

How the Pharmaceutical Industry and Agile Software Development Methods Conflict

A Systematic Literature Review

Ali Hajou

Department of Information and
Computer Science
Utrecht University
Utrecht, the Netherlands
a.hajou@students.uu.nl

dr. Ronald Batenburg

Department of Information and
Computer Science
Utrecht University
Utrecht, the Netherlands
r.s.batenburg@uu.nl

dr. Slinger Jansen

Department of Information and
Computer Science
Utrecht University
Utrecht, the Netherlands
slinger.jansen@uu.nl

Abstract—Software development projects in the context of the pharmaceutical industry are highly restricted. Innovation in terms of process changes within the practice of software development has to manoeuvre through the regulatory wilderness of legislation and industry standards. 49 articles have been selected using the Systematic Literature Review method to evaluate the maturity of this field of research. Agile methods conflict with the generally accepted software development methods within the pharmaceutical industry. Even the acquired set of literature disseminate about a various amount of topics regarding to the conflict of agile methods and the pharmaceutical industry. This research concludes that the appliance of existing agile methods in the pharmaceutical industry is relatively non-existent, possibly causing software development projects to exceed budgets.

Keywords—Agile; Software Development; Pharmaceutical Industry; Systematic Literature Review

I. INTRODUCTION

The global pharmaceutical industry is very large. The WHO (World Health Organization) estimates the global pharmaceutical market to be worth almost 400 billion in 2014 [1]. It is commonly known that the use of medicinal products increased since the introduction of regulated medicine. The IMS institute for Healthcare Informatics estimates a global spending \$1.2 trillion in 2016 on medicines. That is for the developed pharmaceutical markets (US, Japan, Canada, EU and South Korea) an average of \$609 is spent annually per person. While the rest of the “*pharmerging*” (emerging pharmaceutical) markets spend an averaged \$91 per person. The numbers continue to increase every consecutive year [2].

The expanding volume of the pharmaceutical industry combined with the heavy regulation by national and international agencies generate the closed attitude towards IT innovation. Therefore the number of software developers that operate in the pharmaceutical market is limited.

Agile methods are used to develop software in a more pragmatic way. Fitzgerald, Stol, O’Sullivan and O’Brien mention that agile methods initially were seen as “*best suited for small, co-located teams developing non-critical systems*” [3]. However the methods have a high adoption rate across many industries and project team sizes, but fail to fully penetrate the pharmaceutical industry [4].

II. RESEARCH QUESTIONS

This Systematic Literature Review (SLR) has been created to elaborate on whether the combination of agile software development methods and software development practices within the pharmaceutical industry are possible. Due to the general knowledge about the topic of Agile methods, related research questions are omitted. This SLR’s research questions are therefore narrowed down to: (1) *What are the main impediments of software development projects within the pharmaceutical industry?* (2) *In what circumstances are pharmaceutical companies coping with high overhead of software development projects?*

III. ARTICLE AND INFORMATION COLLECTION METHOD

The search process is based on the Systematic Literature Review method which is explained in the paper of Moher, Liberati, Tetzlaff and Altman [5]. The intention of this SLR is to create an academically reproducible template for researching agile methods in a pharmaceutical context. Therefore Google Scholar has been chosen due to its publicly available source references to literature. In order to increase the availability (ability to download) of literature, this search has been conducted with a university license from Utrecht University called ‘UBUlink’.

Figure 1 shows a visual representation of the SLR article selection process.

The work described was sponsored by Magna Medical B.V. in Nijmegen, the Netherlands. We are grateful for the generous input and constructive guidance.

