Software Vendor Selection Using Proof of Concepts

Seminar paper

Laukat, Tim, FH Wedel, Wedel, Germany, bwl104609@fh-wedel.de

Abstract

This paper reviews the current literature for the software vendor selection method Proof of Concept. It covers the description of the main contents of Proof of Concepts, the types of literature that provide information about this topic and the reported best practises. Furthermore, the different approaches described by the literature are summarized as well as the advantages and disadvantages. As a result, the paper shows the advantages overweight and indicates that the method has a good potential to be used more often.

Key words: Proof of Concept, Software Vendor Selection, Prototype, POC, Best practise

Table of Contents

Table of Figures

Table of Tables

List of abbreviations

IT	Information Technology
POC	Proof of Concept
MVP	Minimum Viable Product

1 Introduction

Over the past years IT-Systems are have gained increasing importance. They are getting in touch with an increasing number of organizational parts. Simultaneously the number of software developers and software vendors is rising. Figure 1 shows the development of spending on enterprise software between 2009 and 2020. In total the spending increased from 226 billion U.S. dollar in 2009 to 458 billion in 2019. The value for 2020 is estimated with 426 billion U.S. dollar. An upwards trend can be identified for the entire time. Only the year 2020 is estimated to be 6,9 percent lower than the previous year. This decrease is due to the negative economic impact triggered by the coronavirus (COVID-19).

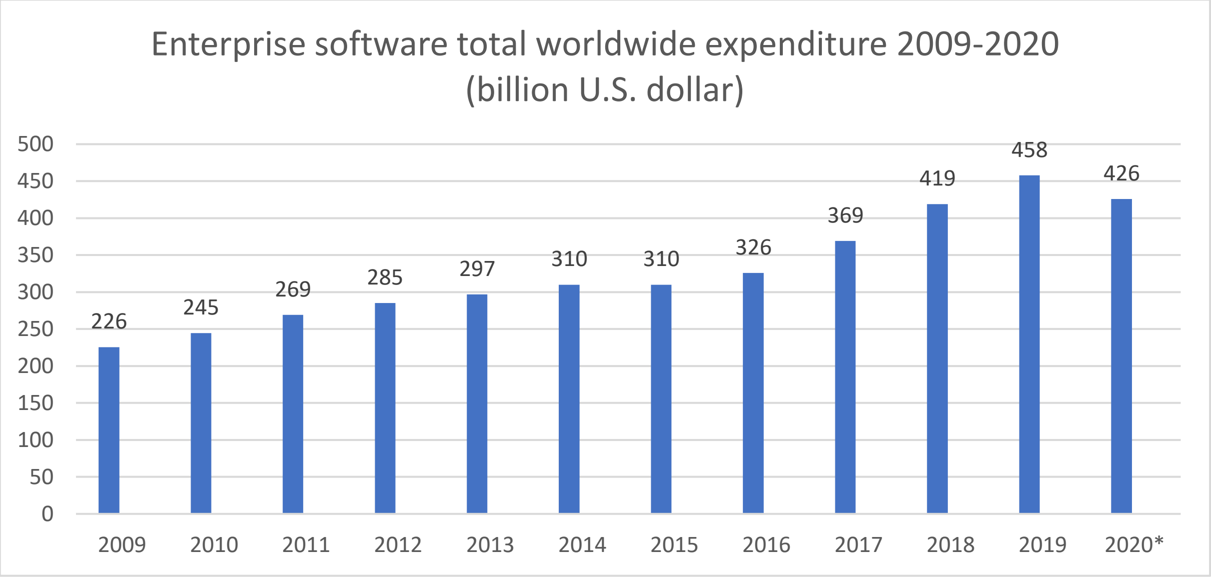

Figure 1: IT Spendings on enterprise software frm 2009 to 2020 (Gartner, 2020)

As the projects behind these systems are getting larger and more complex the risk and cost of failure is increasing equally (Bloch, et al., 2012). Flyvbjerg and Budizer (2011) point out that IT-projects run a high risk for budget overruns. On average 27 % of the project were not on budget.

A research done by the Project Management Institute (PMI, 2017) shows that 39% of the interviewed people identified '*Inaccurate requirements gathering*' as a primary cause for the failure of projects. This leads to the idea of conducting a test run before executing the project. The concept of these prototypes is already common in other industries. In the automotive sector, the development of prototype cars and their testing in specific markets is a standard process. This approach can be similarly applied to the topic of software development and digital transformation. A common name for this principle is Proof of Concept (Lippincott, 2018).

