Agile software development costs – Nobody questions success

Fawaz Alsafadi1, Ayman El Gendy1, Odhran Byrne-Gildea1, John Duffy1

1. School of Computing, Dublin City University, Glasnevin, Dublin, Ireland

{fawaz.alsafadi2, john.duffy34, ayman.elgendy2, odhran.byrnegildea2}

@mail.dcu.ie

**Abstract**. The agile manifesto released in 2001 was an answer to the changing landscape of software development. The size and functionality of modern software required a different model that adapted and evolved throughout the lifecycle of the project, agile development provided the methods and tools to deal with the dynamic nature of these projects. However, as with the importance of creating and delivering software; the cost of a project is an equally important topic. Correct analysis and estimation of cost can determine the outcome of a project to be a success or failure. In order to better understand the cost of agile projects the paper first analyses cost estimation methods currently in use that have been tailored to suit the dynamic nature of agile. The role of the project manager is then examined to understand the impact their management has on cost as well as political and external factors and how these affect decisions made by the project team. Agile allows for frequent changes and revisions of the requirements, the paper examines the consequence of these factors and seeks to analyse the effect on the cost of projects. This paper will be useful for agile users to understand the effect of agile development on project costs.

**Keywords**: Agile software development, Cost, Agile Cost estimation.

# **Introduction**

In modern times since the formation of the agile alliance and the release of the Agile manifesto [26] most software companies have shifted to agile. Agile Software development refers to any software or project that follows the set methods, processes and principles that emphasize adaptive planning, iterative development and cross-communications between teams and the customer; as set out in the agile manifesto [28], [26]. This recent uptake in agile methodology can be traced back to the difficulty to specify and outline all requirements at the start of modern software development (SD) projects [29]. In cases such as these the requirements of the SD project are updated and modified along the duration of the project. As traditional methods of development need clearly defined requirements as a prerequisite [29] traditional models become insufficient and lacking in modern SD projects. Due to the dynamically changing requirements and the need to manage them agile methodology is adopted. However, unlike traditional well documented and researched methods few studies on agile are based on strong theoretical or conceptual foundations, in the absences of this research fragmented and inconclusive knowledge arises which proves problematic for project managers trained with traditional plan-driven development [31].

While the aforementioned lack of conclusive knowledge leads to multiple problems, the problem of accurate cost estimation and planning is among the most significant. In most cases the cost factor is forecasting expected time, effort, manpower and finance that are needed to see the project to completion [33]. Estimating the factors mentioned is central to planning and management of a software project as underestimates will cause schedule pressure which in turn will lead to higher defects and costs; and overestimation will cause waste of resources [3]. Estimation is a challenge for every software project particularly however, owing to the dynamic nature of agile development cost estimation becomes problematic and thus true estimates of the cost of agile become hard to realize [31]. Particularly factors that cause estimation errors are;frequent requirement changes and incomplete ill-defined requirements, complexity, lack of expertise and users lack of understanding of the requirements [28]. As reported by Standish Group CHAOS 2006 nearly 60% of all software projects are either failed or challenged with cost and time overruns.

This paper will seek to further examine the issues surrounding cost estimation of agile projects and attempt to determine whether the cost is positively or negatively impacted by the adoption of agile methodologies. The paper will closely examine how the dynamic nature of agile affects project scope and in turn analyse costs and cost estimation as a result of the constant change. The role of project managers and their interaction with users’ needs will be analysed in order to determine how the close and frequent contact between project and customer affects estimation and costs over the duration of the project. Political pressure and influence will be analysed in order to review how matters outside the control of the project managers and development team affect the outcome of the project and costs. The paper will closely examine various case studies and reports of companies and projects successfully integrating agile and meeting targets as well as projects overrunning their budgets and cost estimates to gain a better understanding of the negatives and positives that led to each outcome.

# **Related literature**

The approach used in finding literature related to this paper was both entering queries on google.com and google scholar. If a paper was found to be relevant and some information was used from it, it was then included in the list of references. Also, if cited papers found within papers were deemed to be relevant, these would also be included in the references.

Each member of the team took their topics and researched to gain a better understanding of the topic. They then went on to write relevant information on their topics, taking views and opinions from other sources and referencing. Caution is advised when referencing non-peer reviewed papers.

