Preface

MÁRIA RAFFAI
Editor in Chief; Chair of GIKOF/SEFBIS SIG
Professor at Széchenyi University, NR in and Councilor of IFIP,
vice chair of IFIP Information Science TC Enterprise Information Systems WG
eMail: maria.raffai@gmail.hu

The Scientific and Educational Forum for Business Information Systems (SEFBIS) was founded in 2001. This Special Interest Group of the John von Neumann Computer Society organizes not only conferences BIS but publishes scientific papers in professional journals both in Hungarian and English languages. The result is 13 Volumes of SEFBIS Journal (in English) and 11 Volumes of GIKOF Journal (in Hungarian). Sorry to say but because of COVID-pandemic the year of 2020 was an inactive period of our community when for lack of papers we could not publish Journals, and even because of the injunctions related to organize events we could not arrange our yearly conference in 2020.The present publication is the volume of the Year 2021; but a special one, because it contains papers both in English and Hungarian. It manifests our efforts and improves that the professionals need opportunity for publishing their works.

The SEFBIS SIG Community celebrates the 20th Anniversary of its existence in 2021 when we look back to our two decades activity and summarize our results. There was published a special volume of our journal-series, the *"2001–2021, Twenty Years attached in Business Information Systems"* gives a detailed overview about these two decades activity and the results. This is a good opportunity for making the significance and the justification of our existence widely known among the professionals. The present bilingual volume aims to show the colorful interest of our members, the many different fields of ICT they are dealing with, and together with the historical volume it calls the attention to the events of 2021 and 2022 that the readers might be interested in.

As we learned from our conferences, from our forums and discussions the readers are interested in

- the effect of COVID pandemic on the business procedures, the production and all other branches of the economy,
- the concepts, methods and tools supporting the IS development processes,
- the role and the impact of IS/IT on business and on society,
- the solutions that satisfy customers' requirements and ensures security and privacy, and last but not least
- the Education Space focusing to the field of business information systems.

Performing our goals also in the future we call the researchers, professionals, developers to report on their results, the efficient business and/or educational solutions. Concluding with my sincere greeting to the Readers I wish to think back to our history and obtain new knowledge, new ideas, about the effective ICT innovations in business and research results from all over the world!

The electronic version of the Journal is citied and downloadable from our HomePages: https://raffa6.wixsite.com/gikof/szakfolyoirat, http://raffa6.wix.com/sefbis/!journals.and also from EBSCO Database.

Maria RAFFAI
Editor in Chief

Monitoring the Duration of Use of Web-based Systems Using Clustering Procedures

BÁLINT MÁTYUS
PhD student, Corvinus University of Budapest
Email: balint.matyus@stud.uni-corvinus.hu

ABSTRACT

Time spent by employees in online training may be a significant cost factor. There is a need for automated measurement of active usage related to e-learning systems. However, in academic literature there is no generally accepted approach for either the definition of system usage or its quantification. This paper reviews the major approaches, and it also attempts to identify active sessions and determine their duration based on user log data related to e-learning systems. Empirical data were provided by students' activity measured during two courses at Corvinus University of Budapest. The results suggest significant activity fragmentation. Based on my own results, it can be clearly established that the overall time spent is extremely sensitive to the initial assumptions and parameters, therefore it is imperative to present the conceptualization and the operationalization of active system usage for future research.

Introduction

In the course of researching information systems, a common demand is that system usage should be measured. There are several approaches for conceptualizing and operationalizing the phenomenon in academic literature, however, there has not been an agreement as to what qualifies as system usage and how it should be measured. This has been a central element of academic interest since the 1970s. At the time, the first models made an attempt to provide an explanation for successful introduction of information systems and their utilization in business environment. [1] From the 2000s on, central aspects of system usage were defined *inter alia* by [2] and [3].

System usage is measured in academic literature using self-reported questionnaires (subjective method) or automated session identification procedures based on activity logs (objective method). The advantage of the former method is that several validated theories (and measurement methods) are available, by means of which a questionnaire can be drawn up with a view to exploring user behaviour related to technology. However, due to memory bias, self-selection bias and confirmation bias, measured data may lead to questionable results. According to [4], respondents tend to overestimate their own activity.

The advantage of automatic identification methods is that they enable to examine the whole population at costs independent of population size. The accuracy, reliability, and validity of the data are determined by the capabilities and limitations of the automated measurement method. The disadvantage of this approach is, however, that the context of system usage may not be learnt by logging procedures. For example, the fact that a software is running does not necessarily mean that the system is being used. [5] points out that there is only a handful of research papers, that measure the very same aspect of system usage applying both subjective and objective methods, as a consequence of which there is not enough data to compare the two methods.

The difficulty of quantifying actual usage is indicated by several examples in academic literature, which do not even attempt to measure actual usage, but only form a hypothesis in relation to the user intention to use. [6]–[8] One reason accounting for this might be that questionnaire measurements of usage

may lead to results of questionable accuracy. Beyond that measuring actual usage requires existing and functioning technology that logs user activity and provides access to them. Enterprise information systems are typically closed source and generally it is not feasible to insert custom metrical procedures. In the absence of an interface, it is not possible to measure usage duration.

This paper examines the possibilities of quantifying system usage with objective methods in the case of a system that is built on a client-server architecture and accessible from a web browser. The objective of the research is to explore active sessions on the server side based on automatically collectible log data, and to quantify system usage on a time-based basis. The paper primarily focuses on system usage related to e-learning services.

The starting point of *online status* is when a user opens the website, and an explicit end is when the user leaves the website or when the IT service terminates. Users are considered to be active when they interact with the website. Within the online status, active presence must be distinguished. Online status and active presence can be defined as a time interval enclosed by a start datetime and an end datetime, which is referred to as a *session* in this paper. Consequently, *online session* that identify a live network connection and *active session* that quantify an active presence can be differentiated.

Logging user activity can be ensured using preinstalled browser plug-ins or web scripts. If there is a continuous live connection between the client and the server (e.g.: via a web socket) more detailed information can be obtained on system usage in addition to logging HTTP navigation events. Examples include events related to switching windows, playing video content, or scrolling the website. The following research questions can be formulated:

- Q1: How is actual system usage quantified using objective methods in the literature?
- Q2: How can active sessions be identified based on log files?
- Q3: How accurately can the duration of active sessions be estimated based on log data?

Practice in Academic Literature

The aspect of time has been permanently represented in the decades of practice for the quantification of system usage. The degree of usage was already expressed by means of usage *frequency* by the Technology Acceptance Model (TAM). [9] System usage was quantified by means of usage *duration* validating the Unified Theory of Acceptance and Use of Technology (UTAUT). [10] In 2008, [3] also considered usage *intensity* in addition to the aforementioned two aspects.

During the validation of the UTAUT, usage of the examined systems was determined based on log data. After 5–15 minutes of inactivity, the system logged out the users, as a result, according to the authors, the majority of the inactive periods have been filtered out. [10] A similar approach was adopted also by [11], and [12] for the quantification of actual usage. The methodology of measuring the duration of system usage was not explained in detail in these papers. It is particularly interesting that in their paper written a few years later, [3] raise the challenges of the measurement of actual system usage. In their view, the issue of system usage quantification is typically addressed superficially in academic literature. The aspects of selecting the method used for quantification is not shared by the authors in sufficient depth, although it is the most important dependent variable of the models.

According to [2], it is important to note that in addition to these three aspects, it should also be considered for what reason users use the system and whether users were able to focus on their task or their attention was divided with other activities. These aspects of activity can be measured to a limited extent by an automated logging procedure; therefore, this paper focuses on the three aspects listed.

[13] conducted academic literature meta-analysis based on 36 papers. The qualitative research sheds light on the magnitude of the proportions observed in the practice of quantifying collaboration system usage. The quantity of messages was examined most often (27 cases), but the content quality of the messages (17 cases) and user perception (14 cases) can also be considered outstanding. System access frequency (5 cases), the number of messages read (3

cases) and the time spent in the system (3 cases) were the least frequent variables researched. The three papers related to the latter variable did not measure the time objectively, either, but considered self-reported values. Although, a small number of papers were involved in the investigation, it is striking that the time spent in the system by means of log procedures was examined practically by none of these papers.

Identifying sessions and determining their duration is not trivial, as the exact end of sessions cannot be determined due to the specification of HTTP. (Closing the browser cannot be logged.) Therefore, to identify sessions, a cut-off point is to be specified. If the time interval between two events exceeds this value, it is considered to be a new session.

[14] found that for the last twenty years the most commonly used cut-off point has been 30 minutes introduced by [15], however, 5 minutes, 15 minutes, and 60 minutes have also occurred in the literature. These values are typically obtained from research based on the distribution, the mean, and the standard deviation of event intervals. For example, the cut-off value was defined by [15] under $\mu + 1.5\sigma$, while [16] defined that under $\mu + 3\sigma$. [14] argues that it is important to distinguish between logical and mechanical sessions. In his view, the focus has shifted towards the logical approach in the last decade, however, the logical approach has not replaced the mechanical approach. Rather, the logical approach can be seen as a complement to the mechanical approach.

Research methodology

This paper attempts to identify active sessions and determine their duration based on user log data. For the research, data is provided by the student activity of two higher education courses. It can be assumed that technology usage related to corporate training systems does not differ significantly from that related to higher education.

Courses involved in the research

The research is based on the recorded student activity of two courses (Software Technology II. and Database Systems) at Corvinus University of Budapest of the academic year 2019/2020/2. Participants in both courses consented to using their data for research purposes. The courses involved in the research differ significantly in the sense that the steps of the course "Software Technology II" are built on each other, as a consequence of which learning should not be interrupted due to the loss of time resulting from the change of context and each assignment should be solved without interruption from the beginning to the end. Whereas the (sub) assignments of the course "Database Systems" are not built on each other, as a result, work can be interrupted after any of the assignments. In that case the relationship between the assignments can only be identified in the monotonous increase in difficulty.

It is important to note that the focus and purpose of this paper is to examine the applicability of the method used for analysis, the courses involved in the research provide merely test data suitable for analysis. Based on the examined sample, general conclusions about students studying in higher education cannot be drawn.

Data collection

Data collection was implemented by means of an external service (Smart Assignments) integrated into the Moodle learning management system developed under this research. The students could access learning materials, assignments and educational videos. Using Smart Assignments, live web-socket connection is maintained between the server and the clients, as a consequence of which the end of the sessions can also be logged in addition to their beginning. The logged events are as follows:

- Service starts or restarts ('server start')
- User has opened/closed assignments ('user joined', 'user left')
- Status of user browsing focus has changed (come into focus, focus lost) ('focus changed')
- User started or stopped playing the recorded video at a given time ('video event')

The focus change event referred at point 3 has binary state, depending on whether the browser tab containing the assignments comes into focus or not. (The focus shift event is triggered by two native Javascript events: window.focus and window.blur). In addition to

the abovementioned, browser-related peripheral events (scrolling, mouse clicks and keystrokes) can also be logged. These events were not recorded by the research due to privacy concerns. For security reasons, a more detailed, higher-quality dataset cannot be retrieved from a browser without installing specialized software monitoring user activity.

Student activity was logged between 20 March 2020 and 22 June 2020. This period included six weeks of term-time, one week of intensive week, and the entire exam period.

Definition of sessions

The number of events related to the assignments of the course "Database Systems" (adatb) was 151,388, for the course "Software Technology II." (szoft2) this number was 322,021. A total of 243 students were involved in the research. Several of these students enrolled in both courses in the semester studied. (Database systems: 199 students; Software Technology II: 198 students) In this period, both courses were held exclusively in the form of online education.

Looking at the descriptive statistics of online sessions generated from exclusively the 'user join' and 'user left' events (Table 1), the distribution of durations is extremely right skewed.

Table 1. Descriptive statistics of online sessions

	Count	Mean	Standard deviation	Minimum	Median	Maximum
Database Systems	4,429	00:35:36	01:19:27	00:00:00	00:04:44	13:29:53
Software Technology II.	25,851	00:38:39	01:38:33	00:00:00	00:03:34	Over 3 days

It is possible that users will reload the website or that clients may be disconnected and reconnected to the assignments for a short time due to a network connection failure or switching between networks. In the case of several users, it was found that they did not close the pages of their browser during the semester, so the previously used assignments were also opened every time they started the browser. Due to similar reasons, many sessions appear in the statistics that do not reflect actual technology use.

Based on the available data, a distinction can be made between active and inactive states if it can be assumed that active system usage generates loggable events at certain intervals. The shift in focus between windows and video-related events are of great importance in this case: learning or active usage is assumed to result in more events than an idle session. The Figure 1 illustrates the relation between actual system usage and log events. The black circle indicates the beginning of a user's online status, while the white circle indicates the end of the online session. The stripped grey rectangles represent actual system usage. The logged events are displayed as lightning symbols on the time axis. The black lightning symbol indicates a join event, the white one an exit event and dark grey ones indicate events generated during actual usage, whereas the light grey one indicates an event generated while the system was not actively used. Focus-related events may be recorded in the log files when a student leaves the webpage open but performs other activities. As a result, the event marked in light grey is to be interpreted as *noise* in terms of the method.

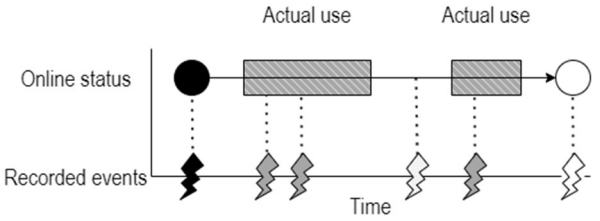


Figure 1. Model for logging actual system usage

In the model, the logged events are known, however, the duration of actual system usage and active sessions are unknown, whose exploration is the purpose of this paper. The probability distribution of the generation of events that can be logged during active usage is not known, either.

It is assumed that:

- active presence of a user generates more events per unit time than the inactive state, and
- at least one event must be generated within a certain time during active usage.

In view of the foregoing, actual usage can be estimated based on the density of the events and the measured time intervals between them. Dense events that are located close to each other in time are likely to be associated with active usage, while events isolated in time are unlikely to reflect the actual solution of assignments. *Training data* is not available, consequently, active sessions can only be identified using an unsupervised machine learning algorithm that arranges the events in groups based on the intervals between the events. The first and last datetimes of the *clusters* mark the beginning and end of an active session.

The question arises whether the join and exit events carry any additional information. If the white (exit) and black (join) symbols shown in Figure 2 are located close to each other in time, the assigned sessions should be combined and treated as one unit. For instance, this may occur when the user reloads the browser.

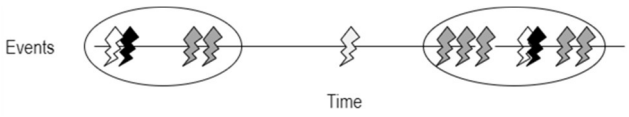


Figure 2. Clustering logged events

A limitation of the approach is that actual system usage can only be inferred indirectly from logged events. If the events assigned to a session are located relatively distant and irregularly from each other in time, the results will also be more inaccurate.

This paper subsequently compares two procedures aimed at identifying active sessions and then evaluate them in terms of arranging the events in groups. The first procedure is a self-developed algorithm that identifies each session by considering the duration of time intervals between events, and the join and exit events. This approach bears the most resemblance to the one described by [3]. The second one is based on a density-based clustering procedure.

Interval-based session identification

The self-developed procedure assumes that session boundaries are basically the events of establishing and terminating network connection. This rule can be overridden by two situations. On the one hand, sessions assigned to exit and join events generated shortly after each other ("bouncing between pages") can be merged, on the other hand, an event generated after a given idle period can be considered the start of a new session, even if the closing exit message is not received. In view of the foregoing, it is necessary to introduce two parameters:

- Merge-threshold t_m [time]: if the time interval between two sessions is less than t_m, they can be considered belonging to the same session, thus they can be merged.
- Inactivity-threshold t_i [time]: if the time interval between two sessions is bigger than t_i, a new session will have to be created.

The comparison with the merge-threshold takes place before the evaluation with the inactivity-threshold. It is important to note that the value of the merge-threshold can be at most the value of the inactivity-threshold. Conversely, sessions that would otherwise count as separate sessions based on the inactivity-threshold may also be merged.

If the two parameters are equal, the join and exit events do not have a quintessential role. The question arises as to whether the cause of events, as such, should be recorded and distinguished from each other, or whether it is sufficient to know the datetimes of events. The answer is difficult to generalize since the added information content of join and exit events depends on student activity. If the time between exit and join events would be more than the merge-threshold but less than the inactivity-threshold, the information content of join and exit events is of greater importance. If the sessions were also merged by the algorithm in this case, the session duration would be presumably overestimated, since the logged-out user is probably not using the technology. It is important to highlight that the events are processed in chronological order by the algorithm based on the order in the log files. The algorithm is deterministic, so the same input parameters and set of events always result in the same set.

Density-based session identification

Quantifying active presence can also be formulated as a clustering problem in which the task is to arrange events that are close to each other in groups. A session can thus be considered as a cluster of events based on the Euclidean distance of datetimes.

Density clustering methods can be applied to arrange points situated close to each other in groups. In addition to the time intervals between observations, the number of events occurring within a given unit of time is also taken into account. One of the major representatives of the method family is *DBSCAN* algorithm, whose popularity lies in its robustness to noise in the data. [17]

Its operation is basically determined by two parameters. The first one is the epsilon (ε) value which is the radius of the neighbourhoods, the centre of which is always an observation. The second parameter is the required minimum number of samples (S_{min}). If the number of observations belonging to a neighbourhood is higher than the value of the second parameter, the observation in the centre qualifies as a *core point*. If there is an observation in a neighbourhood of a core point that contains fewer elements than the S_{min} specified, this observation will be considered to be a *border point*. Core points and their border points are assigned to a cluster. The elements outside this set are identified as *noisy points*.

It should be noted that DBSCAN is not a deterministic algorithm. It does not always provide the same result for the same parameters. This is because if a border point is reachable from two directions as well, the classification depends on the order in which the observations are processed. [18] This phenomenon, however, has a negligible effect on the examined problem.

The two algorithms presented show similarities in that sessions are created based on time intervals between the events. A fundamental difference is that interval-based algorithm considers join and exit events, while DBSCAN may be able to identify the most intense periods by means of the requirement of a minimum number of events within a given time interval. One of the input parameters of both algorithms is the acceptable time interval between the events, as a consequence of which the results obtained by the two methods can be compared. Using density-based estimation, if $S_{min} = 1$ the difference from sessions obtained with the interval-based algorithm may be due solely to the fact that DBSCAN does not distinguish join and exit events from other events. Consequently, sessions obtained with the density-based approach will be longer, as the time elapsed between an exit event and a join event within a "short" period of time is considered to be active time.

Determining parameters

Parameter values for the presented procedures should be selected with great care, as the actual sessions can be easily underestimated or overestimated. *Underestimation* may occur due to strict (low) parameter values. In this case, periods of use that do not or rarely generate events are not tolerated by the algorithms. The *overestimation* of the actual usage can occur due to the parameters allowing longer time intervals when the idle periods can be classified by the procedure as active.

Resolving the problem requires predetermining the input or output expectations based on expert estimation. An input assumption may be that an event is generated by the user in any case within a given period. This is what [3] have also done. An output assumption may be the expected usage time in the system, the number of expected clusters or the ratio of events identified as noise.

In terms of parameter selection, two user scenarios should be distinguished. The first one is when system usage by users is primarily limited to content consumption (e.g.: reading a text). The second one is when the user also uses other software during the learning process. In the former case, the parameters for the time intervals between the events are easier to determine based on the amount of text displayed on screen and the estimated average reading speed (The scrolling event can be particularly informative in this case). In the latter case, the situation is more complicated as the time spent in the external system must also be estimated.

The selection of specific parameter values can be done by means of an iterative process, during which the input parameter values, and the obtained sessions must be evaluated based on their statistical characteristics and their sensitivity related to changes in parameters. This paper examines the sensitivity of the following statistical aspects:

- Distribution of session durations
- Number of sessions
- Ratio of events identifies as noise
- Total time of sessions in the examined period

The listed aspects were quantified per course and per student, but in order to prevent bias resulting from the researcher's selection, the paper presents the results of the sensitivity analysis in an aggregated (averaged) way. If the events describing student activity are located close to each other in time, they can be grouped easily, the listed statistical characteristics are expected to converge as a function of the parameters, i.e. the output is no longer sensitive to input over certain parameter values since the actually related events have already been merged into sessions. Ideally, if there is no fragmentation, no noise and user attention is focused only on system usage, active sessions can be easily identified based on the datetimes of the first and last events. According to the tutors' experience related to the courses suggests that the value t_m should be utmost a few minutes, however, the parameter t_i should be set higher than that. It is also possible that for 5 to 10 minutes users' activity will not trigger any events because they are working in another application (e.g.: Visual Studio), although at least one window change is likely to occur for the assignments within half an hour. For setting the parameters for DBSCAN, the value of ε should also be set to the value of around 5 to 10 minutes. The value of the minimum number of samples is more difficult to determine by expert estimation. The assignments were tailored to 90-minute-long seminars and solving them usually takes approximately 60 to 90 minutes. The range of input parameter values for the sensitivity analysis is shown in Table 2. It can be assumed that at least one event can be observed within 25 minutes during active system usage, consequently, greater durations are not examined. This paper presents the results of the relevant values of the examined parameters in detail

Table 2. Input parameter range of sensitivity analysis

	Name of the parameter	Start value	End value	Step
Interval-based	Merge-threshold (t_m)	30 sec	3 min	30 sec
	Inactivity-threshold (t_i)	5 min	25 min	5 min
Density-based	Epsilon (ε)	5 min	25 min	5 min
	Minimal number of samples (S_{min})	1 event	29 events	7 events

Results

The results were evaluated in three stages, on the one hand, by examining the distribution of time intervals between events, on the other hand, by visually evaluating the clusters created, and on the third hand, by performing a sensitivity analysis of the output of clustering procedures.

Distribution of time intervals between events

Concerning the distribution of event time intervals, it is compelling to observe that both courses are characterized by approximately similar distributions. 80 percent of time intervals are shorter than 10 minutes, however, the rest is widely dispersed (from 10 to 60 minutes). The frequency distribution of time distances is characterized by significant left-hand asymmetry based on Figure 3, and its shape is like the exponential function.

