Agile Software Development and the Incorporation of User-Centered Design

Ermioni Carr
Team: B-Positive
SI 501, Section 006, Liz Marquis
Client: Michigan Medicine—Health and Information Technology Services
Report Due Date: October 21st, 2019
Word Count: 2,120

Health and Information Technology Services (HITS) Mission and Product

The overall mission of HITS is to provide technical support and assistance to Michigan Medicine. This involves the development and management of a wide range of products, software and services that must be secure, reliable and valuable.[1] The specific product being analyzed here is a workflow management tool that handles the appointment, credentialing and enrollment of new faulty into the Medical School called "M-ACE". The long-term goal is to upgrade and integrate M-ACE into Meridian, which is a system that manages the current faculty at the Medical School. This would result in a single system that would be responsible for both the recruiting and hiring of new faculty and the management of current faculty.

The Problem

As HITS begins the process of integrating M-ACE into the Meridian software, they seek to first upgrade the functionality of the M-ACE software since it is considerably outdated. As each part of the process (appointment, credentialing, enrollment) is complex on its own, we will be focusing on the appointment process as this part has the most steps that must be completed and is also the first part in the overall process of hiring new faculty members. In order to upgrade the functionality of the appointment software, it is critical to understand all of the individual feature interactions and how these features are connected and support the larger business process of appointment. By doing so, project goals can be created for developers that will improve the functionality of the resulting system.

The Questions

The following questions will be addressed in this report:

1. What is agile software development?
2. How can user-centered design methodologies be used effectively within an agile software development framework?
3. How can agile software development and user-centered design be applied specifically to the client problem described above?

[1] HITS Mission, Vision & Values. (n.d.). Retrieved from https://hits.medicine.umich.edu/about-hits/hits-mission-vision-values

Introduction

The software industry is continuously evolving.[2] In order to keep up with the changing and increasing demands of software, the use of effective and efficient software development methods is imperative.[3] However, choosing the right method and implementing it effectively is much easier said than done. One factor that makes software development especially challenging is that it is very difficult for software developers to be aware of all of the software requirements at the beginning of the development process.[4] Therefore, software development methods must be adaptive to changing requirements. Also, these methods must consider how to create products that have a high degree of usability as this impacts both acceptance and the market potential of the software.[5] For clarity, in this paper, usability is referring to how the software will be used by end users.[6] Keeping in mind the importance of the software development process being adaptive and resulting in systems with high usability, this paper will discuss agile software development, how user-centered design methodologies can be incorporated into an agile framework and how HITS can leverage these methods as they upgrade the functionality of M-ACE.

Software Development Models

Waterfall and Spiral Models

At a high level, "Software Engineering (SE) is the systematic and manageable development of Software."[7] However, there are many proposed SE models. One such model is the Waterfall Model.[8] This model consists of seven stages completed linearly in which each stage influences the next.[9] As the stages are sequential, this model is likely most effective for projects that are able to be entirely defined at the beginning of the process.[10] Therefore, a benefit of the Waterfall Model is that the process is straight-forward compared to other SE models that tend to have a less certain development process.[11]

Another SE model is the Spiral Model, "which is an enhancement of the Waterfall Model."[12] Unlike the Waterfall Model, the Spiral Model is iterative, incremental and includes four stages

