

# Identify and Classify Critical Success Factor of Agile Software Development Methodology Using Mind Map

Tasneem Abd El Hameed  
MSc student /Information system  
Department  
Faculty of computers and  
information, Helwan University  
Cairo, Egypt

Mahmoud Abd EL Latif  
Assistant Professor of information  
system  
Faculty of computers and  
information, Helwan University

Sherif Kholief  
Assistant Professor of information  
system  
Faculty of computers and  
information, Helwan University

**Abstract**—Selecting the right method, right personnel and right practices, and applying them adequately, determine the success of software development. In this paper, a qualitative study is carried out among the critical factors of success from previous studies. The factors of success match with their relative principles to illustrate the most valuable factor for agile approach success, this paper also prove that the twelve principles poorly identified for few factors resulting from qualitative and quantitative past studies. Dimensions and Factors are presented using Critical success Dimensions and Factors Mind Map Model.

**Keywords**—Agile success factor; Agile principles

## I. INTRODUCTION

Most of the software is used in businesses and organizations all over the world. Nowadays, due to Volatile and unpredictable nature of system projects development, traditional approaches become inflexible, and are unable to adjust to the system projects.

Agile approach is developed to overcome the failures that result from traditional approaches, and to offer a lightweight framework for helping organizations and teams to respond faster and iteratively.

When it comes to adopting agile approach, the first step organizations start with is the agile manifesto stating that agile approach contains four values and twelve principles. Unfortunately, applying these principles, can create challenges, and these challenges result in failures or successes, which help in figuring out factors for success and failure.

Transition from being non-agile to agile in an organization is tricky and difficult. Agile adoption and transformation efforts are experiencing high failure rates in organizations. 84% of respondents in the Agile Development Survey reported that they had experienced a failed Agile project. Only 16% of respondents had not experienced failure[22]. In this paper, we clarify the success factors resulting from different previous studies in a graphical way called Mind Map for easy learning.

Mind map is a graphical way to represent ideas and concepts, and is like a visual thinking tool that helps structuring information, helping in better analysis. Mind Map

represents semantic or other connections between portions of learned material hierarchically. Mind Maps are easy to learn and apply, and provide a concise hierarchic overview, The further advantages of mind map are easy to extend and add further content. They are Idiosyncratic, hard to read for others, can be inconsistent, and can become overly complex (loss of big picture) [8]. To overcome the complexity, we distinct each dimension with related factors in a particular map, to be easy in learning and reading.

The Remainder of the paper is organized into five sections. Section 2 discusses a background study, Section 3 discusses literature review, Section 4 discusses factors, and Section 5 is the conclusion.

## II. BACKGROUND

Software Process Models are based on one of the three models of software Development. The models are *waterfall approach*, *Iterative Development approach* [2].

### A. Waterfall Approach

The waterfall approach emphasizes a structured progression between defined phases. Each phase consists of a definite set of activities and deliverables that must be accomplished before the following phase can begin [3].

The waterfall phases are requirements definition, system and software design, implementation and unit testing, integration and system testing, operation and maintenance [2]. Waterfall model has some disadvantages which are “1) some requirements may emerge after the requirements gathering phase, resulting in some problems. 2) Problems detected at a stage are not solved completely in the same stage. 3) There is no concept of changing (partitioning) the project into multiple stages. 4) New requirements by the client are very expensive and cannot be adjusted in the current edition of the software product. 5) Estimation of time and budget for each stage is very difficult. 6) No prototype before the finishing of the life cycle. 7) Testing in the last stage of the development. 8) If testing reveals some problems, then going to the design stage is very difficult. 9) Very high risk in the entire life cycle development. Not recommended for object oriented projects” [4].

### B. Iterative development approach

Iterative Development starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed. Process models have explicitly been designed to support (iteration development are incremental delivery and spiral development).

- Incremental delivery

Incremental Delivery customer identify, the services to be provided by the system and which are the most and the least important, however, there are some problems with incremental delivery. Increments should be small and each increment should deliver some of the system functionality. "It can be difficult to map the customer's requirement onto the increments of the right size" [2].