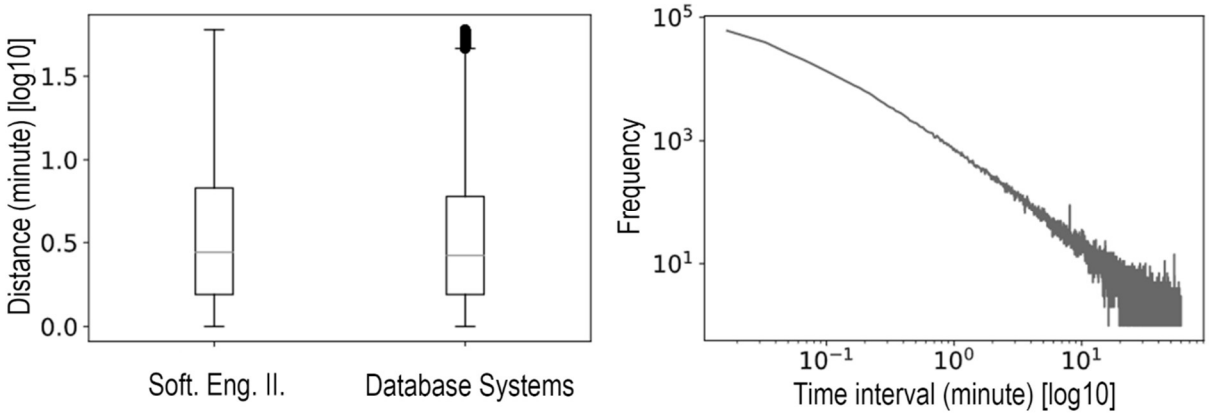


Figure 3. Distribution of event distances between 1-60 minutes

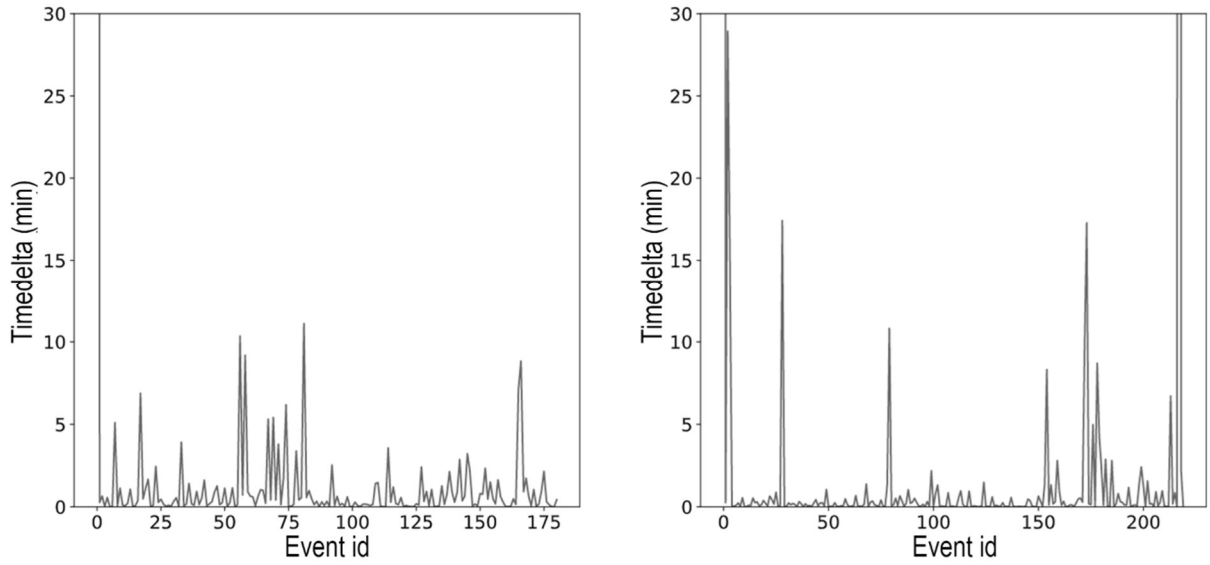


Figure 4. Two users' daily activity during the course "Software technology II."

In the case of students who attended both courses, there is no regression between the event frequency of the examined courses. The number of events related to the course "Software Technology II" is distributed in the range of 0 to 3000 per student, while in the case of the course "Database systems" between the values from 0 to 1750. Significant differences can be identified among students in terms of time intervals between the events. Based on the examples shown in Figure 4, the activity can be divided into active, intense periods with shorter or longer pauses. Time delta on the y-axis represents the time intervals between the datetimes of the current and the preceding events. The result also coincides with [19] findings that user activity can be divided into intensive bursts.

Visual representation of clusters

Representation of student events as shown in Figure 5 can also help determine the parameters. The figure shows the 9-hour activity of 16 users. Each row contains events assigned to one user and along the x-axis the events are displayed depending on their date. The figure shows the output of the interval-based algorithm (t_i = 20; t_m = 1). The events assigned to one cluster are connected from below by a straight line, as a result, the fragmentation within the day becomes visible and the time schedule of the students can be displayed. Events that are not connected from below were identified as noise.

The activity of the first user between 10:43 and 14:17 is enlightening: this period is assigned to one session with the present configuration; however, several major pauses can be observed in the user's activity. The question is whether these periods should be classified as active or inactive periods, or whether these clusters should be further broken down.

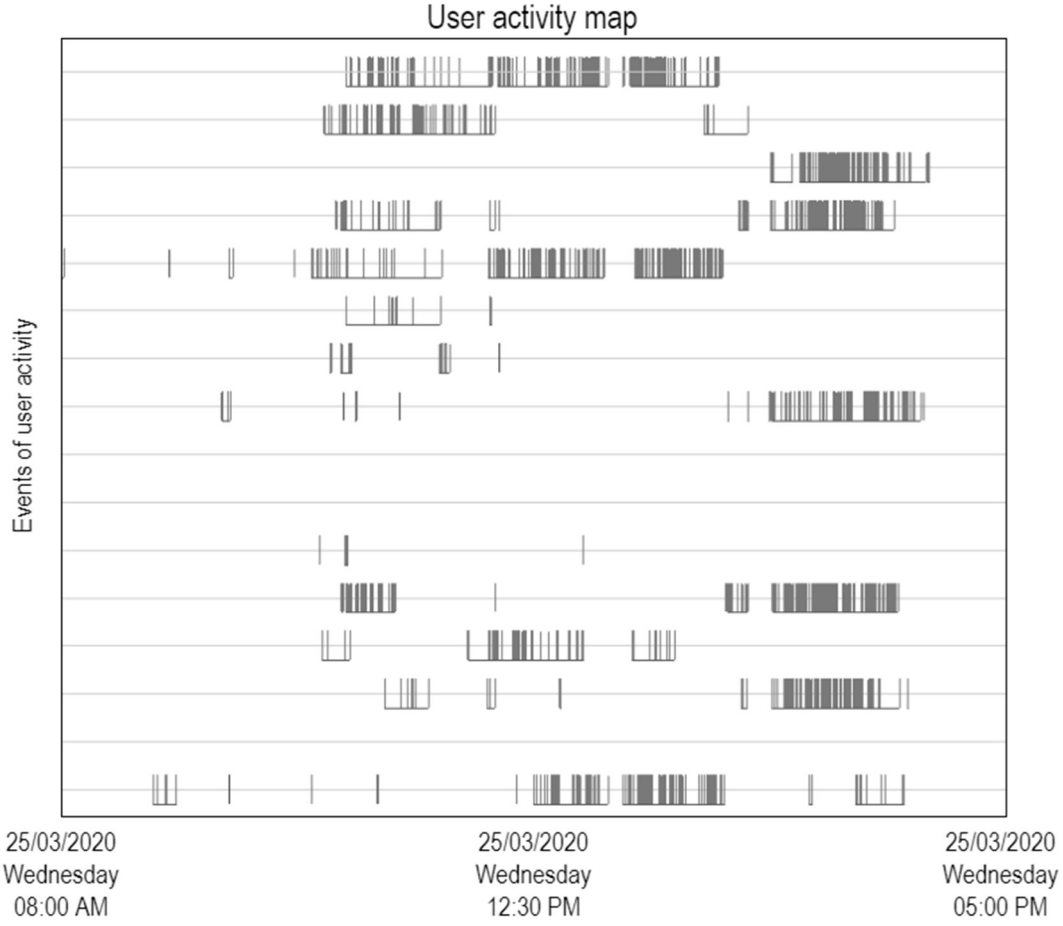


Figure 5. Student activity map (excerpt) 25 March 2020, 08:00 - 17:00 Software Technology II. Interval-based identification (t_i = 20 min; t_m = 1 min)

Output of clustering procedures

Based on the output of the sensitivity analysis, it can be concluded that the duration of the sessions is sensitive to the inactivity-threshold from the two parameters in the examined interval, while it is not sensitive to the merge-threshold. Applying DBSCAN the output is sensitive to both the epsilon and the minimum number of samples.

The session distributions demonstrate that principally the more fragmented (and probably longer) sessions are more sensitive to the increase in the inactivity-threshold. The two courses are characterized by nearly identical session duration distributions. Most of the sessions obtained are significantly shorter than it was expected. Even with a 25-minute-long inactivity-threshold, the interquartile range is between 10 and 35 minutes, which refers to the fragmentation of the students' system usage. The distribution of session lengths is highly right-skewed.

As expected, the output of the density-based estimation (if S_{min} = 1) resulted in a few minutes longer (8 percent on average) sessions than the interval-based estimation. The difference stems from the fact that DBSCAN also considered logged out students to be active if they had returned to the system within the time of epsilon and the condition for the minimum

number of samples had been met. No tendency can be identified for any of the algorithms in the change in the distribution of session lengths that would contribute to the selection of the input parameters. The number of sessions (clusters) created is sensitive to the inactivity-threshold. On average, almost two and a half times as many sessions were created from the course "Software technology" as from the course "Database systems". It is important to note that the number of students and assignments in the two courses was nearly the same. Based on Figure 6, no clear elbow can be identified for any of the courses (S_{min} = 1). With an increasing number of minimum samples, the number of clusters is even less sensitive to epsilon.

The average proportion of events identified as noise is sensitive to both the inactivity-threshold and the epsilon as well as to the minimum number of samples. Noise is an event that either was not assigned to any clusters or forms an independent cluster containing only one element. In the case of interval-based estimation, no clear elbow can be identified. The results of the density-based procedure show that increasing the minimum number of samples significantly increases the ratio of noise (mainly due to shorter sessions). No clear elbow can be identified in this case either, the ratio of noise does not converge for any of the courses.

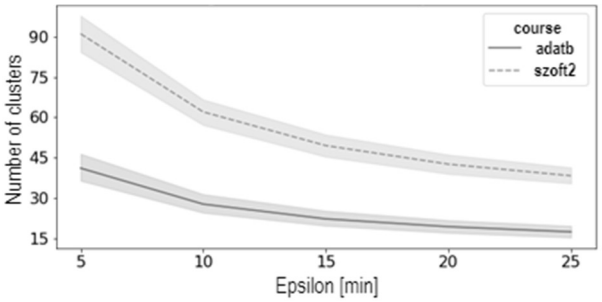


Figure 6. The average number of sessions based on the DBSCAN algorithm (S_{min} = 1)

The total time spent by the students during the examined period is extremely sensitive to parameter selection. Both the interval-based algorithm and the total time spent by the students, obtained through DBSCAN are about two and a half times higher at the highest parameter values examined (t_i = 25 min; ε = 25 min) than at the lowest parameter values examined (t_i = 5 min; ε = 5 min). No clear elbow can be identified.

.

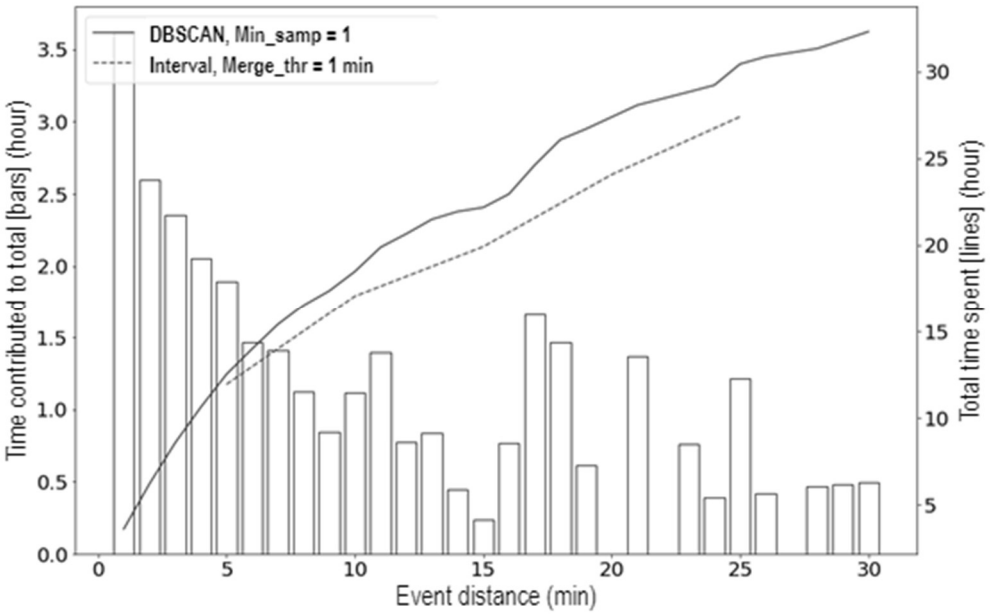


Figure 7. The contribution of a user's event intervals to the total time spent (DBSCAN, S_{min} = 1)

The Figure 7 shows the extent to which each time interval between the events contributes to the measured total time spent by a student. According to the diagram, time intervals between 0–1-minute increase the total time spent by approximately 3.6 hours, the intervals between 1 and 2 minutes by approximately 2.55 hours, etc. The continuous curve represents the cumulative time spent in each of the time intervals. The dashed curve indicates the time spent, obtained with the interval-based algorithm. It can be concluded that the measurement is made more accurate if only the periods with live web-socket connection are taken into account. Although Figure 7 represents only one student's activity, similar trend can be observed for the majority of the students Results / Recommendations.

The results presented clearly demonstrate that the more fragmented the user activity is, the more inaccurate the estimated value of the session time will be. Identifying active sessions is not trivial since activity is a fuzzy phenomenon. A distinction has been made between the concepts of online status and active presence. It is not possible to distinguish clearly which part of the online status can be considered as active usage. The periods, during which the examined students generate a loggable event within a certain time, are considered active by this paper.

Compared to previous studies in the field, in addition to HTTP events, client-side events were also considered and due to the web-socket connection, the termination of live connection is also notable on the server side. Two clustering approaches are presented and applied to identify sessions. Although system usage is considered binary, the identification of the exact boundaries based on the collected data is limited. In the future, events related to scrolling should also be recorded. Additional measurement methods can also be incorporated, but these (mouse movement, tracking keystrokes, camera image) can be considered invasive in terms of privacy and may not be feasible to implement at business environments.

Ideally, where the subject is engaged in only one task, the session can be closed after the user closes the website. Experience has shown that this was not the case for the students of the courses involved in the research. Presumably, other parallel activities may fragmentate the activity related to the system examined. When working from home, home environment disruptions should also be considered. The level of fragmentation can be learned by graphing the frequencies of time intervals, by means of which the parameters for algorithms can be set. Based on the statistical characteristics of the generated sessions, the parameters can be estimated if the statistical characteristics converge as a function of the change in parameters. In the case of the examined courses, this did not happen, which suggests significant noise presence and fragmentation. Event and session fragmentation is a challenge to setting parameters for clustering algorithms.

Concerning parameter selection, it should be noted that longer sessions are more sensitive to the parameter selection. Presumably, user attention is more fragmented over longer sessions. Input or output expectations must be set before the analysis. In the case of the former, it can be specified how much time interval between two events within a session is acceptable. Expected time spent in the system or average session duration can be mentioned, for example, as an output expectation. If no input or output expectation is defined, events cannot be clustered.

Events of students' activity were clustered by two algorithms. Clearly, neither algorithm proved to be dominant over the other in terms of the results obtained. The self-developed interval-based algorithm took into account the join and exit events, which is particularly necessary since the time interval between exit and join events greater than t_m can be considered inactive, however, DBSCAN considers this to be active if the condition for epsilon and the minimum number of samples is met. DBSCAN takes into account the density of events which is another aspect in the analysis of the intensity of student activity.

The parameter of the expected number of samples filtered out principally the otherwise shorter, few minutes long sessions. This, however, did not have a significant impact on the quantified time spent. The parameter of the expected number of samples can be useful in research tasks where these short sections need to be filtered out. The join and exit events should also be considered since this provides – according to the results – on average about an 8 percent lower estimate of active system usage. In the future, a new procedure should be developed to combine the two procedures. Other datasets should also

be involved to research. There are several concerns about generally defined clustering parameters. The frequency of event generation varies within given assignments: there are easier and more difficult assignments, and some parts of the text are easier to process while other parts cause more difficulties in this respect. The intensity of the activity may also differ due to individual habits, and the distribution of the frequency of the time intervals between events may also change over time due to the effect of leisure time, parallel activities and disturbing factors.

The time spent by students cannot be estimated accurately due to fragmentation, both algorithms were very sensitive to the parameters. The method may be appropriate in research tasks where it is sufficient to quantify the time spent on an ordinal or nominal scale. The condition for this is that the frequency distribution of events should not differ significantly per student. This research needs to be conducted in the future.

References

[1] S. R. Barkin and G. W. Dickson, "An investigation of information system utilization," *Inf. Manag.*, vol. 1, no. 1, pp. 35–45, Jan. 1977, doi: 10.1016/0378-7206(77)90007-6.

[2] A. Burton-Jones and D. W. Straub, "Reconceptualizing system usage: An approach and empirical test," *Inf. Syst. Res.*, vol. 17, no. 3, pp. 228–246, 2006, doi: 10.1287/isre.1060.0096.

[3] V. Venkatesh, S. A. Brown, L. M. Maruping, and H. Bala, "Predicting different conceptualizations of system USE: The competing roles of behavioral intention, facilitating conditions, and behavioral expectation," *MIS Q. Manag. Inf. Syst.*, vol. 32, no. 3, pp. 483–502, 2008, doi: 10.2307/25148853.

[4] D. Straub, M. Limayem, and E. Karahanna-Evaristo, "Measuring System Usage: Implications for IS Theory Testing," *Manage. Sci.*, vol. 41, no. 8, pp. 1328–1342, Aug. 1995, doi: 10.1287/mnsc.41.8.1328.

[5] M. Turner, B. Kitchenham, P. Brereton, S. Charters, and D. Budgen, "Does the technology acceptance model predict actual use? A systematic literature review," *Information and Software Technology*, vol. 52, no. 5. Elsevier, pp. 463–479, May 01, 2010, doi: 10.1016/j.infsof.2009.11.005.

[6] C. Ching-Ter, J. Hajiyev, and C. R. Su, "Examining the students' behavioral intention to use e-learning in Azerbaijan? The General Extended Technology Acceptance Model for E-learning approach," *Comput. Educ.*, vol. 111, pp. 128–143, Aug. 2017, doi: 10.1016/j.compedu.2017.04.010.

[7] W. M. Al-Rahmi *et al.*, "Integrating Technology Acceptance Model with Innovation Diffusion Theory: An Empirical Investigation on Students' Intention to Use E-Learning Systems," *IEEE Access*, vol. 7, pp. 26797–26809, 2019, doi: 10.1109/ACCESS.2019.2899368.

[8] A. Revythi and N. Tselios, "Extension of technology acceptance model by using system usability scale to assess behavioral intention to use e-learning," *Educ. Inf. Technol.*, vol. 24, no. 4, pp. 2341–2355, Jul. 2019, doi: 10.1007/s10639-019-09869-4.

[9] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Q. Manag. Inf. Syst.*, vol. 13, no. 3, pp. 319–339, 1989, doi: 10.2307/249008.

[10] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, "User acceptance of information technology: Toward a unified view," *MIS Q. Manag. Inf. Syst.*, vol. 27, no. 3, pp. 425–478, 2003, doi: 10.2307/30036540.

[11] S. Dasgupta, M. Granger, and N. McGarry, "User acceptance of e-collaboration technology: An extension of the technology acceptance model," *Gr. Decis. Negot.*, vol. 11, no. 2, pp. 87–100, 2002, doi: 10.1023/A:1015221710638.

[12] F. D. Davis and V. Venkatesh, "Toward preprototype user acceptance testing of new information systems: Implications for software project management," *IEEE Trans. Eng. Manag.*, vol. 51, no. 1, pp. 31–46, Feb. 2004, doi: 10.1109/TEM.2003.822468.

[13] S. Hrastinski, "What is online learner participation? A literature review," *Comput. Educ.*, vol. 51, no. 4, pp. 1755–1765, Dec. 2008, doi: 10.1016/j.compedu.2008.05.005.

[14] F. Dietz, "The Curious Case of Session Identification," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Sep. 2020, vol. 12260 LNCS, pp. 69–74, doi: 10.1007/978-3-030-58219-7_6.

[15] L. D. Catledge and J. E. Pitkow, "Characterizing browsing strategies in the World-Wide web," *Comput. Networks ISDN Syst.*, vol. 27, no. 6, pp. 1065–1073, Apr. 1995, doi: 10.1016/0169-7552(95)00043-7.

[16] L. Zhuang, Z. Kou, and C. Zhang, “Session identification based on time interval in web log mining,” in *IFIP Advances in Information and Communication Technology*, 2005, vol. 163, pp. 389–396, doi: 10.1007/0-387-23152-8_50.
[17] M. Ester, M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” pp. 226--231, 1996, Accessed: Jan. 29, 2021. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.121.9220.
[18] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, “DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN,” *ACM Trans. Database Syst.*, vol. 42, no. 3, pp. 1–21, Jul. 2017, doi: 10.1145/3068335.
[19] A. L. Barabási, “The origin of bursts and heavy tails in human dynamics,” *Nature*, vol. 435, no. 7039, pp. 207–211, May 2005, doi: 10.1038/nature03459.

Bálint Mátyus is a PhD student at Corvinus University of Budapest. He graduated from the same institute with a master's degree in business information science in 2015. His passion is the education. He is researching on peer-learning and student collaboration networks in higher education. Previously he was developing innovative methods for teachers to create digital handwritten materials and was researching acceptance of mobile government technology by the elderly. He has been actively involved in teaching software development and database management at Corvinus University since 2011. In the last decade he was employed as a C# developer at Siemens, T-Systems Hungary and Evosoft Hungary

Ticketing Data Warehouse System Development Challenges and Experiences

GÉZA MOLNÁR
Corvinus University of Budapest, PhD student
eMail: geza.molnar@stud.uni-corvinus.hu

ABSTRACT

Most of the small or medium-sized companies store their data in multiple systems (databases), usually in different formats. The question arises many times, how can one create either historical or ad-hoc reports from these databases as easy as possible, using available tools. The most common requirements are that these reports can be customised by users, viewed on a computer or mobile device, and shared with other users without any specialised computer knowledge (like self-service BI). In most cases, the reporting process is as follows: the IT-team collects the necessary data and prepares the report, then shares the results with the management on the company's local intranet, mostly in the form of an Excel file. Users, on the other hand, prefer to create and customise reports themselves quickly and efficiently, without any help. This study discusses a case from an IT company struggling with similar problems. The paper presents the improving process of the existing reporting system through a data warehouse development and provides an overview of the related challenges and their handling

.

Introduction

The problem discussed in this paper is from a software development company that provides IT services to clients such as custom application development, maintenance, troubleshooting, and user support. Communication with customers is usually online. To manage clients' requests, the company uses a service-desk application, also known as a ticketing system. In practice, the customer's needs and technical capabilities can be quite different. Some clients - especially the more important ones - insist that their data related to IT services, be stored and managed in a physically separate system. They usually like to customise them according to their needs, e.g., they store additional information related to error messages. So, using one ticketing system is not enough, and these ticketing systems may differ more or less from each other. These systems store data in different databases that are similar in structure but not the same.