[2] Sohaib, O., & Khan, K. (2010). Integrating usability engineering and agile software development: A literature review. *2010 International Conference on Computer Design and Applications.* Qinhuangdao, China: IEEE.
[3] Sohaib, & Khan, Ibid.
[4] Düchting, M., Zimmermann, D., & Nebe, K. (2007). Incorporating user centered requirement engineering into agile software development. In J. A. Jacko (Ed.), *Human-Computer Interaction. Interaction Design and Usability* (pp. 58-67). Berlin, Heidelberg: Springer.
[5] Bai, A., Mork, H., & Stray, V. (2018). How agile teams regard and practice universal design during software development. *Studies in Health Technology and Informatics, 256,* 171-184.
[6] Sohaib, & Khan, Ibid.
[7] Düchting, Zimmermann, & Nebe, Ibid.
[8] Düchting, Zimmermann, & Nebe, Ibid.
[9] Düchting, Zimmermann, & Nebe, Ibid.
[10] Düchting, Zimmermann, & Nebe, Ibid.
[11] Abbas, N., Gravell, A. M., & Wills, G. B. (2008). Historical roots of agile methods: where did "agile thinking" come from? In P. Abrahamsson, R. Baskerville, K. Conbo, B. Fitzgerald, L. Morgan, & X. Wang (Eds.), *Agile Processes in Software Engineering and Extreme Programming. XP 2008* (pp 94-103). Berlin, Heidelberg: Springer.
[12] Düchting, Zimmermann, & Nebe, Ibid.

that are completed in a cyclical manner.[13] At the end of each cycle, there is a planning phase that informs the upcoming iteration.[14] As a whole, the Spiral Model allows for more adaptation to new or changed requirements than the Waterfall Model.[15]

Introduction of Agile

With these two SE models in mind, one could consider the introduction of agile as a "reaction to traditional methods."[16] Similar to the Spiral Model, Agile is iterative and incremental; however, with agile, the iterations are shorter.[17] Also, in contrast to the use of documentation in the other SE models, agile methods focus on communication in order to attain continuous feedback.[18] With this in mind, agile methods should be thought of similarly to the invention process.[19] Comparable to the start of the invention process, it is almost impossible to know all the requirements at the beginning of the development process; therefore, with agile, planning at the beginning of the process is limited and the process is subject to change as requirements do.[20]

Although the components of agile have actually been around for some time, the unique combination of them into what makes up agile is new.[21] Since the beginning of agile, it has gained much attention among development teams and organizations.[22] Today, agile is the most commonly used development process among software developers.[23] The next sections will discuss key elements of agile software development, two of the SE agile models and some benefits and limitations of agile.

Elements of Agile Software Development

What are the elements of agile software development that make it so desirable among software development teams? Beyer, Holtzblatt, & Baker (2004) define the four axioms of agile methods as the following: "Separate Design from Engineering", "Make the User the Expert", "Keep Up-Front Planning to a Minimum", and "Work in Quick Iterations"[24] Separating design from engineering allows developers to focus on writing high-quality code while designers work with

[13] Düchting, M., Zimmermann, D., & Nebe, K. (2007). Incorporating user centered requirement engineering into agile software development. In J. A. Jacko (Ed.), *Human-Computer Interaction. Interaction Design and Usability* (pp. 58-67). Berlin, Heidelberg: Springer.
[14] Düchting, Zimmermann, & Nebe, Ibid.
[15] Düchting, Zimmermann, & Nebe, Ibid.
[16] Abbas, N., Gravell, A. M., & Wills, G. B. (2008). Historical roots of agile methods: where did "agile thinking" come from? In P. Abrahamsson, R. Baskerville, K. Conbo, B. Fitzgerald, L. Morgan, & X. Wang (Eds.), *Agile Processes in Software Engineering and Extreme Programming. XP 2008* (pp 94-103). Berlin, Heidelberg: Springer.
[17] Düchting, Zimmermann, & Nebe, Ibid.
[18] Düchting, Zimmermann, & Nebe, Ibid.
[19] Düchting, Zimmermann, & Nebe, Ibid.
[20] Düchting, Zimmermann, & Nebe, Ibid.
[21] Abbas, Gravell, & Wills, Ibid.
[22] Düchting, Zimmermann, & Nebe, Ibid.
[23] Bai, A., Mork, H., & Stray, V. (2018). How agile teams regard and practice universal design during software development. *Studies in Health Technology and Informatics, 256,* 171-184.
[24] Beyer, H., Holtzblatt, K., & Baker, L. (2004). An agile customer-centered method: Rapid contextual design. In C. Zannier, H. Erdogmus, & L. Lindstrom (Eds.), *Extreme Programming and Agile Methods - XP/Agile Universe 2004* (pp. 50-59). Berlin, Heidelberg: Springer.