- Spiral Development

Spiral Development Consists of Loops where each loop in the spiral represents a phase of software process. Each loop is split into four factors: Objective setting, risk assessment and reduction, development and validation, planning. The main difference between spiral and other software processes is the explicit recognition of risks [2]. Disadvantage of spiral model is a very complex solution suitable for big, complicated projects, and yet undoubtedly more flexible than the original waterfall method. The spiral model is an underlying inspiration for many modern methods, for example, Rational Unified Process [5].

### C. Agile Software Approach

Agile is increasingly becoming the dominating developing method in software industry. For a successful software project, it is essential to identify what leads to success. Projects succeed when enough factors are well defined, and failure teaches us to overcome shortcomings in the future projects [12].

- Agile Manifesto

Agile approaches are introduced to overcome the failure factors for traditional SDLC, "In 2001 the agile manifesto states that four values and twelve principles, four Values are (1) Individuals and interactions over processes and tools (2) Working software over comprehensive documentation (3) Customer collaboration over contract negotiation (4) Responding to change over following a plan and twelve principles are (1) Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. (2) Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. (3) Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. (4) Business people and developers must work together daily throughout the project.

(5) Build projects around motivated individuals give them the environment and support they need and trust them to get the job done. (6)The most efficient and effective method of conveying information to and within a development team is

face-to-face conversation. (7)Working software is the primary measure of progress. (8)Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. (9)Continuous attention to technical excellence and good design enhances agility. (10)Simplicity--the art of maximizing the amount of work not done--is essential. (11)The best architectures, requirements, and designs emerge from self-organizing teams.(12) at regular intervals, the team reflects on how to become more effective , then tunes and adjusts its behavior accordingly". [7]

Agile is a group of lightweight methodologies used to develop highly potential software. Agile methods universally rely on an iterative approach to software specification development and delivery, they are intended to deliver working software quickly to customers, who can propose new and changed requirement to be included in later iterations of the system [2].

- *Agile Methodologies*

Agile software development methodologies share many features and practices which include the practice of whole team, measures, short release, test-driven development, Pair Programming, customer collaboration, Prototyping, refactoring, continuous integration and less documentation to produce valuable software [2]. Agile Methodologies are Extreme Programming (XP), Crystal Methods, Feature Driven development (FDD), system development Methods (DSDM), Scrum.

- Extreme Programming(XP)

Based on a set of practices like pair programming, customer collocation, customer satisfaction [3]. XP is perhaps the best known and most widely used of the agile methods [2].

- Crystal Methods

Crystal methods is a lightweight methodology, Based on premise that people impact software development projects more than tools or processes [2].

- Feature Driven development (FDD)

FDD is an iterative, incremental, and lightweight software development process. It is a combination of a number of industry-recognized best. These practices are all driven from a client- valued functionality. The main purpose it to deliver tangible, working software repeatedly in a timely manner [3].

- Dynamic system development Methods (DSDM)

It is mainly a framework more than a process. Dynamic system development method is about fixing quality, cost and time. DSDM is used for developing software and non-IT companies [3].

- Scrum

Is an incremental and iterative framework where practitioners can employ different processes and techniques to develop a complex product and project. It is specially designed to handle rapidly changing business requirements.

Scrum is a sprint; a time-boxed effort usually from two weeks to four weeks, in a sprint work is divided into parts and to be completed at the end of the sprint time. Scrum focuses more on management of the process than coding techniques, and it is used in small and large projects [3].

### III. LITERATURE REVIEW

Search began in 1996 when Walid and Oya suggested a new framework to determine Critical Success and failure factors. They suggested a new design for critical factors and described the impact of these factors on performance of the project. They used Empirical study to test practicality of using the suggested design and grouped the factors into 4 areas: Project, Managers and team member, organization, and environment. The survey results “demonstrate that project managers, managerial skills, team members, commitment and their technical background, project attributes and environment factors are as viable and can be as a critical as an organization factor and the criticality of these factor varies between industries”[9].

