Implementing Enterprise Content Management Services with an Agile Approach

Emre Erturk[1]

[1] Eastern Institute of Technology, New Zealand

Correspondence: Emre Erturk, Eastern Institute of Technology, New Zealand. E-mail: eerturk@eit.ac.nz

Received: July 15, 2019 Accepted: August 12, 2019 Online Published: August 19, 2019

doi:10.5430/bmr.v8n3p16 URL: https://doi.org/10.5430/bmr.v8n3p16

Abstract

Every organisation needs access to fast online data in order to meet increasing customer demands. The information architecture of a company, how it is designed, and how information is managed are very important. Because the amount of content is growing at a precipitous rate, an online content management system is recommended to manage it with efficacy. How a business content management system is designed and implemented (e.g. with SharePoint, G Suite, or SAP OpenText) determines its adoption and success, especially when using Agile approaches during design and implementation.

This conceptual research paper looks into the current context of Enterprise Content Management (ECM) and the desirable qualities of an Agile approach. First, the literature review explores the definitions, elements, benefits, and recent trends around ECM. Then, the paper discusses the major characteristics of Agile project management. Thirdly, the paper describes how an Agile ECM system should be (e.g., takes advantage of the growing Internet, cloud services, and mobile computing).

This paper covers possible methodologies which can encourage a flexible development approach to ECM implementation, while promoting end user involvement and their needs. An ECM system should be dynamic enough to meet the demands of future organisational growth and the business environment. It should allow integration with other new software, including internal ones and those in the Cloud. Two methodologies, Scrum and MIKE2.0, are explained. One recommendation is to synthesize these. MIKE2.0 can guide information governance, while Scrum can provide an innovative team-based approach and user empowerment for implementing Content Services rapidly.

Keywords: project management, team work, Agile, change management, business content management

1. Introduction

1.1 Problems and Opportunities

Organisations face the challenge of managing constantly increasing volumes and density of data. The challenge for organisations is to implement and manage a dynamic content management system that is responsive to changing requirements, while providing a structure that contributes to organisational efficiencies. Many countries specify government regulations on how long an organisation must maintain certain types of information. ECM can help accomplish those requirements. Generally defined, Enterprise Content Management (ECM) is the ‘strategies, tools, processes and skills’ a company uses for information asset management over its lifecycle (Hullavarad, O’Hare, & Roy, 2015). ECM handles various information including both structured and semi-structured data, unstructured data, code, and metadata in order for companies to manage the ongoing content creation that takes place and to better use this content in automated workflows.

Without a web-based information management strategy, content is often placed in siloed organisational structures, i.e. information is not easily searchable, often disorganised, and replicated across the organisation. However, the overall business system itself no longer has to be on a single platform or a completely centralized ECM software application. The more recent Content Services paradigm is more flexible, with separate and distributed applications, as long as they share APIs (application programming interfaces) and allow multiple content repositories to work in harmony (Hanna, 2019).

Implementing an ECM system can assist organisations to increase employee productivity and customer satisfaction, while simplifying business processes, meeting legal regulations, tracking key data, and reducing the duplication of unnecessary information (Hullavarad et al., 2015). ECM is a core system which should be a part of an organisation's strategic plan to avoid the issues that may arise from a lack of effective content management. The growth of knowledge as an asset has become a subtle success factor for many businesses (Breitner, Dinh, Fill, & Rickenberg, 2015). The hidden financial worth of organisations today is made up of its ideas and information. To maintain their knowledge assets effectively; businesses need to understand information management and ECM, and need to know how to implement these systems (Breitner et al., 2015).

1.2 Objectives and Structure

First, this paper aims to investigate the methods with which systems developers can implement an effective online Enterprise Content Management (ECM) system, made up of diverse Content Service applications. The discussion is around 'Agile' methods and approaches within the IT industry. An ECM can be developed with Agile Project Management or a double approach (using an Agile software development process to create an Agile end product).