A ticketing system has four core functionalities. The first one is to create a new ticket. Users can give its type (incident, change, or request), attaches a detailed description, and record other relevant information such as priority or assignee. The second function is to log the time spent by analysts working on tickets. The third one is to follow and store the changes in the status of tickets. When a ticket is created, its status is open. After that – depending on the current state – the ticket's status can be various, for example, work in progress or waiting. At the end of the ticket management process, the status is usually closed. The last primary function of the ticketing system is to handle the Service Level Agreements (SLAs). A Service Level Agreement is a contract between the service provider and its client. It usually contains the terms and requirements related to the service provided. The service fulfilment can be measured by various metrics such as percentage of availability or average time of troubleshooting [1]. The ticketing system can monitor compliance with the SLA metrics. From this point of view, the SLA status can be OK, warned, or violated. A modern ticketing system has many additional features, such as workflow automation, knowledgebase, or reporting tools [2].

The management needs up-to-date information regarding ticketing systems, for instance, what does the number of unclosed tickets or how many SLA violation occur in a week. To satisfy this goal, we had been facing many challenges:

- The source systems are often different. In practice, this is manifested in the fact that they can contain custom data that are not included in other systems. Moreover, these systems are separated, and there is not always a direct connection between them.
- The number of source systems can change at any time.
- The data quality is usually unsatisfactory. Although every ticketing system tends to avoid inconsistent data entry, data quality can be weakened by several factors. One weak point is data duplication. For example, every system stores data about analysts who deal with the tickets, so a specific one can be stored in more databases. Additionally, the filling of fields is inconsistent. It might happen that in one of them, the birth date is filled out; in another, this is empty. Another weak point is the problem of null values, which is primarily due to differences between systems.
- Preparing reports is not possible without IT support since managing multiple data sources and integrate data requires specialised knowledge. Besides, in case of any modification in reports or data, the IT environment update is needed again.

The previously detailed issues lead us to implement a data warehouse/business intelligence (BI) system to replace the existing reporting system by eliminating its weaknesses. In this article, the process of designing and creating a data warehouse is described, including the procedures for loading and transform data from data sources and developing a BI data model that allows users to create customisable reports. I summarise the challenges of the development and lessons learnt as well.

Literature review

The first management information systems appeared in the early 80s and used pre-defined reports concerning a specific business area (Figure 1). These analytical reports run on the databases of operative systems, so in many cases, they caused a heavy load on it. Later, to eliminate this, data needed for analysis was copied to another system, and the reports were running there. As a result, databases for operational and analytical purposes have become increasingly separated. Data warehouses (DWH) have evolved from these types of analytical databases.

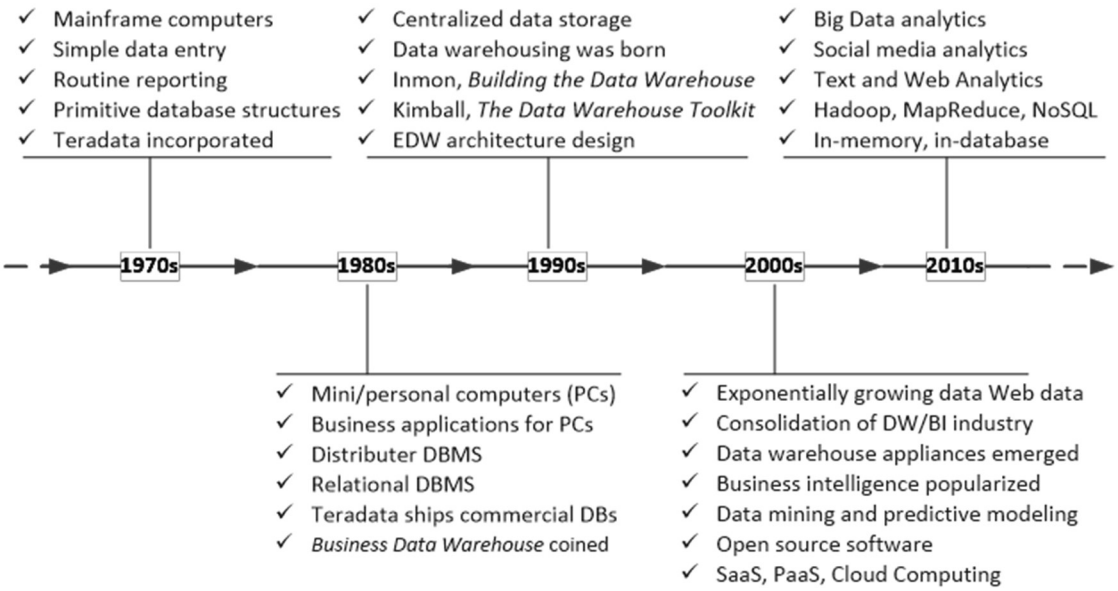


Figure 7: Data warehousing – History [3]

The two best-known classical data warehouse approaches were developed by Kimball and Inmon[3],[4],[5]. Nowadays, there are many other ways to design and create data warehouses. These new DWs mostly contain elements of classic models. Compared to the Kimball's and Inmon's data warehouse models, data warehouses nowadays generally faces new challenges such as Big Data [6], real-time responses [7], and frequent changes in the data model [8].

Kimball's and Inmon's approaches

According to Kimball, "A data warehouse is a copy of transaction data specifically structured for query and analysis [9]". In Kimball's approach, there are four distinct components in the DWH/BI systems (Figure 2): operational source systems, ETL (Extract, Transform, Load) system, data presentation area, and business intelligence applications[10]

The operational source systems are the systems that capture the business's transactions. In many cases, they are special-purpose applications and maintain little historical data.

- The ETL system is between the source systems and presentation area. Extracting means reading the source data and copying them into the ETL system for further manipulation. This latter is usually manifested in a transformation, such as cleansing, deduplicating, or combining the data. The final task of the ETL system is to load the data into the presentation area.
- The presentation area supports business intelligence. To be more specific; it stores and organises data in a form that allows the direct querying by business intelligence applications. The normalised model, which is common in relational databases, is not suitable for this purpose due to its poor query performance. Instead, Kimball suggests the so-called dimensional model which

can be either a star schema or an OLAP (Online Analytical Processing) cube. Both use the same design logic, but they differ in the method of implementation. The star schema is hosted in a relational database; in contrast, the OLAP cube is stored in a multidimensional database.

– The last component is the BI application. Here, this name refers to applications that query the data in the presentation area for business users to make better decisions. A BI application can be either an ad-hoc querying tool, a reporting tool, an analytic app or a sophisticated data mining and modelling application.

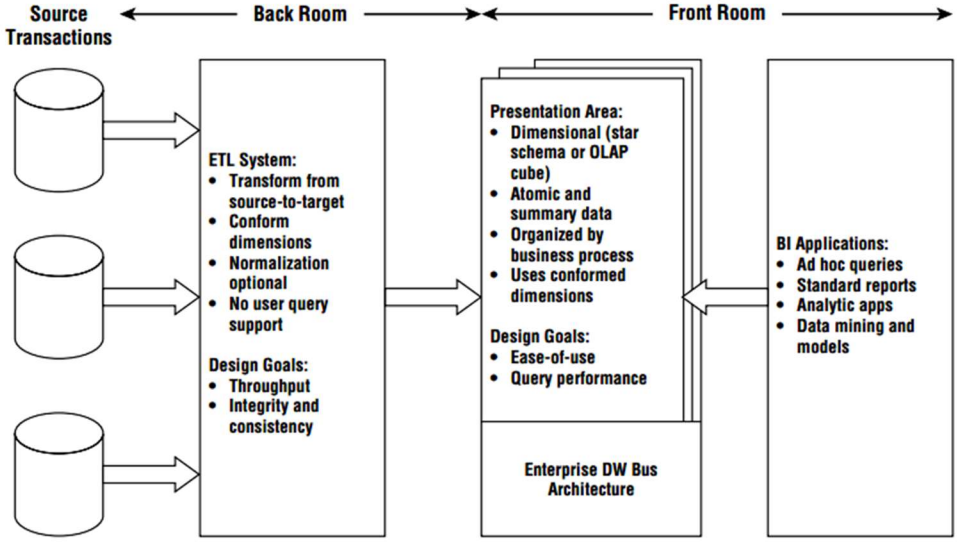


Figure 8: Kimball's DWH/BI architecture [10]

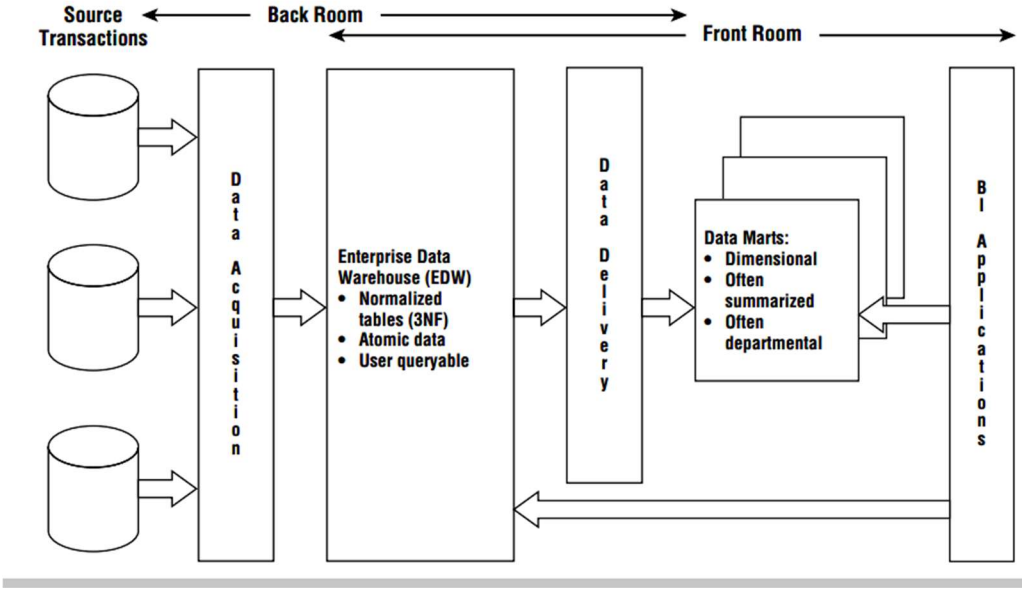


Figure 9: Inmon's architecture [11]

Inmon referred to as the "father of data warehousing". He defined the concept of the data warehouse as follows: "A data warehouse is a subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management's decisions"[11]. At the heart of this architecture is the Enterprise Data Warehouse (EDW), which is a normalised, atomic repository (Figure 3). The data acquisition means the first ETL system that extracts data from operational source systems and processes them to EDW. The second ETL system (data delivery) populates reporting and analytical environment to support business users. The presentation area is usually department-centric and dimensionally structured with summarised data [11]. In terms of how to architect a data warehouse, even today, one of the two classical models is the starting point. When choosing one of them, the following aspect should be considered [12]:

Kimball's model is preferable when
- the primary audience is the IT
- rather tactical decisions are needed
- source systems are relatively stable
- the result is required within a short time with little initial cost and a small team
- data come from individual business areas
- the scope of changes is limited

Inmon's model is preferable when
- the primary audience are end-users
- strategic decisions are in the majority
- source systems often change
- longer start-up time and higher start-up costs are allowed with a larger team of specialists
- enterprise-wide data integration is needed
- the scope of changes may increase.

Data warehouses nowadays

Data warehouses nowadays face several challenges, like managing big data, real-time response, data quality issues, and fast-changing IT environment [13] [14]. The term "Big Data" identifies a specific kind of data set. They usually contain a large amount of data that can come from a variety of data sources such as social media platforms, videos, or IoT devices. In many cases, data is not structured, and their volume can overgrow. Managing Big Data in data warehouses and BI systems pose many difficulties, mainly in two areas. The advanced ETL processes are required to get structured information from low-level, raw data. Another issue is that even state-of-the-art solutions are not capable of dealing with computing OLAP cubes. Several additional problems can be identified:

- the size of the fact table can huge
- the number of dimensions and measures can be multiple as the usual, thus increasing the complexity
- due to the enormous size, traditional computing methodologies do not work
- data quality can be reduced due to the nature of Big Data sources
- the end-user performance quickly becomes poor
- how to design and optimise an analytical process
- integration with classical data-intensive platforms
- suitable development tools for supporting the development of OLAP cubes over Big Data
- mapping an OLAP cube over Big Data in memory.

For many data warehouses, data are updated once or twice a day. It is easy to see that in case of a banking system or a flight ticket system, this is not enough, they need (almost) real-time data. Due to continuous updates, such real-time data warehouses require a non-traditional architecture, whose main components are as follows [17]:

- Transactional replication that pushes each committed transaction towards an intermediate database
- Intermediate database, which has the same structure as the target data warehouse
- Change data capture, which can detect changes in the source systems
- Schema matching to identify semantic correspondences between intermediate database and data warehouse
- Data transformation and cleaning, which are applied to the changed data
- Data load, that load changed data to the data warehouse

Due to continually changing needs, modifications are often required in the data warehouse system. One way to handle this problem is by using the data vault methodology [8]. It allows us to dynamically expand the data model without having a complicated task of

modifying other affected system elements. According to data vault methodology, the data model must be managed at the meta-model level. Its main feature is to separate the business key (hub), the business key transaction (link), and the business key history (also called sat).

Self-service BI development – case study

Data warehouse development as a software development task involves its typical steps such as specification, design, implementation, testing, deployment and maintenance. The easiest way to do these steps sequentially. This development methodology is called the waterfall model [15]. It is usually chosen when the requirements are precise and likely will not change during the development process. Besides the waterfall model, there are many other methodologies. Its common characteristic that the development steps or some of them are performed in parallel or iteratively. Nowadays, agile methods are especially popular, which focus primarily on customer satisfaction [15]. However, these are not beneficial for a specific type of projects, especially larger ones [16]. In the development method selection, the specialities of data warehouse projects had to be taken into consideration such as the high number of use cases, the problem of ensuring data quality, or the difficulties in choosing an effective testing methodology [17].

The first version of the data warehouse was created using a waterfall model. It was made feasible by the fact that being an IT company; there was no problem with the exact specification of the task or the IT-skills of the customers. Accordingly, the first version of the data warehouse has created in four steps described by Figure 4. Perhaps a little surprising is that working with reports precedes ETL processes.

One reason for this was that management wanted to see tangible results as soon as possible. Therefore, test data in appropriate format was also attached to the report specifications. These made it possible to create a raw version of the reports before creating the ETL-processes. Since the company uses Microsoft software, it seemed obvious to use them, as shown in the Figure 4.

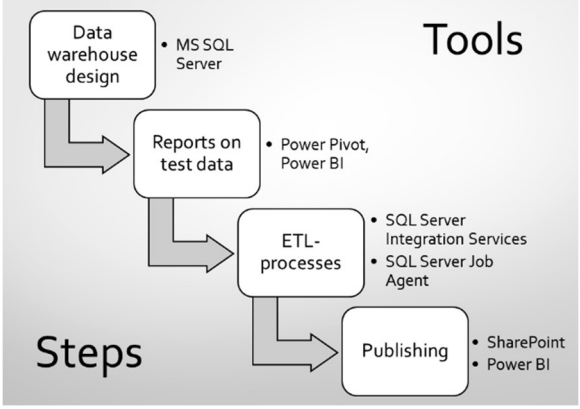


Figure 10: Tools and steps in DW development

Specification

The specification is based on the needs of managers. They focused mainly on the output, i.e. reports. The primary goal is to create a system that generates reports for management from the company's ticketing systems that provide an overview of the status of the open tickets and about the related activities. Data was collected from multiple sources, and it must be updated daily. The dimensional data must be stored historically so that in case of changes, the reports always show the correct values. The following data must be stored in the data warehouse:

- Ticket details, such as ticket type, identifier, creator, source, project code, a brief description of the case, priority, SLA information
- Ticket activities, especially the following: the name of the person performing the activity, the beginning of the activity and completion date, type of activity, short description, overtime
- Users and customers details, mainly user name, email, ID, default group, type. The latter may be employee, vendor, customer, and analyst
- Project data (project name, code, active)

Reports must be able to display the following data (about 30 report definitions):

- Number of tickets assigned to a specific group
- Number of unallocated tickets
- Logged time by analysts
- Number of SLA violations
- Number of open and closed tickets

Each report must be filterable by specific dimensions such as time, user and group. Additionally, some Excel-tables from the previous system were available, which contained the data to be stored and the reports to be created.

Design

The first challenge was selecting data warehouse design principles. There are four main approaches, namely top-down, bottom-up, hybrid and federated [18]. Evaluating the possible solutions, Kimball's bottom-up method seemed to be the most suitable one. The main reason is that Kimball's approach is more in line with the specification. During the design process, the focus was on two issues: what are the business processes, and what are their characteristic [10]?

The answer to the first question helps us to identify the fact tables. Four crucial business processes can be distinguished, such as creating tickets, log activities, store SLA data and follow ticket status changes. Consequently, we will have four fact tables: Ticket, Activity, SLA and ChangeStatus. To identify dimension tables, the following should be considered:

- From what source does the ticket come? (email, phone, user form)
- What is the category of the ticket? (hardware, network, application)
- What is the ticket priority? (low, high, medium)
- What is the ticket status? (open, resolved)
- Which user, which group, which project, which organisation, and which tenant can be assigned to the ticket?
- When was the ticket created?
- What is the type of activity? (attach doc, close request)
- Which user and which tenant can be assigned to the activity?
- When did the activity begin and end?
- Which user, and which group can be assigned to the SLA?

Accordingly, we will have the following dimension tables: ActivityType, Category, CRSource, Group, Organization, Priority, Project, Tenant, User, SnapshotDate, SnapshotTime. The relationships between dimensions and facts are illustrated in Table 1.

Table 3: Process-dimension matrix

	FACT TABLES			
DIMENSIONS	Ticket	Activity	SLA	Status
ActivityType		x		
Category	x			
CRSource	x			
Group	x		x	
Organisation	x			
Priority	x			
Project	x			
Tenant	x	x		
User	x	x	x	x
SnapshotDate	x	x	x	x
SnapshotTime	x	x	x	x

Based on this table, we designed the logical and physical data models of the data warehouse. One more design question left: how to design and implement the related data marts and reports? The simplest solution was to use one of Microsoft's popular BI tools, such as PowerPivot or PowerBI. They allow us, among other things, to connect to the data warehouse and create our data model based on a star schema. In practice, these types of connections were realised by pre-prepared views. Another way to access the data is to connect to OLAP cubes (Fig 5). Since both BI tools can execute the queries in memory, reports that use queried data run fast, even with hundreds of thousands of records.

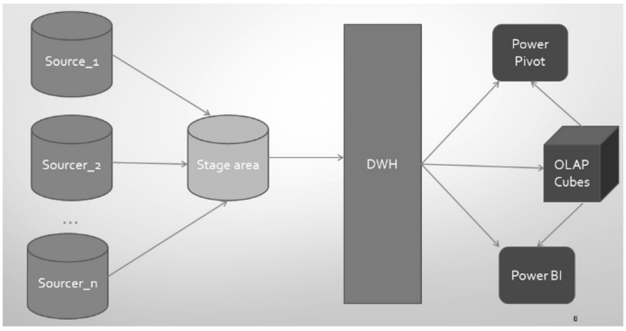


Figure 11: The DWH/BI architecture (Ref. 535780632/h)

Implementation

Creating a data warehouse is often done using visual design tools such as Power Designer or SQL Developer. To avoid using one more software, we created the tables manually with SQL Server Management Studio. Following Kimball's dimensional modelling

first, the dimension tables had to be created, after that the fact tables. Finally, it needed to set the relationships between them.

Initially, the data warehouse contained only test data that allowed the developers to design and create data models using either Power Pivot or Power BI. They form the basis of the self-service BI system. Essential elements of data models are calculations (measures) that will appear in the reports. They are written in DAX (Data Analysis Expressions) language. It is syntactically similar to the Excel-expressions but differs in principle because instead of cell references, only columns or table references are allowed. With the help of the developed data models, it was already easy to create reports and dashboards for managers.

The next challenge was to solve the ETL tasks, including the collection of data from source systems, its transformation, and loading into the data warehouse. The SQL Server Integration Services (SSIS) was used to solve this problem. Apart from the time and date dimensions, all the others are so-called slowly changing dimensions (SCDs).

Loading of dimension tables uses the same logic in all cases (Figure 6). First, we define the data sources and destinations. The primary key of the source table will be the business key. The attributes of the table can be divided into three groups: fixed attributes, historical attributes, and changing attributes.

After that, we check if the current rows already exist in the data warehouse. Accordingly, the process can continue with a new line, historical attribute inserts output, or changing attribute updates output.

Loading of the fact tables is a little bit different because, after loading, there is often for need data transformation (such as data conversions, derived columns). Another difference is one additional step to lookup the right dimension keys from the dimension tables. Finally, at the end of the process, there is a selection. It creates a new record or modifies an existing one depending on the data. All tables must have particular fields and own primary keys to handle historical data and more source systems, such as

- Src: the source system identifier
- InputDateTime: the date of the first load
- LastmodDateTime: the date of the last modification
- ValidToDateTime: the validity of data (only in case of dimensions)

The loading process can be automated using the SQL Server Job Agent. This MS SQL service allows us to run the SSIS-packages on a scheduled, daily basis. Except for the initial database population, every package runs incrementally. Therefore, the total load time is less than two hours. The last task was publishing reports to the company's portal.