Relevance was the most important factor for us to consider whether a paper be included or not. For a paper or study to be included it was decided that it had to be about a similar topic or aspect of the topic that the paper was specifically to be about. If a paper wasn’t peer reviewed then it was more likely not to be included.

# **Analysis**

**3.1 Research**

One of the toughest tasks that project managers deal with is estimating the cost of an IST project. It proves to be a constant flaw in creating new information technology designs. It is regularly reported that new creations are overpriced, above budget and past their release date. Bloch, Blumberg, and Laartz (2012) discovered through research in association with Oxford University that on average large IT projects defined as those with initial price tags exceeding $15 million massively blow their budget. They revealed that large IT projects run 45 percent over budget and 7 percent over budget while delivering 56 percent less value then predicted [1]. Software projects run the highest risk of cost and schedule overruns. This fact is compounded by information found by Flyvbjerg and Budzier (2011) which details the revamping of the technology system for the world-famous jean supplier Levi Strauss. The proposed cost of this revamping was $5 million dollars however due to Walmart requiring that the new system interface worked with its supply chain management system and insufficient procedures for financial reporting meant that Levi’s took a loss of $192.5 million loss against earnings to atone for this project [2]. We find some insight into why these losses occur in a recent study by QSM Associates Michael Mah (2001) of more than 200 software development projects. The findings report that the researchers could not find nearly half of the project's original plans to measure against. The reason for this is that modern projects are no longer focusing on their primary goal. The focus is now on satisfying the customers at the time of delivery, not at project initiation [3]. We decided to look more into why these problems and massive over costs happen. According to John S. Reel of Trident Data Systems (1999) the main reasons why IST projects go over budget is Project managers don’t understand the user's needs, the project's scope is ill-defined, project changes are managed poorly and politics [4].

**3.2 Benefits of agile methodology**

Agile aims to provide a solution to the issue that customer requirements will change over time [11]. A primary benefit of Agile methods is that it allows for rapid response to changes in software requirements without an overbearing workload. Agile software development methods provide a way of dealing with the world of businesses asking for adjustable and fast software development processes [12]. This requirement placed on agile software development projects is a major cause for cost estimation inaccuracies which will be discussed later.

Using Agile, new software releases can be released frequently, e.g. weekly, bi-monthly. It prioritises working software over large amounts of documentation. This usually results in developers writing code that is simple, straightforward and as technical as possible, thus reducing the amount of documentation needed to an appropriate level [12].

The simplicity involved in Agile development is one of its core appealing features. It enables teams to output the quality products while also requiring the least amount of work. The continuous communication between teams allows for effective tackling of new requirements in a straightforward way.

Furthermore, the constant communication with the customer allows for frequent feedback from the customers and users [11]. This communication allows customers to refine requirements throughout the project lifecycle, however frequent changes to requirements consistently lead to inaccurate cost and time estimates.

**3.3 Cost Estimation**

Cost estimation is essentially forecasting the expected time, effort, manpower and finance to complete a project [33]. The cost estimation is a product of the planning phase of a project. Traditionally the estimation was done at the beginning of a project, however for agile the planning and estimation must be done progressively as it is difficult to predict the size of the project since the requirements change during the development of the project [29]. As the agile development technique is based on unconventional concepts, projects following these guidelines are not suited for any of the traditional estimation methods [34]. Using these traditional estimation methods in agile projects would results in considerable inaccuracies due to the lack of historical data [35]. Hence the adoption of new estimation methods is critical to insure the accurate estimation of agile projects. Cost estimation in agile projects has been performed through various mechanisms such as Expert judgement, neural networks, planning poker, linear regression, Wideband delph and many more. Below is a description of each mechanism mentioned.

*(i) Neural Networks (NN)*

NNs are loose simulations of the nervous system, they are designed to recognise patterns and are used for approximations and classification of various functions. There is a high dependency of software cost estimation on multiple factors that are not linearly related thus most dominant factors need to be identified and passed as input to the model being used. The uncertain nature of these inputs is an advantage as NNs produce the best results under such conditions and provide considerably accurate estimation [29], [36].