In order to handle this topic from different perspectives, the paper is organized as follows. The introduction has provided the definition and the benefits of ECM, particularly as a business information system. This helps to understand the terms of reference below and what is meant by being Agile (indicating adaptability and end-user orientation). These should be taken into account while understanding how an ECM system should be constructed in the first place. The literature review in Section 2 covers specific methodologies, the practical activities that are done during the ECM system development project, and the benefits of these Agile activities. The second (minor) aim of the author is to briefly illustrate that the trend toward Agile ECM development has continued in popularity in recent years. This is also done in Section 2 by looking at the summary counts of publicly available informational pages on the Internet, which contain any content related to the topic of this paper.

Another aim of this conceptual paper is to suggest that an Agile ECM system can be produced using the approaches described Section 2. Therefore, Section 3 revisits and discusses how an Agile ECM system should function in actual operation, after the implementation. Section 3 also recaps by providing real life project examples from the literature of content management systems being implemented using Scrum, the most popular Agile development method. Section 4 provides a conclusion, by discussing how web-based tools, social media, mobile applications, and cloud services desired by the end users may be considered and integrated into the system development project. Finally, the paper also offers various recommendations and ideas for future research.

Increasing an organisation's agility is the basis for making prompt and effective responses to market changes, with the aim of gaining a competitive and strategic advantage (Ghilic-Micu, Mircea, & Stoica, 2013). The majority of businesses today are permanently changing due to technological developments and opportunities. The implementation and integration of new technologies may become less time consuming, if an agile implementation approach is adopted. The key advantages of using an agile approach here include, increased collaboration between teams, software quality, customer satisfaction, and quicker delivery, and reduced cost of development. The ability to review and improve existing processes is important when implementing ECM in an organisation, in order to gain competitive advantage in the market. It is not only advantageous to implement technology using an agile approach, but also necessary to maintain agility consistently under all circumstances to deliver rapid solutions.

1.3 Basic Terms of Reference

The Agile concept was initiated in 2001, when a group of software developers published the Manifesto for Agile Software Development. in which they argue in favour of (among other things): human interactions, working software rather than comprehensive documentation, and customer collaboration (Fowler & Highsmith, 2001). According to Karlesky and Vander Voord (2011), there are common project management concerns that Agile-related techniques can offer a solution to, for example, poor estimation, wrong timelines, and users having a lack of interest in the end product. While a product may meet today's customer requirements, it still needs easy adaptation to possible future needs. Otherwise, the product may be used less than anticipated. Therefore, it seems safer to deliver the first version early, and still have resources to continue with the adaptation in the future.

Agile project management (APM) is feature driven and allows software teams to work incrementally and be flexible, instead of permanent controls (as with traditional project management methods). An adaptive APM approach provides: the ability to create prototypes and change them more easily, consideration of each member as a skilled and valuable stakeholder, and reliance on the collective ability of autonomous teams (Augustine, Payne, Sencindiver, & Woodcock, 2005). APM's iterative and incremental development may also provide better risk management. Dynamic adjustments to requirements are handled by frequent and short timeboxes or sprints.

To receive change requests from the customer and regularly thinking of better ideas is a normal part of Agile, whereas other more traditional approaches unrealistically assume that requirements may remain constant for the rest of the entire project. APM gives software developers and customizers an iterative style of working, as well as better tracking any associated risks and timeline changes. An Agile approach organizes and prioritises different pieces of practical work to deliver immediate prototypes with functionality, while traditional approaches try to address documentation and long-term issues.

2. Literature Review

2.1 Specific Development Approaches

The flexibility expected from a new ECM system presents some challenges for both implementation and adoption. Alalwan and Westroffer (2012) conducted a review of ECM literature and identified a number of implementation challenges under some common themes as follows: organisational culture and the resistance to change, mismatch between organisational needs and functions of the vendor solutions, poor integration with existing systems, and a mismatch between the promised system performance and actual performance. As an ECM solution is considered within an organisation, it would therefore be important to note the areas that research has indicated is problematic, and devise appropriate solutions. To address these, AIIM (2016) has developed an ECM deployment checklist that provides some guidance including 14 steps. One of these is to very clearly understand the organisation's various content types and content driven processes. It is also important to communicate the goals and strategies to the stakeholders with sound business cases and the ability to fulfill the high-level requirements. Another important aspect is to align ECM with other internal as well as online technologies, and to have a good contingency plan for any one the legacy and new system components.