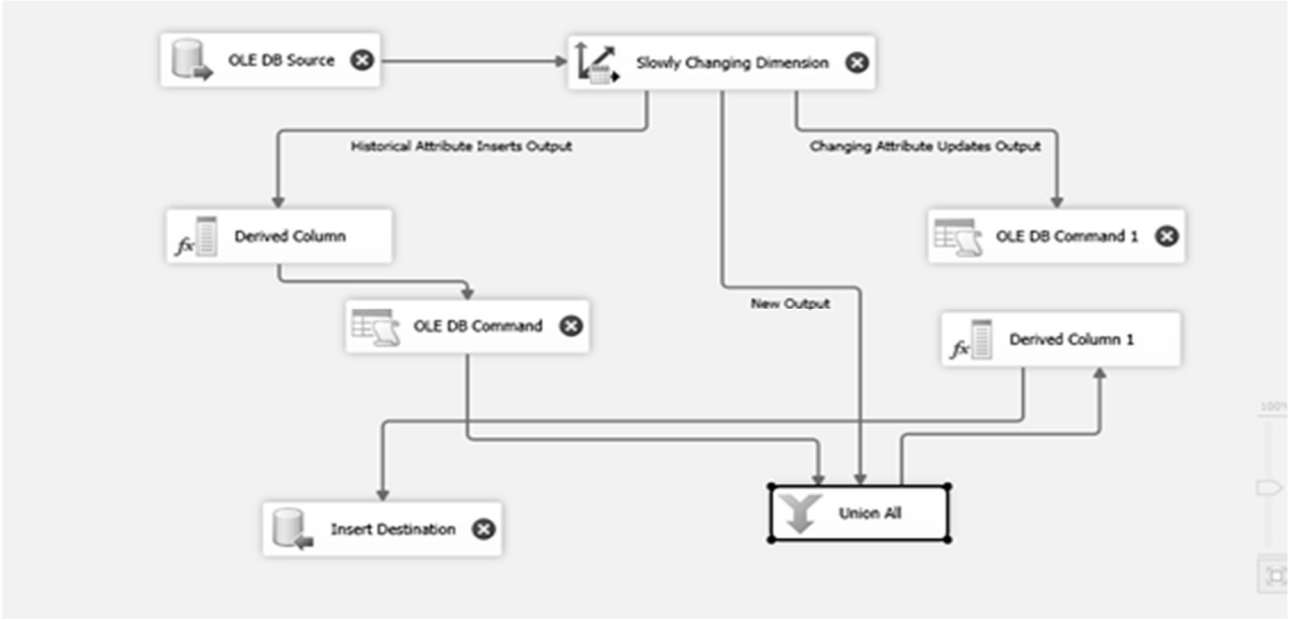


Figure 12: SSIS packages example for loading dimensions

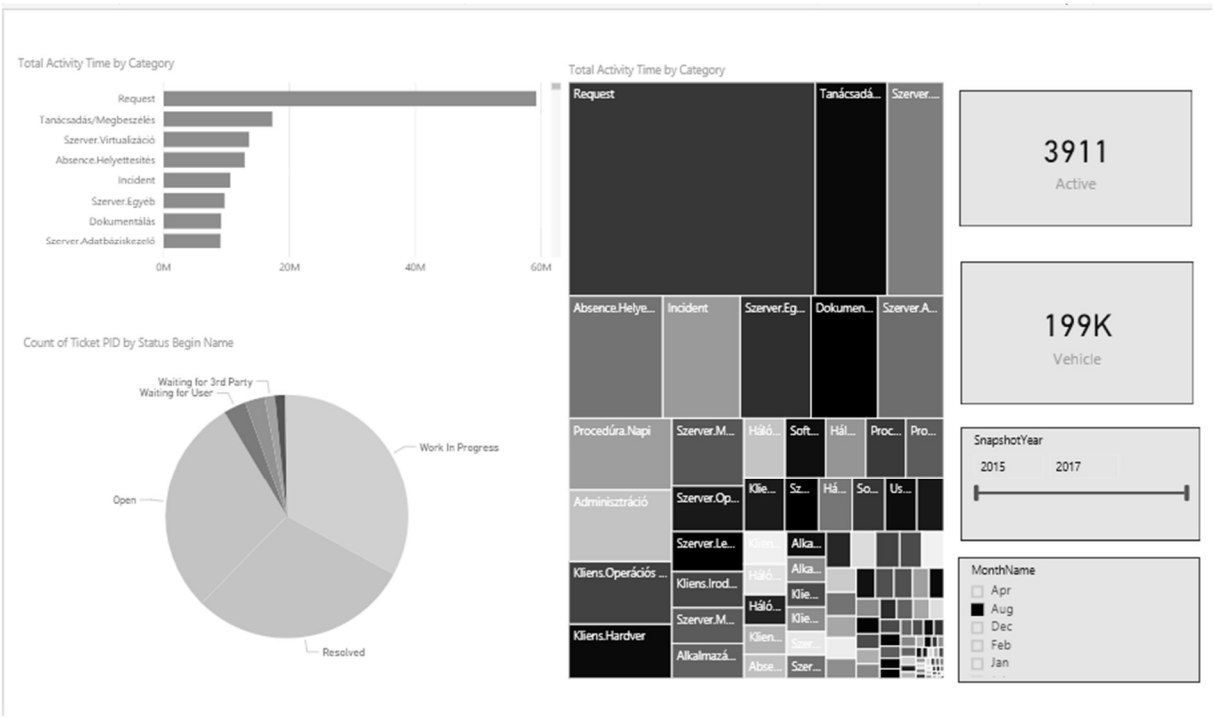


Figure 13: A Power BI report

The created DWH/BI system has solved many previous problems. The data warehouse historically stores data from multiple source systems, ensuring adequate data quality. The managers can use pre-prepared reports or create new ones themselves. An example of the latter is the report (dashboard) shown in the following figure (Figure 7), which was prepared in a simple, drag and drop manner by one of the managers, without any special IT-knowledge

Testing

Testing is always a key point of data warehouse projects. The traditional testing methods, such as structural and functional testing are not always sufficient. Besides them, in most cases, the validation by the customer is indispensable. So, the deployment was preceded by a short test period. It allowed to correct undetected errors and fine-tune the system.

Testing of the DAX-formulas caused the majority of the problems because there is not an automated DAX-testing method or tool yet. Due to lack of this, testing was manageable only with SQL queries performed on a dataset. Comparing the affected data in the data warehouse and the BI reports revealed the errors in the formulas.

Deployment and maintenance

The deployment was preceded by the initial database population during which some of the data was loaded backwards. Then the data warehouse started to work, and the managers recognised the potential in the new system soon. It led to the emergence of new demands which required modifications to the system, like:

- Besides the views, cubes were used to connect to the data warehouse. For this purpose, the Microsoft SQL Server Analysis Services (SSAS) offers two main options, namely the multidimensional and the tabular OLAP cube [19]. Nowadays, the latter method is preferred, so we applied it. Its particular advantage is that any existing Power Pivot data model can be converted into a tabular cube.
- Power Pivot was deprecated, so Power BI became the preferred BI application. It allows us to publish reports to the cloud and share them with other online users. This way, the reports can be viewed in a browser or even on a mobile device.
- It was needed to connect our on-premises systems to the Power BI online services by installing a data gateway software so that the data update also works in the cloud.

Conclusion

Nowadays, data assets are becoming more and more valuable for companies. The managers often suffer from the lack of the necessary information they need to make decisions, whether it is about a strategic decision or a daily problem. In most cases, larger companies have a dedicated management information system or an enterprise resource planning system that partly solves this problem. In contrast, a small or medium-sized company usually must content itself with more straightforward solutions such as using the available office applications or a self-developed system.

In the case presented, we had to face a number of problems, such as extract data from multiple source systems, manage data quality problems, integrate data into a data warehouse and create a self-service BI system for users. The widely used and available software such as Microsoft SQL Server and PowerPivot (or PowerBI) proved to be the right choice, because they have efficient help, various visualisation options, a special programming language (DAX) for managing the unique needs.

The first version of the data warehouse/BI system was available relatively fast in few months ???. Based on experiences and feedbacks, it was possible to improve it by implementing new features that have been not included in the initial specification. Fortunately, these did not require significant structural changes to the data warehouse.

A DWH/BI system is never considered ready. There is a lot of further development opportunities, especially in terms of visualisation, new calculations and semantic enrichment [13], [14], [20], [21].

References

[1] S. I. A. S. Josef Spillner, „Engineering ServiceLevel Agreements: A Constrained-Domain and Transformation Approach," Faculty of Computer Science, Technische Universität Dresden, 2013.

[2] Anonymous, „10 Features Any Good Ticket Management System Should Have," 2019.

[3] R. D. D. &. T. E. Sharda, Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th edition, 2018.

[4] B. V. Gabriella, Az adattárház készítés technológiája, Typotext, 2012, pp. 19-23, 49-51.

[5] K. Andrea, Üzleti intelDöntéstámogató rendszerek 4. fejezet, szerk.: Sántá-né-Tóth Edit, Panem Kiadó, 2008.

[6] L. B. I.-Y. S. Alfredo Cuzzocrea, „Data Warehousing and OLAP over Big Data: Current Challenges and Future Research Directions," International Journal of Business Process Integration and Management, 2015.

[7] W. M. M. Abdelmgeid A. Ali, „Monitoring Business Transactions for a Real-time Data Warehouses," International Journal of Computer Applications (0975 – 8887), 2016.

[8] Y. A. K. M. B. F. Anastasiya V. Demidova, „Designing Multidimensional Information Systems Using the Data Vault Methodology," 2018

[9] R. Kimball, The Data Warehouse Toolkit - Practical Techniques for Building Dimensional Data Warehouses, Wiley, 1996.

[10] M. R. Ralph Kimball, "The Data Warehouse Toolkit Third Edition," Wiley, 2013, pp. 18-22

[11] W. H. Inmon, Building the Data Warehouse, Wiley, 2002, p. 54.

[12] I. Abramson, „Data Warehouse: The Choice of Inmon versus Kimball," IAS Inc, 2004

[13] S. M. F. Ali, „Next-generation ETL Framework to address the challenges posed by Big Data," DOLAP, 2018.

[14] A. Cuzzocrea, „Warehousing and Protecting Big Data: State-Of-The-Art-Analysis, Methodologies, Future Challenges.," Proceedings of the International Conference on Internet of things and Cloud Computing, 2016

[15] W. v. Casteren, The Waterfall Model and the Agile Methodologies, A comparison by project characteristics, 2017, pp. 2-5.

[16] D. M. S. Balaji, „WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC," International Journal of Information Technology and Business Management, 6 2012

[17] Anonymous, Miben más az adattárház projekt, mint egy fejlesztési projekt?, Stratis Kft, 2018.

[18] Anonymous, Four Ways to Build a Data Warehouse, 2007.

[19] Microsoft, „Comparing tabular and multidimensional solutions,"

[20] Microsoft, 2020.G. S. Ko, „A Research Review and Taxonomy Development for Decision Support and Business Analytics Using Semantic Text Mining," Ko, A., & Gillani, S. (2020).

[21] A Research Review and Taxonomy Development for Decision Support and Business AnalInternational Journal of Information Technology & Decision

Making (IJITDM), 1. kötet19(01), pp. 97-126, 2020.

[22] A. L. B. a. I.-Y. S. Cuzzocrea, „Cuzzocrea, Alfredo, Ladjel Bellatreche, and Il-Yeol Song. "Data warehousing and OLAP over big data: current challenges and future research directions." Proceedings of the sixteenth international workshop on Data warehousing and OLAP," ACM, 2013.

Géza Molnár graduated from Eötvös Loránd University in 1995 as a teacher of mathematics, physics and informatics. In the first decade of his career he taught in a high school. Simultaneously, he became an external lecturer at Gábor Dénes College. Later on his interest turned more and more into IT. He has worked, among others, as a system administrator, web developer and data warehouse/BI developer for companies such as Videoton Holding or 4iG Plc. In 2016, he completed a business data analysis postgraduate course at Corvinus University and began his PhD studies at the same institute. His research topic is related to the analysis of textual data in data warehouses. In 2021 he became a full-time assistant lecturer at Corvinus University

Documentation vs. Scraps in the Midst of Digital Transformation
Remarks from a Digital Founder

GÁBOR VITÁLYOS
Vitályos Consulting, Chair of eService Quality of John von Neumann Computer Society
gabor@vitalyos.hu

ABSTRACT

The aim of this paper is to call the ICT experts and decision makers attention to the importance of the interactive quality of professional software documentation, and to the current issues and challenges of the field in the midst of the digital transformation. The paper describes a small, but carefully thought-out set of requirements, worth considering when creating either an interactive documentation or an interactive environment (portal) containing them, and especially when designing document writing technologies for creating interactive environments.

Introduction

Our interest turned to the issue of documentation quality, when encountering the high amount of scraps in search of documentation online. The topicality of this paper is the NJSZT's growing need for high-quality CRM-type software to support his own community's work. For digital transformation we need not only new software, but also accurate and useful documentation for them. Dealing with the standards for the technical documentation, we sums up the situation as follows:

- We have ISO/IEC 26511-26515 standards for the *contents* of software documentation, see [1].
- We also have guides in the internet, one of the most elaborated is [2], standing for the *content*s.
- We have ISO standards based on de facto industrial standards of software vendors defining the file format of the software documentation, e.g. RTF, PDF, DOC.
- At the same time, we have no standards or any conventions for the 'interactive quality' of professional documentation.

For the purposes of this paper, the term *documentation* means reference manuals, tutorials, etc. of professional, consolidated software, or other technical products. emails, correspondence, scraps may be documents, but not documentations in our term.

Pre-digital and post-modern requirements

Our requirements for the quality of the interactive services come from two principles.

- Requirements inherited from the pre-digital world. We think of them as results of the human cognitive capability, the genetic features. One of the most important is the Object Permanency Principle (OPP), [3], coming from the psychology and is discussed for the Usability – or UX - discipline in [4]. It establishes 5 simple axioms – e.g. an object must be in one and only one place at any moment.
- Requirements of our post-modern interactive world: we want to group, filter, select the information and manage the connection between them and manipulate, visualize, public, revoke them, etc. They are being formed by the ICT disciplines in our days.

The essence of these post-modern requirements: they mustn't be in conflict with the pre-digital requirements. Namely the behavior of the virtual *interactive entities* mustn't differ essentially from that of theirs real world ancestors.

Disobeying this principle by the portal builders makes difficult to find and recognize the objects, causes other cognitive difficulties for the clients and finally is the main cause of the usability problems. This paper deals with the pre-digital requirements.

Life cycle of documents or documentation

The next three categories are well known, here we sum them:

- **Transient documents**: read and drop it. From functional point of view these are letters, chief briefings and memos. From technical point of view they are SMS, forum entry, comment, tweet, e-mail in most cases, chats, conversations – namely they are scraps. Naturally, can be archived, later the archive can be analyzed for legal or other procedure. They are not "documentation" for human use.
- **Project documentation**: connected to a project works. It is common to archive them after project close for maintenance problems or legal procedures. Technically they are prepared and used (viewed, searched) by standard Office type applications.
- **Strategic documentation**: May be technical documentation of (or connected to) some technical product, software, etc.; or may be organizational documentation, e.g. a statue of a firm or company; or there are financial, legal documentation and contracts also. Although the conclusions hereinafter are roughly valid for all categories, we concentrate to the technical category. *The life cycle of the strategic documentation has to comply with that of the product – or other entity - it is connected to.*

Unfortunately, a lot of information being project or strategic by their goal, are written via tools of transient documents. And nobody will collect and organize them into any *documentation*. This is the main imperfection, discussed and criticized in this paper. We face it mainly among the technical - more precisely among the software – documentations.

The concept of the 'interactive documentation'

The information building works in the pre-digital era were based on the next information container panels:

- document, scraps, etc
- book,
- dossier, containing many of the previous, the pre-digital form of the directory in the file system
- bookshelf, containing whole sets of these, i.e. whole set of documentation
- office, containing all of these and forming work environment for us,
- library, a public service, out of our office

The 'interactive documentation' is – or would be – the suitable interactive digital successor of all these pre-digital information container panels. Important mission of the ICT industry is - or would be - to elaborate and develop these successors. (More precisely: the software development technologies for building them.)

The Book

Finally, we can specify the goal of this paper: It deals with the pre-digital features of the *book* – for revealing the requirements against the suitable digital successor. The book may be dossier or clipped, stapled papers also. We simply refer them as *book*.

The *book* behaves as an object: we recognize and identify it in the table or shelf of the office, without effort, by its cover, or maybe by its size. The book *introduces itself* for us: shows the title, the author, the impress, we can see them without effort. Furthermore, the table of contents shows the thematic and the inner structure. We find it at its habitual place: near the front or back cover, no long search for it.

Thus, the book is an object, complying with the OPP. In addition, the content of the *book* is a set of objects - e.g. chapters - the book has to arrange in comprehensive way. Moreover, the *book* is a **cultic** object, evolved during centuries in the European culture, counted as aggregate and holder of information and knowledge, used as handheld commodity. The *book* is suitable form of the strategic documentation.

We search the features of the 'interactive document' making it the worthy successor of the cultic *book*. We defined and elaborated the theoretical principles, in other words axioms, of these features. In the other hand the *book* is a product so its production and use need a correct business model. The next requirements or axiom-like points may seem to be trivial. Remember, that the separate axioms are generally trivial, but their set as a whole constitutes the important meaning: the essence of the "bookness".

- **Completeness.** It covers the whole of the topic established in the table of contents. Even if the document is not yet complete, the table of contents must be. The **non-overlapping** (or redundancy-freeness) is a case in point: repeated topics or texts are not allowed. This inner feature is derived directly from the concept *book*.
- **Clarity** (understandability, comprehensibility). In other words: transparent structure. The common tree structure with comprehensible chapter tittles and the table of contents satisfies this principle. There may also be contents registers and indices. This inner feature has evolved during centuries, along with the *book*, and is its attribute. So we derive it directly to the interactive documentation**.** Naturally, there is a conflict between the comprehensible structure and the complexity of the knowledge - especially of technological and business-oriented knowledge - of our days. The author's opinion: we have not yet well-known de facto standard for the structure of the knowledge instead of the tree-formed structure, and the same formed table of contents.
- **Neighboring**. If one finds some topic on the document, one will search the related topics in the neighboring spaces. The feeling of the neighborhood comes in the one hand from the 'space feeling', based on the proper tree structure of the contents, and in the other hand from the proper arrangement of the concepts among the tree. This inner feature is the consequence of the **Clarity**.
- **Orderableness** (capability of the content to be ordered in linear way, in other words "inner findability"). This is a consequence of the tree structures and is connected to the neighboring. The capability of moving, i.e. scrolling the forward or backward direction while scanning or searching the content gives us ergonomic safety feeling or "scroll bar feeling": feeling that we have nothing left out accidentally. It is only feeling, decreasing the mental load, and not technical fact. This inner feature is, in some measure, consequence of the **Clarity**. The Orderableness and the Neighboring make together the most important inner feature: the **inner findability**, i.e. the reader is capable to find information in the *book*.
- **Thisness** (or singularity, identifiability). A unique, individual product (in our cases technical construction, mainly software) needs one and only one documentation, or reference manual. More precisely, one set of manuals, generally: reference, user, operator and owner manuals. Other writings must be of other type: essay, report, review, etc.Naturally a singular, identifiable documentation can only be produced for a product complying itself with the same requirements. This ideal situation is rare, large compromises are to be made in this feature. This external feature is derived directly from the concept of *book* and from the industrial practice.
- **Impartibility** (atomicity). No part of a book is book. If I cut a book in pieces, no peace is a book, for **Completeness** and **Clarity**. This is an external feature, derived directly from the concept of *book*.

- **Outer Findability** - the bookshelf problem. A book needs a well-defined *location* in our physical environment, the office, generally in the bookshelf. If a *book* consists of more tomes, we like them to be in close proximity, to be side by side in the bookshelf. Otherwise, we feel disorder and great mental load while find one of them. We have an interesting paradox: we expect that a *book* (in the physical space) be movable in the space, but at the same time we expect it have a usual locus, where we find it without great effort. This external feature is derived directly from the concept of *book*.
- **Impress.** The books need author and senior editor, being responsible for the compliance of the previous principles. The impress must introduce them. There need also auxiliary information, e.g. the deadline, or, in case of portal-like *book*, will there be upgrades or not. This inner feature is derived directly from the concept *book*.
- **Business model.** The impress must tell, whether on can buy it, or it is sponsored by software vendor or other body. In addition, we would like to know, if the book will be found at the same place in the future, or this is a marketing campaign.This external feature is derived directly from the concept *book*, and this information is needed for its authenticity.

Remarks on de reader's feeling: reader is given the **authenticity feeling,** if the book complies with all previous principles. The **user experience** (or usability) is based mainly on the Completeness, on the Clarity and the Neighboring. The e) Thisness, the f) Impartibility and the g) Outer Findability make together the most important feature that the *book* is an object.

Examples

We analyze some cases of interactive documentation in our virtual environment in accordance with the previous principles. Meanwhile we search the suitable digital successor of the *book*.

- Paper-like documentation: They are prepared and used by different sort of Office and Arcobat applications. To satisfy the inner features principles a)-d) is mainly up to the authors, because the software applications allow them, giving a consolidated, relatively simple and printable structure and having de facto standard interactivity. But this type of the documentation is not a real successor of the *book*. It is simple mirroring of the *book* to the file.
- Wiki-s: Wikipedia, Wikimedia, Wikimedia, etc. form a unique, new structure, based on the Wiki software technology. It is intended for the dictionaries and other cyclopedia-like contents, the special form of the documentation. Trying to consider a whole wiki as documentation, we face the Neighboring and Orderableness are violated.

GitHub-hosted documentation - findability problems; Searching the for example the CiviCRM documentation, we land a page, like FIgure 1: If we need a glance to the Guide, we don't know where to go. If we click eventually to the GitHub link, we face a directory hierarchy, see Figure 2.

The GitHub is a very inventive, well elaborated and famous software project management tool, with version control, source code management, and wiki-bookshelf for every project, etc. But it is more than extravagant idea to use it for documentation bookshelf. The GitHub is rather the programming workbench, witch's brew for programmers, where the reader (the user of the documentation) isn't worthy to be allowed into, for he loses his way there.

In addition, we don't find the well-known documentation formats, for the GitHub uses the new, not yet common so-called MD (MarkDown) format and tool kit for it. The GitHub is tool for developers, rather than users. These may be matter of course for the software experts, but not for the reader of the CiviCRM documentation, who is not even software expert. Information found here can serve as documentation for the experts, but counted as composted scraps for the readers.

Moreover, we also see the most common portal mistake here: the pages - the Figure 1. shows only one of them - make loops among themselves. But it is important role, the portal mustn't allow the user round and round by clicks, because this may cause cognitive difficulties in the understanding where he is, so increases the mental load, the probability of losing the way.

❖ Documentation vs Scraps

If I still done such a solution with a portal-like intro to the CiviCRM documentation, I would do something likes what is in the Figure 3. instead of Figure 1. The bold dots are links to the documentation in question. This is the simplified demo.

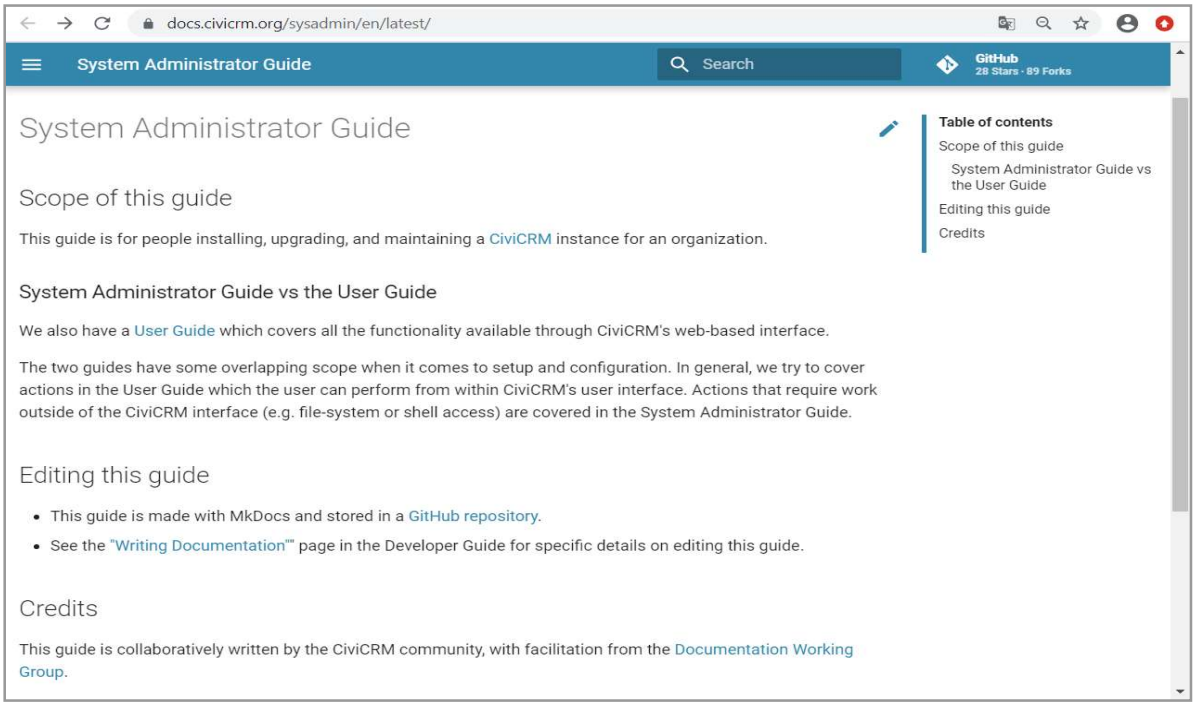


Figure 1. Where can I find the Admin Guide here? Where to go?

