Using Multi-Attribute Decision Making to Define, Prioritize and Manage Software Requirements[1, 2]

Haoyu Wang

ABSTRACT

In the development process of software projects, requirement management runs through the entire life cycle of software projects. In software project management, requirements engineering is the first step in software development, which is a key step and the most difficult step. The quality in requirement management directly affects the quality of software and even the success or failure of software projects. This determines that the project team must have a requirement management strategy and effective implementation.

This paper uses multi-attribute decision making to define and prioritize several behavioral development models that manage software requirements and will help users understand the most critical criteria they should choose (from the customer's point of view), the advantages and disadvantages of various methods in different aspects. This should allow a team to achieve barrier-free communication, error-free delivery.

Keywords: Specification by Example (SBE), Requirements, information technology, Software, System product, agile, behavioral development models

INTRODUCTION

As IT systems become more and more critical competitive factors in the industry, the scale of the IT project is more and more prominent, involves the organization level is higher and higher, the risk of the project is becoming more and bigger. As one of the fastest growing areas in the world today, the failure of the project also represents a large amount of capital loss and time cost and the waste of labor costs. According to the PMI industry statistics, '*the average waste of IT*

[1] Editor's note: Student papers are authored by graduate or undergraduate students based on coursework at accredited universities or training programs. This paper was prepared for the course "International Contract Management" facilitated by Dr Paul D. Giammalvo of PT Mitratata Citragraha, Jakarta, Indonesia as an Adjunct Professor under contract to SKEMA Business School for the program Master of Science in Project and Programme Management and Business Development. http://www.skema.edu/programmes/masters-of-science. For more information on this global program (Lille and Paris in France; Belo Horizonte in Brazil), contact Dr Paul Gardiner, Global Programme Director, at paul.gardiner@skema.edu.

[2] How to cite this paper: Wang, H. (2018). Using Multi-Attribute Decision Making to Define, Prioritize and Manage Software Requirements, *PM World Journal*, Vol. VII, Issue XII (December).

industry project expenditures in 2017 was as high as $78 million per $1 billion. Among them, the average waste was the highest regarding geographical distribution and European project expenditure, and $131 million was wasted for every $1 billion.'[3]

Then in the field, there are many examples of companies being dragged down by the IT project or the returns were much lower than expected. The data in Figure 1 is the result of the IT project in earlier years. We can see that only 39% of IT projects were successful in that year. This figure means that more than 50% of the IT projects did not achieve the expected results.

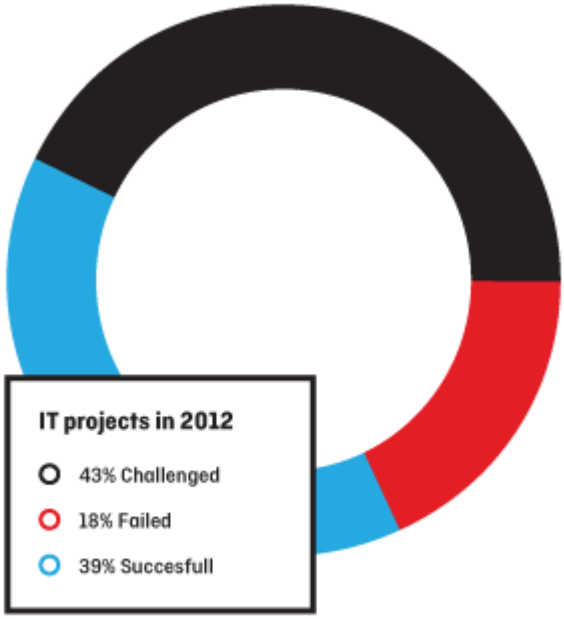

Figure 1: Software development project results[4]

Then the software that has been successfully developed must be no problem, without the doub t? However, it is not. According to statistics, there are still many customers not satisfied with th e final delivery software. And after customers use the software for two months, the percentage of dissatisfaction can double. This figure is a horrifying number and lets us take a look at the sit uation of industry practitioners. In the project development work, many programmers have not understood the target results. With this requirement, people can start development. However, the reality is that people do not know what to do, but they continue to develop. When the proj ect is postponed, or there is a problem with the project test, time and money have been lost, a nd Inconsistent with the original intention.

[3] Pulse of the Profession | Project Management Institute. (November 11). Retrieved from https://www.pmi.org/Pulse

[4] Sweeney, M. (2015). 3 Reasons Why Software Development Projects Fail - Clearcode Blog. Retrieved from https://clearcode.cc/blog/why-software-development-projects-fail/

The technology is advanced, the team is strong, and the products have a market. However, it sti ll failed because – "delivered the wrong software." However, for a project that delivers a piece o f software, "delivering the wrong software" is often the origin of the dispute. However, it is also a difficult problem to solve. For the owner, he is paying for a "right software." However, there a re often hardships for the developers who are entrusted, because they are given instructions to "act the software according to the precise functional description of the owner." "Correct or not " is not their ultimate responsibility. Whether or not "correct" usually has to wait until after the online, the customer can know according to the user's actual measurement response (although the developer is often the development stage can usually guess the group of people who failed). However, this is never within the responsibility of the contract.