A Proof of Concept (POC) can support the risk minimization for implementing new IT Systems. After conducting a POC following three questions should be answered:

- What does the final product should look like?
- Is the product feasible (proof of technology)?
- Are there any potential issues?

The knowledge about this information provides a better understanding of the requirements and can potentially point out that the project is not feasible (Deyan, 2020). A stop of the project in such an early phase can save costs and effort. Besides stopping the project, identified problems can be included into the architecture development and be used as lessons learned. This early discovery can support the success of the entire project (Eeles, 2006).

2 Proof of Concept

The Term Proof of Concept is clearly defined by the literature. It includes the approach to proof that an idea is viable to solve a particular problem (Soni, 2020). This method can be used for various topics and different objectives. In terms of using the Proof of Concept as a software vendor selection tool the target is to assess the vendor claims as well as to determine if the software is capable of satisfying the needs. Usually only a small number of users is included in this process (Add-On Products, 2020). A Proof of Concept is conducted in advance of a project. Thereby the output is a not deliverable product (Malsam, 2019).

Beside the term Proof of Concept, Protype is an accompanying term. As remarked by Singaram and Jain (2018) these two terms are often used interchangeably despite being clearly separable.

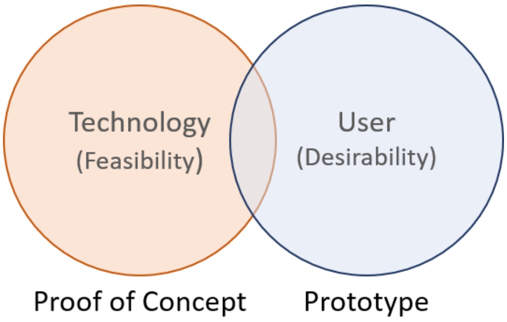

Figure 2: Proof of Concept vs. Prototype (Singaram and Jain, 2018)

As already described, the POC is conducted to prove the feasibility. In spite the prototype is the first developed model of the final product. It provides a variety of the functionalities but may include errors and might be inefficient. Based on this prototype tests can be done and allow the end-user to get an idea of the final product. Furthermore, the vendor has the possibility to determine the development status of the product. Rouse (2020) compares the prototype to the minimum viable product (MVP) based on the build-measure-learn circle (Lenarduzzi and Taibi, 2016).

While conducting a POC for a software vendor selection the earlier differentiated terms Proof of Concept and Prototype are often used as one method. As described by Pham (2020) in his five critical factors to success, by Soni (2020) in step three of the best practise for POC's as well as by Lazarikos (2005) in his second step for a successful POC, the conduction of tests with the tool are part of the POC.

3 Literature Review

3.1 Research Sources and Criteria

For this paper the research was done in the databases EconBiz, GVK, Google Scholar as well as the Katalogplus from the Hamburg University. All databases where searched for the terms *Proof of Concept, POC* and *Proof of Concept for Software* together as separated key words. No fields or result types were excluded. In the Table 1 you can find the number of results for the searched terms separated into the different databases.

Libary	**Total number of results**
EconBiz	5.002
GVK	1.924
Google Scholar	3.870.000
Katalogplus	44.238

Table 1: Number of database research results

For every library the first 100 results were checked by title and if necessary, by the abstract as well. Despite the large amount of search results, a first assessment of the result titles showed that the results cover studies done by the method Proof of Concept, rather than describing it. Within the research the topics of rapid prototyping and prototyping in software development were mentioned. These topics need to be differentiated from the POC. They include the development of software after the contract signing in contrast to the development of prototypes in the process of acquisition.

As the results of these different databases did not provide information that were needed, the research was extended to google search results. They include blog posts, magazine articles as well as publications on company websites. Table 2 shows the used key words and google search results. As the results for the key words are not separable the columns 3 and 4 contains all of them. For narrowing down the number of matches they were sorted by relevance and limited to the 100 most relevant results for every key word.

Key Words	**Total number of results**	**Relevant results by title**	**Relevant results by summary**
proof of concept best practices	270.000.000	**70**	**32**
proof of concept software	165.000.000		
proof of concept software vendor	15.300.000		
proof of concept example	325.000.000		
proof of concept template	47.600.000		

Table 2: Number of google research results

For the evaluation of the relevant results two steps have been done. At first the titles of the links were checked for relation to the search topic. After this step 70 results were remaining. In the second step the summary of the results as well as the list of content was review. Based on this research 32 relevant sources were identified.