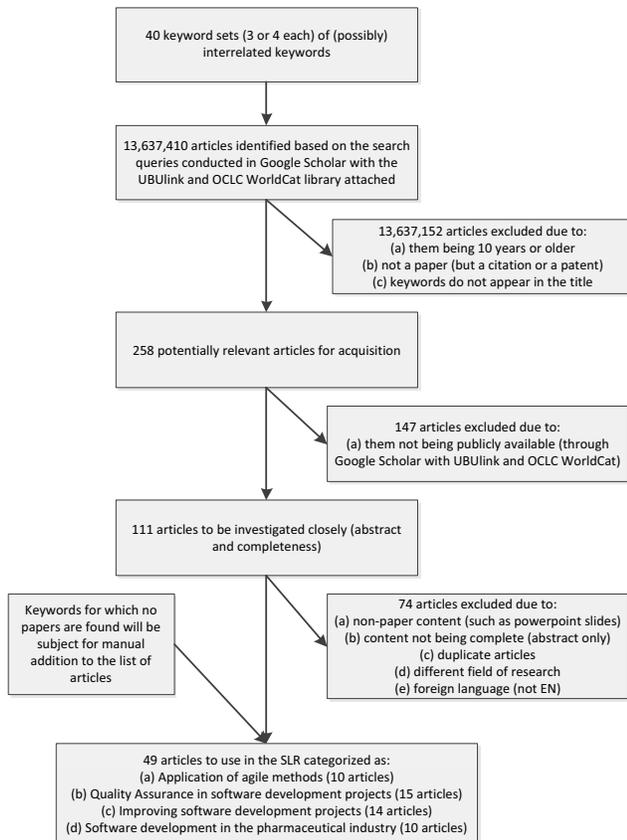


Fig. 1. Search and filter process.

A. Keyword selection

The SLR is initiated by collecting keywords for conducting keyword-sets (sets of keywords that are to be used in searching the literature source). The first ten keywords are based on the research question and the research domain; ‘development’, ‘software’, ‘dynamic’, ‘project management’, ‘effective’, ‘projects’, ‘documentation’, ‘framework’, ‘innovation’ and ‘pharmaceutical’. The reason for not including the word ‘agile’ in this initial list of keywords is to keep the possibility open for other dynamic methods to be included which are not categorized as an agile method.

The ten keywords are used to find more related keywords that can be used in the creation of keyword-sets. These were found in the article-titles that had the highest number of citations of the search results based on the single keywords. The following keywords were added to the list: ‘practices’, ‘technological process’, ‘pharmaceutical industry’, ‘compatibility’, ‘regulatory’, ‘adopting’, ‘pharma’, ‘lean approach’, ‘adapting’, ‘pharmaceutical validation’, ‘agile’,

‘product lifecycle management’, ‘agile development’, ‘fragmented software’, ‘traceability’ and ‘agility improvements’. The list of individual keywords is sorted on the number of individual results every keyword generated (i.e. ‘development’ generated 6,510,000 results, while ‘agility improvements’ generated 56,000 results). The individual keywords were merged together into keyword-sets, resulting into 40 keyword-sets that are used to search with in Google Scholar. The total number of (unfiltered) results of articles was 13,637,410.

B. Filtering Criteria

The filtering criteria were set up to improve the applicability of the results onto the research domain. This meant that the results had to be recent scientific articles in the first place. The recency of the articles is determined by defining a round number based on the mean between the publication dates of the agile manifesto (2001) and the publication of important pharmaceutical regulations and legislations; 1: PIC/S *Good Practices for Computerised Systems in Regulated “GxP” Environments* (2007), 2: ISPE *GAMP5* (2008), 3: The most recent revision of EudraLex Volume 4 *GMP Annex 11: Computerized systems* (2011) and 4: the most recent revision of FDA Code of Federal Regulations Title 21 Chapter 1 *Part 11: Electronic Records; Electronic Signatures* (2013).

Therefore an initial filter has been created to: (a) filter out articles that were published before 2004 (max. 10 years old), (b) filter out patents and citations, and (c) filter out results that do not have the entered keywords in the title.

The second filter is created based on the availability of (i.e. ability to download) the articles. After applying the second filter, all results have been downloaded for close analysis.

Due to the fact that Google Scholar is a worldwide search-engine using multiple article databases and other sources, it generates results that could not be used during the research. The results therefore had to be cleaned up. The following main exclusion criteria were identified after closely analysing the downloaded 111 articles: (a) non-paper content (i.e. PowerPoint slides) should be taken out, (b) the articles should be complete, (c) duplicates should be taken out, (d) the article should be related to the research topic, and (e) the articles should be written in English.

The keyword-sets that did not generate in any results have been used for a subjective search of scientific articles that are closely related to the sets. This resulted in 8 additional articles, bringing the total of usable articles to 49.