*(ii) Expert judgement*

This method draws on the knowledge and experience of senior people working in the software industry and in software projects, who have over time gained rich experience regarding estimation of various projects. They have developed and acquired sound knowledge having been involved in numerous projects throughout their work period. As such expert opinion is often sought out to analyse cost and time estimates for projects. It is important to also be aware of the tendency for the expert judgement method to be influenced by various subjective elements such as bias, the project, influence of work environment, political factors and human error. However, it is a proven efficient method and successful for estimating small and medium-sized projects [37].

*(iii) Planning poker*

This technique involves the analysis of each user requirement individually to create estimates. It is carried out by having the project team discuss requirements to produce estimates and compare them to reach an agreement. If disagreements of estimates occur within the team then these discrepancies are further discussed to find a consensus otherwise the estimates are accepted and the analysis of the next requirement begins [29].

*(iv) Linear regression*

Linear regression is a statistical technique that can be used to estimate costs based on past cost data. The existing data can be used to construct regression models. Due to the cost of development depending on multiple factors multiple linear regression models can be used for representing interrelationships between the different factors. For the estimations from this method to be reliable the data passed needs to be stable and continuous with few missing data. The assumptions need then be verified in order to be applied to the current project.

*(v) Wideband Delphi*

Wideband Delphi is a variation of the planning poker method, it also focuses on and emphasizes interaction between team members. The method is designed to encourage discussion amongst the team to reach a consensus. The team dealing with the project meet to discuss and decide a breakdown structure of all the necessary components to be estimated. The individuals are then encouraged to perform individual estimations for their tasks and then a report is distributed amongst everyone for further discussion and revision. The integrated tasks and estimations are then reviewed for relevance. This method is proven to be reliable and straightforward to implement as it attempts to be as objective as possible due to the agreement of the multiple team members, also taking experts’ opinion ensures this method's quality [38].

The methods outlined above are some of the most popular cost estimation techniques employed in agile development projects. For each individual project the team members should consider each method and their equivalent accuracies and choose the best methods for the project.

**3.4 Project managers don’t understand the user’s needs**

According to (Brown & Eisenhardt, 1995; Katz, 1982; Katz & Allen, 1985; Keller, 1994) in regard to the domain of research and development project work evidence supports the view that positive relationships between internal task-related communication and project performance exist. Agile methods of development have been introduced as ways to shorten the development time along with improving communication. This time shortening is very important to any software development project. It can significantly drop costs if done right however it is not always done in the correct way. (Anderson 2003) wrote that the experiments and surveys on agile methods promise faster development thus improving the communication and collaboration inside agile teams and within teams, customers and business units [6]. Trying to combat the problem of users not knowing or understanding their own requirements, their lack of information technology knowledge and incorrect or inadequate problem definition is a big factor to inaccurate cost estimates. By improving the game of cooperation with different methods, can help keep down the cost of development. With it being common knowledge that the more frequent the communication between all parties the more productive and therefore cheaper the final project will be.

Lederer and Prassad (1995) go into more detail about the causes of Inaccurate Software Development Cost Estimates where they say user communication contains items reflecting the shortcomings of users in their work with information systems analysts; the factor is so named because these shortcomings of the user such as their lack of understanding are often attributed to ineffective communications with them [7]. While Clavadetscher (1998) writes that users should have a requirement control board maintaining a baseline of user requirements, they should have automated requirements tracing and tracking tools to help them manage. This requires the user to give valuable and useful information to the team that can be worked with and not lead them in the wrong direction. It is therefore recommended that a significant portion of time should be allocated to gaining knowledge from the user [8]. It is key to build a strong customer developer relationship so that valid knowledge can be exchanged. This is why agile practises are now recognised by many companies as a way to reduce costs and to respond to the conditions of a dynamic market. This is backed up in a paper by (VIJAYASARATHY and TURK) where they discuss factors that influence agile use. They mention that the most highly rated factors by respondents were corporate/team culture, improved communication and collaboration, availability of required skillsets and the failure of traditional development methods such as the waterfall method. [9]

Năftănăilă Ionel (2009) details how the main problems emerge in communication in agile software development and in turn how the project manager doesn't understand the end users' needs. She states that although communication is considered to make software development more efficient we start to see problems when the communication is not handled correctly. The problems start to occur when each member of the development team ranging from the developer to the user starts viewing and discussing the same issue from different perspectives [10]. For example users require the product to have a large degree of usability, customers seek reliability and low maintenance costs, as well as fast time-to-market, managers seek minimizing costs, maintenance teams seek documentation and reliability, while the development team seek technical challenges and moving to the next project (Boehm & Ross, 1989) [13]. This is one of the reasons why Alistair Cockburn who is one of the co-authors of the agile Manifesto and the creator of the agile movement emphasises that communication is one of the key challenges that needs to be done correctly for agile software development to be done in a cost efficient and correct manner and therefore creates a better final product [3].