the users to develop the interface and address issues of usability.[25] Making the user the expert stresses the importance of the user in defining what a "good system" entails.[26] Keeping up-front planning to a minimum encourages the software development team to be adaptive and responsive to user feedback.[27] Lastly, working in quick iterations allows the development teams to work in short iterations to develop code and get feedback in a timely manner, which is a commonly cited problem with other methods.[28]

Similarly, Abbas, Gravel, & Willis (2008) describe the main features of agile methods as "Adaptive", "Iterative and incremental" and "People-oriented."[29] Compared to the axioms, there is much overlap. For example, "Making the User the Expert" and being "People-oriented" are very similar, as are "Work in Quick Iterations" and "Iterative and incremental."

The Agile Manifesto outlines twelve principles that complement the axioms and features described above.[30] For example, one of the principles is to "Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage."[31] This corresponds to the axiom of "Keep Up-Front Planning to a Minimum" and to the feature of being "Adaptive" as it stresses the importance of being welcoming and responsive to change at any point in the process. Table 1 outlines all twelve principles of the agile manifesto along with evidence on whether or not the principle is new.[32] As evident, there is much consensus on what constitutes the key elements of agile software development.

Agile Methods

Two SE models that leverage agile methodology are Scrum and Extreme Programming (XP).[33] Aligned with agile methodology, both of these models are iterative and incremental in nature.[34] However, there are some differences between the models.

[25] Beyer, H., Holtzblatt, K., & Baker, L. (2004). An agile customer-centered method: Rapid contextual design. In C. Zannier, H. Erdogmus, & L. Lindstrom (Eds.), *Extreme Programming and Agile Methods - XP/Agile Universe 2004* (pp. 50-59). Berlin, Heidelberg: Springer.

[26] Beyer, Holtzblatt, & Baker, Ibid.

[27] Beyer, Holtzblatt, & Baker, Ibid.

[28] Beyer, Holtzblatt, & Baker, Ibid.

[29] Abbas, N., Gravell, A. M., & Wills, G. B. (2008). Historical roots of agile methods: where did "agile thinking" come from? In P. Abrahamsson, R. Baskerville, K. Conbo, B. Fitzgerald, L. Morgan, & X. Wang (Eds.), *Agile Processes in Software Engineering and Extreme Programming. XP 2008* (pp 94-103). Berlin, Heidelberg: Springer.

[30] Abbas, Gravell, & Wills, Ibid.

[31] Abbas, Gravell, & Wills, Ibid.

[32] Abbas, Gravell, & Wills, Ibid.

[33] Düchting, M., Zimmermann, D., & Nebe, K. (2007). Incorporating user centered requirement engineering into agile software development. In J. A. Jacko (Ed.), *Human-Computer Interaction. Interaction Design and Usability* (pp. 58-67). Berlin, Heidelberg: Springer.

[34] Düchting, Zimmermann, & Nebe, Ibid.

Scrum Model

Under the Scrum model, scrum teams of around 7-9 people are created with a Scrum Master who acts as the manager.[35] The iterations are referred to as Sprints and are typically 30 days each.[36] Each Sprint includes the following: Sprint Planning, Daily Scrum, and Sprint Review.[37] Sprint Planning involves picking requirements for the upcoming sprint from the product backlog, which is where the software requirements are defined and prioritized.[38] The Daily Scrum allows for status updates on progress.[39] The Sprint then ends with a Sprint Review where updates are presented.[40]

Extreme Programming (XP) Model

Although XP has some similarities to Scrum, there are different elements as well. First, the iterations are shorter than Scrum as they are typically only 1-3 weeks.[41] Also, the XP model stresses programming as a team; specifically, it utilizes pair programming which involves two programmers working closely together and embraces the common code ownership principle in which editing others code is allowed.[42] In addition, when using the XP model, a user is present at the development organization in order to allow for frequent feedback in a timely manner.[43] From this feedback, user stories are generated to determine software requirements; these stories are later broken into more technical components and are prioritized similarly to Scrum.[44] As evident, Scrum and XP both utilize Agile principles but in slightly different manners.