TABLE I. Inclusion and exclusion criteria for data synthesis

Inclusion criteria	Exclusion criteria
<p>I1. The information should clearly indicate advantages or disadvantages of a software development method. Category relation: a, c, d</p> <p>I2. The information should clearly indicate the applicability of agile methods in a specific environment which require method adaptation. Category relation: a, b, c</p> <p>I3. The information elaborates on the practical implementation of quality assurance, documentation steps, supportive tools and process improvements within the domain of software development. Category relation: b, c, d</p> <p>I4. The information provides insights in software development projects within the context of the pharmaceutical industry. Category relation: c, d</p>	<p>E1. The information related mainly to improve a process that is not to be found within the pharmaceutical industry. Category relation: a, b, c</p> <p>E2. The information that is solely focussed on quality assurance and not specified onto the software development process. Category relation: b, d</p> <p>E3. The information that elaborate more on relatively known general practices of agile methods. Category relation: a, c</p> <p>E4. The information that dictates the use of a method but not elaborates on the applicability of it in a defined context. Category relation: a, b, c</p> <p>E5. The information that does not have anything to do with software development, regulation, process impediments or process improvements. Category relation: b, c, d</p>

C. Categorization and article distribution

The acquired articles have been categorized by reading the article titles and abstracts. By closely analysing the topics of the articles 4 main topics are identified in which the 49 articles can be separated: (a) 10 articles are categorised in ‘Application of agile methods’, of which 1 was subjectively found; (b) 15 articles are categorised in ‘Quality Assurance in software development projects’ of which 1 was subjectively found; (c) 14 articles are categorised in ‘Improving software development projects’ of which 2 were subjectively found; and (d) 10 articles are categorised in ‘Software development in the pharmaceutical industry’ of which 4 were subjectively found. The subjectively found papers were mainly distributed among the category ‘Software development in the pharmaceutical industry’, possibly proving the point about the relatively unknown field of research.

D. Data synthesis

The extraction of data from the acquired articles to be used in the research results has been done by using inclusion and exclusion criteria which are based on the 4 categories in which the articles are grouped. These are shown in Table I.

IV. QUALITY ASSESSMENT

From the 49 acquired articles 22 were published in scientific journals and 9 were published in conference proceedings. Others were either published in a periodical, on a university website or were theses. This means that (at least)

63.26% of the articles can be considered to be sufficiently scientifically validated. Figure 2 displays the distribution of the 49 articles.

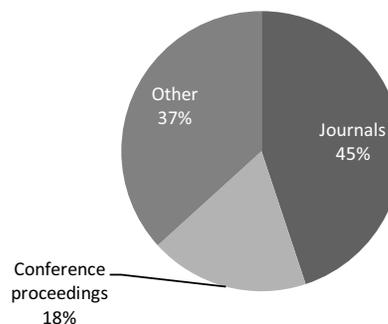


Fig. 2. Type of publications.

The articles have been quite evenly distributed in terms of publications per year, as shown in Figure 3. The publication rate can be considered quite reasonable, as there is an average publish of 4 subject-related articles per year. The article selection did not result in any articles that were published in early 2014.

Based on the list of acquired articles one can consider the research topic not to be popular. No additional research is done to determine whether this trend has been originating from before the 10 year filtering criterion.

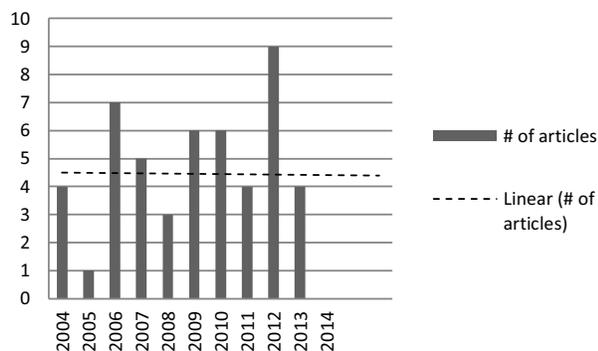


Fig. 3. Number of articles published per year.

Figure 4 shows that the number of citations is slightly skewed due to one single article that has 676 citations, making it the most popular article acquired. Without this popular article [6] the average number of citations per article is 11.9. A total of 18 articles did not have any citations (displayed by Google Scholar) at all.

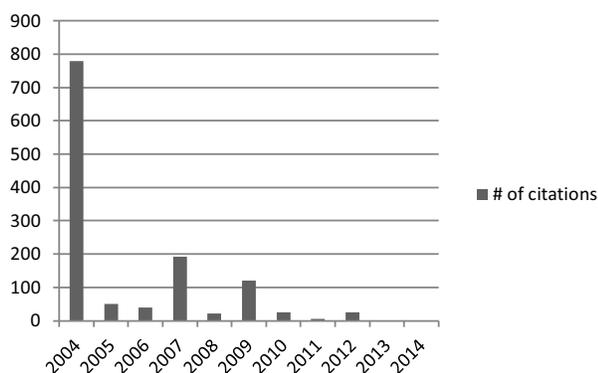


Fig. 4. Aggregated number of citations per year

The state in which this field of research (software development in the pharmaceutical industry) is considered as relatively new. Figure 5 shows a small number of evaluative researches (6%) and a large number of exploratory research (37%). The second largest group of research type is the descriptive research group (31%). However, most of the articles introduce a new method for improving their field of research, which usually is not related to the pharmaceutical industry. The newly introduced methods might be generalizable, but miss the context of highly documentative environments.

