fhwedel
UNIVERSITY OF APPLIED SCIENCES

LEGACY SYSTEMS

IT-Security Seminar

RICHARD WAGENER

Leitung:
Prof. Dr. Gerd Beuster

7. JULI 2019

TABLE OF CONTENTS

Legacy Systems

Richard Wagener
Student of IT-Security

ABSTRACT

Legacy Systems which are not mainted do not only grow in size but also in complexity. Over time chances are high that a company is stuck with a system which can not be extended or adapted without great exspenses. In this case the company is forced to take actions. The legacy system has to be addressed to restore an acceptable risk level and a maintainable state of the system. Using this paper as guidance a company can chose reasonable procedings on how to identify threats and how to address them properly.

KEYWORDS

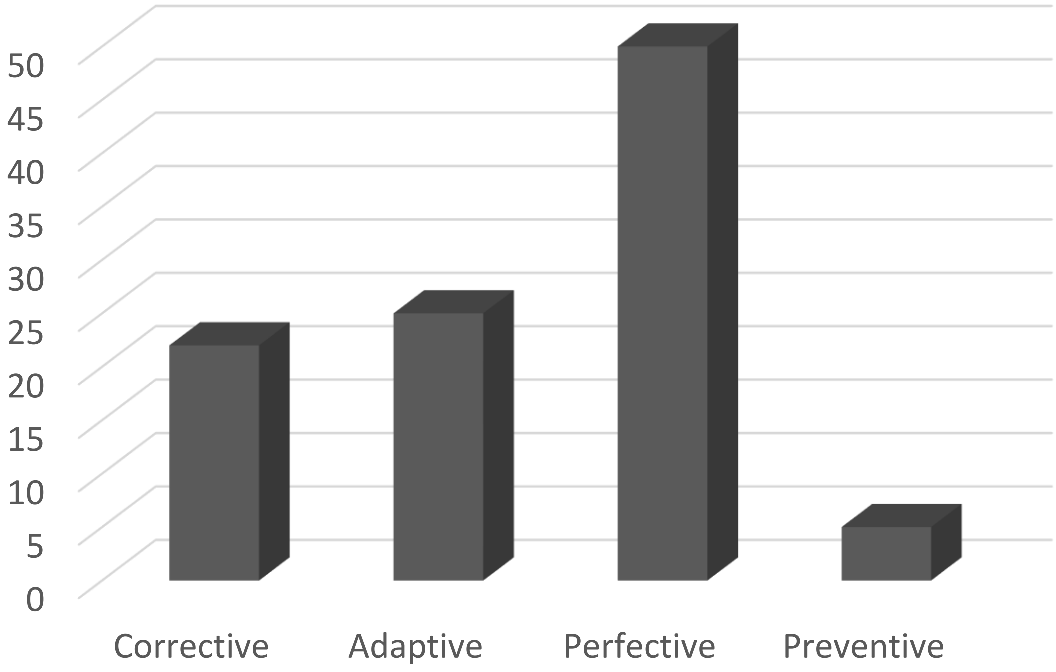
It-Security Seminar; Legacy System Modernization; Risk Management Framework; DSGVO; Black-Box Modernization; White-Box Modernization; SOA; General Data Protection Regulation

INTRODUCTION

The sentence “Security is a process, not a product” [1] , describes the status of information systems in 2019 very accurate. Vulnerabilities found in information systems evolve with time and so do corresponding counter measures. This is why it is important to address the issues of legacy systems in todays business and not view security as a product which does not require further changes since it is completed, because it is not. Information systems grow over time and associated security issues are not addressed resulting in aging legacy systems. Since companies are profit oriented, security is not the highest priority.

Most changes to an information system are perfective measures. New system requirements emerge naturally over time due to changing goals of organization. The company has to adapt to their customers and therefore new features are added to the system. Furthermore some adaptive measures are done to be compatible to new innovations introduced and some corrective to fix error behaviour (Bugs). Only about five percent of the work invested into a information system is actually preventive, see figure 1. This is a huge problem, since these measures prevent information systems from becoming very cost intensive to maintain, adept or evolve or generally insecure.