A lot of researches are done and discussed from 1996 till 2006. Aniket Mahanti made a survey paper of major challenges in adopting agile practices by enterprises. Successful adoption of agile methodology includes obtain management buy-in, education and support, integrating to external processes, starting pilot projects, report and adapt, and sustain agility. The success of agile adoption is directly related to how the new methodology is introduced in the organization [10].

After that, in 2008 a survey study of critical success factors in agile software projects was done by Tsun chow, Dac-Buu Cao using quantitative approach. A study led by agile experts, gathered survey of 109 agile projects from 25 countries all over the world. Multiple regression techniques were used. The survey results obtained that only 10 out of 48 hypotheses were supported, and identified three critical success factors for agile software projects, “Delivery Strategy, agile software engineering techniques and team capability”[11]. They conclude that they should try to define different success factors or try to display the success of agile projects with different method [11].

Then Dragan Stankovica, Vesna Nikolicb, Miodrag Djordjevicc, Dac-Buu Cao continued the study (Chow and Cao, 2008). They tried to verify the classification of critical success factors previously described in study by Chow and Cao (2008). They made a regression analysis on the collected data which introduced three more factors that could potentially be considered as critical success factors. [12]

Subhas Chandra Misra, Vinod Kumar, and Uma Kumar developed a conjectured hypothetical success factors framework to address the research question. They used the data analysis techniques to validate the hypotheses. The study was made utilizing an extensive scale study-based methodology, comprising respondents who practice agile software development and who had experience in practicing arrangement driven programming advancement previously. The study demonstrates that 9 of the 14 hypothesized factors have significant relationship with “Success”. The important

success factors that were found are: “customer collaboration, customer satisfaction, customer commitment, decision time, corporate culture, personal characteristics, societal culture, and training and learning” [13].

In 2010, Zulkefi, Saadiyah and Noor carried out a literature review to gather information from previous study, they found that “Customer involvement, communication, minimum changes of requirement, corporate culture, time allocation, simplicity, active testing, code review and customer collaboration determine the successful in agile software development methodology” [14]. They developed a conceptual model in their study.

Jianping and Routing have designed P company success factors model “leading (recognition of top leaders, participation of top leaders), organization (creating clear vision, building the agile organizational culture, changing the way of management), tools and technology (configuring the necessary tools and infrastructure, using design patterns and other advanced design methods, using software reuse technology), appropriate import (selecting applicable import project, excellence implementation staff, selecting proper agile method practice), training and education (correct understanding and mastery of agile methodologies, enhancing the professional capabilities of the employee), measuring success (flexible and innovative development method, rapid response to demand, forward looking response to changing factors, successfully building learning organization) which they verified them by a questionnaire in P Company” [15]. They conclude that education and training play a positive role in agile improvement. Agile method must be established in agile culture with due attention to the design and application of technology.

In 2011 Claudia, Daniela, Fabio, and Reidar made two case studies in industry and analysed data from two projects. They identified three literature review and present the main factors based on most relevant factors, “the factors include product (reuse, software characteristics), project (resource constraint, schedule, team composition, communication), personnel (team experience and motivation) and process (customer participation, daily builds, documentation, early prototyping, incremental and iterative development, modern programming practice, programming language abstraction, software methods, tool usage)”[16]. They conclude that there are some factors impacting the productivity of agile teams. These factors are team decomposition and allocation, external dependencies, and staff turnover.

Ani Liza and Andrew M Gravel initiated a study that involves 13 participants including CEOs, project managers, founders, and developers. Their study resulted that social and human aspects are very important when they start using the agile methods. During the study, they used qualitative semi-structure interview and concluded that the issues and challenges during adoption were mindset, knowledge, project, people, knowledge transfer, management involvement, communication, technical aspects, and organizational structure [17].

