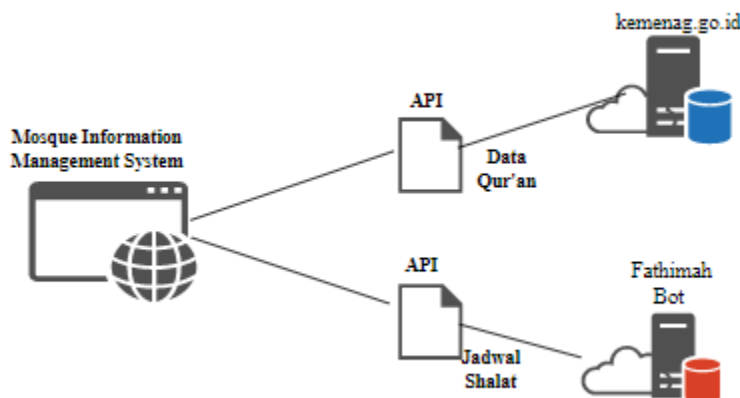


Figure 10: Get data API architecture

CONCLUSION

Software development with high profile level characteristics is very suitable to use the FDD model because FDD has advantages in handling each Feature. From the research that has been done, the researcher presents the appropriate specifications for selecting software development methods using the FDD model. It is expected that from this research, the developers will more easily determine what models are in the agile method based on project specifications and the developers' capabilities. The developers will not be confused in determining a software development model in the agile method. FDD model suit for developing a system with has many features; it is different from the scrum and extreme programming.

REFERENCES

- Albers, A., Stief, P., Dantan, J., Etienne, A., & Siadat, A. (2021). ScienceDirect ScienceDirect ScienceDirect Agile development : a a and Agile p p roduct roduct d d evelopment : a n n a a nalysis nalysis of a cceptance cceptance and a a dded dded v v alue alue May of in in p p ractice ractice A new methodology to analyze the functional and physical architecture of Kaiser product for an assembly oriented family identification. *Procedia CIRP*, 100, 768–773. <https://doi.org/10.1016/j.procir.2021.05.046>
- Alsaqqa, S., Sawalha, S., & Abdel-nabi, H. (2020). *Agile Software Development : Methodologies and Trends Agile Software Development : Methodologies and Trends*. (July). <https://doi.org/10.3991/ijim.v14i11.13269>
- Cherkaoui, A., Guerss, F., Marzak, A., & Sael, N. (2021). ScienceDirect ScienceDirect Smart Irrigation : case study for Hybrid-SIX methodology Smart Irrigation : case study for Hybrid-SIX methodology. *Procedia Computer Science*, 191, 524–529. <https://doi.org/10.1016/j.procs.2021.07.077>
- Damodharan, S. (2020). *Feature Driven Agile Product Innovation Management Framework*. 7, 5–9.
- Edison, H., Wang, X., & Conboy, K. (2021). *Comparing Methods for Large-Scale Agile Software Development : A Systematic Literature Review*. <https://doi.org/10.1109/TSE.2021.3069039>
- Leau, Y., Loo, W. K., Tham, W. Y., & Tan, S. F. (2014). *Software Development Life Cycle AGILE vs Traditional Approaches*. (November).
- Merzouk, S., Elhadi, S., Ennaji, H., Marzak, A., & Sael, N. (2017). A Comparative Study of Agile Methods: Towards a New Model-based Method. *International Journal of Web Applications*, 9(4), 121–128.
- Nawaz, Z. (2021). *Proposal of Enhanced FDD Process Model*. (August). <https://doi.org/10.5815/ijeme.2021.04.05>
- Paluch, S., Antons, D., Paluch, S., Antons, D., Brettel, M., Hopp, C., ... Wentzel, D. (2019). Stage-gate and agile development in the digital age : Promises , perils , and boundary conditions Stage-gate and agile development in the digital age : Promises , perils , and boundary conditions. *Journal of Business Research*, (March), 0–1. <https://doi.org/10.1016/j.jbusres.2019.01.063>

* Corresponding author



-
- Rychlý, M. (2007). *A Tool for Supporting Feature-Driven Development*.
- Saravanan, T. (2020). *Comparative Analysis of Software Life Cycle Models*. 906–909.
- Setyautami, M. R. A., Rubiantoro, R. R., & Azurat, A. (2019). Model-Driven Engineering for Delta-Oriented Software Product Lines. *2019 26th Asia-Pacific Software Engineering Conference (APSEC)*, 371–377. <https://doi.org/10.1109/APSEC48747.2019.00057>
- Šimíčková, J., Buganová, K., & Mošková, E. (2021). ScienceDirect ScienceDirect Specifics of the Agile Approach and Methods in Project Specifics of the Agile Approach and Methods in Project Management and its Use in Transport Management and its Use in Transport. *Transportation Research Procedia*, 55, 1436–1443. <https://doi.org/10.1016/j.trpro.2021.07.130>
- Sofi, K., Sunge, A. S., & Riady, S. R. (2020). *The Implementation of Cost Control System Based on Website in PPIC Department*. 1(2), 69–80.
- Tashtoush, Y. M., Darweesh, D. A., Husari, G., Darwish, O. A., Darwish, Y., Issa, L. B., & Ashqar, H. I. (2022). Agile Approaches for Cybersecurity Systems , IoT and Intelligent Transportation. *IEEE Access*, 10, 1360–1375. <https://doi.org/10.1109/ACCESS.2021.3136861>
- Yakovyna, V., & Symets, I. (2020). *The Relation between Software Development Methodologies and Factors Affecting Software Reliability*. 23–26.

* Corresponding author



This is an Creative Commons License This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0).