**3.5 Project scope is ill defined**

When starting a project, it is of the utmost importance that the project idea is correctly filtered down from owner to developer to customer. If the developers don't have a correct idea of what they are designing then the project is on a straight path to become over budget. According to the Standish Group (1995) who are a primary research advisory organization that focuses on software project performance. They stated that the U.S. government and businesses spent approximately $81 billion on cancelled software projects and another $59 billion for budget overruns [19]. Not all of this is attributed to just ill-defined project scope, however Watts Humphrey claimed that many of the projects he worked on were hampered by people who were directionless in their work. Cockburn and Highsmith (2001) go into detail on how to utilize agile methods to help with feature planning and dynamic prioritization. Dynamic prioritization means that at the end of every iteration, the customer can reprioritize the features desired in the next cycle. This can help with the cost however it doesn't always guarantee an efficient project [3]. This proves to be a massive problem in state or government sponsored ICT projects. For instance, a study done by Heeks(2001) showed that 35% of public sector ICT projects are marked as failure, 50% as partial failures and only 15% were marked as being a success [21]. In a paper by L. Anthopoulos (2016) they go into more detail as to why these projects are marked as failures. They attribute the issues to project management failure [21]. Loukis and Charalabidis (2011) mention that this was one of the main risk factors to a poor e-government project. They claim that in many projects the scope was vague and should be elaborated and clarified, they also mention that the same risk factor is associated with the lack of organizational capacity of public organizations for implementing such large IS projects [22].

One of the biggest examples of the extreme cost for an IST project is the event that happened on October the first 2013. (Johnson, C. & Reed, H) Where the 630 million-dollar HealthCare.gov insurance website went live and then crashed minutes after launch. Heavy traffic from users and problems with the technology implemented meant that millions of Americans were not able to access the site. After an investigation into why this failed it was revealed that the main fault was weak definitions of requirements and scope [23]. In an article by Harvard Business school (2016) they go into more detail as to why this was such a failure. They detail the lack of leadership. An example of this is the contractor that was responsible for the log on system. He estimated a low demand because the website plan initially included the option to shop for products without having to sign up. The option was then removed due to technical difficulties [24].

**3.6 Political influence and Management Control**

Political actions carried out by different factions in an organisation can lead to overly optimistic estimates or pressure to avoid reporting the actual outcome [34]. These pressures usually come from managerial factions within the project team. Managers often produce estimates such as these with the sole intention of keeping customers satisfied, rather than delivering the most accurate estimate [7].

Pressures from managers, customers or other members of the project to alter the estimate, are the main political factors that can negatively impact the accuracy of software cost estimation [18]. This regularly leads to projects going over budget due to the unforeseen circumstances that occur as the project specifications become clearer and more refined.

Constant review of requirements is an inevitable feature when working with Agile systems. This means that managers need to be constantly aware of updates made throughout the project, and their effect on previous cost and effort assumptions.

Failure of management to become involved in the estimation of effort and neglecting to continuously review the accuracy of the results as time progresses are also important factors in the occurrence of underestimating and overestimating project budgets [7]. Furthermore, inaccuracy occurs from the failure of management to refer to the estimate when assessing the estimators and others involved in the project. To produce an accurate assessment, it is important that all members of the development team obtain a mutual understanding of the details. Studies display that the Cost estimation carried out by a member of the development team yields more accurate results than projections formed by staff members from other departments [29].