Benefits of Agile

One of the benefits of agile methods is that it allows developers to leverage their skills: they can focus on coding since other members of the organization are responsible for working with the users.[45] Another related benefit is that agile methods contribute to the rapid creation of high-quality code.[46] In addition, the adaptability of agile methods is a major benefit given the unpredictable nature of the business environment today.[47] To further this point, Abbas, Gravell, & Wills (2008) state that the majority of changes to software requirements happen during the

[35] Düchting, M., Zimmermann, D., & Nebe, K. (2007). Incorporating user centered requirement engineering into agile software development. In J. A. Jacko (Ed.), *Human-Computer Interaction. Interaction Design and Usability* (pp. 58-67). Berlin, Heidelberg: Springer.
[36] Düchting, Zimmermann, & Nebe, Ibid.
[37] Düchting, Zimmermann, & Nebe, Ibid.
[38] Düchting, Zimmermann, & Nebe, Ibid.
[39] Düchting, Zimmermann, & Nebe, Ibid.
[40] Düchting, Zimmermann, & Nebe, Ibid.
[41] Düchting, Zimmermann, & Nebe, Ibid.
[42] Düchting, Zimmermann, & Nebe, Ibid.
[43] Düchting, Zimmermann, & Nebe, Ibid.
[44] Düchting, Zimmermann, & Nebe, Ibid.
[45] Beyer, H., Holtzblatt, K., & Baker, L. (2004). An agile customer-centered method: Rapid contextual design. In C. Zannier, H. Erdogmus, & L. Lindstrom (Eds.), *Extreme Programming and Agile Methods - XP/Agile Universe 2004* (pp. 50-59). Berlin, Heidelberg: Springer.
[46] Beyer, Holtzblatt, & Baker, Ibid.
[47] Beyer, Holtzblatt, & Baker, Ibid.

development process.[48] Therefore, the ability to adapt and react to changes in a timely manner is a major advantage of agile methods compared to other design methodologies in which the inability to adapt is cited as one of the main causes of failures.[49] Lastly, it is argued that usability and agile methods are a natural fit.[50] This is important since as stated earlier, usability impacts if customers will use the software, as well as the market potential of the software.[51]

Limitations of Agile

However, there are also limitations with agile methods. First, there are some concerns about the ability of agile methods to scale effectively.[52] In addition, even though developers are able to produce high-quality code in a rapid manner, they may not always get good direction on what they should be doing.[53] This responsibility falls on the customer, which is why it is important to leverage effective user-centered design methods.[54] However, it is also not well specified how to integrate such methods with agile.[55] With this in mind, the next section will discuss two user-centered design methods and why they may fit well within an agile framework.

Incorporating User-Centered Design into the Agile Framework

There is consensus that user feedback is essential in the agile software development framework.[56] Therefore, integrating user-centered design methods seems appropriate as they make users the central focus in the design process.[57] By working directly with users and utilizing user testing throughout the design process, the requirements of software can be defined and updated as they may change.[58] In regard to agile, this can result in many advantages such as appropriate prioritization of system requirements and increased usability.[59] Although

[48] Abbas, N., Gravell, A. M., & Wills, G. B. (2008). Historical roots of agile methods: where did "agile thinking" come from? In P. Abrahamsson, R. Baskerville, K. Conbo, B. Fitzgerald, L. Morgan, & X. Wang (Eds.), *Agile Processes in Software Engineering and Extreme Programming. XP 2008* (pp 94-103). Berlin, Heidelberg: Springer.