Two key development approaches have been identified, to explore how to best implement an Agile ECM. The first one is Scrum, an Agile approach. The second one is called the Method for an Integrated Knowledge Environment 2.0 (that is, MIKE2.0), which is an open source delivery framework (MIKE2.0, 2019). These are both widely published and serve as the theoretical foundation for this paper.

Kenneth (2012) describes Scrum as an Agile approach to develop innovative products and services. It begins by creating a Product Backlog, typically built on user stories from the Product Owner, ordered by priority. The cross-functional Scrum team completes the work in sprints (iterations) with a specific end time. The outcome of a sprint is to be a potentially shippable or working product. The benefits of Scrum include: happy customers, greater return on investment, reduced cost, faster results, confidence to succeed in a complex environment, and enjoying work.

Two of the primary characteristics of the Scrum methodology are the planning backlog and the development sprint. A planning backlog is the list of product features and functions derived from the organisational and stakeholder requirements. These can take the form of a to-do list for technical requirements, or user stories representing the ways in which the various stakeholders will use the system (Scrum Institute, 2019). Items on the backlog are then actioned during ongoing sprints, which are short development phases where teams work together to ensure a product is fit for purpose.

Scrum builds on the Agile principles of developing a working software that users can see early, as opposed to spending a lot of time planning and writing business requirements and specifications upfront. It is focused on empowering the cross-functional team to make the decisions with rapid iteration and emphasis on continuous customer input, “versus big hierarchies and compartmentalization by function” (Sutherland & Schwaber, 2012).

One of the disadvantages of Scrum is that it may misleadingly seem like a simple framework in theory, and its adopters may think that it is easy to implement (Kenneth, 2012). The cost of training and time needed to set up an environment conducive to Scrum could prove to be a challenge. Scrum can also reveal existing dysfunctionalities and waste in the organisation. If not overcome, these can prove to be barriers to switching to an empowered and problem-solving workforce.

MIKE2.0 is a comprehensive, innovative, and collaborative approach that focuses on Information Management (IM) in an enterprise environment, often around structured or semi-structured data, but it can be applied to create any type of IM system, including those containing unstructured data. It has five phases: Business Assessment, Technology Assessment, and three continuous improvement increments (which focus on Design, Development, and Deployment).

MIKE2.0 has a number of differentiating features. It is comprehensive and architecturally driven (not vendor). It is scalable (suits various enterprise or projects), and takes into account all factors related to data issues (people, process, organization, and technology). Because of its open-source approach, MIKE2.0 is also innovative, inviting and incorporating new approaches to information management. It is also a collaborative approach. Constant feedback between practitioners is shared through online discussions. These allow contributors on a worldwide scale, and their contributions are managed in a wiki. Therefore, MIKE2.0's open source framework aims to balance stability with innovation.

Due to the comprehensiveness of the framework, it may be difficult for an enterprise to decide how to start using this for an implementation, or whether to adopt at a later phase. It also requires a greater knowledge of software theory. The framework's ability to incorporate so many different facets of information management could make it seem like a convoluted approach. The open source nature of the framework could also lead to disagreements.

According to Zelfond (2017) who is a well-known SharePoint expert, the implementation project can be managed in Sprints according to Scrum. He has summarized this with Figure 1. Although Sprints 2, 3, and 4 seem like similar iterations, they serve different purposes in configuring the different parts of the online content management system. Sprint 2 is for gathering the metadata requirements, discussing additional functional requirements, creating subsites, refining web parts, configuring the term store with metadata capability, and configuring views on the lists and libraries. Sprint 3 is for reviewing all pages and web parts, discussing the security model for various parts of the content, creating security groups, adding users, and configuring external sharing (if applicable to the Intranet). The objective of Sprint 4 is to complete administrator training, perform any cleanup as a result of user feedback, create site templates (for easier future expansion), and enable the usage and auditing features in conjunction with Office 365 online.