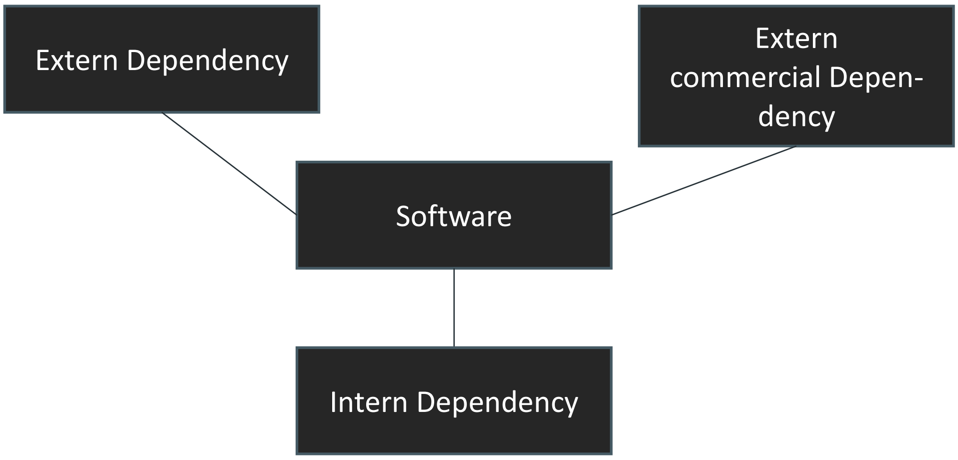
Figure 1: Distribution of maintenance by categories [2]



Legacy Systems

Even during development of an information system the first signs of aging can be found. Most software uses combination of intern, extern or even commercial extern dependencies, see figure 2. Since commercial software is released in specific time intervals and not each feature by itself and a software developer will most likely not update dependencies with a new updates available, it is likely that the developed software is not using the latest version of every dependency it references. If the product is already in the process of quality assurance it is even more unlikely that updates will take place.

Figure 2: Information system components - Illustration



Legacy systems is a wide topic, including multiple different kind of software systems. Typical IT systems have changed over time and so did their associated threats and mitigation strategies.

A company has to address different kinds of legacy systems, from newly developed once to old legacy systems to minimize security risk. In order to no lose the bigger picture or miss important aspects of every single system, standardized proceedings should be used. These are introduced widely in the main part of this paper.

LEGACY SYSTEM PROTECTION

A common approach to legacy system protection is using standardized procedures. The Risk Management Framework serves the purpose of minimizing chances of human error by wrapping necessary proceedings in smaller segments which are easier to supervise.

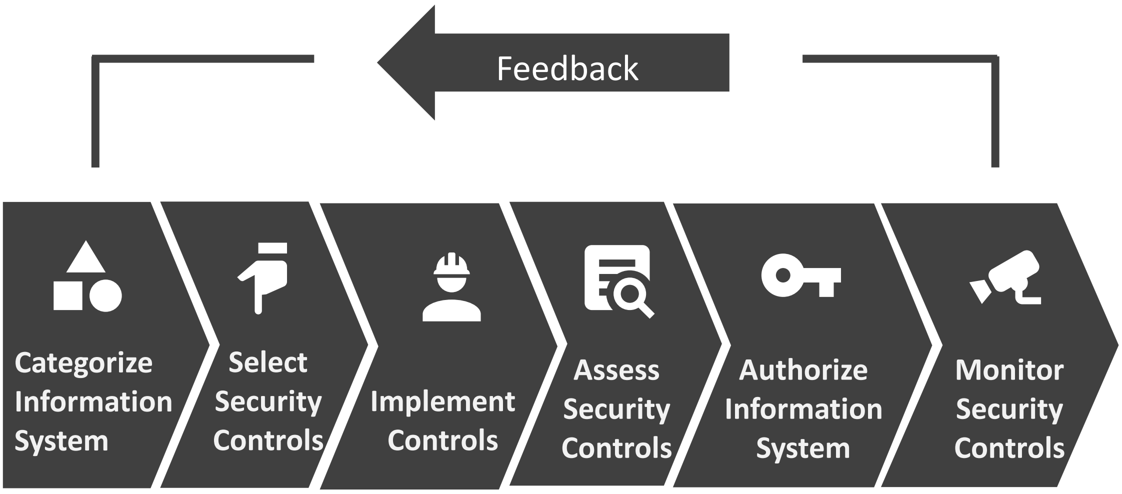
Risk Management Framework

In order to be able to find appropriate measures to secure an information system, its security issues have to be understood first. Only afterwards fitting mitigation strategies can be designed and implemented to minimize the posing thread for a company’s operational goals. A structured approach is best practice to not miss crucial aspects of a system or to optimize the analysing process of information systems. By applying standardized proceedings human errors can be minimized and the overall progress can be accelerated.