Kumar and Goe (2012) illustrate the results of a survey conducted to demonstrate and explain the factors considered by

software practitioners while adopting agile methodologies, and the effects of adopting agile methodologies on customers and business while practicing agile. They presented six hypotheses, which are “impact of team size, impact of requirement gathering for agile methodologies, effective requirement capturing method, time taken to resolve a problem and impact of small response time with customer on software development” [18]. The results of this research indicated that adopting agile increased the productivity of an organization and also increased customer satisfaction [18].

IV. CRITICAL SUCCESS FACTORS

In this section, Critical Success Dimensions that grouped from previous quantitative and qualitative studies are displayed. The factors were arranged based upon the most ranked one to the least ranked ones. Similar factors were collected with each other and given the most common name between studies.

Finally, each principle was given a constant value (see Table 8) where we matched each factor with their related one as emphasized by Maarit [22], see Fig. 9. We found that most of studies are about certain dimension so each dimension has been given a weight to determine which dimension is the most important table [1-6]. The average displayed is the weight of each dimension as it determine how important is. All factors and dimensions are collected and displayed in one Mind Map to be more memorized, then Each dimension are illustrated with their related factors, at the end of this section, all Dimensions with their factors are presented in only one CSDF Model (Critical Success dimensions and Factors).

Fig. 1 includes Process, People, Project, Product, Organization, technical (4P OT), where each dimension is one

of the most critical success dimensions grouped from previous researches.



Figure 1: Mind Map Model

Fig. 1. CSF Mind Map Model

A. People

Most of the researchers agreed that People Dimension plays an important role in any software development project. People dimensions are classified into eight factors, and these eight factors are demonstrated in Table 1.

People factors may include Education where team should learn agile techniques and how to apply and adopt them in non-agile companies, accomplished by learning how to support teams, as the project manager should have the ability to tradeoff, ability to coordinate, and participate in all aspects. Also the team member and the manager must be committed for their tasks and project.

Table 1 displays People Dimension and all factors according to the highest one. [9][10][11][12][13][14][15][16][18][19].

TABLE I. PEOPLE FACTORS

Dimensions	Factors	{Belassi1996}	{Mahanti2006}	{Chow2008}	{Misra2009}	{MansorSandNoor2010}	{Wan2010}	{Melo2011}	{Kumar2012}	{Sheffield2013}	{Hummel2015}	Rate	Principles	AVG
People	Education and support	√	√	√	√		√	√			√	7	P5	4.5
	Customer Centric issues	√		√	√	√			√	√	√	7	P1,2,4	
	Management Style	√	√	√	√		√	√				6	All	
	Communication Skills	√		√	√	√		√			√	6	P6	
	Motivation			√	√			√	√	√		5	P5,11	
	Commitment	√		√							√	3	P3	
	Report and Adapt		√									1	P2	
Project Champion	√										1	P2		

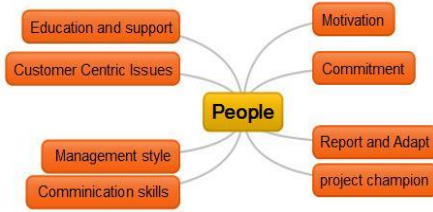


Fig. 2. People Mind Map

**B. Organization**

When an organization starts to adopt agile, the way of organizational culture and mindset have to change. An agile way of working that deliver along new practices for teams and managers, and usually agile impacts the organizational culture

and mindset. Although these factors are important but also changing everything at the same time might be too big challenge for an organization. Organization dimension with their factors are displayed in Table 2 [9][10][11][12][13][14][15][16][18][19]



Fig. 3. Organization Mind Map

TABLE II. ORGANIZATION FACTORS

Dimensions	Factors	{Belassi1996}	{Mahanti2006}	{Chow2008}	{Misra2009}	{MansorSandNoor2010}	{Wan2010}	{MeIo2011}	{Kumar2012}	{Sheffield2013}	{HummeI2015}	Rate	Principles	AVG
Organization	Corporate Culture				√	√	√		√	√		4	P4	2.25
	Organizational Environment (Political economical, technological environment)	√									√	2	P5	
	Collocation of Whole team			√				√				2	P4,6	
	Sustain Agility		√									1	P8	