STEP 1: Problem Definition

How did this happen, why did it become like this? Let us have a look at some of the features of IT project to help us sort out the problems.

In the IT project, the following points are different from the general project:

1) **Time urgency**. This aspect is especially important for software because the timing of the market is fleeting. Before starting a project, it is necessary to clarify the time constraints of the project, and even specific time requirements must be specified for each task.
2) **Project uniqueness**. The IT project not only provides products to customers but more i mportantly, offers different solutions according to customer requirements. Even if ther e is a ready-made solution, it is necessary to carry out specific customization work accor ding to the unique needs of the customer.
3) **Uncertainty**. In the implementation process of IT projects, we often encounter a variety of unexpected problems, and often there is no ready-made processing method, and we have to make frequent adjustments.
4) **Requirements frequently change in the project**. It is difficult to define clear software re quirements when a software project begins development.

Based on the unique characteristics of the above IT project, This research paper will mainly foc us on answering the following questions:

1) What are the possibilities in the process of centrally managing software development requirements?
2) Which are the most critical criteria that Software development team should choose to select an alternative?
3) 3.The benefit of calculating quantitative requirement management feedback from the beginning

METHODOLOGY

First of all, we will use the Pareto diagram to explore the root causes of the problem as figure 2, and Then we will focus on the main issues found. In this section, we will use Multi-Attribute Decision-making models to quantify each alternatives solution.

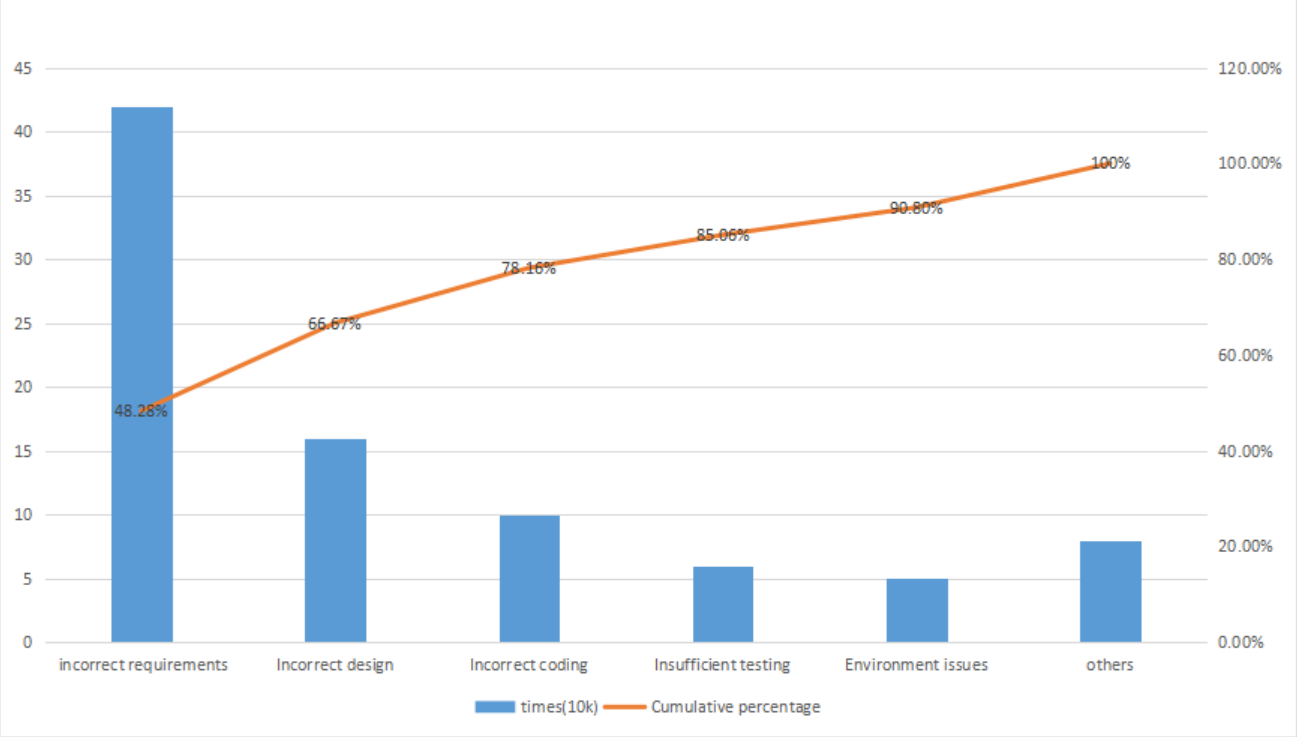

Figure 2: Pareto chart of failed software development[5]

Based on the above figure, among the many failed software projects, about 50% is due to requirement.