3.2 Types of results

Only one paper could be found regarding the topic of Proof of Concepts as a software vendor selection tool. Figure 3 shows most of the results are in form of blog posts and records on company websites. The website results are mainly provided by consulting companies which use the articles as a marketing resource. The 12 results of blog posts were written by persons involved in the IT and project management department. One result is an article from the magazine *computerworld* and another one is a book source.

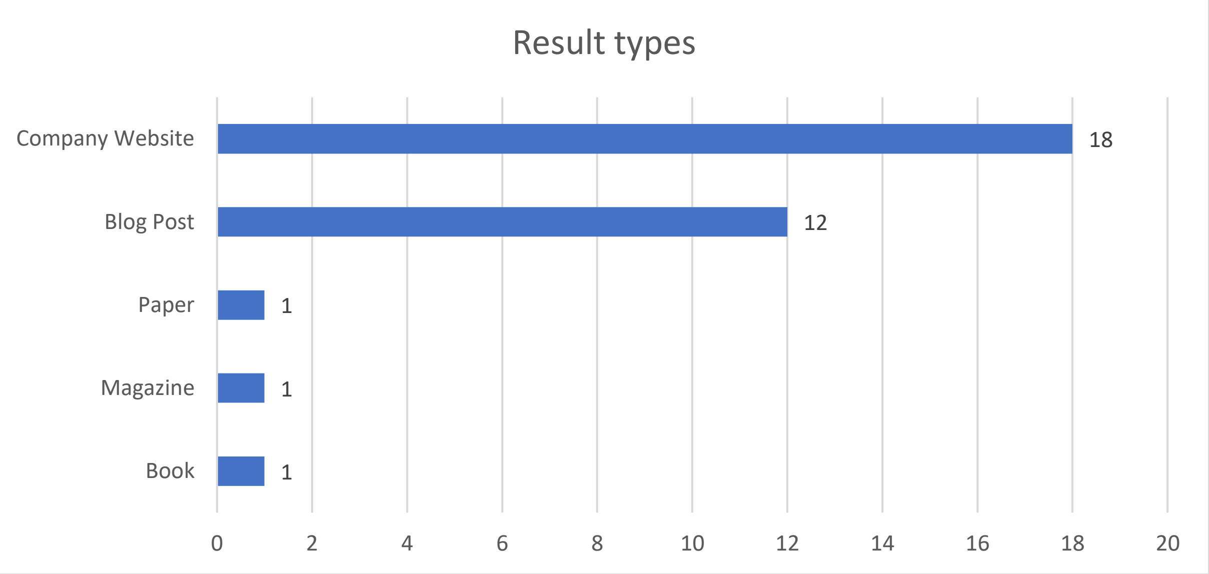

Figure 3: Relevant research results per year

Most of the results are not older than 2 years. Results older than 2005 can also be found but significantly increase in the last couple of years. As the results are mostly in form of websites the content is constantly updated as well as the timestamp of the article.