When management fails to take into consideration estimations proposed by the development team, inaccuracies in the timeline and estimated costs of the project are more likely to exist. It is not unusual for developers to receive pressure from management to stay within agreed estimations, but if these estimates were originally made to appeal to the manager or customers, they will usually lead to over-runs and shortfalls [29]. As a result, project managers must be aware of the negative effects that political factors place on cost estimation. In teams where the developers are responsible for estimating their own tasks, issues can occur if they feel pressure to please managers or customers, and as a result underestimate their own tasks, leading to inaccuracies.

It is important for managers to acknowledge the impacts of political decisions and inadequate management control on the cost of Software development projects [25]. Acknowledging such issues can help organizations minimize inaccuracy and exceed their budget when using Agile software development strategies. Accurate predictions on the cost of a project will be challenging for companies that have recently switched to Agile methodologies as a result of lack of experience and lack of data. Project managers should be aware of the uncertainty of such organizational changes when contemplating the cost of switching to an Agile based project development style [25].

**3.7 Frequent project changes**

Agile methods aim to reduce the cost of changes throughout the development of a system, but not necessarily to reduce the occurrence of changes [29]. However, if the amount of changes exceeds the rate of progress, original cost management attempts will be inaccurate [39]. A survey carried out by Phan [39] asked participants what they believed to be the main cause of projects going over budget and over schedule. 50% of the participants stated that they believed that changes in design or implementation led to the collapse of initial cost boundaries, while 33-36% attributed schedule overruns to frequent major or minor changes in project specification [40].

An issue arises in agile projects when “vital” features appear at different points in the project and changes to the amount of work to be completed occur. These problems should not be attributed to errors in the original estimation process, but rather errors in the user’s requirements specification [29].

Customers rarely have well defined needs, especially at the beginning of a new project. Consequently, new features that need to be completed appear at multiple points in the development phase [15]. This will have an impact on the specified timeframes and the customer then must understand the delays that will result from these late requests. If such delays are not possible, refinement of the current task requirements will be necessary [29]. Often features may be removed or postponed as a result of these changing requirements. For example, “in a project involving 30-iterations, less than 50% of the tasks were found during the exploration and analysis phase, 25% new tasks were found during the next ten iterations, and 25% were found in the last ten iterations” [16]. This shows underestimation of the effort to be extremely high, which may result in unexpected and undesirable costs.

The high probability of change of user requirements over time puts pressure on software development methods to allow for such changes. A core idea that is associated with Agile principles is that changes to software requirements are welcomed throughout the development lifecycle [29]. As a result of this, initial cost estimations can be subject to change as the requirements change. However, if there is high pressure from the customers, management will often try to keep estimates close to the original, despite changes to the workload. This leads to over-optimistic and unrealistic cost goals, which in turn leads to overruns/underruns in costs [40].

The number of requirement changes within a project is also a factor in determining the scale of the effect it will have on estimations and whether these estimations need to be refined. The company BrightSoft has stated that they follow the principal of ‘welcoming requirement changes’ but that too much of such changes can have negative consequences on any aforementioned estimates they have provided, in some cases these estimates are rendered completely invalid and are discarded [29]. This can have a major impact on the profitability of the project.

In order to keep costs as low as possible, it is important for clear communication between the members of the team and the customer to be established early. The major and core features of the software should be identified in the beginning, so that only minor requirement changes occur later in the development process. This will lead to more accurate budgeting.

# **Research Limitations**

This research paper was conducted by a team of four final year computer application and software engineering students. Due to their lack of research experience as undergraduates it is inevitable that there will be certain aspects of project topic that would be left unexplored. To try and balance this the students were given training in how to properly conduct a research paper and were provided with help by the lecturer through templates and techniques. The students had a timeframe of 7 weeks from when they were first given their title to properly research, investigate and write up their paper. Another Limitation that hampered the research was the lack of papers from recent. Many of the papers that related to Agile Software were outdated and seems to be too old to relate to. Technology grows at such a fast pace, that the always evolving sector can change considerably in a short space of time. Some papers can be looked at as obsolete compared to their modern counterparts.

An additional limitation arises when trying to find papers directly related to the topic assigned. There are many results based on agile development but appropriate peer reviewed papers on agile development costs proved harder to find than expected. During the research it was difficult to keep the research on scope and not to get side-tracked by information that is not necessarily related to the title of the research paper. It should also be taken into account that due to our lack of experience the scope and depth of our analysis would not be of the standard of experienced scholars such as postgraduates and masters of the chosen topics.