We can see that the root cause of the problems we have discovered is still regarding initial requirements and next we will focus on how to solve this problem effectively.

STEP2: Feasible Alternatives

We selected four feasible solutions.

The first alternative is **Specification by Example (SBE)**

***'Specification by example** is a collaborative approach to defining requirements and business-oriented functional tests for software products based on capturing and illustrating requirements using realistic examples instead of abstract statements. It is applied in the context of agile software development methods, in particular, behavior-driven development. This approach is particular*

[5] By the Author, and the data from 中国统计网. (n.d.). Retrieved from http://www.itongji.cn/

ly successful for managing requirements and functional tests on large-scale projects of the signif icant domain and organizational complexity.'[6]

The second alternative is **Behavior-driven development(BDD)**

Behavior-driven development is a technology for agile software development that encourages collaboration between developers, QA and non-technical or business participants in software p rojects. Mainly from the user's needs, emphasize system behavior. Named initially by Dan Nort h in 2003, BDD includes extreme programming practices such as acceptance testing and custom er test drivers as a response to test-driven development.

The third alternative is **Test-Driven Development(TDD)**

Test-driven development is a core practice and technology in agile development and a design m ethodology. The principle of TDD is to write a unit test case code before the function code is de veloped. The test code determines what product code needs to be written. The basic idea of TD D is to promote the whole development through testing, but test-driven development is not jus t a simple test work, but a process of quantifying demand analysis, design, and quality control. T DD first considers the use requirements (objects, functions, procedures, interfaces, etc.), mainly by writing a test case framework to design the functional processes and interfaces, and the test framework can be continuously verified.

Finally, the last one is **Acceptance Test Driven Development (ATDD)**

'***Acceptance test-driven development** (ATDD) is a development methodology based on commun ication between the business customers, the developers, and the testers.*[7]' Before preparing to i mplement a function or feature, the team first needs to define the desired quality standards an d acceptance rules to identify and reach a consensus acceptance test plan (including a series of test scenarios) to drive the developer's ATDD practices and testers' testing. Script development. For developers, emphasize how to implement the system and how to verify it.

STEP 3: Development of the Outcomes

First alternative: **Specification by Example (SBE)**, could describe the functions and behavior of c omputer systems in a way that is helpful to the development team (ideally expressed as an exe cutable test). So that stakeholders who do not understand the technology can understand it, ev en if the customer's needs are regularly Change, it also has good maintainability and can mainta in the relevance of the requirement, the feedback loop in software development can be signific antly reduced, thereby reducing rework and improving product quality.

[6] Adzic, Gojko (2011). Specification by example: How successful teams deliver the right software. Manning. ISBN 9781617290084.

7 Pugh, Ken (2011). Lean-Agile Acceptance Test-Driven Development: Better Software Through Collaboration. Addison-Wesley. ISBN 978-0321714084.

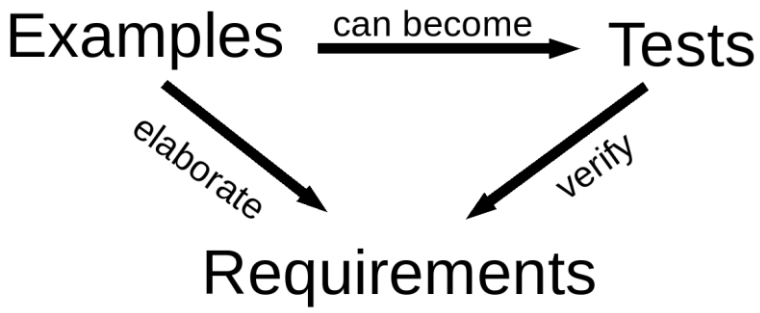

Figure 3: The Key process of SBE[8]

Second alternative: **Behavior-driven development(BDD)**, The focus of BDD is to understand the expected software behavior through discussions with stakeholders clearly. Focus on scenarios t hat make system behavior. The main task is to build consensus between project stakeholders a nd delivery teams through collaboration and requirements clarification. It extends the test-driv en development approach by writing non-programmer-readable test cases in natural language. Minimize the cost of converting the technical language of code writers back and forth between business customers, users, stakeholders, project managers, and other domains.