The Risk Management Framework is a policy designed by the National Institute of Standards and Technology (NIST) which can help in this regard [3]. Similar guidance can be the ISO 31000 or the BSI-Standard 200-3, Risk Analysis based on IT-Grundschutz [4] [5]. Subsequently only the RMF will be covered in greater detail. In short the RMF does help understanding the business context by extracting technical risks in order to determine effects on goals of the organization.

The Risk Management Framework includes six steps: Categorize, Select, Implement, Assess, Authorize and Monitor, see figure 3. These steps are designed as a cycle, see figure 3, illustrating, that the concept of IT-Security is a process and not a product. The main focus lies on the identification of threads, selection and implementation of suitable control measures. The remaining steps are related to quality management and they depict only an amendment to the RMF concept.

Figure 3: Risk Management Framework [3]


Feedback
Categorize Information System
Select Security Controls
Implement Controls
Assess Security Controls
Authorize Information System
Monitor Security Controls

Categorize

Categorizing covers the process of initial information gathering. An information system is analysed regarding information processed or stored. This is usually done by using impact analysis. For more information in regard to impact analysis the ISO 27001/02 can be consulted for example. To give an idea, methods to compare likelihood of an security incident to effect on the operational goals can be utilized.

A company, not knowing which of the information systems it possesses has to determine the extent necessary to identify the status quo and document it. The following exemplary categories can be used to split all information systems of a company in distinct groups [6]:

- Describe the systems tasks
- Release Date (Go Live)
- Development circumstances
    - In-House
    - Contractor-developed
    - Commetcial off-the-shelf (COTS)
- System’s type
    - Main frame system

        - Client server application
        - Browser based application
        - Desktop application
    - Software facts
        - Language
        - Framework
        - Type of database
    - Internal or external (e.g. Internet) usage
    - Sensitive information
        - Health related data
        - Financial data
    - Other information
        - Existing documentation (user, developer)
        - Scope of testing
        - Context person

After collecting all relevant information about existing IT systems, this information can be used to identify ones, which are critical for the company's organizational goals. Examining all existing systems would provide the most comprehensive protection but is not cost efficient in most cases. Considering the financial aspects and limited posibilities of such an examination on all running IT systems within the organization or company shows, that only the major systems can be examind effectively, since broad focus leads to delay of action.

To conduct a system ranking based on the risk level, the company's specific requirements come into play. Important at this point is, that no analysis of the systems have to take place. The assessment should start with the information at hand. If processed or stored data is sensitive for example, it might be reasonable to prioritize this system for further modernization.

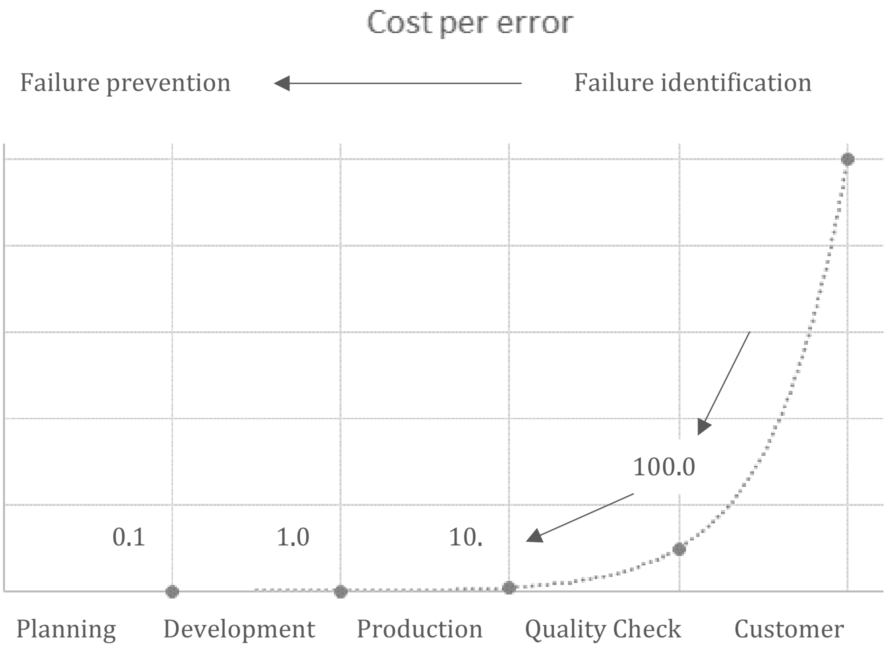
Select controls

For identified high risk information systems controls have to be chosen which will minimize the residual risk below the acceptence level. In order to do so a complete investigation of the targeted system has to take place first.