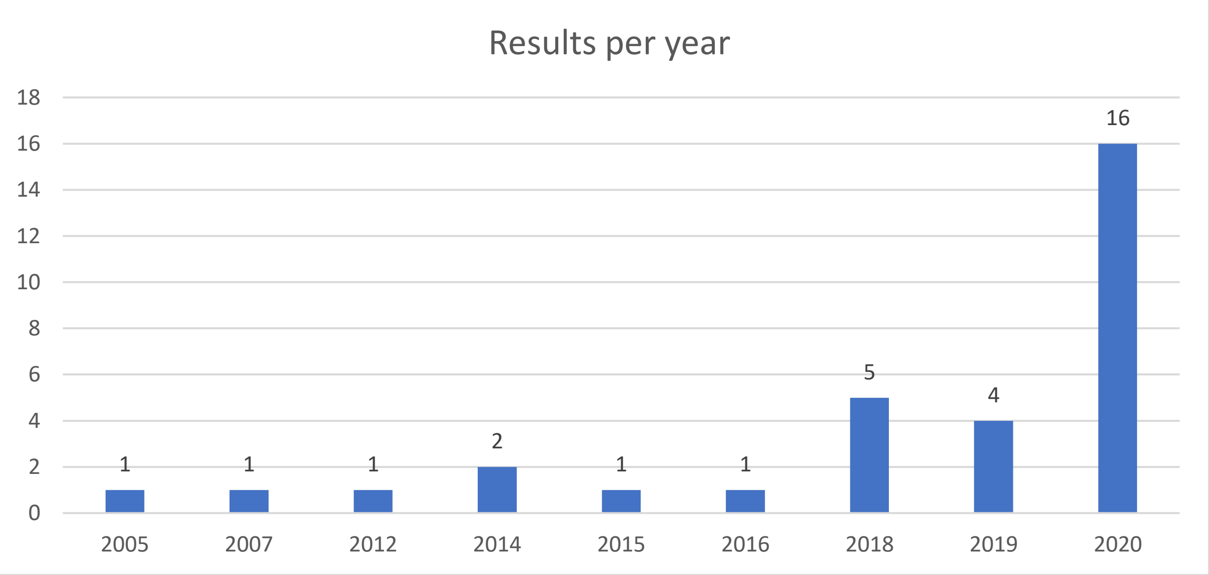

Figure 4: Relevant research results per type

4 Results

4.1 Best practise

From the 32 researched results 25 recommended the usage of specific best practise to conduct a successful Proof of Concept. 21 of them are using fixed steps for their approach.

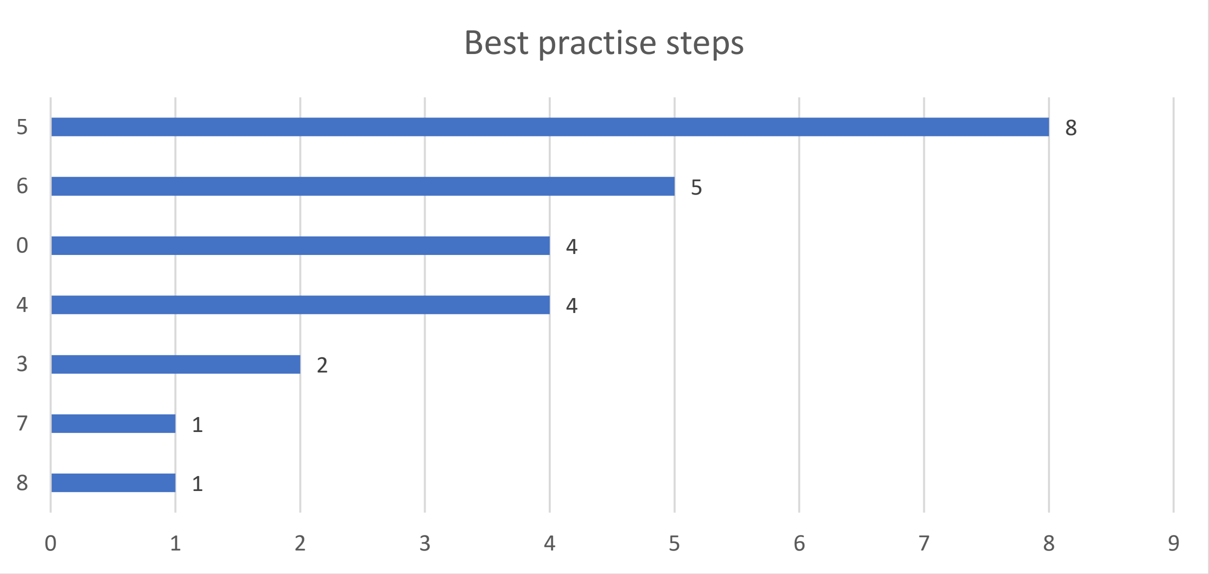

Figure 5: Best practise results per step

Figure 5 shows the number of results that recommend the usage of a certain number of steps displayed on the Y axis. Most of the results use 5 steps. There are two outliers describing more than 6 steps as well as two mentioning only 3 steps. The most steps are described by Shacklett (2019). Not all results recommend using a certain number of steps in general. 4 results just give advice and mention critical key factors.

Best Practise Step 1

The first step in conducting a POC is described in various ways. Malsam (2019) argues that you should begin by defining the duration as well as the effort. As the timeline for a POC is limited the resources need to be used efficiently. Furthermore, the identification of the process owner, relevant IT Systems and success factors should be done here (George, 2014). Divergently Soni (2020) is starting from the product side. At first all information regarding the product should be collected to get a clear picture of the expected result. A similar starting point is outlined by Accenture (2020). In a first POC workshop the functional and technical requirements should be defined, interfaces described, and sample date provided. Shacklett (2019) additionally mentioned the need to identify business cases and provide them to the software vendor. Thereby the quality of the result and testing capability on vendor side can be improved.