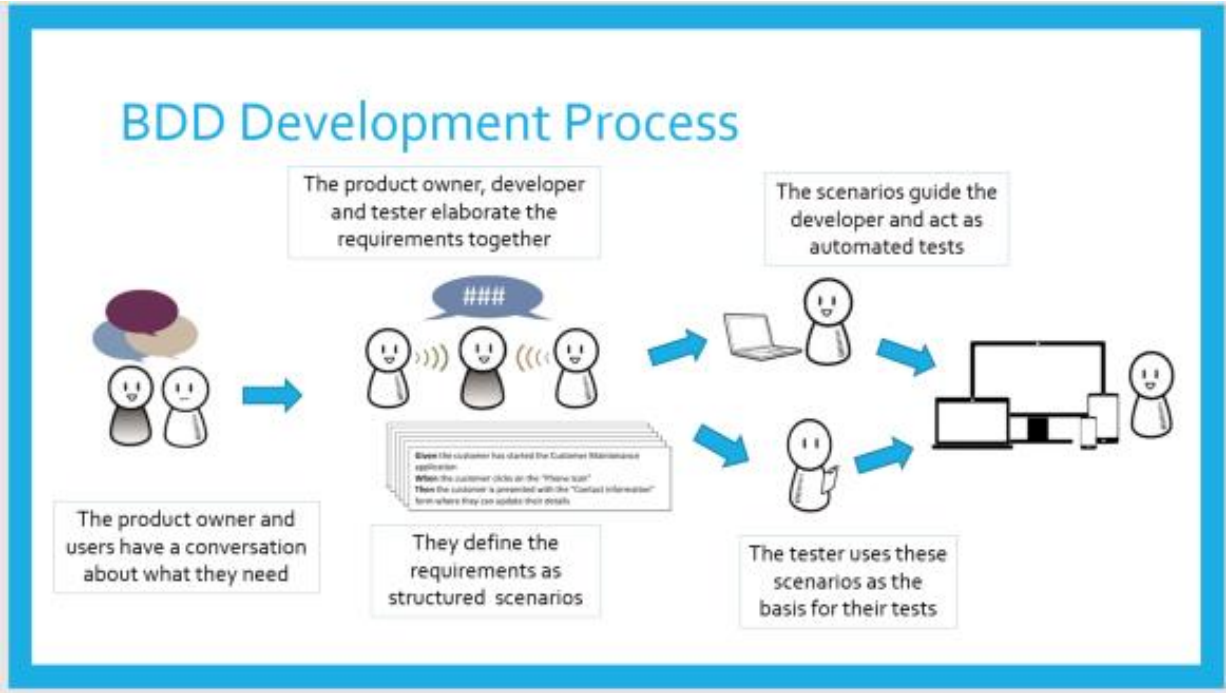

Figure 4: The Key process of BDD[9]

[8] Neuri Consulting LLP. (n.d.). Specification by Example -- Neuri Consulting LLP. Retrieved from https://neuri.co.uk/specbyexample.html

[9] Behaviour Driven Development (BDD) – Closing the loop on a Great Fiori UX | SAP Blogs [Web log post]. (2017, April 12). Retrieved from https://blogs.sap.com/2017/04/12/behaviour-driven-development-bdd-closing-the-loop-on-a-great-fiori-ux/

Third alternative: **Test-Driven Development (TDD)**, The basic idea of TDD is to promote the wh ole development through testing, but test-driven development is not just a simple test work, bu t a process of quantifying demand analysis, design, and quality control. Increase the amount of code in the development process to help customers and programmers remove ambiguous requ irements. Consider the use of elements first.

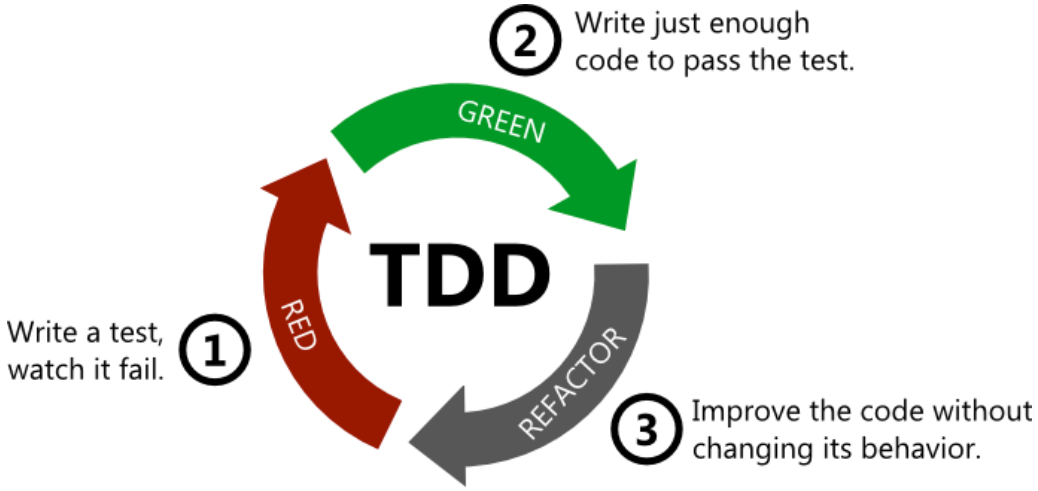

Figure 5: The Key process of TDD[10]