The categorization done at the beginning of the Risk Management Framework realization can be used to simplyfy further evaluation and might even be used for partially automizing the process by providing common threads for certain categories. Checklists might also help reducing the factor of human error. The most common thread of browser based applications for example is injection according to the OWASP Top 10, but might be irrelevant for a desktop application without database [7]. By addressing known threads of certain legacy systems a smaller workload remains which has to be analysed manually. In order to address threads not covered by the categorization step threat modelling can be used to analyse the system further.

Based on the result of the comprehensive inspection controls can be selected or it might be possible to have to reconsider the decision of labeling the legacy system as high risk system. Since costs tend to grow with a factor of ten from each product step (development, testing, production, etc.), see figure 4, this is the best possible moment to consider decisions made earlier. If it turns out the inspected system is rather secure and should not be modernized or at least not be modernized first.

Figure 4: Rule of Ten [8]



Using the threat modeling results, controls to reduce the overall thread of chosen systems to acceptable level, can be described. Furthermore recognized advantages and pitfalls should be noted as well. At least final mitigation options should be thoroughly weighed and considered.

The most basic ones are enhancing, hardening, replacing the system or deciding that nothing is to be done.

Implement controls

A company has two main options when it comes to securing legacy systems, the Big Bang - (Cold Turkey) or the incremential approach [9]. By following a Big Bang approach all action take place at once. This could for example result in a complete redevelopment of the system using new tools, architectures, databases and hardware. The big disadvantage of this approach is the high cost factor and the fact that found bugs have to be fixed in the active legacy system and the on in development. Same goes for adaptations. Furthermore it takes a longer time until the system can go live. Using the incremential approach a company can introduce improvements step by step. Early changes can be implemented in the live system while others are still in development. No redundant system changes are necessary.

In oder to chose the right approach the following risks and benefits can be analysed [10]:

- Cost benefit analysis
- Obstacle estimation
- Business value estimation
- Life expectancy
- Maintenance costs
- Difficulty of migration
- Relation to other systems
- Stability of system
- Reuseable components

Redevelopment:

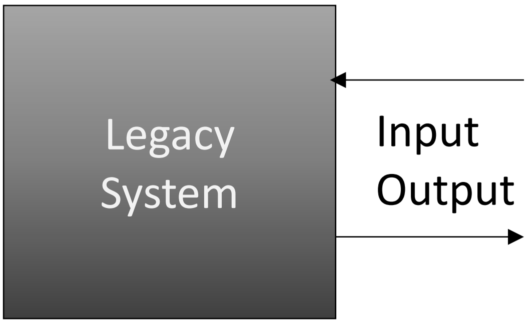
If modernization is not sufficient and a company decides to redevelop a system, the legacy information system in use has to be understood first. Afterwards the target system can be developed and after a testing phase it can be migrated and go live. These steps would be in accordance with the PDCA cycle. A company would most likely want new features to be added to the system as well, making this approach even more costly [2].

Modernization:

There are two groups of modernization proceedings, Black-Box and White-Box modernization. White-Box modernization, illustrated in figure 5, requires complete or partial knowledge of the legacy system. Using this knowledge a partial code replacement could take place or critical security issues could be fixed for example.

Black-Box modernization, illustrated in figure 6, uses only the system's input and output and no further information about the system, which makes this approach very cost efficient and fast. The downside is, that it is not always sufficient. Often further changes -with knowledge of the system - are required, making the modernization process more or less a White-Box modernization.