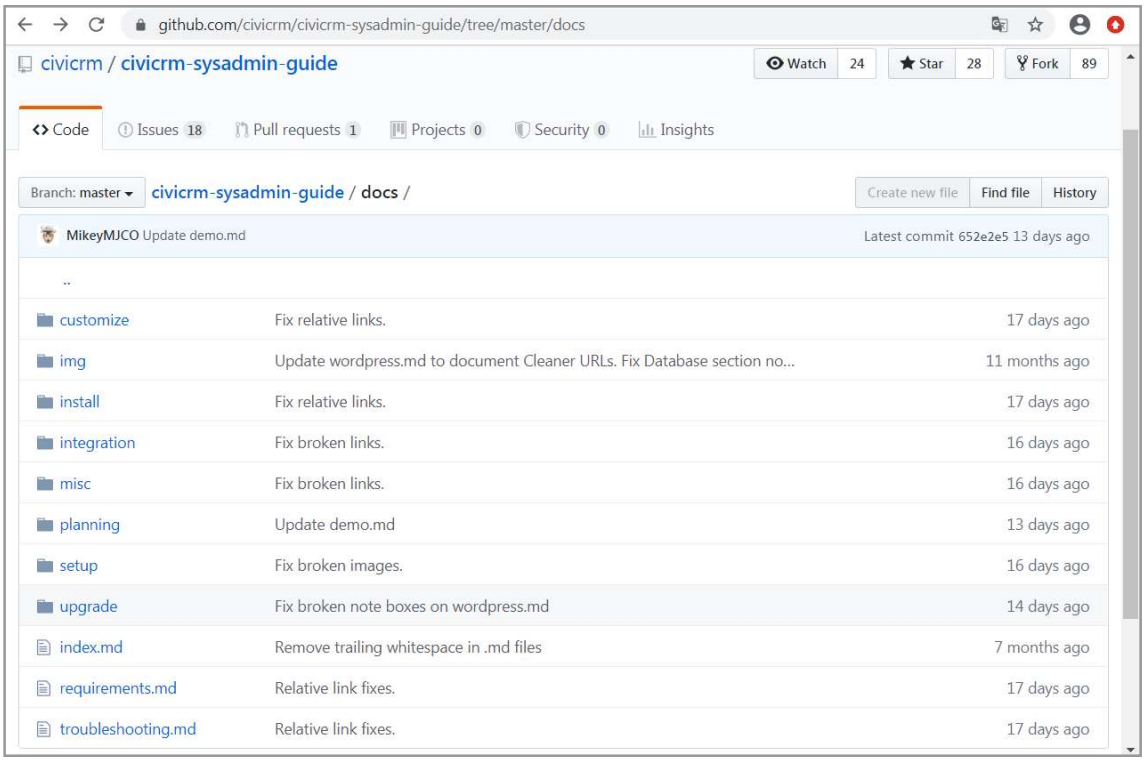


Figure 2. Is it the Sysadmin Guide? These are the chapters of the Guide, or 8 guides?

CiviCRM documentation

	Read in GitHub	Write in GitHub	Get the PDF
CiviCRM user's guide	●	●	●
CiviCRM System Administrator Guide	●	●	>400 pages
GitHub Guide for documentation readers	●	●	●
GitHub Guide for documentation writers	●	●	●
....			

The native place of the documentation is within the GitHub. Getting the PDF is via on-line conversion, may take some minutes and no version control information. It works only small files, till 400 pages.
....

Figure 3. The possible main screen, giving a solution of the Outer Findability and the Thisness.

A good example: a portal-like (HTML) handbook; The figure 4. shows the front page of the [2] book. This guide complies with almost all requirements of our definition (we don't see the imprint, and business model). The unique URL satisfies the Thisness. The Impartibility comes from the write inhibition of the containing portal. The Outer Findability is from the containing portal also. Its well-formed and detailed table of contents is worth to be considered for Document writers.

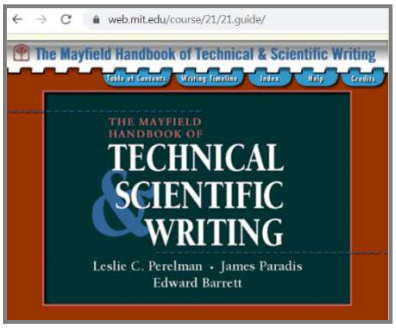


Figure 4. The front page of [2]

Other, nonstandard documentation methods: These are generally distributed portal-like HTML solutions; we can read them by browser, in a portal. They actually are a part of the portal, and suffer from the common portal building lacks. So, these solutions generally can't be counted as documentation.

Results

We hardly find de facto standard for interactive documentation which can be worthy digital successor of the *book*. We see no new ideas or any consensus on knowledge structure, and so we still use the obsolete tree structures. We can make the prognosis that *the one of the most interesting movement of the ICT development will be on the knowledge structures and on their interactive representation and manipulation.*

For example, the **Orderableness** requirements can be weakened if we allow the partial ordering [5]. This ordering doesn't contain loop, more complex than the tree, and may have comprehensible visual representation. See Figure 4.

This paper is part of the research of the HCI (Human Computer Interaction) of the e-services by the 'Community of e-service quality, John von Neumann Computer Society, Budapest.

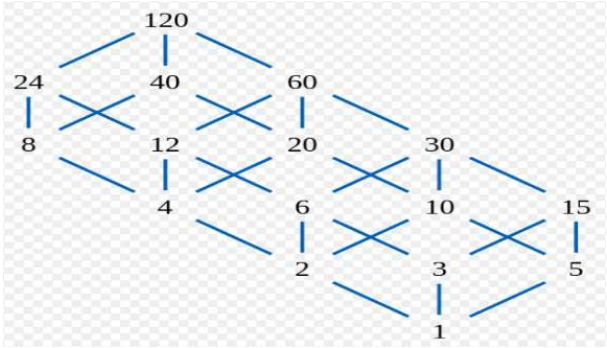


Figure 4. The divisors of 120 ordered partially
Source Wikipedia.

References

[1] ISO/IEC 2651x at a glance: https://www.technical-communication.org/technical-communication/software-documentation
[2] Perelman-Paradis-Barrett: Technical & scientific writing https://web.mit.edu/course/21/21.guide/ https://en.wikipedia.org/wiki/Object_permanence
[3] Vitályos 2011: The object permanency principle in the usability discipline SZTAKI, IEEE, 2011https://en.wikipedia.org/wiki/Partially_ordered_set

Gábor Vitályos has graduated in Electrical Engineering (MSc), postgraduated Special Engineering in Information Technology (MDD), Budapest University of Technology. He is working as an expert of customized, high quality software solutions, researcher-developer in information technology of the digital founder generation. As a co-founder and manager of the Vitalyos Consulting firm he managed the development of complex, high quality information systems. His work covers the whole life cycle of the systems: from the conceptual plan through the system analysis to the testing, documentation, training and the maintenance. He performed among others great development projects on real time technological data gathering, data warehouse for power-plant and bank middleware. Since 2010 he is working on designing, building and evaluating the Information Architect Reference Model for high quality interactive information systems. He is the founder of the eService Quality Community of John von Neumann Computer Society (Budapest)


S·E·F·B·I·S


GIKOF JOURNAL
G·I·K·O·F


GIKOF Journal
G·I·K·O·F


Gazdaságinformatikai Kutatási és Oktatási Fórum
G·I·K·O·F

Előszó

RAFFAI MÁRIA
Editor in Chief; Chair of GIKOF/SEFBIS SIG
Professor at Széchenyi University, former NR and Councilor of IFIP,
vice chair of IFIP Information Science TC Enterprise Information Systems WG

A Gazdaságinformatikai Kutatási és Oktatási Fórumot (GIKOF) 2001-ben alapítottuk, a Neumann János Számítógép-tudományi Társaság egy speciális szakmai közösségeként. Közösségünk az elmúlt két évtizedben nemcsak fórumokat, konferenciákat szervezett, de szakmai folyóiratokat is megjelentetett, magyar és angol nyelven. Az eredmény 13 SEFBIS angol nyelvű és 11 GIKOF Journal magyarnyelvű kiadvány. A COVID-járvány sajnos több, mint egy évre visszavetette a közösségi munkát, a szakmai tevékenységet, így a kötelező elzártság, a karantén megakadályozta a személyes találkozást, a benyújtott cikkek kis száma miatt pedig 2020-ban nem tudtuk megjelentetni a szakfolyóirataink. 2021 első félévének végefelé azonban egyre növekedett az igény a szakmai közösségre, a megjelenésre, az együttlétre, így nemcsak a jelen szakfolyóiratot tudtuk megtölteni cikkekkel, de novemberben megszervezzük Veszprémben a hagyományos Gazdaságinformatikai Konferenciát is.

A GIKOF húsz éves jubileuma alkalom arra is, hogy visszatekintsünk az elmúlt időszakra, hogy értékeljük munkánkat és eredményeinket. Ehhez egy összefoglaló anyagot készítettünk, amely nemcsak a múltat tekinti át, de a jövőre vonatkozóan is komoly terveket vázol fel. A GIKOF-Journal sorozat külön köteteként megjelent *"2001-2021 Húsz év a gazdaságinformatikai szolgálatában"* c. füzet tükrözi mindazt a sok munkát, erőfeszítést, a nehézségeket és problémákat, és nem utolsósorban az eredményeket, amiket a gazdaságinformatikáért elkötelezett kollégák elértek.

Jelen kötetünk kétnyelvű, hűen tükrözi a szerzők sokrétű érdeklődését az ICT-szakma üzleti, gazdasági alkalmazása iránt és a történeti visszatekintő kötettel együtt felhívja a figyelmet és az érdeklődőket a 2021 évi konferenciánkra, amelynek fókuszát a COVID-járvány üzleti hatásai, a járvány alatti on-line megoldások és a nagyon bizonytalan jövő megoldandó kérdései jelentik...

Ezúton is ösztönözzük a téma iránt érdeklődőket, hogy nyújtsák be cikkeiket a Szerkesztőséghez! Legyen a Journal-unk lehetőség arra, hogy a szakemberek megismerhessék a legújabb kutatási eredményeket, hogy tudomást szerezzenek az új fejlesztésekről, az üzleti innovációt elősegítő és támogató alkalmazásokról, a felhasználói tapasztalatokról és hogy ötletet merítsenek a további munkájukhoz![1]

Raffai Mária
GIKOF Journal főszerkesztő

[1] A GIKOF-Journal elektronikus változata letölthető a GIKOF Szakmai Szervezet honlapjáról: http://raffa6.wix.com/gikof

Az információrendszerek értékelésének interpretatív megközelítése

NAGY GÁBOR SZABOLCS
Pécsi Tudományegyetem, Gazdálkodástani Doktori Iskola, PhD hallgató
eMail: nagy.gabor.szabolcs@gmail.com

ABSTRACT

Bár az információrendszerek értékelésével régóta foglalkozik a szakirodalom, a téma napjainkban is aktuális. A publikációk többsége jellemzően formális-racionális módszereket ajánl az információrendszerek értékelésére, de emellett akadnak olyan tanulmányok is, amelyek az interpretatív megközelítés fontosságát hangsúlyozzák, ami egyfajta szemléletváltást jelent a hagyományos megközelítéssel szemben. A dolgozat rövid áttekintést ad arról, hogy miben más az interpretatív megközelítés, miért lehet létjogosultsága, pontosan hol a helye, mi a szerepe az információrendszerek értékelésében.

Bevezetés

Az 1990-es évek termelékenységi paradoxonként elhíresült vitája két kényes kérdést is felvetett az információrendszerek értékelésével kapcsolatban: (i) egyrészt, hogy nem a módszerek hiányosságaira vezethető-e vissza a paradoxon léte, (ii) másrészt, hogy nem az értékelési módszerek gyengeségei tehetők-e elsősorban felelőssé azokért a rossz döntésekért, amelyek később alacsonyabb jövedelmezőséghez vezetnek [1] [8]. A Gartner előrejelzéseiből, valamint a Standish Group Chaos Report című jelentéséből azt látjuk, hogy évről évre egyre többet költünk IT/IS beruházásokra, ugyanakkor a projektek jelentős része kudarccal zárul, vagy nem az eredeti terveknek megfelelően kerül megvalósításra. Egy információrendszer informatikailag sikeres bevezetése még nem feltétlenül jelenti azt, hogy a rendszer üzleti, vagy stratégiai szempontból is sikeres. Mindez még inkább ráirányítja a figyelmet az értékelés fontosságára.

Az információrendszerek értékelésével foglalkozó szakirodalom nagy része különböző formális-racionális módszereket használ az információrendszerek értékelésére. Nagyon sok modell született, ugyanakkor kevés az olyan tanulmány, amely a modellek gyakorlati alkalmazhatóságát vizsgálja. Akadnak olyan publikációk is, amelyek az interpretatív megközelítés fontosságát hangsúlyozzák, és egyfajta szemléletváltást sürgetnek az információrendszerek értékelésében.

A különböző interpretatív módszerek jól ismertek a társadalomtudományokból, de az informatika világától sem állnak távol, hiszen a szoftverfejlesztés folyamatában is használnak ilyen módszereket. Valójában nem önmagában az interpretatív módszertan alkalmazása az érdekes, hanem a mögöttes szemlélet, hogy miként gondolkodunk az információrendszerekről, illetve az értékelés céljáról, szerepéről. Ha az információrendszerek nem csupán gépek, berendezések összességeként fogjuk fel, hanem figyelembe vesszük az emberi, szervezeti szempontokat is, akkor a hagyományos értékelés módszerek már nem elegendőek.

A dolgozat rövid áttekintést ad az információrendszerek értékelésének módszereiről, különös tekintettel az interpretatív megközelítésre, elsősorban arra koncentrálva, hogy milyen szerepe lehet az információrendszerek értékelésében, illetve hogy miként járulhat hozzá ahhoz, hogy jobban megértsük a rendszer működését, és fejlesszük használatát annak érdekben, hogy ténylegesen realizálódjanak az üzleti előnyök.

Információrendszer-értékelésről általában

Az értékelési szempontok változása

Az IT/IS szerepének változásával párhuzamosan az információrendszerek értékelésének fő szempontjai is változtak [36]: (1) A kezdetektől a ’60-as évek végéig leginkább az *automatizálás* és a *hatékonyság növelése* volt az elsődleges cél. (2) A ’70-es évektől

az *információnyújtásra* és az *eredményesség* növelésére került a hangsúly. (3) A '80-as évek közepétől az *új üzleti lehetőségek kiaknázása*, és a lehetőségek megvalósításához szükséges *változások támogatása* került előtérbe. (4) A '90-es években az Internet megjelenése, illetve az utóbbi évtizedek látványos technológiai fejlődése még inkább kiszélesítette az üzleti lehetőségeket, és új távlatokat nyitott, és ezzel együtt az információrendszerek köre is jelentősen bővült.

Az értékelés a hagyományos költség-haszon elemzés helyett egyre inkább az üzleti értékre, a hosszú távú célokra, valamint a szervezeti és környezeti hatásokra koncentrál. Nehéz megmondani, hogy mit hoz a jövő, de a jelenlegi trendek alapján úgy tűnik, hogy a technológia költsége egyre kevésbé fontos szempont (egyelőre úgy látszik, hogy a hardver és szoftver árak tovább csökkennek), ugyanakkor egyre fontosabbá válnak az egyénekkel, a szervezettel, valamint a belső és külső környezettel kapcsolatos hatások.

Az IT/IS életciklusához igazodó értékelés

Az információrendszerek „életciklusa" kapcsán általában a szoftverfejlesztés életciklus modelljeire, vagy a beruházási projektek életciklusára gondolunk, de a rendszer életciklusa ennél jóval tágabban értelmezhető, a rendszer használatát is magában foglalja. Az informatikai beruházások általában valamilyen fejlesztési/beruházási projekt keretében valósulnak meg, így az értékelés rendszerint a projekt életciklusához igazodik (ex ante, mid term, ex post), s többnyire egy utólagos értékeléssel véget is ér.

Az információrendszerek bevezetésének sajátossága, hogy a fejlesztés általában hosszadalmas folyamat, az implementáció és a fenntartás költségei magasak, a várt hozamok hosszú idő után jelentkeznek, és nem mindig az eredeti elképzelésnek megfelelő mértékben. Ahhoz, hogy a használatból származó előnyök ténylegesen realizálódjanak, szükség van a rendszer használatának fejlesztésére, ehhez azonban meg kell érteni a rendszer működését, illetve tudatosítani kell a rendszerrel kapcsolatban állókkal, hogy milyen módon és milyen mértékben keletkeznek a hasznok. Az értékelés nem érhet véget a projekt lezárásával, olyan értékelésre van szükség, amely nem csupán a projekt életciklusához, hanem az információrendszer életciklusához igazodik.

Egyre nagyobb igény van egy olyan folyamatos értékelésre, amely a már használatba vett rendszer működésének megértését, illetve a rendszer használatának fejlesztését szolgálja annak érdekében, hogy valóban realizálódjanak a használatból származó előnyök, és minél többet ki tudjunk hozni a már működő rendszerekből.

- Az értékelés során nem a projekt életciklusához, hanem az információrendszer*életciklusához kell igazodni*, az értékelés nem érhet véget a projekt lezárásával.
- Az értékelés nem egy egyszeri tevékenység (mint a projektekhez kapcsolódó előzetes vagy utólagos értékelés), hanem egy *folyamatos tevékenység*, melynek alapvető *célja a rendszer működésének megértése, illetve a használat fejlesztése, az üzleti érték növelése érdekében*. Ebben a megközelítésben az értékelés egy iteratív szervezeti tanulási folyamatként is felfogható.
- Az értékelés során nem önmagában egy konkrét üzleti érték meghatározása a cél, hanem a *működés megértése*, hogy ténylegesen hol és miként keletkezik az üzleti érték, hogyan befolyásolja a rendszer a szűkebb és tágabb értelemben vett környezetét, és ez hogyan hat a rendszer működésére, hogyan lehet többet kihozni a rendszerből, azaz hogyan lehet növelni a rendszer értékét a használat során.
- Az *értékelés fókusza* folyamatosan változhat, attól függően, hogy éppen milyen részterületet vizsgálunk, de végső soron mindig a hosszú távú stratégiai szempontokat kell szem előtt tartani. A cél, hogy az informatikailag sikeres projektek végül üzletileg, stratégiailag is sikeressé váljanak.
- Egy ilyen értékelésben a hagyományos kvantitatív és kvalitatív módszerek mellett kiemelt szerepet kaphatnak a kifejezetten interpretatív módszerek is, de nem a módszertan az igazán lényeges, hanem a szemlélet, hogy miként gondolkodunk az információrendszerről, illetve magáról az értékelésről.

Az IR-értékelés nehézségei

Az értékelés önmagában egy bonyolult és összetett folyamat, és ez különösen igaz az információrendszerek értékelésére. A rendszerek sokfélesége, a nézőpontok, szempontok, értékelési célok különbözősége csak tovább bonyolítja a helyzetet. Azt értékelés során számos nehézséggel kell szembenézni, a leggyakrabban említett problémák az információrendszerek sajátosságaival, az információtechnológia fejlődésével, az IT fejlesztésekkel, a költségek és a hozamok számszerűsítésével, a külső hatásokkal, a rendszert használó egyénekkel, valamint a szervezettel kapcsolatosak [4].

Az IT/IS rendszerek sajátosságai

Sokféle információrendszer létezik, melyek értékelése eleve eltérő módszereket igényel [19]. Ráadásul az újabb rendszerek egyre *összetettebbek* és *szofisztikáltabbak*, mind a funkcionalitás, mind az alkalmazási lehetőségek tekintetében. A különböző rendszerek gyakran szorosan *kapcsolódnak egymáshoz*, ami szintén nehezíti az értékelést [41]. Ahogy időről időre változott az IT szerepe, úgy *változtak az értékelés szempontjai* is [36] [6]. Az értékelés egyre inkább interdiszciplináris megközelítést igényel, az informatikai és a pénzügyi szempontok mellett stratégiai, szervezeti, humánpolitikai szempontokat is figyelembe kell venni.

Az IT/IS beruházások sajátosságai

A beruházások célja általában a költségek csökkentése, a termelékenység növelése vagy valamilyen stratégiai versenyelőny elérése, ezek az eredmények rendszerint hosszú távon jelentkeznek. Az informatikai beruházások *gyakran más befektetési/beruházási alternatívákkal versenyeznek* [6], ezért még inkább fontos, hogy az értékelés minél reálisabb képet adjon a beruházásról. Az informatikai projektek körében szokatlanul *magas a kudarcok aránya* [6], ami szintén szükségessé teszi az értékelési módszerek alkalmazását már az implementálást megelőzően. Általában *több fejlesztés is folyik párhuzamosan*, és ezek gyakran kapcsolódnak is egymáshoz [22]. Az IT beruházások bizonyos értelemben *visszafordíthatatlanok*, bár magát a fejlesztést le lehet állítani, a befektetett összegeket már nem lehet visszanyerni [22].

A költségek és a hasznok számszerűsítése

Az IT beruházások értékelésénél a legtöbb gondot a költségek és a jövőbeni hozamok számszerűsítése jelenti, ami komplex és bonyolult módszereket igényel [17]. Az IT/IS beruházások egyik sajátossága, hogy a rendszerből származó hasznok rendszerint *hosszú idő után mutatkoznak meg* [33]. Emellett az is nehezíti a hasznok számszerűsítését, hogy nem minden mérhető közvetlenül pénzben, sok *immateriális* formában megjelenő hozadéka is van az információrendszereknek, melyeket sok esetben nehéz pénzben kimutatni [4]. Léteznek olyan rendszerek is, amelyek *nem termelnek ugyan hasznot közvetlenül*, de az alapját képezik más rendszereknek, így közvetve hozzájárulnak az értékteremtéshez [35]. Végül sok esetben a *nem megfelelő használat* miatt nem mutatkoznak meg a hasznok [8].