**C. Technical**

Technical Practices such as continuous integration, test-driven development, pair programming, refactoring, and collective ownership are what has made it possible for most of organization to deliver what the customer need efficiently and effectively. Teams will be become twice as productive, if they adopt some of these practices [20]. Table 3 displays the technical dimension and factors. [9][10][11][12][13][14][15][16][18][19]



Fig. 4. Technical Mind Map

TABLE III. TECHNICAL FACTORS

Dimensions	Factors	{Belassi1996}	{Mahanti2006}	{Chow2008}	{Misra2009}	{MansorSandNoor2010}	{Wan2010}	{MeIo2011}	{Kumar2012}	{Sheffield2013}	{HummeI2015}	Rate	Principles	AVG
Technical	High expertise for team and organizational factor	√		√	√		√		√		√	4	P9	3.5
	Practices					√	√	√			√	2	P9	
	Trouble shooting for team	√		√					√			2	P9	
	Tool Usage		√				√	√				1	P9	

D. Process

There are many agile processes: SCRUM, Crystal, Behavior-Driven Development (BDD), Test-Driven

Development (TDD), Feature-Driven Development (FDD), Adaptive Software Development (ADP), Extreme Programming (XP), and more. [9][10][11][12][13][14][15][16][18][19]

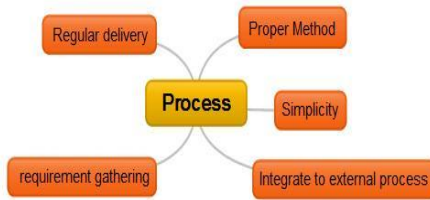


Fig. 5. Process Mind Maps

TABLE IV. PROCESS FACTORS

Dimensions	Factors	{Belassi1996}	{Mahanti2006}	{Chow2008}	{Misra2009}	{MansorSandNoor2010}	{Wan2010}	{Melo2011}	{Kumar2012}	{Sheffield2013}	{Hummel2015}	Rate	Principles	AVG
Process	Regular Delivery of software			√	√	√		√	√		√	5	P9	2.8
	Effective requirement gathering method			√				√	√			3	P9	
	Select proper Methodology			√							√	2	P9	
	Simplicity			√		√						2	P9	
	Integrate to external processes		√	√								2	X	

E. Project

Agile Methods are most applicable to projects where requirements are ill-defined and fluid since they seek to accommodate change easily. Projects that are unprecedented within an organization or use cutting-edge technology are examples of projects where change is likely to have a significant impact on the project [21]. Table 5 displays project factors [9][10][11][12][13][14][16].



Fig. 6. Project Mind Maps

TABLE V. PROJECT FACTORS

Dimensions	Factors	{Belassi1996}	{Mahanti2006}	{Chow2008}	{Misra2009}	{MansorSandNoor2010}	{Wan2010}	{Melo2011}	{Kumar2012}	{Sheffield2013}	{Hummel2015}	Rate	Principles	AVG
Project	Project type	√		√	√						√	4	X	2.4
	Schedule		√	√				√				3	X	
	Tem Size			√		√		√			√	3	X	
	Pilot Project		√									1	X	
	Minimum changes of Requirements					√						1	X	

F. Product

Most of studies merge the product with project and process due to the big similarities between them but only few studies separate them, we illustrate it separate to be more specific and Precise. Table 6 displays product factor [15][16].