Figure 5: Black-Box Modernization



The modernization includes in both cases the following steps:

- Risk and benefit evaluation
- Legacy system understanding
- Implementing
- Testing
- Go Live

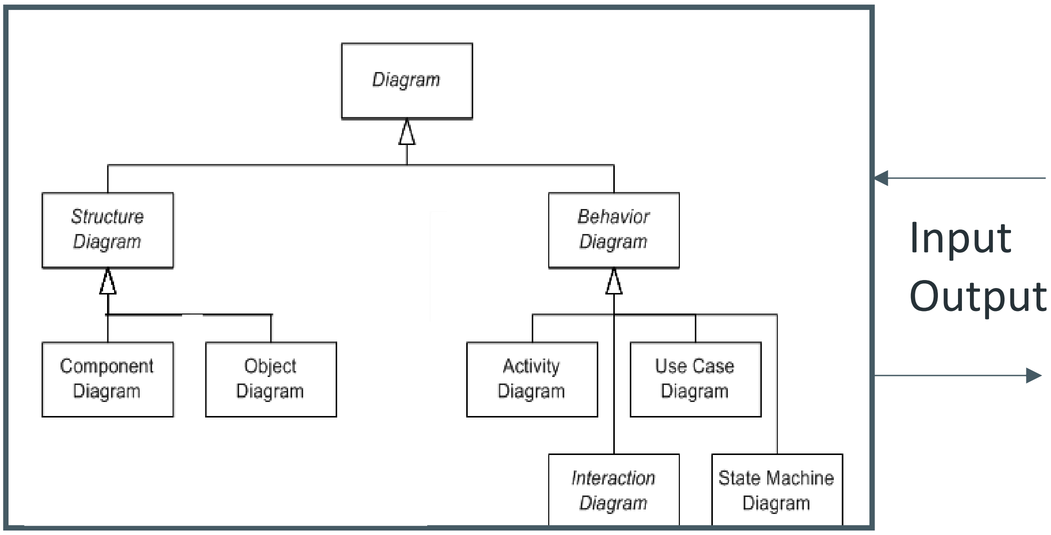
But the actual way the implementation is done differs significantly. For the Black-Box modernization the input and output is examined, with the goal of adding a Wrapper around the legacy system translating old in and output to modern standardized interfaces.

The White-Box approach consists of multiple possible strategies, for example:

- Examine Documentation, …
- Modeling the domain
- Extracting information from the code
- Create abstractions describing underlying system structure

With these strategies a system code restructuring, a part code replacement or a complete code migration are viable. [2]

Figure 6: White-Box Modernization

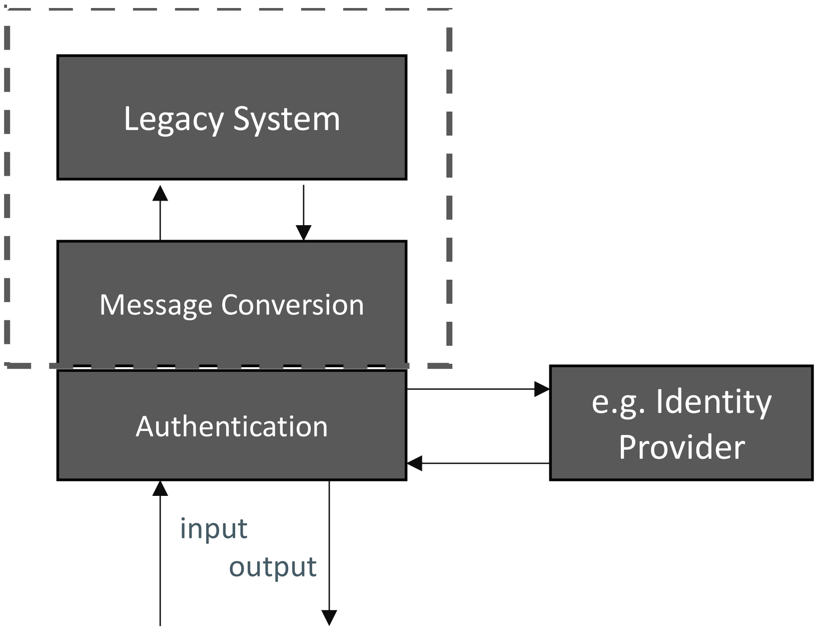


SOA:

Modern legacy system modernization goes one step further and introduces application logic and security as a reusable services [11]. With standardized interfaces authentication can be done using an external identity provider, see figure 7. This approach is usually combined with wrapping the existing legacy system, since the systems in and output must match the standardized interfaces of security or application logic services. By splitting a large problem into smaller concerns both smaller problems can be addressed independently using a baseline of conventions. This makes the SOA approach very convenient to use as it is better contructed, carried out and managed and therefore allows distribution [12].