Fourth alternative: ***'Acceptance Test Driven Development (ATDD)** is a practice in which the whole team collaboratively discusses acceptance criteria, with examples, and then distills them into a set of concrete acceptance tests before development begins. It is the best way I know to ensure that we all have the same shared understanding of what it is we are building. It is also the best way I know to ensure we have a shared definition of Done.'*[11]

[10] Onyaem, C. (2016, December 24). Test-Driven Development (TDD) Resource Site (tdd.pacroy.com). Retrieved from https://medium.com/pacroy/test-driven-development-tdd-resource-site-tdd-pacroy-com-a02e02396f32

[11] Hendrickson, E. (2014). *Explore it!: Reduce risk and increase confidence with exploratory testing*. Frisco, TX: The Pragmatic Programmers.

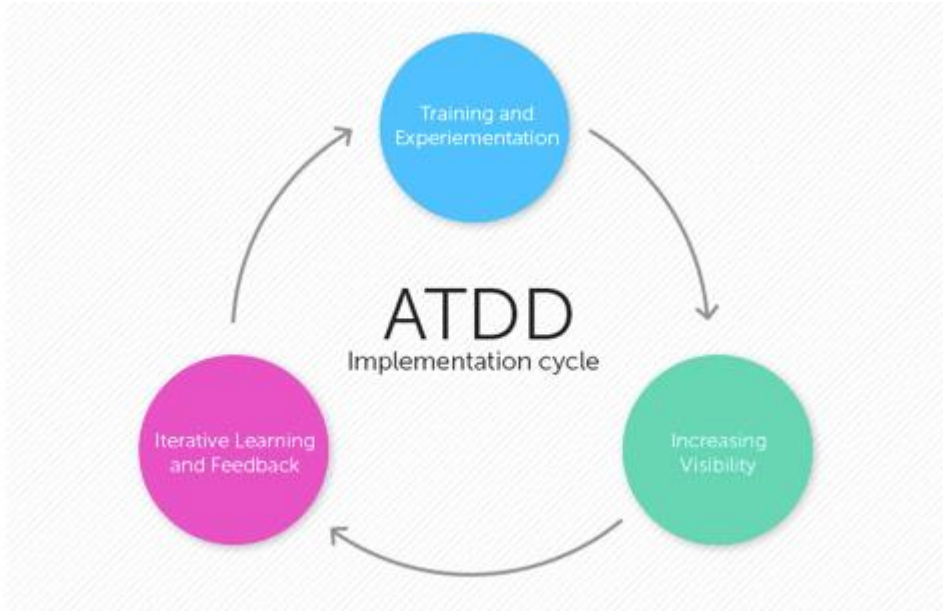

Figure 6: The Key process of ATDD[12]

STEP 4: Selection of the Criteria

To better evaluate each alternative solution. At the same time combined with the characteristi cs of its software, we identified six attributes to assess them:

1) Rapid development: this is the essential characteristics of the IT project. Check if it is a gile enough for development?

2) Ability to meet the uniqueness of the project: Meet the diversity of procedures and di fferent aspects of the requirements

3) Ability to respond to unexpected situations: Is it capable of handling accidents during development?

4) Ability to achieve demand response: Is it possible to complete the demand response accurately?

5) Automated test: is it possible to perform automated testing? An essential part of IT development Because it involves the amount of code.

6) Implementation cost: How much does it cost to achieve growth?

[12] A Quick Guide to Implementing ATDD. (2018, May 2). Retrieved from https://www.infoq.com/articles/quick-guide-atdd

Attributes	SBE	BDD	TDD	ATDD
RAPID DEVELOPMENT	GOOD	GOOD	EXCELLENT	EXCELLENT
ABILITY TO MEET THE UNIQUENESS OF THE PROJECT	EXCELLENT	EXCELLENT	GOOD	GOOD
ABILITY TO RESPOND TO UNEXPECTED SITUATIONS	EXCELLENT	FAIR	POOR	FAIR
ABILITY TO ACHIEVE REQUIREMENT RESPOND	EXCELLENT	EXCELLENT	POOR	FAIR
AUTOMATED TEST	FAIR	GOOD	GOOD	EXCELLENT
IMPLEMENTATION COST	FAIR	EXCELLENT	EXCELLENT	GOOD

Figure 7: Multi-Attribute Decision making models-analysis of the dominance of each alternative[13]

To distinguish the importance of these standards, we evaluate them and give them a rating to understand the most important criterion for the alternative.