The two articles that are the closest related to the research subject were from Fitzgerald, Stol, O’Sullivan and O’Brien [3] and from Heeager and Nielsen [7].



Fig. 5. Research type of acquired articles

V. RESULTS

During the analysis of the articles and therefore during the close inspection of the theory, three main topics are identified which provide a holistic overview of the researched topic.

A. The regulatory complexity of software development in the pharmaceutical industry

“The pharmaceuticals industry is characterised as being an innovative but highly regulated sector” [8]. The investments into innovation (especially in pharmaceutical product development) have been increasing since the early 50’s. The success of many complex medicines (such as antipsychotics) resulted into the setup of large research projects to find the next ‘blockbuster’. The prospected income of a successful product usually is many times higher than the money invested. These research projects are subject to very strict regulation and legislation, which is one of the reasons that they therefore are very expensive. The pharmaceutical industry is ever since searching for other area’s to innovate in. One area might be the area of software development, as computerized systems can positively influence the creation, monitoring and distribution of pharmaceutical products. However, the pharmaceutical industry conflicts with the software industry. The efforts to be done by the software industry for producing software more efficiently are difficult to mix with the demanding regulatory standards of the pharmaceutical industry [3][4][7][9]. Software developers must take different regulatory and legislation into account which might come from overarching agencies (e.g. EMA, ICH), country specific agencies (e.g. FDA, MHRA, CBG-MEB, PMDA), industry-specific agencies (e.g. EudraLex, ISPE) and subject specific regulations (e.g. HIPPA, GxP, CFR Part 11). It therefore requires a large initial investment for the software developer to understand the field, index the regulatory requirements and match these with the clients’ requirements. To add a layer of difficulty; the regulatory requirements change relatively fast [10]. These changes are usually opposed by country-specific laws and are usually introduced after a reoccurring incident involving a pharmaceutical product. Law making in the pharmaceutical

industry can therefore be considered to be a reactive process and therefore have the risk to be changed again anything soon.

The fast changing regulatory requirements, generation of traceability with preferably ‘wet-ink signatures’ (of reviewed and acknowledged documentation possibly by other stakeholders) put high pressure on software development. The overhead makes software development projects marginally profitable while this ‘extra work’ is mandatory [8].

B. Differences between the agile and the highly documentative approaches

The pharmaceutical industry and the software development industry differ not only in terms of business, but especially in terms of perspective and mentality. Highly documentative or regulated environments desire “*defined*” logic, while the agile approach desires the “*empirical*” logic [3]. Instead of failing and adapting to improve products and processes, regulated environments tend to overdo tasks to prevent project impediments. This means that any intermediate deviation of the stated project plan has to wait until the project is finished. It is a paradox in itself, as overdoing tasks is an impediment for the project due to the exceeding of budget and deadlines. This traditional approach provides “*stability*” because the changes in the environment are temporarily neglected to not disrupt the approved plan [11]. Intermediate changes to the plan or to software requirements are not tolerated as, according to the FDA’s and GAMP5’s extensive traceability requirements [4][12], they might influence the eventual outcome. The focus therefore is more on the (initial) prospected outcome of the project and merely on the creation of a solution that eventually is fit for purpose.

Software development stakeholders in the pharmaceutical industry are required to develop a high amount of documentation in order to fully remain compliant, while agile approaches emphasize “*working software over comprehensive documentation*” [13]. This highly documentative approach is based on the essence of working according to a defined plan. These plans are often created using traditional project management techniques [14][15] and therefore are not adapted to the reality that might be changed over time. This allows for the illusion of correctly estimating a project, while estimating software development projects is mainly “*based on gut-feeling*” [16]. However, such documentation is required in order to achieve regulatory compliance [4][17].