SOA is very cost efficient. Security features are implemented once and can be used by multiple different systems. Even further replacements of these services can be done with ease using the standarized communication interfaces, making this approach agile and easy to maintain.

Figure 7: SOA - Visualization



Assess controls

After all necessary controls have been implemented the system has to be evaluated again. All requirements for the system have to be met and the security acceptence level of the information system should be on the desired level. In short the asses control step verifies that all product requirements in regard to functionallity and security are met

Authorize system

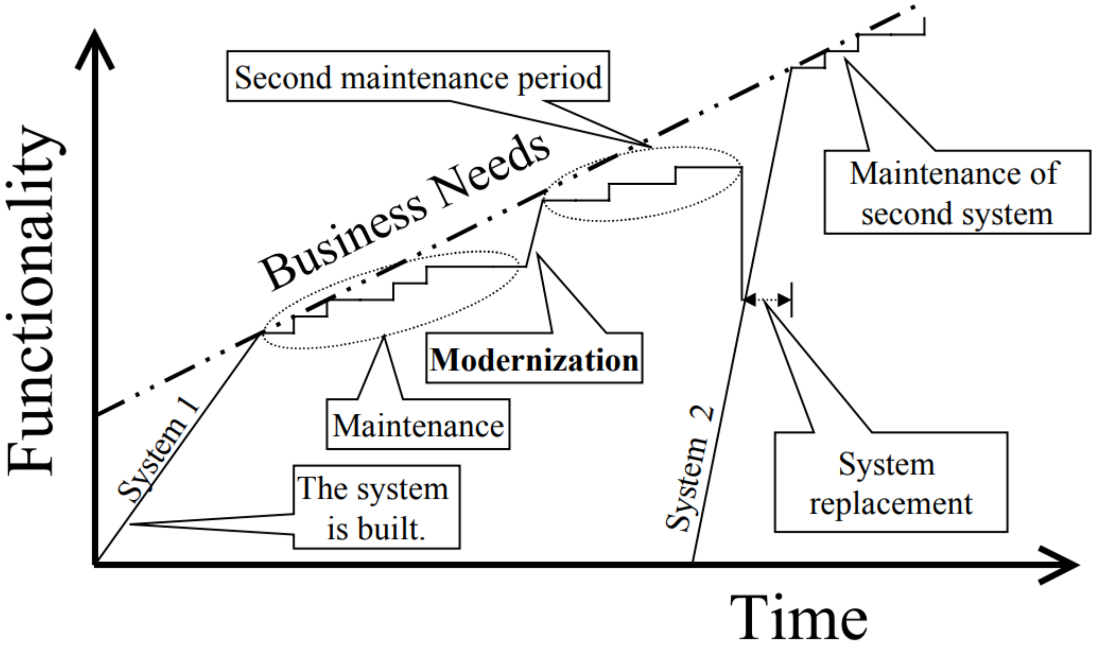
A responsible personal has to check if the control assessment was successful and authorize the commissioning. The responsible personal can act as contact point for errors of any kind and minimues the risk of upcoming threads beeing unaddressed.

Monitor controls

Its common in quality management to not stop after implementing required controls but to continue to monitor results, implemented controls in this case. Similar to the PDCA cycle (Plan, Do, Check, Act), the Risk Management Framework advises to continue to monitor the information system in order to prevent the system to become a legacy system again. By investing more in preventive measures a company can save costs and mimimize thread for future usage as well.

Similar to figure 5, by applying preventive measures the timespan from system replacements can be extended and the maintenance cost can be reduced.

Figure 8: Information System's Functionality over time [13]



General Data Protection Regulation

In regard to legacy systems the changes to german law through the introduction of the european directive General Data Protection Regulation (GDPR) is important. According to Art.4, Nr.2, sensitive user data now have to be managed for the entire life cycle of the system [14]. This includes newly developed systems as well as legacy systems. Furthermore organizations have to provide information about size, source, procession and security of personal data at any time [15]. In conclusion the GDPR forces companies to address legacy system's security threads.