Fig. 7. Product Mind Map

TABLE VI. PRODUCT FACTORS

Dimensions	Factors	{Belassi1996}	{Mahanti2006}	{Chow2008}	{Misra2009}	{MansorSandNoor2010}	{Wan2010}	{Melo2011}	{Kumar2012}	{Sheffield2013}	{Humme12015}	Rate	Principles	AVG
Product	Using Software reuse technology						√	√				2	X	1.5
	Software characteristics							√				1	X	

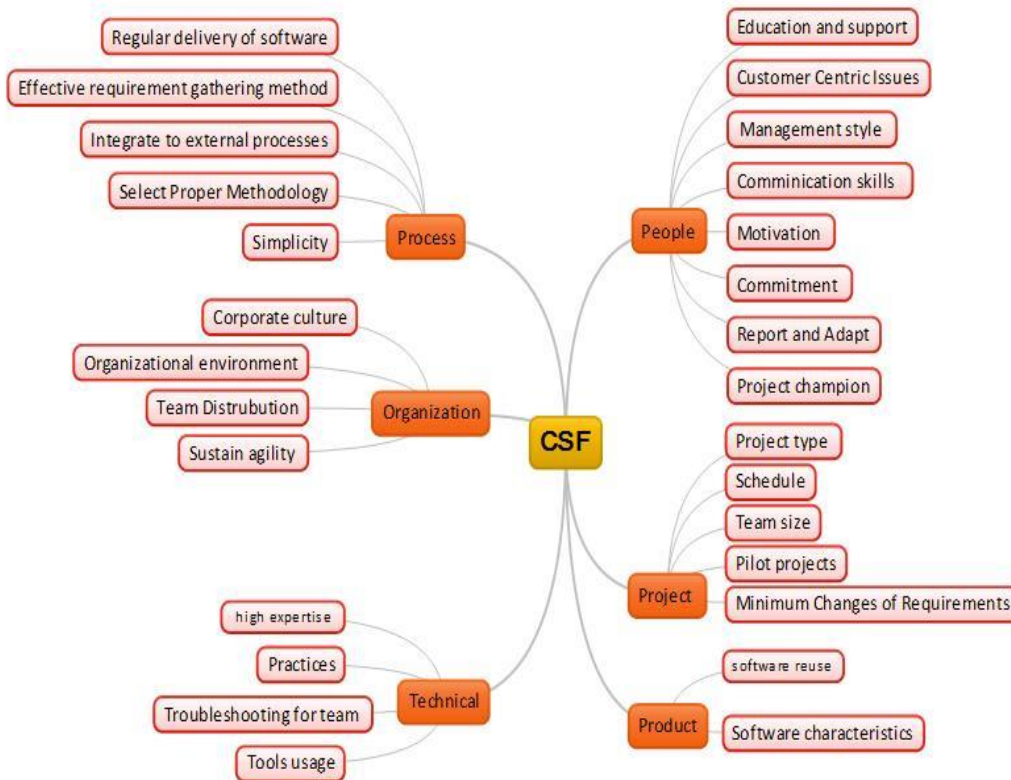


Fig. 8. CSDF Mind Map Model

TABLE VII. PRINCIPLES AND WHAT THEY EMPHASIZE[22]

Agile Principles	Emphasis
Our highest priority is to satisfy the customer through early and continuous delivery of valuable software	Customer satisfaction, continuous delivery, value, early deliveries
Welcome changing requirements, even late in development. Agile harness change for customer's competitive advantage	Adaptability, competitiveness, customer benefit
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter time scale.	Working software frequently, shorter time scale frequent deliveries
Business people and developers must work together daily throughout the project.	Work together daily, collaboration
Build Projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done	Motivated individuals, good environment, support, trust
The most efficient and effective method of conveying information to and within a development team is face to face conversation	Efficiency, Communication
Working software is the primary measure of progress	Measure progress via deliverable
Agile Processes promote sustainable development. the sponsors, developers, and users should be able to maintain a constant Pace identifiability	Sustainability, People
Continuous attention to technical excellence and good design enhance agility	Focus on technical excellence, good design as enabler of agility
Simplicity – the art of maximizing the amount of work not done is essential	Simplicity – optimizing work
The best architectures, requirements, and design emerge from self-organizing teams	Self-organization
At regular interval, the team reflect on how to become more effective, then tunes and adjusts its behavior	Built-in improvement of efficiency