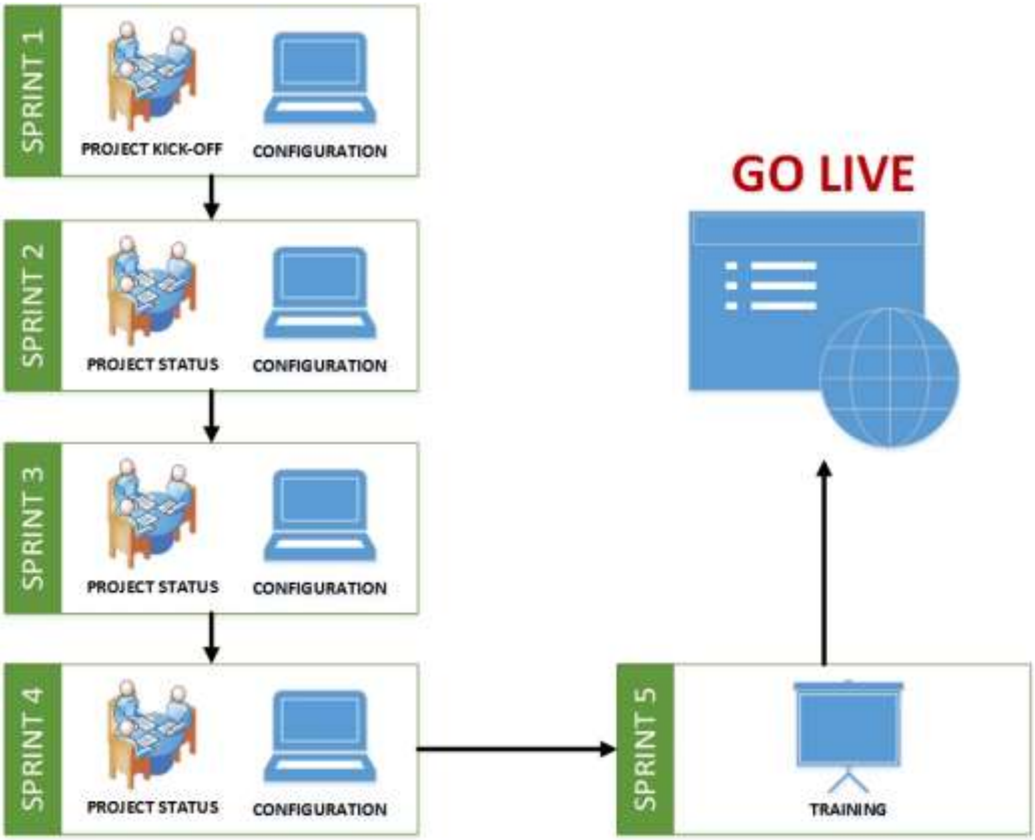

Figure 1. Greg Zelfond's SharePoint Implementation Model

In effect, Zelfond's model can replace the last three phases of MIKE2.0 and can be done incrementally and iteratively, i.e. multiple times for bigger projects. For large organisations with intricate system architectures, the first two phases of MIKE2.0 (which are traditional, sequential, and documentation-based) can be helpful. Therefore, this combination would create a current and realistic hybrid development approach.

2.2 The Popularity and Currency of Agile Approaches

Just as with any business information systems development, the application of Agile approaches to content management websites has received publicity and support. This type of thinking first rose in popularity 2011-12 (http://mike2.openmethodology.org/wiki/Guidelines_for_adopting_SCRUM_in_Information_Management_Projects). The main question in this section is whether that was a fad, or the concept of "Agile Content Management" has been able to maintain its relevance since then. The author of this paper has decided to look at the results of an advanced search in Google (manually – with these keywords), according to the publication years of the web pages containing information about this topic. The impression from the search engine data (as summarized below in Table 1) is that the popularity of this concept has grown over the years, with more information published about this topic as well as the previous pages being updated.

Table 1. Page counts from Google search according to the years

Year *Pages*	*2018*	*2017*	*2016*	*2015*	*2014*	*2013*	*2012*	*2011*
"Agile Content Management"	50	30	20	23	20	20	8	5

Furthermore, the author of this paper has been interested in how people perceive "Agile ECM" and what ideas they associate with it. Google is commonly known as the world's most popular search engine by far. It is also a source of open and free data. The following is an additional analysis of Google search engine data, this time using wordtracker.com. It is interesting and highly relevant to see which other keywords are the most popular alongside this topic, indicating to us what (especially) the users and professionals want to learn about and may do in the future. In Table 2, Volume shows the quantitative appearance of those words whereas KEI (keyword effectiveness index) refers to how well the keywords appear to be attracting new traffic from online searches and other websites. This shows strong interest in Scrum, as well as interest in agile methodologies overall.