	RAPID DEVELOPMENT	ABILITY TO MEET THE UNIQUENESS OF THE PROJECT	ABILITY TO RESPOND TO UNEXPECTED SITUATIONS	ABILITY TO ACHIEVE REQUIREMENT RESPOND	AUTOMATED TEST	IMPLEMENTATION COST	ORDINAL RANKING
RAPID DEVELOPMENT	x	0	0	0	1	1	3
ABILITY TO MEET THE UNIQUENE	1	x	0	0	1	1	4

[13] By the Author

SS OF THE PROJECT							
ABILITY TO RESPOND TO UNEXPECTED SITUATIONS	1	1	x	0	1	1	5
ABILITY TO ACHIEVE REQUIREMENT RESPOND	1	1	1	x	1	1	6
AUTOMATED TEST	0	0	0	0	x	1	2
IMPLEMENTATION COST	0	0	0	0	0	x	1

Figure 8: Comparison of the importance of the attributes[14]

Based on the above figure, we can have an understanding of the performance of each alternative solution based on each criterion.

Following the previous analysis through the figures, it is now possible to create a relative ranking of each alternative based on the attribute. We decided to choose lexicography to have an overview of the best option for our topic, according to the characteristics selected.

After the previous figures, it is now possible to create a relative ranking for each attribute. Based on the selected characteristics, we decided to choose a lexicographical compilation to outline the best alternatives for our theme.

[14] By the author

Ordinal Ranking		The relative ranking of each alternative based on attribute
6	ABILITY TO ACHIEVE REQUIREMENT RESPOND	SBE=BDD>ATDD>TDD
5	ABILITY TO RESPOND TO UNEXPECTED SITUATIONS	SBE>BDD=ATDD>TDD
4	ABILITY TO MEET THE UNIQUENESS OF THE PROJECT	SBE=BDD>ATDD=TDD
3	RAPID DEVELOPMENT	TDD=ATDD>SBE=BDD
2	AUTOMATED TEST	ATDD>BDD=TDD>SBE
1	IMPLEMENTATION COST	BDD=TDD>ATDD>SBE

Figure 9: Lexicography to choose the best metric[15]

FINDINGS

Step 5: Analysis and comparison of alternatives

In this section, we will compare each alternative, which will enable the recommendation of the best option, using the following figures:

VALUE	FORMULA	DIMENSIONLESS VALUE
EXCELLENT	Relative rank = (4-1)/(4-1)	1
GOOD	Relative rank = (3-1)/(4-1)	0.67

[15] By the author

FAIR	Relative rank = (2-1)/(4-1)	0.33
POOR	Relative rank = (1-1)/(4-1)	0

Figure 10: dimensionless value[16]

Attributes	SBE	BDD	TDD	ATDD
RAPID DEVELOPMENT	0.67	0.67	1	1
ABILITY TO MEET THE UNIQUENESS OF THE PROJECT	1	1	0.67	0.67
ABILITY TO RESPOND TO UNEXPECTED SITUATIONS	1	0.33	0	0.33
ABILITY TO ACHIEVE REQUIREMENT RESPOND	1	1	0	0.33
AUTOMATED TEST	0.33	0.67	0.67	1
IMPLEMENTATION COST	0.33	1	1	0.67
TOTAL	4.33	4.67	3.34	4

Figure 11: Summary of Nondimensional scaling results[17]

[16] By the author

[17] By the author

			ALTERNATIVES							
ATTRIBUTES	RELATIVE RANK	NORMALIZED WEIGHT (A)	SBE		BDD		TDD		ATDD	
			(B)	(A)×(B)	(C)	(A)×(C)	(D)	(A)×(D)	(E)	(A)×(E)
RAPID DEVELOPMENT	3	0.14	0.67	0.09	0.67	0.09	1	0.14	1	0.14
ABILITY TO MEET THE UNIQUENESS OF THE PROJECT	4	0.19	1	0.19	1	0.19	0.67	0.13	0.67	0.13
ABILITY TO RESPOND TO UNEXPECTED SITUATIONS	5	0.24	1	0.24	0.33	0.08	0	0	0.33	0.08
ABILITY TO ACHIEVE REQUIREMENT RESPOND	6	0.29	1	0.29	1	0.29	0	0	0.33	0.10
AUTOMATED	2	0.10	0.33	0.03	0.67	0.07	0.67	0.07	1	0.1

TEST										
IMPLEMENTATION COST	1	0.05	0.33	0.02	1	0.05	1	0.05	0.67	0.03
TOTAL	21	1		0.86		0.77		0.34		0.58

Figure 12: The additive weighting technique results[18]

Step 6: Selection of the preferred alternative

From the analysis above, we can find out that in figure 7, BDD performs best. However, after weighing in Figure 8, the performance of SBE has tasted BDD. From the above results, SBE is 111% better than BDD (0.86 / 0.77 * 100). However, BDD is the second-best choice because it performs better by 226% (0.77 / 0.34 * 100) and 133% (0.77 / 0.58 * 100) than TDD and ATDD, respectively. Therefore, we strongly recommend applying the SBE method to deal with demand-related issues in software projects.