The main originator of the high overhead is the pharmaceutical industries’ focus on risk-management and quality assurance [7]. Heeager and Nielsen mention the increasing costs of approval of medical devices and thus the increasing desire to be compliant from an early stage of development. But using traditional methods for project estimation and resource allocation often result in exceeding the budget and deadlines. The cause of these exceeds is usually due to fact that the estimations are done by other people which are not part of the development team [18]. This

is an effect of large project teams with many stakeholders and limited intermediate communication. These large teams are set up so that ‘everyone has a specific task to do’, and can focus on keeping their documents and their products of a high quality. The cost of conformance to quality requirements is often more manageable than the indirect costs that are related to poor quality [19].

C. Lack of attempts to be agile in the pharmaceutical industry

The number of published case studies, experiences and scientific articles about the adoption of agile methods within highly documentative environments is limited [8]. The ones that are published (and cited from the acquired literature for this SLR) mention that agile adoption in these environments is restricted by elements that define these environments [4][7][9].

An agile method has to be perfectly defined in SOPs (Standard Operating Procedures) in order to be accepted. Furthermore the mind-set of stakeholders in pharmaceutical organizations has to change [20]. The development team is not able to “*embrace change*” (altering the requirements or even slightly improving the development process) if changes require additional documentation and reviews from external stakeholders such as software project portfolio managers [7][10][21]. This behaviour can be perceived as if project or software faults are not tolerated, and therefore resulting in a mind-set of registering and managing all intermediate steps to trace bugs or software anomalies [15][22]. Project related faults can and should be evaluated during or after the project, “*but they (project evaluations) are seldom done*” [23].

Managing software development projects is a difficult task in every organizational environment [21]. One of the main difficulties in the adoption of agile methods is managing uncertainty, such as project estimation, minimal viable products and resource allocation [24]. The introduction of “*living traceability*” and a method of engaging into “*continuous compliance*” prove to be key in ‘becoming agile’ [3][25]. In order to accomplish this, appropriate supportive tooling is needed [26][28]. Without supportive tools it will come down to software developers having to create and maintain the extensive amount of documentation, which is counterproductive in an agile environment [9]. The generation of “*Agile Documents*” might be an option. The generation of documentation in ‘an agile way’ proves to be hard to implement, but eventually very successful to maintain [29]. However, no literature has been found about this practice in the context of the pharmaceutical industry.

The research of McHugh, McCaffery and Casey [4] revealed that the pharmaceutical industry is actively searching for alternatives to traditional plan driven software development methods. The sponsor of this research is one of these actively searching companies for methods to reduce the overhead of software development projects, but keeps a safe distance from rigorous changes in order to remain their

compliance to regulations. Unfortunately no figures have been provided regarding to the percentage of overhead or success rate of software development projects. Controlling and managing these regulatory obstacles are difficult to such extent that it creates the situation in which organizations that have been able to accomplish the optimization of a process are acquired by the larger players in the industry [10].

VI. DISSEMINATION

The collected articles vary in topics which do not always correspond with each other. The fact that these topics do not have collective arguments might indicate root causes for the problems that occur in this field of research.

After performing a topic analysis, the main topics on which the findings and/or opinions vary are; (1) the method of estimation of software development projects; (2) the compatibility of documentation with agile practices; (3) the tooling or processes to successfully support software development projects; (4) the method of ensuring quality assurance; and (5) the fundamentals on which quality should be based for software development projects.

A. Estimation of software development projects

Traditional estimations of software development projects seem to be eventually incorrect. The majority of the literature is in consensus about this observation. However the reason of these incorrect project estimations (and therefore budget-exceeds) vary. Lindström and Lidvall [16] mention that the project estimations are usually done based on past experience but mostly on gut-feeling. Brown and Grundy [14] mention that the use of any traditional project estimation method (i.e. for real estate construction) is not applicable on software development projects. Especially if the project is executed using an iterative method. Project estimation should be an iterative process [28][30]. These articles mention the constantly changing learning-curve which provide an increasingly better estimation. Mainly because the developers (that create the estimation) need the time to discover their development speed. The articles of Kostelac, Vukomanovi and Ikonić [25] and Torkar, Awan and Khadem [27] mention that iterative project estimation is only realisable when both the management and the software developers are collaborating in becoming fully transparent and traceable.

According to the research of Heeager and Nielsen [7] project estimation are usually incorrect due the lack of discipline of other stakeholders. One example that was mentioned is the habit of constantly “*breaking the sprints*” due to the perceived ability to directly intervening the software development process when a bug is found. Heeager and Nielsen categorize this behaviour of directly searching for the problem as ‘typical quality assurance’. Another reason for the estimation failures is the fact that these usually are not created by the developers themselves [18]. Milanov and Njeguš mention that intermediate changes (i.e. intervening

stakeholders) can be tolerated and acted upon when the developers themselves create and maintain the project estimations.