TABLE VIII. AGILE PRINCIPLES AND CONSTANT VALUES

Principles	Factors	Constant
Our highest priority is to satisfy the customer through early and continuous delivery of valuable software	Customer satisfaction, continuous delivery, value, early deliveries	P1
Welcome changing requirements, even late in development. Agile harness change for customer's competitive advantage	Adaptability, competitiveness, customer benefit	P2
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter time scale.	Working software frequently, shorter time scale frequent deliveries	P3
Business people and developers must work together daily throughout the project.	Work together daily, collaboration	P4
Build Projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done	Motivated individuals, good environment, support, trust	P5
The most efficient and effective method of conveying information to and within a development team is face to face conversation	Efficiency, Communication	P6
Working software is the primary measure of progress	Measure progress via deliverable	P7
Agile Processes promote sustainable development. the sponsors, developers, and users should be able to maintain a constant Pace identifiability	Sustainability, People	P8
Continuous attention to technical excellence and good design enhance agility	Focus on technical excellence, good design as enabler of agility	P9
Simplicity – the art of maximizing the amount of work not done is essential	Simplicity – optimizing work	P10
The best architectures, requirements, and design emerge from self-organizing teams	Self-organization	P11
At regular interval, the team reflect on how to become more effective, then tunes and adjusts its behavior	Built-in improvement of efficiency	P12



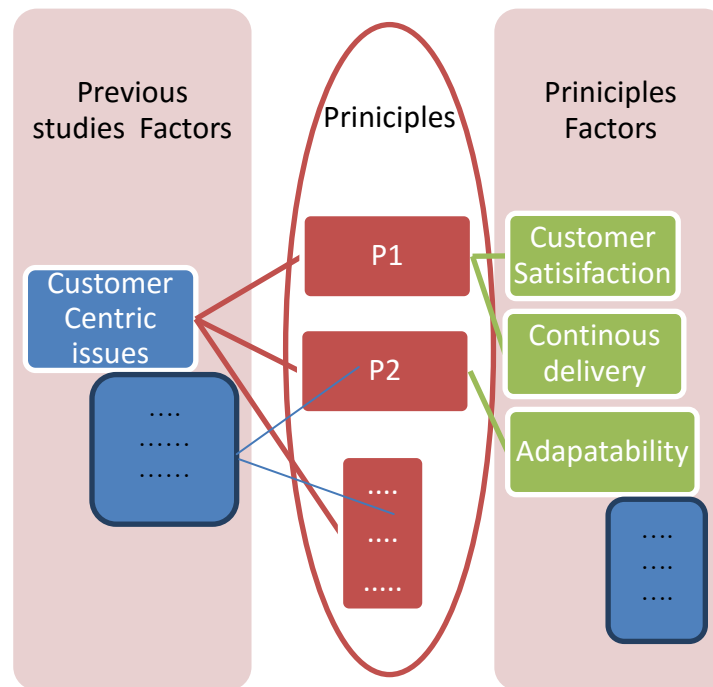


Fig. 9. Factors Matches Agile Principles

## V. CONCLUSION AND FUTURE WORK

This paper represents and identifies the success dimensions and factors for agile and how it matches the agile principles and concluded that few of factors resulted from people experts and companies' statistics and surveys. In the future, a quantitative study will be conducted by these factors to measure how these factors succeeded in Egypt companies.

In Process Dimension, the Factor (Effective requirement gathering method) is not explicitly appeared in P2 & P11, and also, continuous integration factor mentioned in the principles, but nothing mentioned about integration with external process and how adopting the agile in an old process, this means that gathering requirement and integrate with external process, is poorly identified in agile principles. In Process and Product Dimensions, there is no straightforward principle to clarify these factors, and all these factors result from qualitative and quantitative surveys resulting in turn from people and companies experiences.

Although there are some factors that are poorly identified, the remaining are clear, and all factors might have the probability of success and failure, but it depends on how, where, and when we use them in the organization.