Az előnyök egyik legszemléletesebb kategorizálását Brown írja le az alapján, hogy az adott előny mennyire mérhető, illetve milyen mértékben tulajdonítható az információrendszerbevezetésének [7]. Az előnyök egy része egyértelműen az információrendszerbevezetéséhez köthető és jól mérhető, de ezeken kívül, az úgynevezett „puha" előnyöknek legalább három csoportját lehet megkülönböztetni, melyekhez már különböző kvalitatív módszereket kell alkalmazni [7] [8]. Az *immateriális* előnyök viszonylag egyértelműen hozzárendelhetők bizonyos alkalmazásokhoz, azonban a mérésük, számszerűsítésük már nem ennyire egyszerű. A *közvetett* vagy *potenciális* előnyök könnyen mérhetők, de nem tulajdoníthatók teljes egészében és kizárólag az adott rendszernek, és csak akkor számszerűsíthetők, ha már megvalósult a rendszer bevezetése. Végül a *stratégiai* előnyök hosszú távon jelentkeznek, s nagyrészt visszavezethetők az információrendszerre.

Hivatkozás	A hasznok taxonómiája
Gustafsson et al. (2008)	Üzleti érték: (i) rugalmasság, (ii) hatékonyság, (iii) eredményesség, (iv) integráció és koordináció, (v) döntési folyamat fejlesztése, (vi) szervezeti kultúra fejlesztése.
Ward and Daniel (2006)	Az információs rendszerek háromféle módon járulnak hozzá a szervezet működéséhez: lehetővé teszik (i) új tevékenységek végzését, (ii) meglevő tevékenységek fejlesztését, (iii) nem szükséges tevékenységek kiváltását, felszámolását.
Kusters and Renkema (1996)	Hasznok, előnyök: a hatékonyság előnyei, az eredményesség előnyei, szervezeti transzformáció, technológiai szükség és/vagy rugalmasság, külső elvárásoknak való megfelelés, tágabb értelemben vett humánpolitikai és szervezeti hatások.
Ward et al. (1996)	Hasznok, előnyök: költségcsökkentés, menedzsment információ, folyamatok hatékonysága, változások lehetővé tétele, versenyelőnyök, szolgáltatások minősége, stb..
Shang and Seddon (1996)	Hasznok, előnyök: működési, menedzseri, stratégiai, IT infrastrukturális, szervezeti.
Farbey et al. (1995)	Az információs rendszerek típusai alapján különböztetik meg a hasznokat: üzleti transzformációval kapcsolatos, stratégiai, infrastrukturális, menedzsment információs vagy döntéstámogató, közvetlen hozzáadott érték, stb.
Brown (1994)	Az előnyök egy része egyértelműen az információs rendszer bevezetésének köszönhető, jól mérhető. Emellett az ún. „puha" előnyök legalább három csoportba sorolhatók: immateriális előnyök, közvetett vagy potenciális előnyök, valamint stratgiai előnyök.
Renkema and Berghout (1996)	**Hozzájárulás** / **Pozitív** / **Negatív** / **Összesen** **Pénzügyi**: hozamok / költségek / profitabilitás hasznok / kiadások / megtérülés **Nem pénzügyi**: pozitív hozzájárulások / negatív hozzájárulások / hozzájárulás **Összesen**: hasznok / áldozatok / érték

11. ábra: A hasznok taxonómiája

Környezeti és szervezeti hatások

A hosszú távú sikeresség szempontjából nem lehet figyelmen kívül hagyni azt sem, hogy az információrendszer milyen hatást gyakorol az egyénekre, a szervezetre, a külső környezetre, illetve, hogy ezek a tényezők miként befolyásolják a rendszer működését, miként mozdítják elő, vagy akadályozzák a hasznok realizálását. Az értékelés során *több nézőpontot is figyelembe kell venni*. Különböző stakeholderek részére készül az értékelés (menedzsment különböző szintjei, felhasználók, szakértők, projekt csoportok, stb.), akik eltérő szempontokat tartanak fontosnak, s ez gyakran érdekütközést eredményez [41].

Az IT beruházásokkal kapcsolatos döntések nem mindig racionálisak, gyakran erős hatást fejt ki rájuk a szűkebben vagy tágabban értelmezett *környezet*. Az információrendszerek bevezetése *jelentős változásokat hozhat a szervezetben* (a struktúrában, a folyamatokban), *a szervezeten belüli kapcsolatokban* (megváltoztatja a szociális interakciókat, az életminőség javulását eredményezi, hatással van a szervezeti kultúrára) illetve *a menedzsment folyamatokban* (információhoz való hozzáférésben, illetve a döntéshozatali folyamatok racionalizálásában), de, ezeket a hagyományos értékelési módszerek rendszerint figyelmen kívül hagyták [47].

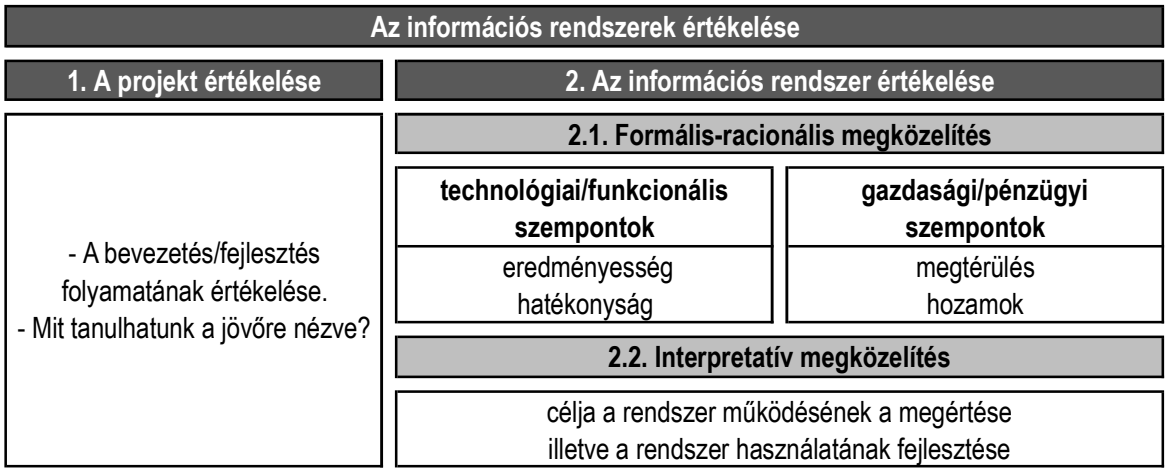
Végül fontos szempont a *munkavállalók reakciója* is. Az értékelés érzelmi reakciókat vált ki, mert a felhasználók fenyegetve érezhetik magukat, ezért amikor megkérdezik a véleményüket, hajlamosak alulértékelni a rendszer teljesítményét, ami torzítást eredményezhet az értékelés során [29]. Ezekre a szempontokra a hagyományos értékelés nem fordít kellő figyelmet, ami részben érthető, mivel más a célja, illetve az alkalmazott módszerek sem teszik lehetővé ezeknek a hatásoknak az értékelését.

Az értékelés formális-racionális megközelítése

Az információrendszerek értékelése egyrészt a projekt értékelését, másrészt magának az információrendszernek az értékelését jelenti [23]. Önmagában a projektek értékelése is fontos visszajelzés, ami tanulsággal szolgálhat a jövőre nézve, viszont a hasznok hosszú távú tényleges realizálása szempontjából legalább ennyire fontos, hogy a működés során is folyamatosan értékeljük a rendszert. Az információrendszerek értékelésével foglalkozó szakirodalom meglehetősen szerteágazó. A tanulmányok többsége formális-racionális módszerekkel vizsgálja az információ értékét, az értékelés elsődleges célja alapján Serafeimidis szerint nagyjából két irány különböztethető meg, az egyik technikai/funkcionális szempontokat vizsgál, a másik pedig gazdasági/pénzügyi jellegű [39]. A hagyományos megközelítés mellett létezik egy másik irány is, amely az interpretatív szemlélet fontosságát hangsúlyozza (1. ábra).

Gyakran nehéz meghúzni az éles határvonalat, mivel a technikai/funkcionális értékelés is rendszerint valamilyen pénzügyi kontextusban jelenik meg, de az alapvetően gazdasági/pénzügyi értékelésben is helyet kaphatnak a technikai és funkcionális szempontok is. Song és Letch 25 év szakirodalmát áttekintve három irányt különböztet meg aszerint, hogy a hatékonyságot (*efficiency-driven*), az eredményességet (*effectiveness-driven*) vagy a megértést (*understanding-driven*) szolgálja-e az értékelés [42]. Ez utóbbi kategória egyébként lényegében az interpretatív megközelítésnek felel meg.

Valójában az éles határvonal nem annyira az elsődleges célokban vagy az alkalmazott módszertanban van, hanem inkább a mögöttes filozófiai szemléletben mutatkozik meg. A Serafeimidis által említett technikai és pénzügyi értékelés, valamint Song és Letch kategorizálásában a hatékonyság-vezérelt és eredményesség-vezérelt értékelés alapvetően egy *formális-racionális* megközelítés, a megértést célszó értékelés viszont alapvetően *interpretatív* megközelítés.



2. ábra: Az információrendszerek értékelése

Technikai/funkcionális szemléletű értékelés

A t*echnológiai/funkcionális* értékelés elsősorban a rendszer technológiai, illetve funkcionális működésére helyezi a hangsúlyt, középpontjában a *hatékonyság* és az *eredményesség* mérése áll. A publikációk jelentős része az értékelés során alkalmazható kritériumokkal, illetve ezek lehetséges kategorizálásával foglalkozik [32]. A legtöbb szerző DeLone és McLean IS Success modelljére hivatkozik, amely az információrendszerek sikerességének 6 dimenzióját ragadja meg. Ezek közé tartozik (i) a rendszer minősége, (ii) az információ minősége, (iii) az információ felhasználása, (iv) a felhasználók elégedettsége, (v) az egyénekre gyakorolt hatás, illetve (v) a szervezetre gyakorolt hatás [15], majd később (vi) a szolgáltatás minőségével egészítik ki modelljüket [16]. Sajnos kevés az empirikus tanulmány, amely

alátámasztja, hogy a jellemzők közül melyek azok, amelyek valóban relevánsak az üzleti érték szempontjából, illetve mi az, amit a felhasználók, menedzserek ténylegesen fontosnak tartanak. A technikai/funkcionális értékelésben megjelenik ugyan a kontextus, vagy a felhasználók elégedettsége is, azonban a formális-racionális megközelítés korlátai miatt mégsem foglalkozik kellő mélységben az értékelés ezekkel a tényezőkel.

Gazdasági/pénzügyi szemléletű értékelés

A *gazdasági/pénzügyi* értékelés elsősorban az IT beruházások megtérülését, jövedelmezőségét vizsgálja, olyan jól ismert módszerekre támaszkodik, mint a diszkontált cash flow (DCF), a nettó jelenérték (NPV), a belső megtérülési ráta (IRR), a befektetés megtérülésének mutatója (ROI), a menedzsment által hozzáadott érték (ROM), a költség-haszon elemzés (CBA), a tevékenység alapú költségszámítás (ABC), vagy a gazdasági hozzáadott érték (EVA). A módszerek közös jellemzője, hogy a hozamok és a költségek szembeállításával próbálják meghatározni az informatikai beruházások értékét.

A pénzügyi módszerekről több átfogó tanulmány is készült, melyekben több mint 100 különböző módszert számoltak össze, bár ezek között vannak olyanok, amelyek csak kismértékben térnek el egymástól [34] [43]. A leggyakrabban használt módszerek nagyjából három-négy csoportba sorolhatók. (i) A *hagyományos pénzügyi módszerek* alapvetően a költség-haszon elv alapján értékelik a projekteket, a bejövő és a kimenő pénzáramokat állítják szembe egymással (DCF), a legfontosabb szempontok a megtérülési idő, a belső megtérülési ráta (IRR), valamint a nettó jelenérték (NPV). (ii) A *többféle kritériumon alapuló értékelési módszerek* különböző kvantitatív és kvalitatív módszereket használnak, és rendszerint egy score értéket adnak eredményül. Ezek a módszerek a pénzügyi hatások mellett olyan értékelési szempontokat is figyelembe vesznek, amelyeket már nem olyan könnyű pénzben kifejezni. (iii) A *rátákon alapuló módszerek* általában a hatékonyságot, eredményességet mérik, különböző mutatószámokat használnak összehasonlításra, vagy benchmarkként (például ROI, ROM). (iv) Végül a *portfolió módszereknél* a hosszú távú stratégiai szemlélet is megjelenik. A menedzsment irodalomból származnak, különböző kritériumok szerint csoportosítják az IT beruházási projekteket.

A hagyományos *formális-racionális* megközelítés beruházási projektként tekint az információrendszerekre, s bár különbséget tesz pénzügyi és nem pénzügyi hatások között [34], alapvetően pénzügyi módszereket használ az értékelés során, így figyelmen kívül hagyja a nem pénzügyi hatások jelentős részét. A legtöbb kritika éppen azért éri a hagyományos pénzügyi módszereket, mert nem tudják megragadni a pénzben nehezen kifejezhető üzleti hasznok jelentős részét, illetve az IT projektek rejtett költségeit, valamint elsősorban magára a fizikai értelemben vett rendszerre koncentrálnak, és nem veszik figyelembe kellő mértékben sem a működési környezetet, sem a rendszerekkel kapcsolatban álló egyéneket.

Az IR-értékelés interpretatív megközelítése

Részben az említett hiányosságok, részben az értékelés céljának, illetve fókuszának változása miatt többen is egyfajta szemléletváltás sürgetnek az információrendszerek értékelésében, és az *interpretatív* megközelítés szükségességét hangsúlyozzák. A hagyományos és az interpretatív megközelítés között nem az alkalmazott módszerekben van a lényeges különbség, hanem a mögöttes szemléletben, hogy miként gondolkozunk az információrendszerekről, illetve az információrendszerek értékelésének céljáról, szerepéről.

- Az interpretatív megközelítés az információrendszereket egy olyan társadalmi-technológiai entitásként kezeli, amelybe „beleágyazódik” az információ-technológia. A rendszerrel kapcsolatban álló egyének nem csupán felhasználók vagy üzemeltetők, hanem maguk is a rendszer részét képezik, akárcsak a technológia vagy a folyamatok. Minden rendszer valamilyen egyedi környezetben (kontextus) jelenik meg, amely hatással van a rendszer működésére, ugyanakkor a rendszer is befolyásolja a környezetét.
- A másik fontos különbség, hogy az értékelést egy iteratív tanulási folyamatként értelmezi, amelynek célja végső soron a döntéshozatali folyamatok támogatása, illetve a rendszer használatából származó üzleti hasznok minél jobb kiaknázása. Az értékelés célja, hogy megértsük

azokat a folyamatokat, amelyek az információ-rendszerben zajlanak, illetve azt a belső és külső kontextust (környezetet), amelyben az információrendszer működik. A végső cél a rendszer használatának fejlesztése, hogy ténylegesen realizálódjanak a működésből származó hasznok, illetve tudatosuljon az IT/IS szerepe az értékteremtésben.

Tudományfilozófiai háttér

Valójában az értékelés során alkalmazott módszerek kiválasztása csak másodlagos kérdés ahhoz képest, hogy milyen szemléletben, milyen paradigma alapján végezzük az értékelést [20]. Akárcsak a tudományos kutatásban, az értékelés területén is két uralkodó paradigma érvényesül: a (poszt-)pozitivista, illetve a konstruktív/interpretatív megközelítés [2] [9]. Filozófiai szempontból a hagyományos formális-racionális módszereket alkalmazó értékelésre a pozitivista megközelítés, az interpretatív értékelésre pedig a konstruktív/interpretatív szemlélet jellemző.

	FORMÁLIS-RACIONÁLIS MEGKÖZELÍTÉS	INTERPRETATÍV MEGKÖZELÍTÉS
PARADIGMA	pozitivista	konstruktív / interpretatív
FILOZÓFIAI HÁTTÉR	realizmus, idealizmus, kritikai realizmus	hermeneutika, phenomenológia
EPISZTEMOLÓGIA - a megszerzett ismeret - az értékelés fókusza - mi 'vezeti' az éretékelőt	objektív ismeret ami általánosítható és absztrakt hipotézisek, elméletek	a rendszer működésének megértése ami specifikus és konkrét a kontextust próbálja megérteni
ONTOLÓGIA - az értékelő - hogyan vizsgálja a rendszert - a rendszer környezete	objektív önmagában alapvetően a fizikai rendszer értékelésével foglalkozik	szubjektív a használat során a környezet és a felhasználók is a rendszer részét képezik
AXIOLÓGIA - az igazság - mit várunk az értékelőtől	verifikálható, megfigyelésen és mérésen alapul független, objektív	kontextus-függő része az értékelésnek
METHODOLÓGIA - adatgyűjtés módja	mérés, kérdőv, teszt	kontextusba helyezett megkérdezés közvetlen megfigyelés résztvevő megfigyelés dokumentumok elemzése esettanulmányok

2. ábra: Az információrendszerek értékelésének filozófiai háttere

❖ Az IR-értékelés interpretatív megközelítése

A pozitivista megközelítés a rendszert önmagában vizsgálja, nem fordít kellő figyelmet a környezetre, illetve a rendszerrel kapcsolatban álló egyénekre. Az értékelő objektív, az értékelés főleg számokon alapul, ha a beruházás értékelése a cél, akkor pénzügyi mutatókat, ha pedig a rendszer teljesítményét kell értékelni, akkor különböző egyéb mérőszámokat alkalmaznak.

Az interpretatív megközelítés szerint az információrendszerek összetettek, ráadásul gyakran más rendszerektől is függenek, ami még nehezebb feladattá teszi az értékelést. Azok az információk, ismeretek, amelyeket az értékelés során nyerünk, nem objektíven kerülnek meghatározásra, hanem szubjektíven, ráadásul több különböző "forrásból" származnak, mivel többen vesznek részt az értékelésben. Az értékelés folyamata kevésbé merev, sokkal rugalmasabb, lehetőség van menet közben új irányelvek megfogalmazására is, ami egy pozitivista megközelítésben elképzelhetetlen lenne. Az interpretatív megközelítés nagy hangsúlyt helyez az emberekre (akik a rendszer üzemeltetői, felhasználói), fontosnak tartja szubjektív tapasztalataik, véleményük megismerését. Az értékelés végső soron a rendszer működésének megértését szolgálja.

Társadalomtudományi háttér

Az interpretatív értékelés olyan tudományos elméletekre támaszkodik, mint a technológia társadalmi felépítése, a cselekvőhálózat elmélet, a hermeneutikai hagyomány vagy a kritikai perspektívák elmélete.

- A *technológia társadalmi felépítése* (Social Construction of Technology, SCOT) a technológia és a társadalom viszonyát vizsgálja. A technológiai determinizmussal szemben az elmélet azt hangsúlyozza, hogy a technológiát alapvetően társadalmi folyamatok határozzák meg, a technológia használatát csak úgy lehet megérteni, ha megértjük, hogy a technológia hogyan ágyazódik bele a társadalmi kontextusba [5].
- A *cselekvő-hálózat elmélet* (Actor-Network-Theory, ANT) szerint a heterogén természetű (emberi és nem emberi) cselekvők különböző heterogén hálózatokat alakítanak ki [10] [27] [28]. A cél a cselekvő-hálózatok működésének bemutatása. Az elmélet alapgondolata, hogy a cselekvések nem vezethetők vissza az önálló cselekvők döntéseire, ezért nem a cselekvőket, hanem a cselekvések által meghatározott hálózatot és annak heterogén összetevőit kell vizsgálni [46].
- A *hermeneutikai hagyomány* (hermeneutic tradition) alapvetően szövegek értelmezésével foglalkozik, három fontos alapfogalma a megértés, az értelmezés és az alkalmazás [14] [20].
- Végül a *kritikai perspektívák elmélete* (critical perspective) szintén nagy hatást gyakorolt az interpretatív megközelítésre. Ennek lényege, hogy a dolgokat különböző szemszögből (különböző perspektívákból) vizsgálja, és ahelyett, hogy egyetlen nézőpontot tekintene helyesnek, igyekszik minden nézőpontban megtalálni az értékes gondolatot. Az értékelőnek nem kell azonosulnia az egyes nézőpontokkal, és a különböző nézőpontokat se feltétlenül kell kibékíteni egymással, hanem minden egyes nézőpontot külön kell vizsgálni [25].

Az elméletek közös vonása, hogy a kontextus értékelése során nem elegendő technológiai/gazdasági szempontok érvényesítése, szociális, társadalmi tényezőket is figyelembe kell venni

Az interpretatív megközelítés jellemzői

- Az interpretatív szó jelentése „értelmező" vagy „magyarázó", ami jól kifejezi a megközelítés lényegét. Az értékelés célja kettős: értelmezni, megérteni a rendszer működését az adott kontextusban, illetve elmagyarázni az érintetteknek. Az értékelés elsősorban a rendszer használatának, illetve a szervezeti tanulási folyamatnak a fejlesztésére szolgál annak érdekében, hogy elősegítse az előnyök realizálását.
- Az értékelés célja, megérteni azt a kontextust, amelyben az IT/IS működik, illetve megérteni azokat a folyamatokat, amelyeken keresztül a rendszer befolyásolja környezetét, illetve a környezet a rendszert [47].
- Az „értékelő" feladata, hogy megszervezze és támogassa az értékelési folyamatot, tervszerűen meghatározza a stakeholderek számára az értékelés szempontjait, illetve, hogy felhívja a

figyelmet azokra a tényezőkre, amelyek az információrendszerek értékelésével foglalkozó szakirodalom szerint szignifikánsak, de a résztvevők hajlamosak figyelmen kívül hagyni.

- Az értékelő mellett a rendszerrel kapcsolatban álló stakeholderek (megrendelők, felhasználók, érdekeltek) is aktív résztvevői az értékelésnek. Az értékelőnek tisztában kell lennie azzal, hogy az egyes stakeholderek milyen szerepet töltenek be az értékelésben, és milyen kapcsolatban állnak a rendszerrel. A stakeholderek bevonásának két fontos következménye van: egyrészt mivel az információk különböző forrásokból származnak, az értékelésbe szubjektív elemek is kerülnek, másrészt az értékelés során különböző nézőpontokból kell vizsgálni az adott kérdést, így előfordulnak egymással ellentétes vélemények, és ezeket nem feltétlenül kell kibékíteni egymással.
- A megközelítés egyaránt alkalmaz kvantitatív és kvalitatív módszereket, illetve kifejezetten interpretatív módszereket (kontextusba helyezett megkérdezés, résztvevő értékelés, kooperatív értékelés, dokumentumok elemzése, esettanulmányok), de önmagában nem ezek alkalmazása érdekes, hanem a szemlélet, a mögöttes filozófia.
- Az adatgyűjtés és a feldolgozás egy szimultán és iteratív folyamat. Ez azt jelenti, hogy az elemző lefolytat egy interjút, majd gyakran ki is értékeli, mielőtt rátérne a következőre. Az interpretatív megközelítésbe az is belefér, hogy menet közben változtatunk a kérdéseken, ami egy pozitivista szemléletű értékelés során elképzelhetetlen lenne [3].
- A kívánt eredmény elérése érdekében világosan kommunikálni kell a szervezeti / üzleti célokat. Az interpretatív megközelítésben fontos szerepet kapnak a narratívák is, amelyek használata kétoldalú: egyrészt inputként szolgálnak az értékelés során, másrészt outputként is használhatók, amennyiben a megértést és a szervezeti tanulást szolgálják [24].