[49] Düchting, M., Zimmermann, D., & Nebe, K. (2007). Incorporating user centered requirement engineering into agile software development. In J. A. Jacko (Ed.), *Human-Computer Interaction. Interaction Design and Usability* (pp. 58-67). Berlin, Heidelberg: Springer.

[50] Sohaib, O., & Khan, K. (2010). Integrating usability engineering and agile software development: A literature review. *2010 International Conference on Computer Design and Applications.* Qinhuangdao, China: IEEE.

[51] Bai, A., Mork, H., & Stray, V. (2018). How agile teams regard and practice universal design during software development. *Studies in Health Technology and Informatics, 256,* 171-184.

[52] Beyer, H., Holtzblatt, K., & Baker, L. (2004). An agile customer-centered method: Rapid contextual design. In C. Zannier, H. Erdogmus, & L. Lindstrom (Eds.), *Extreme Programming and Agile Methods - XP/Agile Universe 2004* (pp. 50-59). Berlin, Heidelberg: Springer.

[53] Beyer, Holtzblatt, & Baker, Ibid.

[54] Beyer, Holtzblatt, & Baker, Ibid.

[55] Beyer, Holtzblatt, & Baker, Ibid.

[56] Beyer, Holtzblatt, & Baker, Ibid.

[57] Humayoun, S. R., Dubinsky, Y., & Catarci, T. (2011). A three-fold integration framework to incorporate user–centered design into agile software development. In M. Kurosu (Ed.), *Human Centered Design* (pp 55-64). Berlin, Heidelberg: Springer.

[58] Najafi, M., & Toyoshiba, L. (2008). Two case studies of user experience design and agile development. *Agile 2008 Conference*. Toronto, Canada: IEEE.

[59] Humayoun, Dubinsky, & Catarci, Ibid., Najafi, & Toyoshiba, Ibid.

development organizations may be concerned about this lengthening the design process, it has been shown that incorporating user-centered design has almost no effect on the timing of releases.[60] Two examples of these methodologies will be introduced below: Rapid Contextual Design (CD) and Usability Engineering (UE).

Rapid Contextual Design (CD)

CD is a methodology for identifying and understanding the workflow and needs of users.[61] Although CD and agile have some differences in their approaches, they appear to work well together.[62] For example, considering the four axioms of agile described earlier, CD encourages the separation of design from engineering since there is a distinct user-interface design team.[63] Furthermore, CD takes the agile principle of getting early feedback throughout the design process even further by suggesting that user testing should begin even before any coding occurs.[64] This can help limit any unnecessary work, which helps developers conserve their time and effort.[65] With this in mind, Rapid CD is a modified version of CD that "is a fast, effective, customer-centered way to design a product when quick turnaround is desired."[66] Therefore, Rapid CD includes modifications to CD that may make it a better fit with agile methodology as it allows for more rapid development.[67] However, in the cases of more intricate projects, a typical CD approach may be necessary in order to attain enough detail.[68]

Usability Engineering (UE)

UE methods provide strategies for incorporating user-centered design into agile with the end-goal of producing software that has a high level of usability.[69] UE consists of three sequential stages: "Requirement Analysis", "Design/Testing/Development Phase," and "Installation of the product."[70] Similar to agile, the stages are completed in an iterative manner and are reliant upon user engagement.[71] Also, as both agile and UE are centered around requirements and deliverables, this presents a uniting point for the two methodologies.[72] Strategies for integrating UE more effectively into an agile framework include testing consistently throughout the development process rather than just during the second phase of UE and "incorporating user