B. Project documentation and agile practices

Project documentation is an important subject in agile software development projects. Takashi, Takashi, Fukue and Tsuda [9] mention that the creation of documentation is against the principles of agile, and thus should be minimized or managed outside the development team. Heeager and Nielsen affirm to this notion, but it is very much contradicting with the articles of Vogel [17] and McMichal and Lombardi [20]. Vogel, McMichal and Lombardi explain the ability to create documents in an iterative way, which therefore can become a standard element of agile software development projects. Mazni, Sharifah-Lailee and Azman [29] even mention the concept of “*Agile Documents*”, which are documents that are created iteratively and are bound to finish-criteria such as the ‘Definition of Done’ in the ‘Scrum’ software development method. In their research the introduction of agile documents seemed to be highly effective.

C. The need of supportive tools in agile software development

All articles mention the need of supportive tools to accommodate software development and connect cross-functional stakeholders. There is a difference in how the articles specify the need of tooling. Misra, Kumar and Kumar mention the need for tools to “*support software development*” [26][30], while Nithya, Lavanya and Gokila Deepa mention it “*creates traceability*” [28]. Van de Weerd, Brinkkemper, Nieuwenhuis, Versendaal and Bijlsma confirm this fact, but add the requirement that tooling should support “*a definitive process*” [31].

The need for a definitive process should be the main focus point, and not the supportive tools. Therefore the focus should be primarily on process management [8][21][30][32].

D. Quality is key

The definition of quality is often mentioned in the literature. According to Børretzen and Dyre-Hansen [22] is quality the reliability to depend on a (computerized) service. According to GAMP5, the pharmaceutical industry standard for software development projects, quality software is “*fit for purpose*” [12].

Quality is based on project scope, schedule and cost [19]. That makes quality assurance an overarching concept that should become a vital part of all handlings related to software development projects, or even of any project. Rauch, Kuhn and Friedrich [30] focussed more on the product quality assurance which, in their opinion, is based on the lines of code (LOC). Børretzen and Dyre-Hansen [22] and Winkler and Biffel [15] focus on a slightly higher level, mentioning that quality is based on the definition of requirements (functional,

non-functional requirements and software faults) as these mainly define the usability and applicability of the software.

There are different views on how to enforce quality assurance in software development projects. Torkar, Awan, Khadem and Afzal [27] and Alecu [19] mention that quality indicators should be defined on beforehand (before starting a software development project) to eliminate ambiguity at an early stage. Other literature mention the complete opposite; quality assurance is a process which runs parallel to software development projects [30][33][34]. Quality indicators might change during a software development project and by making ‘quality checks’ a concurrency with software development the development team can adapt to these changes.

Colomo-Palacios, Soto-Acosta, Mishra and García-Crespo [35] support the view of having a concurrent quality assurance process during the software development project, but mention the need of a starting point. A small quality or validation plan should be created on beforehand and continuous quality controls should be done during the software development. However, this article also revealed the fact that these quality indicators are usually not followed during the software development.

VII. DISCUSSION

Using a definitive process for selecting scientific articles cannot be done without taking the risk of missing potentially interesting and topic-related articles. But by explaining the search process thoroughly this research can be reproduced or expanded to generate a new status update of this particular field of research. However, two elements are open for discussion.

A. Validity

The lack of maturity of the field of research might influence the internal validity (elimination of bias). The two research areas have not been connected often with one another. This is resulting in articles that are related to agile but are not written in the context of the pharmaceutical industry. Therefore the prescribed applicability of agile methods might not cohere with the software development practices in the pharmaceutical industry. This means that reflecting elements from the two research areas is subject to nuance.

The external validity of the research is mainly subjected to time and the ability to use the Utrecht University’s licence for acquiring scientific articles. A relatively new field of research might inhabit more research and scientific publications over time, which will have effect on the results as they are described in this SLR. What is an obstruction now does not have to be an obstruction in the future as there might be solutions by that time.

Furthermore the research has been conducted using articles that might not be publicly available for researchers that do not

have a university-enhanced article index. Just using Google Scholar without any additional catalogues will significantly reduce the number of available (relevant) articles.