Step 7: Performance Monitoring and Post Evaluation Results

To monitor the performance better, we could think about track the production from the following steps:

- whether the purpose and use of the project development are clear
- whether the user needs are sorted out?
- whether the demand changes are monitored?
- whether the demand is confirmed?
- whether the demand risk is prevented?
- whether the tools are applied well during demand management?
- whether there is a targeted request management process?
- whether requirements, test cases, bug management links?
- whether there is Quantitative project management feedback?

CONCLUSIONS

We can now answer to the questions asked at the beginning of this research:

1. What are the possibilities in the process of centrally managing software development requirements?

[18] By the author

2. Which are the most critical criteria that Software development team should choose to select an alternative?
3. 3.The benefit of calculating quantitative requirement management feedback from the beginning

We have developed four different alternative solutions. They are SBE, BDD, TDD, ATDD. To choose a superior alternative, we compare them to six attributes and minimum acceptance criteria. In the end, none of them are eliminated, and they can all retained. They can be excellent. However, if a mature software development team needs to choose one of them, then using the SBE method can make the stakeholders who understand the technology understand the participation, and the SBE method has good maintainability and can maintain the relevance of the requirements. Any software project starts with conditions and ends with whether the software meets the requirements, nothing is more important than this.

The benefit is to enable all stakeholders in the project to collaborate and communicate effectively, to illustrate requirements in an instance, to verify requirements frequently through automated testing, and to evolve a set of live examples from the instantiation of elements and automated test cases. Document system." This "live document system" can adequately explain the system and can be used as the standard for delivery acceptance.

In general,'*The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later'.*[19]

BIBLIOGRAPHY

1. Adzic, G. (2011). *Specification by Example: How successful teams deliver the right soft ware*. Manning Publications.

2. Brooks, F. P. (1975). *The mythical man-month: Essays on software engineering*.

3. Wiegers, K. E., & Beatty, J. (2015). *Software requirements*. Redmond, Washington: Mic rosoft Press.

4. Robertson, J., & Robertson, S. (2017). *Mastering the requirements process*. Harlow: Ad dison-Wesley.

[19] Brooks, F., & Kugler, H. J. (1987). *No silver bullet* (pp. 1069-1076). April.

5. Maciaszek, L.A. (2005, January 30).*Requirements Analysis and System Design*

6. Crawley, E. (2015, April 25). *System Architecture: Strategy and Product Development f or Complex Systems*

7. Feng, X. (2017, January). *Effective requirements analysis*

8. Brooks, F. P. (1987). *No Silver Bullet—Essence and Accidents of Software Engineering*

9. Burk, S. (2011, November).Gathering software requirements: The importance of sequ ence. Retrieved October 2018, from https://searchsoftwarequality.techtarget.com/an swer/Gathering-software-requirements-The-importance-of-sequence

10. Hirsch, J.(2013, November 22).successful requirements gathering. Retrieved October 2018, from https://www.phase2technology.com/blog/successful-requirements-gathe ring

11. Eriksson, U. (2016, March 3). 5 essential agile techniques to improve your requiremen ts documentation. Retrieved October 2018, from https://reqtest.com/requirements-b log/5-essential-agile-techniques-to-improve-your-requirements-documentation/

12. Harschnitz, L. (2011, October 14).Effective Requirements. Retrieved October 2018, fro m https://www.sas.com/content/dam/SAS/en_ca/User%20Group%20Presentations/ Hamilton-User-Group/LesleyHarschnitz-EffectiveRequirements-October2011.pdf

13. Makar, A. (2015, July 22). 7 tools to gather better software requirements. Retrieved O ctober 2018, from https://www.liquidplanner.com/blog/7-tools-to-gather-better-soft ware-requirements/

14. Richter, L. (2011, September 26).a guide to gathering and analyzing project requireme nts. Retrieved October 2018, from https://www.brighthubpm.com/project-planning/1 24948-a-guide-to-gathering-and-analyzing-project-requirements/