[60] Najafi, M., & Toyoshiba, L. (2008). Two case studies of user experience design and agile development. *Agile 2008 Conference*. Toronto, Canada: IEEE.
[61] Beyer, H., Holtzblatt, K., & Baker, L. (2004). An agile customer-centered method: Rapid contextual design. In C. Zannier, H. Erdogmus, & L. Lindstrom (Eds.), *Extreme Programming and Agile Methods - XP/Agile Universe 2004* (pp. 50-59). Berlin, Heidelberg: Springer.
[62] Beyer, Holtzblatt, & Baker, Ibid.
[63] Beyer, Holtzblatt, & Baker, Ibid.
[64] Beyer, Holtzblatt, & Baker, Ibid.
[65] Beyer, Holtzblatt, & Baker, Ibid.
[66] Beyer, Holtzblatt, & Baker, Ibid.
[67] Beyer, Holtzblatt, & Baker, Ibid.
[68] Beyer, Holtzblatt, & Baker, Ibid.
[69] Düchting, M., Zimmermann, D., & Nebe, K. (2007). Incorporating user centered requirement engineering into agile software development. In J. A. Jacko (Ed.), *Human-Computer Interaction. Interaction Design and Usability* (pp. 58-67). Berlin, Heidelberg: Springer.
[70] Düchting, Zimmermann, & Nebe, Ibid.
[71] Düchting, Zimmermann, & Nebe, Ibid.
[72] Düchting, Zimmermann, & Nebe, Ibid..

scenarios and including usability specialists in the team," which are UE concepts.[73] Overall, usability and agile appear to align well in many aspects.[74] Table 2 provides a more complete comparison between agile and UE along with suggested approaches for integration.[75]

HITS: Using Agile Development and User-Centered Design

How can HITS incorporate user feedback to ensure that the resulting M-ACE software supports an efficient workflow and has a high level of usability? As M-ACE has been in use for many years now, users of the system are very familiar with the current system. Therefore, these users have likely developed opinions about which features work well and which features could be improved. As these users are experts in using the software, HITS should use this to their advantage by forming a group of expert users to provide feedback to guide each iteration. This will likely result in higher user satisfaction and a smoother software release process. In order to effectively incorporate the users' feedback, user-centered design methodologies, such as Rapid CD or UE, can be used to facilitate the process.

Conclusion

As discussed in this paper, agile software development is a design methodology that is iterative, incremental and emphasizes user involvement and adaptability. By utilizing user-centered design methodologies within an agile framework, user feedback can be incorporated into the design process effectively. Two such methodologies are Rapid Contextual Design and Usability Engineering. HITS can utilize agile and user-centered design methodologies as they upgrade the functionality of M-ACE in order to incorporate user feedback into project goals that will improve the functionality of the system. Overall, the use of methodologies that support development teams in fulfilling software requirements, even as they may change throughout the process, is necessary.

[73] Sohaib, O., & Khan, K. (2010). Integrating usability engineering and agile software development: A literature review. *2010 International Conference on Computer Design and Applications.* Qinhuangdao, China: IEEE.
[74] Sohaib, & Khan, Ibid.
[75] Sohaib, & Khan, Ibid.

Bibliography

Abbas, N., Gravell, A. M., & Wills, G. B. (2008). Historical roots of agile methods: where did "agile thinking" come from? In P. Abrahamsson, R. Baskerville, K. Conbo, B. Fitzgerald, L. Morgan, & X. Wang (Eds.), *Agile Processes in Software Engineering and Extreme Programming. XP 2008* (pp 94-103). Berlin, Heidelberg: Springer.

Bai, A., Mork, H., & Stray, V. (2018). How agile teams regard and practice universal design during software development. *Studies in Health Technology and Informatics, 256,* 171-184.

Beyer, H., Holtzblatt, K., & Baker, L. (2004). An agile customer-centered method: Rapid contextual design. In C. Zannier, H. Erdogmus, & L. Lindstrom (Eds.), *Extreme Programming and Agile Methods - XP/Agile Universe 2004* (pp. 50-59). Berlin, Heidelberg: Springer.

Düchting, M., Zimmermann, D., & Nebe, K. (2007). Incorporating user centered requirement engineering into agile software development. In J. A. Jacko (Ed.), *Human-Computer Interaction. Interaction Design and Usability* (pp. 58-67). Berlin, Heidelberg: Springer.