Best Practise Step 2

In difference to George (2014), Pham (2020) proposes the selection of the staff as well as the timeline planning in the second step. This approach is also described by Soni (2020) and Add-On Products (2020). Malsam (2019), George (2014) and Ricco (2020) use the second step to align the scope and success factors of the project. This includes the description of the actual status and the measurement of success. A further approach is proposed by Lazarikos (2005) and Accenture (2020). Organizational needs are fulfilled in the first step. Next direct interaction with the product is described. This interaction includes testing with provided sample data as well as discussing first feedbacks.

Best Practise Step 3

Soni (2020) assigns testing and gathering feedback the third step. As there are only three steps mentioned by Lazarikos (2005) and Craine (2019) this is the earliest described finish point of the POC best practise. After aligning business requirements and testing the product, the last step includes the documentation of the results. This document can help to provide the findings to other colleagues and summarize the process. Accenture (2020) uses the third step for a detailed evaluation of the product. Collected strengths and weaknesses can be aligned with the software vendor. While these authors are already getting in touch with the product, others described the organizational needs in more detailed steps. Malsam (2019) and Add-On Products (2020) assign the resources to this step. In another sorting of the best practise steps the setup of performance goals is done in this third step (Shacklett, 2019).

Best Practise Step 4

Step four is considered the last step by Accenture (2020), Malsam (2020), Worthwile (2020) and Ricco (2020). This step contains the final evaluation of the results as well as the documentation. The findings can be used to identify improvements and compare the product to others. As described by Soni (2020) this step is used to improve the product based on the results form step three. Another approach has its first touchpoint with the product in this step. Designing and definition is done on the previous three steps (Qodestack, 2020; ETL Solutions, 2020; Shacklett, 2019).

Best Practise Step 5

Most of the best practises end at the fifth step. Soni (2020) finishes with the evaluation of the results and the creation of a roadmap for the next steps. One crucial factor of this is the documentation. The importance of the documentation is underlined and described as the last steps by Add-On Products (2020) and Qodestack (2020) as well. The detailed best practise steps by Shacklett (2019) and Livingstone (2018) use the fifth step for tracking the metrics and managing the POC process.

Best Practise Step 6

The last step of the process is similar between the described best practises. Next steps, evaluation and documentation are the key activities for completing the process (Wentworth,2014) (Livingstone, 2018) (George, 2014). Shacklett (2019) and ETL Solutions (2020) propose the presentation of the results for this step. This includes the communication of the key results to all stakeholders of the project. Based on the collected metrics the POC can be evaluated as a success or a failure.

Best Practise Step 7 and 8

Only two of the analysed best practises proposed 7 or more steps. The seventh step is the last step of the ETL Solutions (2020) process. It includes the delivery of all results and a documentation the stakeholders. Furthermore, this includes the description of possible next steps. In case of a successful POC, the 7th step by Shacklett (2019) includes the determination of a possible implementation. For this a cost proposition by the vendor is needed. Based on this the assumptions of the finance department can show the cost of the real project. The 8th and final step then includes the budget approval, the clarification of organizational needs and the planning of a timeframe.

4.2 Reported advantages and disadvantages

The usage of a POC has advantages but also shows some risks. As reported by Famuyide (2013) one common misunderstanding between the user and the developer is the development status of the POC. Often the result that is presented to the customer already includes many functionalities and has a proper design proposal. This can lead the user to think, that the development is nearly finished and only a small effort for completion is needed. In reality the product is only a showcase and often needs significant development effort. A further potential risk is the inaccurate definition if the product requirements. As the POC follows the idea of creating a better understanding of the product, changes to the specifications are implied. This can lead to a higher complexity and thereby extend the original scope of the project. The needed resources need to be extended and the cost may rise or the POC will fail completely (tryqa, 2020).

In spite of the small number of reported disadvantages there are several reasons why the conduction of a POC is beneficial. Mentioned as a fundamental advantage of the POC and often described in the literature is the proof that the software can fulfil the requirements. Product demos can only provide a general idea about the functionality as well as the usability. The development of a prototype can show the solutions to the individual business cases. It can also proof that the implementation into the customer specific system landscape is possible (Accenture, 2020; Turban et al., 2018, p.373).

A further advantage can be seen in savings of money and time. Developing a prototype is much faster than developing the final product. The user can interact with the result after a relative short development time. It can be evaluated if the development or the vendor fits the requirements of the business. This gives the possibility to stop the project if necessary. Since there were no investments in terms of software licenses done yet, the costs of stopping the project are relatively low. Furthermore, different software vendors can be compared at the same time (itdesign GmbH, 2020). Lindeboom (2015) underlines the importance of comparing the software products of different vendors. The POC enables the testing of the products at the same time for identical business cases. Advantages and disadvantages between the different solutions can be identified.