We also suggest a framework to display the factors called (4P OT) which stand for People, Process, Product, Project, Organization, and Technical Model. In addition to another framework collect all dimensions with factors (CSDF) Model.

## REFERENCES

- [1] Li, Tong. An Approach to Modelling and Describing Software Evolution Processes. UK: Software Technology Research Laboratory - De Montfort U, 2007.
- [2] Sommerville, Ian. Software Engineering. 8th ed. Harlow, England: Addison-Wesley, 2007
- [3] Ahmed Awad, Mohamed. A Comparison between Agile and Traditional Software Development Methodologies. U of Western Australia
- [4] Ullah, Mehar. Comparison and Problems between Traditional and Agile Software Development Methods. Lappeenranta U of Technology, 2014
- [5] Fianta, Roman. Iterative Web Systems Development. Brno, Spring, 2009.
- [6] Boehm, B.W.(1988). A Spiral model of software development and enhancement. IEEE Computer, 21(5), 61-72. (Chs. 4, 5)
- [7] "Manifesto for Agile Software Development." Manifesto for Agile Software Development. 2001. Web. <<http://www.agilemanifesto.org/>>
- [8] J Eppler, Martin. "A Comparison between Concept Maps, Mind Maps, Conceptual Diagrams, and Visual Metaphors as Complementary Tools for Knowledge Construction and Sharing."Information Visualization 5 (2006): 202-10.
- [9] W. Belassi and O. I. Tukul, "A new framework for determining critical success/failure factors in projects," Int. J. Proj. Manag., vol. 14, no. 3, pp. 141-151, 1996.
- [10] A. Mahanti, "Challenges in enterprise adoption of agile methods-A survey," CIT. J. Comput. Inf. Technol., vol. 14, no. 3, pp. 197-206, 2006.
- [11] T. Chow and D.-B. Cao, "A survey study of critical success factors in agile software projects," J. Syst. Softw., vol. 81, no. 6, pp. 961-971, 2008.
- [12] Stankovic, D., et al., A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. Journal of Systems and Software, 2013. 86(6): p. 1663-1678.

- [13] S. C. Misra, V. Kumar, and U. Kumar, "Identifying some important success factors in adopting agile software development practices," *J. Syst. Softw.*, vol. 82, no. 11, pp. 1869–1890, 2009.
- [14] Z. Mansor Saadiah Yahya, and Noor Habibah Arshad, "Success Determinants in Agile Software Development Methodology\_ICMLC\_Updated." 2010.
- [15] J. Wan and R. Wang, "Empirical Research on Critical Success Factors of Agile Software Process Improvement," *J. Softw. Eng. Appl.*, vol. 3, no. 12, p. 1131, 2010.
- [16] C. Melo, D. S. Cruzes, F. Kon, and R. Conradi, "Agile team perceptions of productivity factors," in *Agile Conference (AGILE)*, 2011, 2011, pp. 57–66.
- [17] A. L. Asnawi, A. M. Gravel, and G. B. Wills, "Empirical investigation on agile methods usage: issues identified from early adopters in Malaysia," in *Agile Processes in Software Engineering and Extreme Programming*, Springer, 2011, pp. 192–207.
- [18] A. Kumar and B. Goel, "Factors Influencing Agile Practices: A Survey," 2012.
- [19] J. Sheffield and J. Lemétayer, "Factors associated with the software development agility of successful projects," *Int. J. Proj. Manag.*, vol. 31, no. 3, pp. 459–472, 2013.
- [20] Oluwole, Dele. "Agile X Factors." <https://www.scrumalliance.org>. Web. <<https://www.scrumalliance.org/community/articles/2013/december/agile-x-factors>>.
- [21] M. Coram and S. Bohner, "The Impact of Agile Methods on Software Project Management A Brief Look at Agile Methods," *Engineering*, pp. 363–370, 2005.
- [22] M. Sahota, *An Agile Adoption and Transformation Survival Guide: Working with Organizational Culture*. 2012.