Tartalom-Kontextus-Folyamat keretrendszer

Többen is felvetették, hogy egyfajta paradigmaváltásra van szükség az értékelésben, el kellene mozdulni az interpretatív megközelítés irányába, ez a változás azonban nehezen megy végbe. Léteznek átfogó keretrendszerek (pééldául: Farbey Structure in Fives, Seddon IS Effectiveness Matrix, DeLone McLean IS Success Modell, Balanced Scorecard, Levy Analytical Framework, Farbey Benefits Evaluation Ladder), amelyek elvileg rugalmasan alakíthatók, így bármilyen értékelésre jól használhatók [12], Ennek ellenére a gyakorlatban kevés példát találunk olyan értékelési stratégiákra, amely az információrendszereknek ezt a szélesebb kontextusát is átfogja, és az interpretatív szemléletnek is teret ad. A legnagyobb kihívást egy olyan keretrendszer kialakítása jelenti, amelyek kellően általános ahhoz, hogy minél több rendszer értékelésére használható legyen, ugyanakkor kellően részletezett, hogy világos útmutatást adjon az értékeléshez [44].

Az információrendszerek értékelésével foglalkozó szakirodalom gyakran hivatkozik a Tartalom-Kontextus-Folyamat keretrendszerre, melyet eredetileg Pettigrew dolgozott ki a szervezeti tanulmányozására [30]. Információrendszerekre először Symons alkalmazta, majd később Stockdale és Standing fejlesztette tovább a modellt [44] [45]. A keretrendszer nevében szereplő három dimenzió olyan kérdéseket fogalmaz meg, amelyek támpontot adnak az értékelés megtervezéséhez:

- MIT értékelünk,
- MIÉRT, KINEK, és MILYEN CÉLLAL készül az értékelés, végül
- MIKOR végezzük az értékelést és konkrétan HOGYAN történik a megvalósítás.

Mivel egy keretrendszerről van szó, nem ad támpontot az elemzés részleteinek kidolgozásához, ugyanakkor rugalmasan használható mindkét megközelítésben. Az alábbiakban arra használjuk a keretrendszert, hogy összehasonlítsuk a formális-racionális és az interpretatív megközelítést, és áttekintsük a lényeges különbségeket (4. ábra).

❖ Az IR-értékelés interpretatív megközelítése

TARTALOM-FOLYAMAT-KONTEXTUS KERETRENDSZER		FORMÁLIS-RACIONÁLIS MEGKÖZELÍTÉS		INTERPRETATÍV MEGKÖZELÍTÉS
		tecnológiai / funkcionális szempontok	**gazdasági/pénzügyi szempontok**	
Az értékelés TARTALMA	MIT értékelkünk?	elsősorban magát a rendszert	elsősorban magát a rendszert	a rendszert használat közben
	MIT mérünk?	hatékonyság eredményesség	jövedelmezőség megtérülés	a stakeholderek határozzák meg, kontextustól függ
Az értékelés KONTEXTUSA	KI vesz részt?	menedzsment, IT, néhány felhasználó, stb.	menedzsment, IT, pénzügy, néhány felhasználó, stb.	sokkal szélesebb a stakeholderek köre, be vannak vonva az értékelésbe
	MIÉRT értékelünk?	a cél a hatékonyság, eredményesség mérése, illetve annak értékelése, hogy a rendszer mennyire felel meg az elvárásoknak	a cél rendszerint a beruházás megvalósíthatóságának értékelése	a cél a rendszer működésének megértése, illetve a rendszer használatának fejlesztése
Az értékelés FOLYAMATA	MIKOR értékelünk?	ex ante, ex post, a projekt életciklusához igazodva	ex ante, ex post, a projekt életciklusához igazodva	folyamatos tevékenység, a rendszer életciklusához igazodva
	HOGYAN értékelünk?	célok / elvárások / kritériumok alapján	különböző gazdasági / pénzügyi módszerek alkalmazásával	kontextustól függően interpretatív módszerek alkalmazásával

3. ábra: A két megközelítés összehasonlítása a Tartalom-Kontextus-Folyamat keretrendszerben

Az értékelés tartalma

MIT értékelünk?

Az első legfontosabb kérdés annak tisztázása, hogy pontosan MIT is értékelünk, ami a gyakorlatban legalább két dolgot jelent: (i) hogy az információrendszert önmagában értékeljük-e, vagy pedig a használat során [11] [13], (ii) illetve, hogy mit szeretnénk mérni az értékelési folyamatban, azaz miről szeretnénk információkat gyűjteni.

Amikor egy információrendszert önmagában értékelünk, akkor az értékelésből kihagyjuk a felhasználókat, csak maga a rendszer, illetve az értékelő vesz részt a folyamatban. Az értékelés elsősorban az információrendszerre, illetve a meglevő dokumentációkra támaszkodik. Az eredmény elsősorban azon múlik, hogy az értékelő mennyire érti meg az információrendszerműködését, illetve hogy ez miként szolgálja a szervezet működését [11] [13].

A másik lehetőség, hogy a rendszert a használat közben értékeljük, olyan szituációkat vizsgálunk, melyben a felhasználók kapcsolatba lépnek a rendszerrel. Ez sokkal komplexebb megközelítés, ugyanakkor lényegesen több információt nyújt a rendszer működéséről. A meglevő dokumentációk mellett az adatgyűjtésre olyan módszerek is felhasználhatók, mint a felhasználókkal folytatott interjú, vagy a felhasználók megfigyelése a rendszer használata közben. Értelemszerűen ez a megközelítés nagyobb teret ad a szubjektív értékelésnek, ami nehezebben mérhető. A végeredmény egyrészt az értékelő megértésén múlik, másrészt azon, hogy a stakeholderek miként értékelik a rendszert, mennyire érzik úgy, hogy az segíti a munkájukat [11] [13].

Az értékelés során egyrészt az üzleti célokat kell figyelembe venni, illetve hogy az információrendszermely tulajdonságai támogatják ezeket a célokat, másrészt a megvalósítás dokumentumait (követelmény specifikáció, költség-haszon elemzés), a változások folyamatait, a konfliktusok menedzselését. Fontos megérteni, hogy MIT mérünk, a stakeholderek bevonásával kell kiválasztani a kritériumokat, velük egyetértésben kell eldönteni, hogy mi az, ami helyet kaphat az értékelésben, és mi az, ami nem. A hagyományos, alapvetően a költségeket számszerűsítő értékeléstől el kell mozdulni olyan módszerek irányába, amelyek lehetővé teszik, a nem anyagi, immateriális hasznok mérését is, valamint az információrendszerben rejlő kockázatok és a lehetőségék feltérképezését [44]. Természetesen ez nem jelenti azt, hogy a korábbi módszerek haszontalanok lennének, de mindenképpen a szemlélet átformálásra és az eszköztár bővítésére van szükség.

Az értékelés kontextusa

KINEK, MIÉRT, MILYEN CÉLLAL készül az értékelés?

A kontextus foglal magában minden olyan külső és belső tényezőt, amelyek befolyásolják az értékelést [31] [37] [38]. Ez egyaránt jelenti a társadalmi-, politikai-, gazdasági- vagy versenykörnyezetet, a szervezeti struktúrát, illetve a vállalati kultúrát. A kontextus kapcsán a legfontosabb kérdés, hogy MIÉRT készül az értékelés, KINEK a részére készül, illetve végső soron MI A CÉLJA az értékelésnek.

A *belső kontextus* maga a vizsgált szervezet. Ez lehet önálló vállalat, egy folyamat, egy nagyobb szervezeti egység, vagy más alrendszer, amelynek a határait egyértelműen meg lehet határozni. Minden szervezetben van egy „története" az információrendszerek fejlődésének, amiből sokat lehet tanulni, de ebből láthatjuk azokat a hiányosságokat is, amelyek megszüntetése cél lehet egy új rendszer számára. A belső kontextushoz tartozik az infrastruktúra, mindazok az erőforrások, amelyek biztosítják a rendszer működését, valamint az információs folyamatok és az információáramlás útjai. A *rendszer kontextus* a vizsgált szervezet közvetlen környezete, az a rendszer, amelynek maga is részét képezi. Az értékelt szervezet nem tudja teljes mértékben irányítani közvetlen környezetét, de hatással van rá, tudja befolyásolni, és a környezet is befolyásolja a vizsgált szervezetet. Végül a *külső kontextushoz* tartozik mindaz, amit a szervezet csak kismértékben, vagy egyáltalán nem tud kontrollálni [36].

Értékelés célja általában: egy konkrét érték meghatározása, a siker mérése, vagy hozamok/hasznok számszerűsítése, figyelembevétele (Guba and Lincoln, 1998, House 1980).

Az értékelés folyamata

MIKOR és HOGYAN végezzük az értékelést?

A FOLYAMAT egyfajta összekötő híd szerepet tölt be a TARTALOM és a KONTEXTUS között. Az értékelés folyamatával kapcsolatban két lényeges kérdés merül fel: hogy MIKOR történik az értékelés, ami az információrendszerek esetében azt jelenti, hogy már a fejlesztési folyamat során, vagy csak az implementálást követően történik-e az értékelést, illetve hogy HOGYAN végezzük az értékelést, azaz konkrétan milyen elveket, módszereket alkalmazunk az értékelés során. Az interpretatív értékelés inkább egy folyamatos tevékenység, amelynek fókusza a kontextustól függően változik ugyan, de végső soron mindig az üzleti érték van a középpontban. Az értékelés folyamata magában foglalja a stakeholderek (menedzserek, IT szakértők, felhasználók stb.) akcióit, reakcióit és interakcióit. Be kell vonni a folyamatba a stakeholdereket, el kell érni, hogy érdekeltek és elkötelezettek legyenek, illetve biztosítani kell számukra az adatokhoz való hozzáférést. Az értékelés során biztosítani kell a stakeholderek különböző csoportjai közötti kommunikáció lehetőségét, ami lehetővé teszi, hogy minden csoport adjon visszajelzéseket.

Összegzés

Az információrendszerek értékelése bonyolult és összetett folyamat, ami abból is látszik, hogy a témával foglalkozó szakirodalom mennyire szerteágazó. Az értékelésre javasolt modellek többsége hagyományos formális-racionális megközelítést alkalmaz, de emellett létezik egy másik irány is, amely az interpretatív megközelítés fontosságát és szükségességét hangsúlyozza. Az interpretatív módszereket régóta használják a különböző társadalomtudományok, önmagában nem ezeknek a módszereknek az átvétele az érdekes, hanem egy újfajta szemlélet meghonosítása, amely másként gondolkodik az információrendszerről, illetve az információrendszerek értékeléséről. Az interpretatív megközelítés az információrendszert egy bonyolult társadalmi-technológiai entitásként értelmezi. Az értékelés elsődleges célja nem egy konkrét érték meghatározása, hanem a működés megértése, illetve a rendszerben zajló folyamatok feltérképezése, ami hozzájárulhat ahhoz, hogy jobban megértsük az üzleti érték keletkezését, illetve hosszú távon még jobban ki tudjuk használni a rendszerben levő lehetőségeket.

Hivatkozások

[1] Aranyossy, M. (2011). Az információtechnológia üzleti értékének nyomában. Hitelintézeti Szemle, 10 6, pp. 554-574.

[2] Babaheidari, S. M. (2007). Reviewing Interpretive Approaches for Evaluation of Information Systems Investments. Master Thesis in Informatics, IT University of Gotheburg, Sweden

[3] Bhattacherjee A. (2012). Social Science Research: Principles, Methods and Practices. 2nd Edition,

University of South Florida, Tampa, USA, 2012.
[4] Bakis, N., Kagioglou, M., Aouad, G. (2006). Evaluating the Business Benefits of Information Systems. In: 3rd International SCRI Symposium on Salford Centre for Research and Innovation (SCRI), University of Salford, Salford, pp. 280-294.
[5] Bijker, W. E., Hughes, T. P., Pinch, T. eds. (1987). The Social Constructions of Technological Systems. Cambridge, Massachusetts: MIT Press
[6] Bőgel, Gy. (2009). Üzleti elvárások – Informatikai megoldások. Budapest: HVG Kiadó
[7] Brown A. (1994). Appraising intangible benefits from Information Technology Investment, in: Proceedings of the First European Conference on IT Investment Evaluation, Henley, England, September, pp. 187-199.
[8] Brynjolfsson, E. (1993). The productivity paradox of Information Technology. Communications of the ACM, 36 12, pp. 67-77.
[9] Calidoni-Lundberg, F. (2006). Evaluation: definitions, methods and models. An ITPS framework. Working Paper: R2006:002, ITPS, Swedish Institute For Growth Policy Studies
[10] Callon, M. (1986). The Sociology of an Actor-Network: The Case of the Electric Vehicle. in: Callon, M., Law, J. & Rip, A. (Eds) Mapping the Dynamics of Science and Technology, Macmillan Press,19-34
[11] Chen, S., Osman, N. M., Nunes, J. M. B., Peng, G. C. (2011). Information system evaluation methodologies. in: Proceedings of the IADIS International Workshop on Information Systems Research Trends, Approaches and Methodologies (ISRTAM), 20 July (2011), Rome, Italy
[12] Costello, P., Sloane, Moreton, R. (2007). IT Evaluation Frameworks – Do They Make a Valuable Contribution? A Critique of Some of the Classic Models for use by SMSs. The Electronic Journal Information System Evaluation, 10 1, pp. 57-64.
[13] Cronholm, S., Goldkuhl, G. (2003). Strategies for Information Systems Evaliuation: Six Generic Types, in: Proceedings of the Tenth European conference on information technology evaluation, Madrid, Spain, pp. 65-74.
[14] Davis, G.B., Lee, A.S., Nickles, K.R., Chaterjee, S., Hartung, R., Wu, Y. (1992). Diagnosis of an information system failure; a framework and interpretive process. Information Management, 23 5, pp. 293-318.
[15] De Lone, W. H., McLean, E. R. (1992). Information Systems Success, The Quest for the Dependent Variable. Information Systems Research, pp. 60-95.
[16] DeLone, W. H. , McLean, E. R. (2002). Information Systems Success Revisited. in: Proceedings of the 35th Hawaii International Conference on System Sciences
[17] Erdős, F. (2012). Different Techniques to Quantify the Yield of IT Projects. SEFBIS Journal, 7, pp. 11-17.
[18] Farbey, B., Land, F., Targett, D. (1993). IT investment: A study of methods and practices, UK: Published in association with Management Today and Butterworth-Heinemann Ltd.
[19] Farbey, B., Land, F., and Targett, D., (1995). A taxonomy of information systems application: The benefits' evaluation ladder. European Journal of Information Systems, 4 4, pp. 41-50.
[20] Guba, E. G., Lincoln, Y. S. (1996). Competing Paradigms in Qualitative Research. in: Denzin, Lincoln (Eds.), Handbook of Qualitative Research, USA, Sage Publishers
[21] Gustafsson, P., Franke, U., Johnson, P., Lilliesköld, J. (2008). Identifying IT impacts on organizational structure and business value. in: Proceedings of the Third International Workshop on Business/IT Alignment and Interoperability, pp. 44-57.
[22] Hallikainen, P., Nurakämi, K. (2000). Post-Implementation Evaluation of Information Systems. Initial Findings and Suggestions for Future Research. in: Svensson, L., Snis, U., Sørensen, C., Fägerlind, H., Lindroth, T., Magnusson, M., Östlund, C. (eds.) Proceedings of IRIS 23, Laboratorium for Interaction Technology, University of Trollhättan Uddevalla
[23] Hallikainen, P., Chen, L. A. (2005). A Holistic Framework on Information Systems Evaluation with a Case Analysis. The Electronic Journal Information Systems Evaluation, 9 2, pp. 57-64.
[24] Hedman, J., Borell, A. (2005): Broadening Information Systems Evaluation Through Narratives, The Electronic Journal of Information Systems Evaluation, 8 2, pp. 115-122.
[25] Klecun, E. and Cornford, T. (2005). A Critical Approach to Evaluation. European Journal of Information Systems, 14, pp. 229–243.
[26] Kusters, R. J., and Renkema, T. J. W., (1996). Managing IT investment decisions in their organisational context: The design of 'local for local' evaluation models. in: A. Brown and D. Remenyi (eds.), Proceedings of the 3th European Conference on IT Evaluation, University of Bath, 133-141.
[27] Latour, B. (1987). Science in Action: How to Follow Engineers and Scientists Through Society. Open University Press, Milton Keynes
[28] Law, J. (1992). Notes on the Theory of the Actor-Network: Ordering, Strategy and Heterogeneity. Systems Practice, 5 4, pp. 379-393.

[29] Lederer, A., Mirani, R., Neo, B., Pollard, C., Prasad, J., Ramamurthy, K. (1990). Information System Cost Estimating: A management Perspective. MIS Quarterly, 14 2, pp. 159-176.
[30] Pettigrew, A. (1985). The awakening giant, continuity and change in ICI. Oxford, Blackwell.
[31] Piotrowicz, W., Irani, Z. (2008). Information system evaluation in context-inpact of the corporate level. European and Mediterranean Conference on Information Systems 2008, May 25-26 (2008), Al Bustan Rotana Hotel, Dubai
[32] Platiša, G., Balaban, N. (2009). Methodological Approaches to Evaluation in Information System Functionality Performances and Importance of Successfulness Factor Analysis, Management Information Systems, 4 2, 11-17.
[33] Remenyi, D., Money, A., Sherwood-Smith, M. (2000). The Effective Measurement and Management of IT Costs and Benefits, 2nd Edition, Oxford: Butterwoth-Heinemann
[34] Renkema, T. J. W., Berghout, E. W. (1997). Methodolgies for information system investment evaluation at the proposal stage: a comparative review. Information and Software Technology, 39 1, pp. 1-13.
[35] Renkema, T. (2000). The IT Value Quest. Chichester, John Wiley and Sons
[36] Serafeimidis, V., Smithson, S. (1996). Understanding and Supporting Information Systems Evaluation. Journal of Computing and Information Technology, CIT, 4 2, pp. 121-136.
[37] Serafeimidis, V., Smithson, S. (1999). Rethinking the approaches to information systems evaluation. Logistics Information Management, 12 1/2, pp. 94-107.
[38] Serafeimidis, V., Smithson, S. (2000). Information systems evaluation in practice: a case study of organizational change. Journal of Information Technology, 15 2, pp. 93-105.
[39] Serafeimidis, V. (2002). A Review Of Research Issues In Evaluation Of Information Systems. In: Grembergen, W. V. (Ed.), Information Systems Evaluation Management, London: IRM Press
[40] Shang, S., and Seddon, P.B., (2002). Assessing and managing the benefits of enterprise systems: the business manager's perspective. Information Systems Journal, 12 4, pp. 271-299.
[41] Smithson, S., Hirschheim, R. (1998). Analysing Information Systems Evaluation: Another Look at an Old Problem. European Journal of Information Systems, 7, pp. 158-174.
[42] Song, X., Letch, N. (2012). Research on IT/IS Evaluation: A 25 Year Review. Electronic Journal of Information Systems Evaluation, 15 3, pp. 276-287.
[43] Stix, V., Reiner, J. (2004). IT Appraisal Methods and Methodologies – A Critical Literature Review. Proceedings of the Information Resources Management Association (IRMA). pp. 37-44.
[44] Stockdale, R., Standing, C. (2006). An interpretative approach to evaluating information systems: A content, context, process framework. European Journal of Operational Research, 173, pp. 1090-1102.
[45] Symons, V. J. (1991). A review of information systems evaluation: Content, context and process. European Journal of Information Systems, 1 3, pp. 205-212. DOI: 10.1057/ejis.1991.35
[46] Szabari, V. (2007). A társulások szociológiája. Bruno Latour: Reassembling the Social An Introduction to Actor-Network-Theory. Oxford: Oxford University Press, 2005. Szociológiai Szemle, 1-2, pp. 109-118.
[47] Walsham, G. (1993). Interpreting Information Systems in Organizations. Chichester: Wiley
[48] Ward, J., Taylor, P., and Bond, P., (1996). Evaluation and realisation of IS-IT benefits: An empirical study of current practice. European Journal of Information Systems, 4 4, pp. 214-225.
[49] Ward, J., and Daniel, E., (2006). Benefits management: Delivering value from IS & IT investments. John Wiley & Sons Ltd., Chichester, ISBN: 978-0-470-09463-1

Nagy Gábor Szabolcs a Janus Pannonius Tudományegyetem Közgazdaságtudományi Karán szerzett közgazdász diplomát 1999-ben monetáris szabályozás és vállalatfinanszírozás szakirányokon. A kezdeti évek útkeresését követően érdeklődése egyre inkább az üzleti informatika felé fordult. Eleinte önállóan képezte magát, később a Hageni Távegyetem (Fernuniversität in Hagen) üzleti informatika képzésén szerzett BsC diplomát. PhD tanulmányait a Pécsi Tudományegyetem Gazdálkodástani Doktori Iskolájában folytatta, abszolutóriumát 2017-ben szerezte. Kutatási területe az információ értéke, az információrendszerek és az információs vagyon értékelésének módszerei. Az elmúlt években az NN Biztosító és a CIB Bank munkatársaként dolgozott CRM területen, jelenleg a HBO Europe üzleti intelligencia osztályán hasznosítja megszerzett tapasztalatait.

A kibertérből érkező fenyegetések elleni védekezés vállalati környezetben

NÉMETH RICHÁRD
Széchenyi István Egyetem
eMail: nemeth.richie@gmail.com

ABSZTRAKT

Az USDoD definíciója szerint a kibertér egy „informatikai (információs) környezetben értelmezett globális tartomány (domain), amely magába foglalja az IT infrastruktúrák egymással összefüggő elemeinek hálózatát, beleértve az Internetet, a telekommunikációs hálózatokat, számítógépes rendszereket, valamint a beágyazott feldolgozó és vezérlő elemeket. A kibertér a közös játék, szórakozás, kapcsolattartás és munka színtere, elsődleges funkciója azonban az információk cseréje, amely e virtuális térben hihetetlen sebességgel zajlik. A fejlődés felgyorsulásával szinte minden munkavállaló kapcsolatba kerül adatokkal és számítógépekkel. A legtöbb támadás az információrendszerekben gondatlanságból vagy hozzá nem értésből megmaradt sebezhetőségeket használ ki. Napjainkban már nem csak a nagyvállalatok, de kis- és középvállalkozások is célpontok lehetnek. A kibertámadók eszközök rendkívül széles tárházát veszik igénybe a digitális kártevőktől a technológiai megoldásokig, de a vállalat szempontjából a legnagyobb veszélyt az emberi tényező képviseli. A fenyegetések összetett volta miatt a kibervédelemnek is jól kiépítettnek kell lennie. Írásom célja egy olyan védelmi modell bemutatása, amely képes az információrendszerek védelmének megalapozására, és a biztonság minden szintjére kiterjed.