The development process can be supported by the POC as well. While designing the prototype errors and inconsistent definitions can be identified. This knowledge allows an early fixing of those problems and can affect the later software architecture. Furthermore, the continuous cross-checking of the development increases the software quality (Gosh, 2019). In conclusion the information gained in a successful but also from unsuccessful POC increase the quality of the final product. Based on the learnings the implementation process can be conducted much faster and the number of errors is significantly lower (Huebner, 2018).

4.3 Common mistakes and problems

In the process of conducting a POC some common mistakes are reported in the literature. To proof that the software vendor is able to fulfil the specific business requirements it is necessary to use concrete and real data. This data should be provided by the customer and includes an extract from his current system (Moreschetti, 2019). The data might include outliers which cannot be handled by the system. Furthermore, the user can test with data that is already known and can be validate. This provides a higher credibility for the product (Hoppe, 2018).

Involving the right people at the right time is a key success factor. Ricco (2020) describes that one of the main mistakes is not to involve the IT team early enough. Because the IT acts as the key resource of knowledge, they need to participate right from the very beginning. Besides the IT team Moreschetti (2018) points out that the team composition in general is crucial to the success. Too few people but also too many people can affect the chances of success.

Another common mistake is done by not clearly defining the requirements and deliverables of the POC. The lack of a clear communication can lead to a misunderstanding, resulting in a product that does not fit to the requirements and does not solve the issues. Before starting the development there should be enough time for exactly understanding the problem. Therefore, a clear ownership of the project needs to be defined. Inconsistency at this point endangers a fluent process (Abbasi, 2015). Defining deliverables can help evaluating whether the product fulfils the needed functionality (Moreschetti,2019).

Further reported problems are:

- Lack of Sponsorship (Abbasi, 2015)
- Poor change management (Abbasi, 2015)

- Unclear cost estimation (Abbasi, 2015)
- Too long timeframe (Moreschetti,2019)
- Vendor does not demonstrate claims (Hoppe, 2018)

5 Discussion

This literature review of the Proof of Concept as a tool for software vendor selection described a variety of best practise steps. Even though all the descriptions include a different number of steps, they all have the same essence. The different steps differ mostly in the degree of detail in the description. Summarizing the best practises 4 main steps can be identified.

First the organizational needs for the POC must be clearly defined. This includes the setup of a time frame, the identification of the process owner and the involved resources. The needed resources can be in form of staff but also IT-Infrastructure. In a second step the expectations need to be aligned. A clear picture of the product is created, success factors are defined, and metrics are implemented. These first two steps are directly related to each other and can be done in any order. On the one hand you cannot define the requirements to the resources properly without a clear product definition. On the other hand, the identification of the product characteristics needs the input from the assigned resources.

In the third step the product is then tested and evaluated. The software vendor needs to provide a prototype of the product including the basic functions to proof that the requirements are feasible. Strengths and weaknesses are identified, and the defined metrics are applied to the product. The last step than includes the documentation of the whole POC process. Results from the evaluation as well as the POC setup is described. Based on the evaluation the product can be identified as a success or a failure. Either way the next steps need then to be setup. In case of a failure a new POC can be planned. If the result satisfies all needs, the financial calculation of a real project needs to be done.

Conducting a POC does not include a high risk. The financial costs of the conduction are relatively low therefore a failure does not lead to a major loss. A well-documented failure can infact result in lessons learned and improve the business. In general, the reported advantages outweigh the risks. The knowledge about the skills of the software vendor and its product enables a detailed evaluation. An online research for software products can only reveal the information the marketing of the vendor shows, where as a POC allows deep insights. Furthermore, the POC can also serve as a basis for the development of the final product. Thereby the selection process can be seen as the first phase of the tool development as well. Furthermore, different products can be evaluated at once and the best solution can be selected.

As most processes and frameworks do include some pitfalls so does the POC. A clear definition about the scope, needed resources and measures need to be defined. In addition, the data needs to meet the requirements of the business case. Inadequate data can lead to a false positive or a false negative result. If the provided data do not include outliers the system might be able to handle it well. While working with real data the system then fails. The false positive includes a high risk, because at this point the project is completed and paid. A false negative result does not pose such a risk. The project would not be started at all. This case can occur when the data is in unrealistic bad condition or includes impossible outliers. The system is not able to handle them and is evaluated as not feasible. Whereby it would be capable of handling the real live data. Summarising the literature describes the usage of POCs as a good solution for the software vendor selection. As the number of software vendors as well as the need of software solutions rise, the problem of a software vendor selection records a rising importance. The chances of POCs increasing in their importance and usage is high. It is therefore most important to know the best practises as well as the common mistakes of conducting a POC.