15. Goldsmith, R.F. (2015, January). Engage developers in the requirements definition pro cess. Retrieved October 2018, from https://searchsoftwarequality.techtarget.com/an swer/Engage-developers-in-the-requirements-definition-process

16. Reichert, A. (2015, February). Use story mapping for more customer-focused apps. Retrieved October 2018, from https://searchsoftwarequality.techtarget.com/tip/Use-story-mapping-for-more-customer-focused-apps

17. Goldsmith, R.F. (2014, August). Understanding the requirements process vs. requirements management. Retrieved October 2018, from https://searchsoftwarequality.techtarget.com/answer/Understanding-the-requirements-process-vs-requirements-management

18. Reichert, A. (2014, December). How do requirements relate to acceptance criteria? Retrieved October 2018, from https://searchsoftwarequality.techtarget.com/answer/How-do-requirements-relate-to-acceptance-criteria

19. Goldsmith, R.F. (2016, November). Use elicitation techniques to discover software requirements. Retrieved October 2018, from https://searchsoftwarequality.techtarget.com/feature/Use-elicitation-techniques-to-discover-software-requirements

20. Goldsmith, R.F. (2014, October). Is Agile requirement gathering that different from waterfall? Retrieved October 2018, from https://searchsoftwarequality.techtarget.com/answer/Is-Agile-requirement-gathering-that-different-from-waterfall

21. Goldsmith, R.F. (2015, August). Jennifer LentDual-track Agile keeps requirements in check. Retrieved October 2018, from https://searchsoftwarequality.techtarget.com/opinion/Dual-track-Agile-keeps-requirements-in-check

22. Reichert, A. (2015, January). How Agile teams manage continuous requirements. Retrieved October 2018, from https://searchsoftwarequality.techtarget.com/answer/How-Agile-teams-manage-continuous-requirements

23. Goldsmith, R.F. (2010, August). How to handle requirements creep. Retrieved October 2018, from https://searchsoftwarequality.techtarget.com/answer/How-to-handle-requirements-creep

24. Goldsmith, R.F. (2010, July). How to settle conflicting software requirements between users and stakeholders. Retrieved October 2018, from https://searchsoftwarequality.techtarget.com/answer/How-to-settle-conflicting-software-requirements-between-users-and-stakeholders

25. Goldsmith, R.F. (2009, June). Differentiating between Functional and Nonfunctional R equirements. Retrieved October 2018, from https://searchsoftwarequality.techtarget.com/answer/Differentiating-between-Functional-and-Nonfunctional-Requirements

26. Goldsmith, R.F. (2009, January). How to avoid requirements creep. Retrieved October 2018, from https://searchsoftwarequality.techtarget.com/tip/How-to-avoid-requirements-creep

27. Goldsmith, R.F. (2008, August). Why-you-should-test-requirements-definitions. Retrieved October 2018, from https://searchsoftwarequality.techtarget.com/answer/Why-you-should-test-requirements-definitions

28. Sweeney, M. (2015). 3 Reasons Why Software Development Projects Fail - Clearcode Blog. Retrieved October 2018, from https://clearcode.cc/blog/why-software-development-projects-fail/

29. Pulse of the Profession | Project Management Institute. (November 11). Retrieved Oc tober 2018, from https://www.pmi.org/Pulse

About the Author

Haoyu Wang

Paris, France

Haoyu Wang, Chinese, 23 years old, major in Project and Programme Management & Business Development (PPMBD) at Skema Business School in Paris, France. He has a background in business and IT education.

Before coming to France, Haoyu completed his studies at ZhaoQing College in China and hold a bachelor's degree in the Internet of Things. Haoyu has been working for 'Network Information and Security Research Institute' (China) and 'Internet of Things Collaborative Innovation Center Embedded Systems Application Institute' (China) for many years. During this time, he assisted in the completion of the 'SMS-based smartphone virus propagation analysis and simulation system research' and the 'Unmanned Helicopter Quadrotor,' 'WiFi student attendance system.' From 2015 to 2016, Haoyu hosted the provincial assignment tasks 'climbing plan' and 'Challenge Cup.' Finally, the task is perfectly delivered with 'small block type wireless LED display design' and 'smart home remote monitoring system development.' After that, he assisted the professor in publishing an article entitled "Spi / Uart and Zigbee Protocol Conversion Module Design" in the IoT Journal. In 2016, he became an executive intern at Hong Kong AXA Insurance Company. Participate in the 'fund selection plan and solve the difficulties and countermeasures of financial planning in mainland China.'

Haoyu has the potential to be a promising project manager. At the same time, he has great enthusiasm and interest in the IT field.Haoyu is now working on a master in sciences in Skema business school.He lives in Paris now and can be contacted at haoyu.wang@skema.edu or https://www.linkedin.com/in/haoyu-wang-0a21b1161/