Table 2. "Agile ECM" tangent keyword analysis using wordtracker.com

Keyword	***Volume***	***KEI***
agile methodology	1,300	83
agile scrum	1140	71
agile manifesto	590	82
agile project management	480	69
agile principles	170	81
agile software development	230	66
agile certification	90	78

3. Discussion

3.1 Practical Features of an Agile ECM

According to Orienge (2019), the Agile ECM concept has appeared as a reaction to new challenges. An Agile ECM provides greater adaptability to constant changes, support for different business management methods and styles, and more effective control and clarification of business processes. An Agile ECM system, like other ECM platforms, is used to incorporate solutions for automating routine business tasks. However, it should do this without extensive effort and programming, by providing powerful and user-friendly configuration tools. The implemented ECM application should provide easy and meaningful access to the stored data and support the business processes. Another desirable characteristic of an Agile ECM would be the ability to support both small teams and large groups of enterprise users, face-to-face and online.

There are several distinctive features typical of Agile ECM. Support for business processes should be flexible and updateable. Content should be matched easily with business processes. The underlying goal should be to focus on enabling business tasks. Rapid application development and rapid deployment are also part of an evolving and Agile ECM. The Agile ECM should also support for mobile devices (Orienge, 2019). Similarly, Roe (2010) defines Agile

ECM as a combination of modern features and effective process management, allowing users to access their content accurately and promptly. An Agile ECM should try to improve on following main capabilities of the system: ability to provide reports and share information, publishing applications and information online, and consolidating (synergizing and synthesizing) business content and processes.

According to Pelz-Sharpe (2009), agility can be viewed generally to indicate ease of redeployment. Therefore, an Agile ECM should be able to add and deploy new features, processes, and technology with ease so that the system itself can grow alongside future requirements. Laserfiche (2011) offers the following description, “Agile ECM organically expands throughout your organisation, crossing departmental boundaries and integrating easily with line-of-business applications and legacy systems to eliminate data silos”. An agile ECM should demonstrate the ability the scale with organisational growth and respond to major technical changes, e.g. integrating cloud resources, and enabling mobile and remote workers (AIIM, 2016).

3.2 Practical Implementation Evidence from the Literature

SharePoint is a popular platform from Microsoft which is used to create intranets and websites for organisational collaboration, document management, business blogs, and company news distribution (https://www.pcmag.com/encyclopedia/term/63621/microsoft-sharepoint). It can be deployed to extend certain password-protected information, when used in tandem with Microsoft's SQL Server and Web Server. SharePoint also interfaces with Microsoft Office and other third-party applications through its programming interfaces (APIs). SharePoint Online provides Content Services in the Cloud (rather than from a local server).

According to Spiik (2014), an APM method such as Scrum can help to implement a SharePoint ECM project. By using Scrum, a SharePoint project can be better implemented, and the team’s stress can be reduced through better communication. Scrum provides project tracking ideas and tools to the team to carry out their SharePoint solution, with notable transparency for them and their stakeholders. The author also emphasizes that Scrum can suit a variety of system development teams containing people with different skills levels.

Another good example of a SharePoint implementation where Scrum is used as an APM method, is the Intranet for project and document management developed by the company RealTest AB in Västerås, Sweden. RealTest AB used Scrum, which helped to get valuable feedback from the demonstrations and presentations they conducted during the Sprint Review meetings. These reviews often led to better collaborations and changes, in order to satisfy the users’ needs within a relatively shorter period (in comparison with the time taken by the company’s other non-Agile implementation projects). In the end, RealTest AB got exceptional feedback from both the software team and from all the other stakeholders (Lundqvist, Johansson, & Yildiz, 2013).

Aside from using an Agile approach along with a content management system, what makes RealTest AB (https://rocketreach.co/realtest-ab-profile_b5de74f9f42e4e00) interesting is its placement within a regional technology incubator (Västerås Science Park), which currently continues to play an important role for business and technical development. As the author of this paper is also writing from a regional perspective, this aspect will be further discussed in Conclusions and Recommendations section.