Lastly another limitation was dealing with conflicts arising from biased views and other personal issues. Finding legitimate studies was helped by looking at how the data gathering process was completed for each of the papers. This was also helped by following the templates and information given to us by the lecturer overseeing this paper.

# **Future Work / Directions for Future Research**

For future papers if they were to be written, some suggestions would revolve around the following. Looking into what success means, is it overall success, is it success in the eyes of the employer or customer, what factors get taken into account when considering success measures etc. Another suggestion would be trying to get a clear picture of what you want to discuss before starting the paper as it would be very easy to go down an irrelevant avenue of research in this subject. If at all possible it would be important to emphasize on tailoring the workload and topic coverage to the scope of the paper as this is something that could be found to be difficult as the topics that could be covered on this subject are so vast it would be more advisable to stick to fewer subjects and delve into these in more detail. There are many papers available on this subject and the direction of future research could go in any amount of directions.

# **Conclusions**

Agile Software development can be traced back to as early as the mid-nineties where people started to work inside companies and organizations to develop software products. These developers created a new methodology by combining some old and new ideas. These new ideas consisted of close relations between the development team and the stakeholders along with the end users. The people who invented these methodologies came to the realization that other software development teams might be interested in gaining some of the benefits. This led to the creation and development of new frameworks such as Scrum and Extreme Programming. In review Agile development costs are deemed relative instead of the absolute values that other basic more traditional development methods would use. By looking at the methodology that occurs in agile software development it is apparent that personnel factors are weighted heavier then project factors in terms of estimation of cost of an agile project. There are many reasons as to why agile development costs are badly estimated including politics, project scope and bad communication however the methods of estimating costs in Agile is getting better and better over time. This is due to managers learning more about the project over time. It can also be improved by selecting the right type of cost estimation methods so that the development can stay true to what has been set out in the first place. Considering all these factors it has been concluded that the benefits of agile software development far outweigh the costs if and only if as a team you are methodical in your development process. Some benefits thought to be most apparent from this development method are that agile, increases Transparency, promotes an early and predictable delivery as well as costs and schedule, allows for change, focuses on users and business value, improves transparency, increases stakeholder engagement etc. These benefits will not always outweigh the costs as of course there will be teams or groups that stray from the norm, but these groups contribute to a small portion of the teams using the agile software development approach. It is believed that teams/groups can and should take measures to promote the use of agile software development as the process is a worthy method to consider if not adopt.