CONCLUSION

In order to delay the aging process of an information system periodic refreshes are strongly recommended. Adding new features to a system without optimizing the given code bases most likely results in more interconnected code. Old technology might not be compatible with new features, forcing developers to introduce workarounds which leads to more written code overall. If these code parts are not documented further improvements are even more work intense until a modernization or replacement is unavoidable. The 2. Lehman's law states "as am E-type system evolves, its complexity increases unless work is done to maintain or reduce it" [16]. The use of Risk Management Framework or similar approaches can help a company to make the difficult decision wheather to modernize or replace a legacy system. Analysing existing legacy systems systematically and assessing their threads is important in order to lay a foundation for further actions. Using the techniques described a company can evaluate which one fits their requirements best. Furthermore the important risks and benefits of the improvements are described giving advise which pitfalls could emerge.

FIGURES

BIBLIOGRAPHY

[1] H. P. Solutions (2014): Mitigating Cyber Security Risks in Legacy Process Control Systems, [Online] https://www.honeywellprocess.com/library/marketing/whitepapers/cyber-security-legacy-systems.pdf [access on 22.06.2019].

[2] Seacord R. C. et al. (2003): Modernizing Legacy Systems - Software Technologies Engineering Processes, and Business Practices, C. M. S. E. Institute, Hrsg., Boston: Addison-Wesley.

[3] J. T. F. T. Initiative (2010): Guide for Applying the Risk MAnagement Framework to Federal Information Systems: a Security Life Cycle Approach. [Online] https://csrc.nist.gov/publications/detail/sp/800-37/rev-1/final [access on 05.06.2019].

[4] I. o. R. Management (2018): A Risk Practitioners Guide to ISO 31000:2018. [Online] https://www.theirm.org/media/3513119/IRM-Report-ISO-31000-2018-v3.pdf [access on 05.07.2019].

[5] B. f. S. i. d. Informationstechnik (2017): BSI-Standard 200-3 [Online] https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/International/bsi-standard-2003_en_pdf.pdf?__blob=publicationFile&v=2 [access on 10.06.2019].

[6] Weber C. C. (2006): Assessing Security Risk In Legacy Systems. [Online] https://www.us-cert.gov/bsi/articles/best-practices/legacy-systems/assessing-security-risk-in-legacy-systems [access on 12.06.2019].

[7] O. Foundation (2017): OWASP Top 10 - 2017 [Online] https://www.owasp.org/images/9/90/OWASP_Top_10-2017_de_V1.0.pdf [access on 06.06.2019].

[8] Barth M. (2009): Object-oriented engineering data exchange as a base for automatic generation of simulation models. [Online] https://www.researchgate.net/publication/224115846_Object-oriented_engineering_data_exchange_as_a_base_for_automatic_generation_of_simulation_models [access on 22.07.2019].

[9] Kaps S. (2016): Migrationsstrategien im Vergleich. [Online] https://www.heise.de/developer/artikel/Migrationsstrategien-im-Vergleich-3283418.html?seite=all [access on 10.06.2019].

[10] Bisbal J. und Richardson R.: A Survey of Research into Legacy System Migration. [Online] https://pdfs.semanticscholar.org/62f6/e11e3153a3be7e5aef509cc00c9aa2bf6bac.pdf [access on 03.07.2019].

[11] Bundesamt für Sicherheit in der Informationstechnik: Architecture for an SOA security framework. [Online] https://www.bsi.bund.de/EN/Topics/OtherTopics/SOAsecurity/SOAsecurityframework/Architecture/fw-arch.html [access on 20.06.2019].

[12] Erl T. (2005): Service-Oriented Architecture - Concepts, Technology and Design. Prentice Hall.

[13] Corrêa P. (2015): Architecture to Application Gateway to access Electronic Government Legacy System. [Online] https://www.researchgate.net/publication/237303936_Architecture_to_Application_Gateway_to_access_Electronic_Government_Legacy_System [access on 21.07.2019].

[14] E. P. a. Council, (2016): DSGVO. [Online] https://dsgvo-gesetz.de/art-4-dsgvo/ [access on 06.05.2019].

[15] Tmaxsoft (2017): Legacy Datensysteme machen Compliance-Risiko bei DSGVO unkalkulierbar.

[16] Lehman M. M. (2003): On understanding laws, evolution, and conservation in the large-program life cycle. Journal of Systems and Software, Nr. Volume 1, pp. 213-221.