HITS Mission, Vision & Values. (n.d.). Retrieved from https://hits.medicine.umich.edu/about-hits/hits-mission-vision-values

Humayoun, S. R., Dubinsky, Y., & Catarci, T. (2011). A three-fold integration framework to incorporate user–centered design into agile software development. In M. Kurosu (Ed.), *Human Centered Design* (pp 55-64). Berlin, Heidelberg: Springer.

Najafi, M., & Toyoshiba, L. (2008). Two case studies of user experience design and agile development. *Agile 2008 Conference*. Toronto, Canada: IEEE.

Sohaib, O., & Khan, K. (2010). Integrating usability engineering and agile software development: A literature review. *2010 International Conference on Computer Design and Applications.* Qinhuangdao, China: IEEE.

Appendix

Table 1

Principle	**New or not with Evidence**
Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	EVO first principle: deliver something to the real end-user [17]
Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	Relatively new, the problem always existed but without a real solution
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	In EVO the frequent and early delivery is essential , also RAD recommended quick delivery [17, 24]
Business people and developers must work together daily throughout the project.	Relatively new as some approaches recommended good relation with customer, however, the idea of daily communication and on-site customer is new
Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	These ideas were raised in the psychology of computer programming book which was published in 1985; the author empathized on the importance of motivation which is the inner directing force (chp10). In addition, he mentioned that the richness of the environment gives it a self–maintaining quality which resists the imposed changes.(chp4) [37]
The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	The previous book focused on the importance of how the working space can affects the social interaction which in turn will affect the work. The author emphasized on how face to face communication helps transmitting useful information [37]
Working software is the primary measure of progress.	EVO second principle: measure the added- value to the user in all criteria dimensions [17]
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	In the Death March book, Edward Yourdon pointed the importance of managing and controlling progress and he suggested the "daily build" concept to succeed that mission [39]

Continuous attention to technical excellence and good design enhances agility.	Probably we could find the same idea of the importance of doing a much better programming job (technical excellence) in Dijkstra' s famous article "Humble programmer" [40]
Simplicity the art of maximizing the amount of work not done--is essential.	The famous saying on simplicity of design comes from Antione de Saint-Exupery: "Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away". [30]
The best architectures, requirements, and designs emerge from self-organizing teams.	We could find the idea of self-organizing team in open source projects which were out roughly at the same time as Agile methods. In the Cathedral and the Bazaar paper, Raymond referred to the developers as people bring their own resources to the table [28]
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.	The idea of process improvements was presented in CMMI level 5 with different emphasize as in Agile all the team will reflect on improving the process not only the management [36]

From Abbas, N., Gravell, A. M., Wills, G. B. (2008). Historical roots of agile methods: where did "agile thinking" come from? In P. Abrahamsson, R. Baskerville, K. Conbo, B. Fitzgerald, L. Morgan, & X. Wang (Eds.), *Agile Processes in Software Engineering and Extreme Programming. XP 2008* (pp 94-103). Berlin, Heidelberg: Springer.

Table 2

Agile Methods Concepts	**Usability Engineering**	**Suggested Approach**
Deliver working software frequently	Traditional software approach but iteration within phases	Iterative development throughout the project
Requires generalists	Requires specialists	Assemble a multidisciplinary team to ensure complete expertise
Customer focus	User focus	Collaboration between customers, users, product managers, Business analysts, developers, will maximize overall team efficiency for usable product
Test driven development and continuous integration	Contextual inquiry, field surveys, usability inspection methods for testing	Unit Testing + User Acceptance Testing + Usability testing throughout the process
Using onsite customer, functional requirement are encapsulated as user stories	Scenario based design for requirement analysis	Integrate user stories with scenario based design

From Sohaib, O., & Khan, K. (2010). Integrating usability engineering and agile software development: A literature review. *2010 International Conference on Computer Design and Applications.* Qinhuangdao, China: IEEE.