Sef (2012) also explains in his research study, how Scrum can help deal more successfully with a Sharepoint project, for example, with this insight from one of the participants in the study: "The methods are working to give us focus, keep us on track with the most important priorities, and provide transparency to everyone involved in the process including the stakeholders and product owner".

4. Results

This section presents the hybrid Agile approach to content management system development. This is done first by contrasting the new approach with a structured approach such as the System Development Life Cycle and the original MIKE2.0. The typical activities in the two different approaches are shown in Table 3 below. The new approach formulated in this paper is shown in the last column.

Table 3. Suggested Approach: Agile MIKE2.0

Phases and Activities	***Structured Approach***	***Agile MIKE2.0***
1. Planning (or Business Assessment)	Write Charter, Delivery Plan. Conceptual Model. Business Requirements.	Gather user stories (business requirements). Set up cloud environment.
2. Analysis (or Technology Assessment)	Investigate technical feasibility. Compare candidate solutions.	Get enough information to make an initial mockup (landing and main pages).
3. Design *Design, development, and deployment may be done during each Agile sprint.*	Design and document metadata. Create plans for data quality and validation.	Create subsites, add web parts/widgets, create the metadata store, add library views.
4. Development *This also relates to sprints 2, 3, and 4 of Figure 1 and helps to describe them.*	Create the site without concurrent end user feedback or testing. Keep management and administrators updated.	Perform refinements as a result of user feedback, create site templates. Administrator training.
5. Deployment	Comprehensive testing. Comprehensive training.	Create security groups, add users and external sharing. End user training.

Even if the end product of the two approaches were similar, there are substantial differences in how it would be achieved. The first positive aspect of Agile MIKE2.0 is creating greater user engagement by involving them earlier. The second difference is that the new approach is more modular and gradual with practical development, by spreading it over multiple phases and sprints. This is quite logical with contemporary web development which is flexible to expand or to be contained as needed. It also makes common sense not to invest so much time and effort on planning and documentation on a business system which may be partially deferred or if there is any likelihood of a future decision to go in a different direction.

The main challenge in the Agile approach would be the reliance on ongoing and good communication between everyone involved; whereas in the structured approach, different people can complete their particular tasks in isolation, taking turns and passing it on to the next group. However, the traditional structured approach is potentially susceptible to misunderstandings and gaps within the end product, while Agile MIKE2.0 tries to prevent these through greater communication.

5. Limitations and Implications

Small to medium sized teams may be best suited to support an agile methodology while larger teams may struggle to maintain communication as frequently and effectively as smaller teams. A further limitation of an agile approach may have to do with the complexity of the project. Depending on the structure of existing software in place, there may be critical aspects that cannot be easily incrementally modified (France, Rumpe, & Turk, 2014).

The research study itself also has some limitations. The Agile development approach to content management is still evolving. In the content management space, changing technologies will also continue to shape the specifics of this approach. Therefore, there is not a large body of empirical evidence to make formal quantitative and financial comparisons and analyses.

Secondly, many of the findings in the academic and professional literature review are self-reported. How successful the authors and the participants saw a particular project or methodology as a success is possibly subjective and after the fact, rather than taking into equal account the difficulties and gaps they encountered during the process itself. Quite often, success is based on the general feelings of participants rather than purely based on financial or quantitative measures. In addition, some of the projects involve commercial consultants. Therefore, projects that end up not being successful may not be discussed in the literature as frequently or in similar detail.

In summary, the limitations of this study also provide potential hints for these authors, other researchers, and everyone else interested in Agile project management approaches. Academic researchers can collaborate with industry practitioners in the future, during the actual course of their work, to collect feedback on the success of their project management approaches.

6. Conclusions and Recommendations

End users of ECM systems are often interested in extending functionality of this system through web-based tools, social media, and mobile applications. These could include mobile applications that access the ECM system, interfaces for transferring profiles and digital content from social media, and data analytics web applications which can work with an ECM. Furthermore, Content Services are now using Artificial Intelligence (AI) functionalities to a greater extent. Another area of development is managing content in multiple languages and using reliable tools for content translation (Batova & Clark, 2015). As the most important stakeholders of the resulting system are the numerous and diverse online end users, these types of extensions, interfaces and applications that are desired by the end users should be addressed during the system development project, and these should either be integrated (if possible) or some flexibility should be provided in the system for future expansion.