# **References**

1. Bloch, M. & Blumberg, S. & Laartz, Jürgen. (2012). Delivering large-scale IT projects on time, on budget, and on value. McKinsey Quarterly. 27. 2-7.
2. Flyvbjerg & Budzier. (2011). Why Your IT Project May Be Riskier Than You Think. Harvard Business Review
3. J. Highsmith and A. Cockburn, "Agile software development: the business of innovation," in Computer, vol. 34, no. 9, pp. 120-127, Sept. 2001.
4. J. S. Reel, "Critical success factors in software projects," in IEEE Software, vol. 16, no. 3, pp. 18-23, May-June 1999.
5. Felix C. Brodbeck (2001) Communication and performance in software development projects, European Journal of Work and Organizational Psychology, 10:1, 73-94
6. DJ Anderson (2003) Agile management for software engineering: Applying the theory of constraints for business results.
7. Lederer, Albert L. (1995) Causes of inaccurate software development cost estimates; The Journal of systems and software.
8. C. Clavadetscher, "User involvement: key to success," in IEEE Software, vol. 15, no. 2, pp. 30-, March-April 1998.
9. Vinaja, Roberto (2008) Agile software development: A survey of early adopters
10. N Ionel(2009) Agile Software development methodologies: An overview of the current state of research.
11. K.N. Rao, G.K. Naidu, P. Chakka, A Study of the Agile Software Development Methods, Applicability and Implications in Industry, International Journal of Software Engineering and Its Applications. 5 (2011) 12.
12. P. Abrahamsson, O. Salo, J. Ronkainen, Agile Software Development Methods: Review and Analysis, (2002) 112.
13. B. W. Boehm and R. Ross, "Theory-W software project management principles and examples," in IEEE Transactions on Software Engineering, vol. 15, no. 7, pp. 902-916, July 1989.
14. L. Cao, Estimating Agile Software Project Effort: An Empirical Study, (2008) 11.
15. Fuqua, A., and Hammer, J.M. "Embracing Change: An XP Experience Report," XP 2003, 2003.
16. Chapman, C., and S. Ward. 2002. Managing Project Risk and Uncertainty: A Constructively Simple Approach to Decision Making. Chichester, UK: John Wiley & Sons.
17. The Standish Group, "Chaos," 1995,
18. L. Anthopoulos, C. Reddick, I. Giannaidou, N. Mavridas (2016) Why e-government projects fail? An analysis of the Healthcare.gov website
19. R. Heeks (2001). Building E-Governance for Development: A Framework for National and Donor Action
20. Loukis and Charalabidis (2011), Why Do e-Government Projects Fail? Risk Factors of Large Information Systems Projects in the Greek Public Sector: An International Comparison.
21. Johnson, C. & Reed, H., 2013. Why the Government Never Gets Tech Right. The New York Times.
22. HBA (2016), The Failed Launch Of www.HealthCare.gov
23. Winklhofer, H. 2002. Information Systems Project Management during Organizational Change. Engineering Management Journal 14 (2):33-38
24. History: The Agile Manifesto, (n.d.). <http://agilemanifesto.org/history.html>.
25. Shams, S. Böhm, P. Winzer, R. Dörner, App Cost Estimation: Evaluating Agile Environments, in: 2019 IEEE 21st Conference on Business Informatics (CBI), 2019: pp. 383–390. <https://doi.org/10.1109/CBI.2019.00050>.
26. S. Saurabh, S. Sagnika, S. Mishra, M.N. Das, A Systematic Review on Software Cost Estimation in Agile Software Development, JOURNAL OF ENGINEERING SCIENCE AND TECHNOLOGY REVIEW. 10 (2017) 51–64. <https://doi.org/10.25103/jestr.104.08>.
27. L. Cao, B. Ramesh, Agile Requirements Engineering Practices: An Empirical Study, IEEE Softw. 25 (2008) 60–67. <https://doi.org/10.1109/MS.2008.1>.
28. M. Lang, K. Conboy, S. Keaveney, Cost Estimation in Agile Software Development Projects, in: R. Pooley, J. Coady, C. Schneider, H. Linger, C. Barry, M. Lang (Eds.), Information Systems Development, Springer New York, New York, NY, 2013: pp. 689–706. <https://doi.org/10.1007/978-1-4614-4951-5_55>.
29. M. Shepperd, C. Schofield, Estimating software project effort using analogies, IEEE Transactions on Software Engineering. 23 (1997) 736–743. <https://doi.org/10.1109/32.637387>.
30. S. Ziauddin, S.K. Tipu and S. Zia, “An Effort Estimation Model for Agile Software Development”, Advances in Computer Science and its Applications (ACSA), Vol. 2, No. 1, pp. 314-324, 2012
31. R. Popli and N. Chauhan, “Cost and Effort Estimation in Agile Software Development”, International Conference on Optimization, Reliability, and Information Technology (ICROIT), Faridabad, India, pp. 57-61, 2014
32. S. Sequeira et al., “Simple Method Proposal for Cost Estimation from Work Breakdown Structure”, Conference on Enterprise Information Systems/International Conference on Project Management/Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN / HCist , Elsevier, Algarve, Portugal, Vol. 64, No. 1, pp. 537-544, 2015
33. M. Jorgensen, “A Review of Studies on Expert Estimation of Software Development Effort”, Journal of Systems and Software, Elsevier, Vol. 70, pp. 37-60, 2004.
34. T. J. Gandomani, K. T. Wei and A. K. Binhamid, “A Case Study Research on Software Cost Estimation Using Experts Estimates, Wideband Delphi, and Planning Poker Technique”, International Journal of Software Engineering and Its Applications, Vol. 8, No. 11, pp. 173-182, 2014
35. Phan, Dien Dean. "Information systems project management: An integrated resource planning perspective model." (1990).
36. K. Molokken, M. Jorgensen, A review of software surveys on software effort estimation, in: 2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings., 2003: pp. 223–230.