B. Data extraction

Empirical research (especially in literature reviews) is subjected to topic analysis which in this research is used for categorization of relevant subjects into research chapters. This categorization is very much influenced by the scope of the research, and may therefore differ in future reproductions of this research.

VIII. CONCLUSION

The SLR resulted in both trivial and non-trivial answers. The first research question; “*What are the main impediments of software development projects within the pharmaceutical industry?*” is answered many times. The incompatibility of existing agile software development methods within a highly documentative and procedural environment ask for large adaptation of the agile method. To be more specific; there is no room for creating less documentation and performing less quality assurance-related tasks. The regulatory complexity cannot be altered as it is a mandatory element which allows an organization to be or remain operational in the pharmaceutical industry. The number of articles related to applying agile software development methods within a pharmaceutical organization is very limited. This might be the effect of the closed attitude of pharmaceutical organizations, but also the apparently unattractive field of research for scientists. The number of citations underlines this fact. This also has its effect on answering the second research question; “*In what circumstances are pharmaceutical companies coping with high overhead of software development projects?*”. There are limited case studies or observations in this field, resulting in the inability to provide a definitive answer. The limited information reveals the difficulties software developers have to cope with. High costs, incorrect project estimations and the necessity to leave out the essential documentation tasks (to finish software development or testing) put high stress on the software developers. According to the acquired articles, the current method of coping with these problems seems to be; accepting the high pressure, high overhead and inefficiency of projects.

This SLR’s results give the impression that the pharmaceutical industry is an unattractive and unpopular industry to work for as a software developer or as a software developing organization. It therefore is in need for a novel methodological approach to embrace the need for quality control, strive for regulatory compliance and utilize the strong features of agile software development.

IX. FURTHER RESEARCH

The pharmaceutical industry is not the only highly documentative industry in the context of software

development practices. Therefore to elaborate more on this topic, research can be done with a wider scope in terms of highly documentative industries such as the governmental, aircraft and aerospace industry. Special focus can be done on the level (and reason) of legislatively required documentation, the ability to implement continuous integration and the added value of various quality control processes.

Furthermore, as described in the conclusion, there is a need for a software development method or process description which inhabits the novelties of agile software development methods and complies with the overarching regulations regarding to software development within the pharmaceutical industry.

X. REFERENCES

- [1] World Health Organization. (2013). *WHO | Pharmaceutical Industry*. Retrieved 12 2, 2013, from who.int: <http://www.who.int/trade/glossary/story073/en/>
- [2] IMS Institute. (2012). *The global use of medicines: outlook through 2016*. IMS Institute for Healthcare Informatics.
- [3] Fitzgerald, B., Stol, K., O'Sullivan, R., & O'Brien, D. (2013). Scaling agile methods to regulated environments: an industry case study. *ICSE 2013, Software Engineering in Practice* (pp. 863-872). San Francisco, CA, USA: IEEE.
- [4] McHugh, M., McCaffery, F., & Casey, V. (2012). Barriers to adopting agile practices when developing medical device software. *Software Process Improvement and Capability Determination*, 141-147.
- [5] Moher, D., Liberati, A., Tetzlaff, J., & Altman, D. (2009). Preferred reporting items for systematic reviews and meta-analyses: the Prisma Statement. *Annals of Internal Medicine*, 264-269.
- [6] Poppendieck, M., & Poppendieck, T. (2004). *Lean Software Development - An Agile Toolkit*. Upper Saddle River: Pearson Education.
- [7] Heeager, L., & Nielsen, P. (2009). Agile Software Development and its Compatibility with a Document-Driven Approach? A Case Study. *20th Australasian Conference on Information Systems* (pp. 205-214). Melbourne: Aalborg University.
- [8] Carleysmith, S., Dufton, A., & Altria, K. (2009). Implementing Lean Sigma in Pharmaceutical research and development: a review by practitioners. *R&D Management*, 95-106.
- [9] Takahashi, M., Takahashi, S., Fukue, Y., & Tsuda, K. (2006). Computer Validation Oriented Development of Pharmaceutical Automatic Control Software. *10th WSEAS International Conference on Computers* (pp. 856-863). Athens: Shimane University, Tsukuba University, Tokushima University.
- [10] Hynes, M. (2012). Project Management in the Pharmaceutical Industry: Competencies of the Future. *DIA - Global Forum*, pp. 50-55.
- [11] Mateos-Garcia, J., & Sapsed, J. (2008). *Adopting 'Agile' and 'Scrum' Practices as 'Organisational Becoming': Cases from the UK Video Games Industry*. Eastbourne: Brighton University.
- [12] ISPE. (2008). *GAMP5 - A Risk-Based Approach to Compliant GxP Computerized Systems*. Tampa: ISPE.
- [13] Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., et al. (2001). *Manifesto for Agile Software Development*. Retrieved 1 6, 2014, from agilemanifesto.org: <http://agilemanifesto.org/>
- [14] Brown, L., & Grundy, T. (2004). *Project Management for the Pharmaceutical Industry*. Gower.
- [15] Winkler, D., & Biffel, S. (2012). *Improving Quality Assurance in Automation Systems Development Projects*. Vienna: Vienna University of Technology.
- [16] Lindström, C., & Lindvall, A. (2006). *An Estimation Improvement Method Aiming for Successful Deliveries of Software Projects*. Göteborg: Chalmers University of Technology.
- [17] Vogel, D. (2006). *Agile Methods: Most are not ready for prime time in medical device software and development*. DesignFAX Online.
- [18] Milanov, G., & Njeguš, A. (2012). Analysis of Return on Investment in Different Types of Agile Software Development Project Teams. *Informatica Economică*, 7-18.
- [19] Alecu, F. (2011). Quality Perspective: Managing Software Development Projects. *Economy Informatics*, 99-110.
- [20] McMichael, B., & Lombardi, M. (2007). ISO 9001 and Agile Development. *Agile*, 262-265.
- [21] Heller, M., & Wörzberger, R. (2006). *Management Support for Interorganizational Cooperative Software Development Processes based on Dynamic Process Views*. Aachen: RWTH Aachen University.
- [22] Børretzen, J. , & Dyre-Hansen, J. (2007). Investigating the Software Fault Profile of Industrial Projects to Determine Process Improvement Areas: An Empirical Study. *EuroSPI* (pp. 212-223). Potsdam: Norwegian University of Science and Technology.
- [23] Verner, J., & Cerpa, N. (2005). Australian Software Development: What Software Project Management Practices Lead to Success? *Australian Software Engineering Conference*. Sydney: IEEE.
- [24] Harris, M. (2006). *Using emergent outcome controls to manage dynamic software development*. Tampa: University of South Florida.
- [25] Misra, S., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in