This paper suggests that IT professionals who are involved in an ECM implementation (e.g. SharePoint software) may take a hybrid development approach, combining the more flexible Agile Scrum with MIKE2.0 which is sequential and structured approach. MIKE2.0 can provide some comprehensiveness, metadata documentation, and information governance aspects; while Scrum can enable rapid development, teamwork, and communication.

Furthermore, the ideas covered in this paper may be included or mentioned in courses related to business information systems. The curriculum of these courses in recent years have been featuring more content related to teaching Agile approaches. The use of SharePoint or another ECM software platform would provide a good example of a case study or practical project for students. Similarly, some of the ideas in this paper may be relevant to the implementation of cloud-based systems (Erturk, 2017). SharePoint and SAP OpenText are available through Software-as-a-Service (SaaS), which suits the financial and infrastructure circumstances of many businesses. According to the Agile MIKE2.0 approach, the design and configuration of the cloud related requirements, content structures, and workflows should be handled gradually over time (rather than being delayed to a later phase).

Future research may investigate the applicability of Agile project management approaches to developing computer networks. Creating a sophisticated enterprise network (for example, a large Local Area Network or an Internet of Things Network) requires significant planning, problem-solving, configuration, installation, quality control, and testing. The network development phases and issues to be considered may include requirements formulation, risk assessment, feasibility analysis, architecture and topology, overall IT strategies, equipment procurement, detailed design, software creation or configuration, stakeholder feedback, and testing. Network performance, versatility, reliability, scalability, maintainability, and security are among the design priorities when managing company networks, smart communities, and networks of intelligent devices and sensors. These qualitative criteria can be extrapolated in depth with quantitative measures and concrete goals so that the success of the network development project can be monitored. Here an Agile approach can be more beneficial than a traditional planning approach (systems life cycle, structured project management methodologies); since the general Agile philosophy focuses on the essential goals, aims at a faster completion, works to ensure stakeholder satisfaction, and tracks the goals regularly. Such future research may also offer potential recommendations for other project managers.

Using Google data to find and follow new web content on this topic can be helpful for Agile consultants and practitioners and in the future. A further extension to the keyword analysis in this paper would be to look at a large number of keyword combinations and their distributions according to their volume. Certain combinations of keywords may be used interchangeably by similar users and may be leading to discovering the same web pages. It is possible that the volume distribution may not be normal, but rather a skewed or heavy tailed distribution, for example, as in Guo (2017). A flatter non-exponential distribution could hint that the overlaps between the keywords may be greater than one may initially realize.

Creating an Agile business content management system is quite possibly linked to greater user empowerment, which is typical of Agile approaches. While the goal is to identify and incorporate innovative system features, different methodologies may do this at different times. Traditional methodologies would typically have to identify them early, whereas Agile MIKE2.0 is more open to adding new and interesting features along the way. Although similar Agile approaches to enterprise content management systems have already been reported in the literature, it is recommended that the specific approach (i.e. Agile MIKE2.0) postulated in this paper be used in an actual content management project. A good place to try this experiment could be within a technology incubator or business hub, by an innovative and enthusiastic start-up company with good interpersonal relationships between the stakeholders. Then its success could be measured with a follow-up research paper in the future.

References

AIIM (2016). *The Agile Enterprise: A Welcome Change*. Retrieved from https://www.project-consult.de/files/AIIM_Studies_Fall_2016/AIIM_IW_Agile_Enterprise_2016_Jul16.pdf

Alalwan, J. & Westroffer, H. (2012). Enterprise content management research: a comprehensive review. *Journal of Enterprise Information Management, 25*(5), 441-461. https://doi.org/10.1108/17410391211265133

Augustine, S., Payne, B., Sencindiver, F. & Woodcock, S. (2005). Agile Project Management: Steering from the Edges. *Communications of the ACM, 48*(12), 85-89. https://doi.org/10.1145/1101779.1101781

Batova, T. & Clark, D. (2015). The Complexities of Globalized Content Management. *Journal of Business and Technical Communication, 29*(2), 221-235. https://doi.org/10.1177/1050651914562472