5.1 Limitations

The literature used for this review is mostly based on non-reviewed articles from blog posts and publications on company websites. In the process of the literate search nearly no journal articles could be fond describing this topic. Thus, the paper is limited to the quality of the used sources. The authors of these texts might have advertising intentions which steer the content into a more positive direction.

Furthermore, it cannot be ruled out that the different methods of software vendor selection are described by other terms than searched here.

References

Abbasi, M. A. (2015). *How to prevent a failed proof of concept* URL: https://blogs.sas.com/content/sascom/2015/05/26/how-to-prevent-a-failed-proof-of-concept/ (visited on 06/14/2020).

Accenture (accessed 2020). *6 BUILD AN AGILE DIGITAL CUSTOMER EXPERIENCE MANAGEMENT PROTOTYPE* URL: https://www.accenture.com/de-de/service-build-agile-digital-customer-experience-management (visited on 06/08/2020).

Add-On Products (accessed 2020). *5 Steps to a Successful Software Proof of Concept* URL https://www.add-on.com/blog/entry/5-steps-to-a-successful-software-proof-of-concept (visited on 06/07/2020).

Bloch, M. and Blumberg, S. and Laartz, J. (2012). *Delivering large-scale IT projects on time, on budget, and on value* URL: https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value (visited on 06/08/2020).

Craine, K. (2019). *Conducting a Proof of Concept: Three Best Practices Crucial to Your Document Automation Software Implementation* URL: https://cdn2.hubspot.net/hubfs/332414/Market-Intelligence/Tipsheets/01-Parascript-Tipsheet-2019.pdf?utm_campaign=MI.Intelligent%20Automation%20Trends.Infographic&utm_medium=email&_hsmi=76882634&_hsenc=p2ANqtz--kFomAUz2KiVVpui2nQF0weDlIMKHdjz15nTq2PrLdS_RdAP59Ki-mVHNuXGCxkT7prqd4TE-FOchS9k1OX-GVvfpYdKg&utm_content=76882634&utm_source=hs_automation (visited on 06/09/2020).

Deyan, G. (accessed 2020). *What Is Proof of Concept and Do You Need One in 2020?* URL: https://techjury.net/blog/what-is-proof-of-concept/#gref (visited on 06/08/2020).

ETL Solutions (accessed 2020). *An example of a successful proof of concept* URL: https://www.etlsolutions.com/new/an-example-of-a-successful-proof-of-concept/ (visited on 06/09/2020).

Eeles, P. (2006). *The process of software architecting* URL: https://www.ibm.com/developerworks/rational/library/apr06/eeles/index.html (visited on 06/14/2020).

Famuyide, A. (2013). *Proof of Concept: Benefits & Risks of Prototyping in Business Analysis* URL: https://businessanalystlearnings.com/blog/2013/9/1/proof-of-concept-benefits-risks-of-prototyping-in-business-analysis (visited on 06/14/2020).

Flyvbjerg, B. and Budizer, A. (2011). *Why Your IT Project May Be Riskier Than You Think* URL: https://hbr.org/2011/09/why-your-it-project-may-be-riskier-than-you-think (visited on 06/08/2020).

Gartner (2020). *Information technology (IT) spending on enterprise software worldwide, from 2009 to 2020 (in billion U.S. dollars)* URL: https://www.statista.com/statistics/203428/total-enterprise-software-revenue-forecast/ (visited on 06/14/2020).

Geroge, M. (2014). *6 Steps to Make your Software Proof of Concept a Success* URL: https://www.linkedin.com/pulse/20140828131241-10893657-the-software-proof-of-concept-6-vital-components-for-success (visited on 06/08/2020).

Gosh, T. (2019). *Proof of Concept in Testing: What Should You Know About It?* URL: https://aphawks.com/blog/proof-of-concept-in-testing/ (visited on 06/14/2020).

Hoppe, G. (2018). *4 Signs Your Business Intelligence Software Proof Of Concept Is a Dud* URL: https://blog.capterra.com/signs-your-business-intelligence-software-proof-of-concept-is-a-dud/ (visited on 06/14/2020).