- adopting agile software development practices. *The Journal of Systems and Software*, 1869-1890.
- [26] Kostelac, D., Vukomanović, M., & Ikonić, M. (2012). Integrating Enterprise Project Portfolio Management with the Balanced Scorecard: A Case From the Pharmaceutical Industry. *Tehnički vjesnik*, 303-316.
- [27] Torkar, R., Awan, N., Khadem, A., & Afzal, W. (2010). Predicting Software Test Effort in Iterative Development Using a Dynamic Bayesian Network. *21st IEEE International Symposium on Software Reliability Engineering*. San Jose, CA: IEEE.
- [28] Nithya, V., Lavanya, S., & Gokila Deepa, G. (2013). To Ameliorate Quality in Software Development Projects By Using Contemporary Agile Bug Tracking System. *International Journal of Emerging Technology and Advanced Engineering*, 197-201.
- [29] Mazni, O., Sharifah-Lailee, S., & Azman, Y. (2010). Agile Documents: Towards Successful Creation of Effective Documentation. *11th International Conference XP 2010* (pp. 196-201). Trondheim: Springer.
- [30] Rauch, N., Kuhn, E., & Friedrich, H. (2008). *Index-based Process and Software Quality Control in Agile Development Projects*. CompArch2008.
- [31] van de Weerd, I., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., & Bijlsma, L. (2006). *On the Creation of a Reference Framework for Software Product Management: Validation and Tool Support*. Utrecht: Utrecht University.
- [32] Biswas, K. (2007). Process Transformation for Better IT Service Delivery in the Pharmaceutical Industry. *Journal of GxP Compliance*.
- [33] Rothenberger, M., Kao, Y., & Wassenhove, L. v. (2010). Total quality in software development: An empirical study of quality drivers and benefits in Indian software projects. *Information & Management*, 372-379.
- [34] Roemer, M. (2011). New Annex 11: Enabling Innovation. *Pharmaceutical Technology Europe*.
- [35] Colomo-Palacios, R., Soto-Acosta, P., Mishra, A., & García-Crespo, Á. (2011). *Software Quality Management Improvement through Mentoring: An Exploratory Study from GSD Projects*. Madrid: Universidad Carlos III de Madrid.