Breitner, M., Dinh, T., Fill, H. & Rickenberg, T. (2015). Enterprise Content Management Systems as a Knowledge Infrastructure: The Knowledge-based Content Management Framework. *International Journal of e-Collaboration, 11*(3), 49-70. https://doi.org/10.4018/ijec.2015070104

Erturk, E. (2017). An Incremental Model for Cloud Adoption: Based on a Study of Regional Organizations. *TEM Journal, 6*(4), 868-876. https://doi.org/10.18421/TEM64-29

Fowler, M. & Highsmith, J. (2001). *The Agile Manifesto*. Retrieved from http://andrey.hristov.com/fht-stuttgart/The_Agile_Manifesto_SDMagazine.pdf

France, R., Rumpe, B. & Turk, D. (2014). *Limitations of Agile Software Processes*. 3rd International Conference on Extreme Programming and Flexible Processes in Software Engineering, Alghero, Italy. Retrieved from https://arxiv.org/ftp/arxiv/papers/1409/1409.6600.pdf

Ghilic-Micu, B., Mircea, M. & Stoica, M. (2013). *Software development: Agile vs. Traditional, Informatica Economică, 17*(4). https://doi.org/10.12948/issn14531305/17.4.2013.06

Guo, Z. (2017). Heavy-tailed Distributions and Risk Management of Equity Market Tail Events. *Journal of Risk & Control, 4*(1), 31-41. https://dx.doi.org/10.2139/ssrn.3013749

Hanna, T. (2019). *Content Management vs. Content Services: What's the Difference?* Retrieved from https://solutionsreview.com/content-management/content-management-vs-content-services-whats-the-difference/

Hullavarad, S., O'Hare, R. & Roy, A. (2015). Enterprise Content Management solutions—Roadmap Strategy and Implementation Challenges. *International Journal of Information Management, 35*(2), 260-265. https://doi.org/10.1016/j.ijinfomgt.2014.12.008

Karlesky, M. & Vander Vood, M. (2008). *Agile Project Management (or Burning your Gantt Charts)*. The Embedded Systems Conference, Boston, Massachusetts.

Kenneth, R. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Upper Saddle River, New Jersey: Addison-Wesley.

Laserfiche (2011). *Agile ECM as a Shared Service*. Retrieved from https://laserfiche.tamu.edu/files/Agile_ECM_Shared_Service.pdf

Lundqvist, A., Johansson, N. & Yildiz, S. (2013). *Intranet for Project and Document Management* (Master's thesis, Mälardalen University, Sweden). Retrieved from http://www.diva-portal.org/smash/get/diva2:648984/FULLTEXT01.pdf

MIKE2.0 (2019). *Governance Association, 'Differentiating Features*. http://mike2.openmethodology.org/wiki/Differentiating_Features

Orienge (2019). *Understanding the Importance of Agile ECM*. Retrieved from http://www.orienge.com/agile-ecm

Pelz-Sharpe, A. (2009). *Can ECM ever be Agile?* Retrieved from https://www.realstorygroup.com/Blog/1688-Can-ECM-ever-be-Agile

Roe, D. (2010). *An Agile Enterprise CMS - What's That?* Retrieved from https://www.cmswire.com/cms/information-management/an-agile-enterprise-cms-whats-that-007279.php

Scrum Institute (2019). *What is Scrum?* Retrieved from https://www.scrum-institute.org/What_Is_Scrum.php

Sef, J. (2012). *Agile Scrum for SharePoint Projects*. Retrieved from http://joelsef.blogspot.co.nz/2012/02/agile-scrum-for-sharepoint-projects.html

Spiik, K. (2014). *SharePoint and Scrum* (Master's thesis, Turku University of Applied Sciences, Finland). Retrieved from https://www.theseus.fi/bitstream/handle/10024/79236/Spiik_Karl-Johan.pdf

Sutherland, J. & Schwaber, K. (2012). *The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework*. Retrieved from https://www.researchgate.net/publication/242437392_The_Scrum_Papers_Nuts_Bolts_and_Origins_of_an_Agile_Process

Zelfond, G. (2017). *How to Implement SharePoint in 5 Meetings*. Retrieved from https://sharepointmaven.com/how-to-implement-sharepoint-in-5-meetings/