Huebner, C. (2018). *Proof of Concept: A waste of time and money?* URL: https://www.mirantis.com/blog/proof-of-concept-a-waste-of-time-and-money/ (visited on 06/14/2020).

Itdeign GmbH (Accessed 2020). *Best Practice für Ihre Software-Auswahl: Proof of Concept statt Ausschreibung* URL: https://ppm.itdesign.de/know-how/best-practice-fuer-ihre-software-auswahl-proof-of-concept-poc-statt-ausschreibung/ (visited on 06/14/2020).

Lazarikos, D. (2005). *How to pull off a successful proof of concept* URL: https://www.computerworld.com/article/2559158/how-to-pull-off-a-successful-proof-of-concept.html (visited on 06/07/2020).

Lenarduzzi V. and D. Taibi, (2016). “*MVP Explained: A Systematic Mapping Study on the Definitions of Minimal Viable Product”* In *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Limassol, pp. 112-119

Lindeboom, R. (2015). *So, what is the added value of a Proof of Concept or Pilot?* URL: https://www.linkedin.com/pulse/so-what-added-value-proof-concept-pilot-rene-lindeboom (visited on 06/14/2020).

Lippincott, J. (2018). *How a 'proof of concept' can help businesses minimize risk, save costs* URL: https://www.bizjournals.com/losangeles/news/2018/08/30/how-a-proof-of-concept-can-help-businesses.html (visited on 06/14/2020).

Livingstone (2018). *6 Steps to Make your SAM Proof of Concept A Success* URL: https://www.livingstone-tech.com/make-your-sam-proof-of-concept-a-success/ (visited on 06/09/2020).

Malsam, W. (2019). *Proof of Concept: Definition & Best Practices* URL https://www.projectmanager.com/blog/proof-of-concept-definition (visited on 06/07/2020).

Moerschetti, J. (2019). *Top 6 Mistakes to Avoid When Running a Data Science POC* URL: https://blog.dataiku.com/top-6-mistakes-to-avoid-when-running-a-data-science-poc (visited on 06/14/2020).

Pham, T. (accessed 2020). *Proof Of Concept In Software Development: 5 Critical Factors To Success* URL: https://saigontechnology.com/blog/proof-of-concept-in-software-development-5-critical-factors-to-success (visited on 06/07/2020).

Project Management Institute (2017). *9th Global Project Management Survey: Success Rates Rise - Transforming the high cost of low performance* URL: https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf (visited on 06/08/2020).

Quodestack (accessed 2020). *How to Conduct Proof of Concept* URL: https://qodestack.com/how-to-conduct-proof-of-concept-and-evaluate-automation-tools/ (visited on 06/09/2020).

Ricco, E. (2020). *HOW TO RUN A SUCCESSFUL PROOF OF CONCEPT – LESSONS FROM HUBSPOT* URL: https://learn.filtered.com/thoughts/how-to-run-a-successful-proof-of-concept-hubspot (visited on 06/09/2020).

Rouse, M. (accessed 2020). *Proof of Concept* URL: https://whatis.techtarget.com/de/definition/Proof-of-Concept (visited on 06/07/2020).

Shacklett, M. (2019). H*ow to set up a proof of concept project* URL: https://www.techrepublic.com/article/how-to-set-up-a-proof-of-concept-project/ (visited on 06/08/2020).

Singaram, M. and Jain, P. (2018). *What is the Difference between Proof of Concept and Prototype?* URL https://www.entrepreneur.com/article/307454 (visited on 06/07/2020).

Soni, C. (2020). *A Complete Guide for Proof of Concept In Successful Software Development* URL: https://www.softwaresuggest.com/blog/proof-of-concept-in-software-development/ (visited on 06/07/2020).

Tryqa (accessed 2020). *What is Prototype model- advantages, disadvantages and when to use it?* URL: http://tryqa.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/ (visited on 06/14/2020).

Turban, E. and Pollard, C. and Wood, G. (2018). *Information Technology for Management: On-Demand Strategies for Performance, Growth and Sustainability*. 11th Edition. Hoboken, NJ: John Wiley & Sons.

Wentworth, T. (2018). *Don't Skip the Proof Of Concept* URL: https://www.cmswire.com/cms/information-management/dont-skip-the-proof-of-concept-026523.php (visited on 06/09/2020).

Worthwile (accessed 2020). *Four Steps That Will Guarantee a Profitable Technology Proof of Concept* URL: https://worthwhile.com/insights/2018/01/22/technology-proof-of-concept/ (visited on 06/09/2020).