Bevezetés

Kibertér, kibertámadás, kibervédelem, hackerek… ezek a kifejezések napjainkra nem csak az számítástechnikai terminológiába, de mindennapjaink szókincsébe is beépültek. Az első kibertámadás a ’80-as évek végén történt, egészen pontosan 1988-ban, okozója az ún. Morris-féreg volt, az első hacker pedig tulajdonképpen Alan Turing volt, aki a második világháború során megfejtette az Enigma kódját – bár ekkor még nem hackernek hívtuk azokat az informatikai szakembereket, akik számítógéprendszerek vagy – programok feltörésével foglalkoznak.

Maga a hacker szó a ’60-as években született; az MIT-n tanuló diákok „piszkáltak” bele vasúti modellek programjába, némi plusz teljesítményt kipréselve a modellekből – az eljárást „hackelésnek” nevezték, és művelői lettek a hackerek [1]. A talán leginkább ismert hacker, Kevin Mitnick nem a legelső, és talán nem is a legnagyobb tudású kiberbűnöző volt, de a róla szóló filmek, az általa írt könyvek, és persze a média érdeklődése okán talán ő az, akiről a legtöbben hallottunk. Manapság biztonsági tanácsadóként dolgozik, és több észrevételével, tanácsával is találkozhatunk jelen tanulmányban. Mitnick „játszótere” a számítógépek, a hálózatok és az Internet által megteremtett virtuális világ, a kibertér volt [9] [10]. De mit is jelent maga a szó, miért mondhatjuk, hogy létrejötte legalább akkora jelentőséggel bír, mint a tűz vagy a kerék felfedezése? Nos, a kibertér egy egyedi és összetett kifejezést takar, amit számtalan különböző módon megközelíthetünk, hiszen fizikai, geográfiai, virtuális és társadalmi karakterisztikákkal is rendelkezik – ugyanakkor hatással van éntudatunkra is; a kibertér, globális természete okán a hálózathoz kapcsolódás érzetét teremti meg.

Az egymással kapcsolatban álló számítógépeket összekötő Internet segítségével eddig nem tapasztalt mértékben nyílt lehetőségünk az adatcserére, online tartalmak megtekintésére, játékra, közös munkára, vagy éppen kapcsolattartásra e-mailek, chat- és videochat-alkalmazások, közösségi hálózatok révén. Ugyanakkor azt sem szabad elfelejtenünk, hogy eme virtuális tér nem csak lehetőségeket, de veszélyeket is rejt magában –l ehetőséget nyújt visszaélésekre, adatlopásra, az informatikai rendszerek, szoftverek és hálózatok illetéktelen használatára, manipulációjára vagy rongálására is.

A kibertér definíciója és jellemzői

Mint a világon oly sok innováció, az Internet létrejötte is katonai projektig vezethető vissza. 1968-ban jött létre az ARPANET, a hadsereg, a haditengerészet és a légierő közös gyermekeként, célja egy olyan csomagkapcsolt hálózat volt, melynek résztvevői decentralizáltan kapcsolódnak egymáshoz, esetleges támadás esetén is képesek maradnak a kommunikációra. Az ARPANET-ből született később az Internet (egyetemek és nagy techóriások fokozatos bekapcsolódásával), és végül ez vezetett a kibertér létrejöttéhez is, ami az Interneten kívül számos egyéb eszközt és környezetet foglal magába (például virtuális és kiterjesztett valóság, IoT, okoseszközök, navigáció, online streamelés stb.).

Vállalati és kormányzati szinten egy, az Internethez nagyon hasonló, azonban a külvilágtól elzárt belső hálózatot, ún. intranetet használnak. A technológiák fejlődésével egyre jelentősebbé vált a vizuális megjelenítések területének fejlődése. A mesterséges tér rapid fejlődésnek indult szeletét alkotják az úgynevezett virtuális és kiterjesztett valóság-alapú alkalmazások, melyek célja az érzékelés szimulációja a digitális térben, nagy hangsúlyt fektetve a vizuális környezet leképezésére, de léteznek már megoldások az érintés (pl. haptikus visszajelzés), a hallás szimulációjára, sőt a szaglás és ízlelés is megvalósítható (utóbbi kettő egyelőre kevésbé elterjedt). Napjainkra az okoseszközök körbevesznek minket, összekapcsolt hálózatuk az ún. IoT, azaz a tárgyak internete, amely elektronikai, kommunikációs és egyéb eszközök hálózati összekapcsolása az interaktivitást megvalósító hardverek, szoftverek, szenzorok és aktorok összehangolása révén – szintén hozzájárulva a kibertér bővüléséhez.

Az Amerikai Védelmi Minisztérium (USDoD) definíciója szerint a kibertér egy „informatikai (információs) környezetben értelmezett globális tartomány (domain), amely magába foglalja az IT infrastruktúrák egymással összefüggő elemeinek hálózatát, beleértve az Internetet, a telekommunikációs hálózatokat, számítógépes rendszereket, valamint a beágyazott feldolgozó és vezérlő elemeket" [7]. A tág értelemben vett kibertér fogalom az évek során számtalan új ágenssel bővült – a kibertér egyre átfogóbbá, összetettebbé és komplexebbé vált – azonban sérülékenységét éppen ez a nagyfokú komplexitás adja.

A téma ismert kutatója, Susanne C. Nielsen, a West Point katonai akadémia tisztje szerint a kibertér számos olyan biztonságkritikus jellemzővel bír, melynek révén kifejezetten veszélyes lehet az egyénre és társadalomra. Ezek a biztonságkritikus jellemzők a következők: (1) *ember készítette, dinamikus* mesterséges tér; (2) *gyors* – szinte minden esemény azonnal történik; (3) *nincsenek* geográfiai értelemben vett *határok*; (4) *könnyű belépni* és aktorrá válni; (5) *gyorsan növekszik*, így nehézkes a prevenció; továbbá (6) *gondolkodást, viselkedést formál* [12]. Eme tényekből egyértelműen levonhatunk jó néhány olyan következtetést, amely arra figyelmeztet bennünket, hogy a kibertér nyújtotta szolgáltatások tág spektrumának élvezetén túl ennek a világnak is megvannak a maga veszélyei.

A kibertér aktorai

A fejlődés felgyorsulásával szinte minden munkavállaló kapcsolatba kerül adatokkal és számítógépekkel. A legtöbb támadás az információrendszerekben gondatlanságból vagy hozzá nem értésből megmaradt sebezhetőségeket használ ki. E fenyegetés kivédése érdekében a hálózatépítők és programozók is sokat tehetnek a megfelelő előírások, óvintézkedések betartásával. Céges környezetben a rendszergazdák, üzemeltetők nagy felelősséggel bírnak, az ő feladatuk a loginek biztonságos kezelése, a szoftverek és a működtető környezet karbantartása. Emellett a biztonságért felelős személyek kategóriájába sorolhatók a fizikai bejutást biztosító személyzet tagjai is. Napjainkra a köztudat az összes kibertámadót – tévesen – hackerként tartja számon, függetlenül attól, hogy ártó vagy segítő szándékkal cselekednek. Képzettségük alapján a következő típusaikat különböztethetjük meg:

A *trollok* „az internet szociális vandáljai, akik megsemmisítve áldozataik online identitását, áldozataikat a halálba képesek üldözni" [2] – cselekedeteik a zaklatástól a bűncselekményre felbujtásig rendkívül változatosak lehetnek. A *csalók* (scammers) rendelkeznek IT alapokkal, de nem értenek a biztonsági rendszerekhez, így elsősorban adathalászattal és spam-ek küldésével foglalkoznak. A *szriptkölykök* elsősorban defacement támadásokat hajtanak végre.

A *„wannabe" hacker* „még nem valódi hacker, de arra törekszik, hogy azzá váljon" (…), így hát „más hackerek által kitalált úgynevezett hack-programokkal munkálkodnak" [8]. Az *újonc hackerek* (newbie-k) folyamatosan tanulnak, és már ez idő alatt is veszélyt jelentenek a gyengén védett számítógéprendszerekre. A hacker tool-ok megírásával *kódolók* foglalkoznak. Ők „jól ismerik a számítógépes rendszerek és programok belső működését" [5]. A vandál vagy *cracker* „olyan kárt okozó személy, aki számítógépes rendszereket rongál, illetve adatokat tulajdonít el, vagy bármilyen egyéb módon kárt okoz" [9]. Ők azok, akik a forgalomba kerülő alkalmazásokat, játékprogramokat feltörik. A *valódi hackerek* komoly veszélyt jelentenek, bonyolult rendszerekbe képesek betörni. Kiválóan eligazodnak a rendszerbiztonság területén, ismerik a lehetséges támadási formákat.

A legképzettebb hackerek területük igazi szakértői. A *varázslók* egy-egy területen kiemelkedő tudással rendelkeznek, és igazi nagymesternek számítanak. A *phone phreaker-ek* nagy része is varázsló; választott mesterségük szakavatott művelőiként a telefonhálózatok feltörése a szakterületük. A *guru* „mindent tud egy bizonyos dologról, beleértve a dokumentálatlan részeket is, és rájött, hogyan lehet a feltételezett limiten is túljutni. Ha ez a terület egy felhasználás, valószínűleg többet tud róla, mint az, aki kifejlesztette" [5]. A *HPAV (Hacking, Phreaking, Anarchy, Virus) csoportok* „a létező legkártékonyabbak – vírusokat írnak, állami szervek munkáját teszik tönkre, magánszámítógépekbe törnek be, mindezt csak azért, hogy másoknak gondot okozzanak" [1]. Ők a kibertér legveszélyesebb, legártalmasabb aktorai.

A *social engineer-ek* egy újabb kategória, mivel ők jellemzően csak kis mértékben használják a technológiát. A hackerek egy jól elkülönített csoportjánál cél a hálózatokba történő betörés, elsősorban információ- vagy pénzszerzés céljából. Ők a *fekete kalapos (black-hat) hackerek*. Közülük kerülnek ki a kiberkémek és információ-brókerek is. *A fehér kalapos (white hat) hackerek* célja a kiberbiztonsági hibák, hiányosságok felderítése, megelőzve az esetleges betöréseket. Az *etikus hackerek* a fehér kalapos hackerek közül kerülnek ki, ők – sokszor megbízásból – egy-egy rendszer biztonságát teszik próbára. A kettő közti átmenet ún. *grey-hat hacker*. Ők megkeresik a biztonsági réseket, behatolnak a rendszerekbe, de kárt nem okoznak, hanem értesítik a rendszer üzemeltetőjét, gyakran felajánlva segítségüket is. Operációs rendszer oldalról elsősorban a Microsoft Windows termékek a hackerek kedvelt célpontjai, az ún. *red hat hackerek* azonban kifejezetten a Linux-rendszerek támadására szakosodtak.

Kibertér-sebezhetőségek

A 2020-as évben pusztító útjára indult Covid-19 járvány az informatikai biztonság területén is kifejtette hatását – globális szinten jelentősen megnövekedett a kibertámadások száma. A Kaspersky felmérése szerint „az Európai Unióban az internetet használó számítógépek 13,7 százalékán tapasztaltak legalább egy böngészőalapú, rosszindulatú programtámadást", (és a támadások számát tekintve) „az első tíz között találjuk Magyarországot is." [17]. Nagyságrendileg ugyan az otthoni gépek vannak leginkább kitéve kémkedésnek, adatlopásoknak, rongálásnak és egyéb támadásoknak, de céges környezetben a statisztikák nem kevésbé lesújtóak. Az amerikai CSI egy korábbi felmérése szerint a válaszadók 85%-a észlelt már számítógépes betörési kísérleteket az adott naptári évben, sőt, 64% esetében ez anyagi veszteséget is jelentett. Mitnick máig érvényes megállapítása szerint tízből kilenc szervezet vált már sikeres betörési kísérlet áldozatává [9].

Egyértelműen kijelenthető, hogy a védelem és az okozott kár mértéke között egyenes arányosság van. Ugyanakkor a megfelelő szintű kibervédelem kifejlesztéséhez jelentős anyagi ráfordítás szükséges. Vállalati szinten komoly fejtörést okoz az egyensúly megteremtése. Egy szervezet kibervédelmi gyengeségeinek feltárásakor többféle megközelítést is figyelembe kell vennünk. „Az információs infrastruktúrák fenyegetései lehetnek humán, fizikai, logikai vagy épp a rendszer életciklusa során jelentkező kockázatok" [14], a támadás történhet fizikai, információs vagy tudati dimenzióban is. A lehetséges elkövetők spektruma is rendkívül tág tartományt ölel fel a felbérelt ipari kémektől a gondatlan alkalmazottig. „Ha egy támadó bejut a szervezet épületébe, nem csak bizalmas és belső információk birtokába juthat, hanem például rongálhatja eszközeinket, adatállományokat törölhet vagy megsemmisíthet, illetve belső és bizalmas információkat továbbíthat saját magának, átvizsgálhatja a szemetesünket, kártékony programokat telepíthet eszközeinkre" [3].

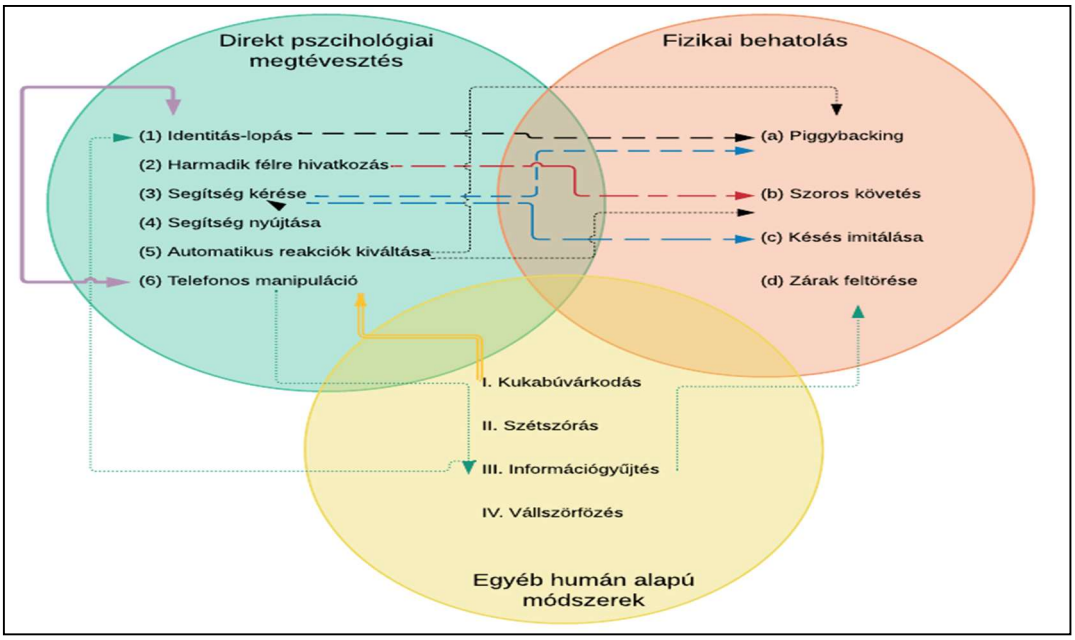
Social engineering támadások

IT biztonság szempontjából a fő támadási felületet az ember jelenti. Ezt felismerve a támadók gyakran élnek a *social engineering* nyújtotta lehetőségekkel. Ez „felettébb sarkalatos pontnak bizonyul a vagyonvédelem szempontjából. Különös tekintettel az őrzésvédelemre, és adat- és információvédelemre, továbbá az IT biztonságra" [1]. A pszichológiai manipuláció „a különféle belső és bizalmas információk megszerzését célzó támadások egyik válfaja, amely az emberi tényező befolyásolására, manipulálására, valamint kihasználható tulajdonságaira épít" [3]. A támadó célja olyan információk szerzése, melyek felhasználása lehetővé tesz egy későbbi, kibertérben megvalósuló támadást.

Az identitás-lopás, azaz megszemélyesítés a leggyakoribb eszköz. Ennél a trükknél a támadó valaki más személyiségét ölti magára; kézbesítőnek, karbantartónak, céges partnernek vagy jelenlegi (sőt néha elhunyt/felmondott [16]) dolgozónak adja ki magát. Ez esetben a szakzsargonok és a vállalat belső ismerete is alátámasztja a megtévesztést. A hatás fokozható azzal, ha a behatoló a felvett szerepnek megfelelő ruhába bújik.[13] A módszer úgy is működhet, hogy ismeretlenként mutatkozik be, de olyasvalakinek adja ki magát, aki egy kolléga, vagy éppen a főnök kérésének tesz eleget. „Ezt hívják »dobálózás a nevekkel« technikának, és a módszert arra használják, hogy gyorsan alakítsanak ki kapcsolatot azáltal, hogy hangsúlyozzák a célpontnak: a támadó kapcsolatban áll valakivel" [10]. A fentiek ellenpontjaként felfogható „reverse social engineering" bizalmat igyekszik kelteni az áldozatban. Ilyenkor a támadó egy fiktív problémára, biztonsági hiányosságra, új fenyegetésre hívja fel a figyelmet – és persze ő az, aki segíteni tud. „Érzelmi állapotunk vagy szellemi fáradtságunk könnyen eltereli a figyelmünket. Így gyorsan, az információ gondos és teljes elemzése nélkül hozunk döntést[10]." Az automatikus reakciók kiváltása bűntudatot vagy együttérzést kelt az áldozatban, de nem ritka a megfélemlítés használata sem.

A céges területre, létesítményekbe történő behatolásnál a legismertebb praktika a más jogosultságainak felhasználásával történő ún. piggybacking. Ilyen esetben „a támadó, aki nem jogosult a belépéshez kiadja magát egy olyan személynek, aki beléphet az adott helyre. Jellemzően az otthonmaradt kulcs, belépőkártya történetet játsszák el"[16]. Szintén gyakran használt trükk az ún. tailgating, a szoros követés is. A technika alkalmazásakor „a támadó úgy tesz, mintha egy vendég- vagy munkás csoport tagja lenne, majd hozzájuk csapódva egyszerűen besurran az épületbe." Nem ritka, hogy a támadó késést imitál – például azt színleli, hogy egy megbeszélésre siet – így sokszor elkerülheti az alaposabb vizsgálatot. Némileg kilóg a sorból a zárfeltörés, azaz lockpicking – itt kifejezetten a kapcsolatfelvétel elkerülése, a feltűnés nélküli behatolás a cél: a „lockpicking az említett hengerzárbetétek, lakatok, s egyéb mechanikai védelmi eszközök kulcs nélküli, roncsolásmentes nyitását jelenti. Elég gyakorlatigényes eljárás, melyet csupán készség szinten elsajátítva lehet hatásosan alkalmazni" [16].

Napjainkban a cégek már nem dobnak ki mindenféle iratot, adathordozót csak úgy a szemétbe, de így is meglepően sok információt szerezhetnek a támadók az ún. „dumpster diving", azaz kukabúvárkodás módszerével: a támadó a kidobott szemetet kutatja át információszerzés céljából. A kukabúvárkodás az ipari kémkedés egyik kedvelt trükkje [9]. A szétszórás (baiting) módszerének lényege, hogy a „támadó egy fertőzött adathordozót (pendrive, CD, DVD, SD kártya) »véletlenül« elhagy, és amikor a gyanútlan felhasználó megtalálja azt, csatlakoztatja a számítógépéhez az eszközt, hogy kiderítse, kié lehet, vagy mit tartalmaz az eszköz"[3]. Az adathordozóra telepített trójai pedig egyből megfertőzi a céges hálózatot. Az ember nem is sejtené, hogy a szinte mindenki által használt internetes oldalak is egy sajátos támadásforma eszközei. Emberi jellemvonás, hogy „szeretnénk közölni a külvilággal létezésünket (…). Fontos szempont, hogy nem csak saját magunk oszthatunk meg a világgal információkat, hanem tőlünk függetlenül, a mi akaratunkon kívül ezt más is megteheti. Sajnos ez nagy kockázati tényező, hiszen névtelenül, felelősség nélkül áramlanak az információk, melyeket nagyon könnyen ki tudnak használni a rosszakarók" [16]. A „shoulder surfing" (magyar megfelelője talán vállszörfözés lehet) technikát használva „a támadó úgy szerzi meg például a célszemély azonosítóját, jelszavát, esetleg pénzfelvételnél a PIN kódját, hogy egyszerűen csak átnéz a válla felett, miközben az áldozat begépeli azt" [3].



1. ábra Social Engineering támadások összefüggései (saját szerkesztés)

Megszámlálhatatlanul sok műszaki, technológiai és digitális kártevő alapú támadási forma létezik; az alábbiakban terjedelmi korlátok miatt csak a leggyakoribb módszerek kerülnek ismertetésre.

Műszaki/technológiai támadásfajták

Azzal nem árt tisztában lennünk, hogy sosem közvetlenül érünk el egy szervert, hanem általában tűzfalon keresztül. A támadás egyik kulcslépése a jelenlét elrejtése. Az álcázás elemi módja az ún. *ugrálás* (firewall-bouncing). Ilyenkor a hacker „több tűzfalon keresztül éri el a célgépet, mégpedig úgy, hogy először bejelentkezik az egyik tűzfalra, azután arról a másikra, és így tovább" [1]. A behatolás elrejtésében segít az úgynevezett *spoof technika*. Így ha sikerül is valahogy visszakövetni a támadót, a kapcsolat túloldalán valaki mást találnak. Szintén erre használják a Wingate gépeket.

A *phone phreaking* a hackelés egy olyan fajtája, ami a telefonrendszer vizsgálatával deríti fel a telefonos hálózatot, azonosítva a gyenge pontokat, a belső vonalakat, a csatlakoztatott eszközöket – ennek révén a támadók hívásokat irányítanak át, vagy lehallgatják mások hívásait. A *közbeékelődéses (MITM) támadás* esetében a cél szintén a rendszerek közötti adatkommunikáció lehallgatása. A támadó mindkét féllel elhiteti, hogy egymással kommunikálnak, de az üzeneteket ő is megkapja. Szintén ismert módja a lehallgatásnak az úgynevezett *Network Monitor* (sniffer) használata. „Egy sniffer program úgy működik, mint egy számítógép-lehallgató. A vezetéken keresztül küldött minden forgalmat titokban megszerez; a hacker általában egy olyan helyre küldi az adatokat, ahol valószínűtlen, hogy észrevennék" [10].

Az információrendszertámadásának egyik leggyakoribb módja az *exploit alapú betörés*. Az exploit „egy külső program, amely a kódban található programozási hibákat használja ki" [10]. Léteznek úgynevezett exploit-scanner segédprogramok, amik végigellenőrzik az operációs rendszert és a telepített gyakoribb alkalmazásokat, hiányzó frissítések – és ez által fennálló, be nem tömött lyukak – után kutatva. A webalapú támadások ismert módja az *URL hamisítás* – ilyenkor a támadók egy, az eredetivel látszólag megegyező weblapot hoznak létre, aminek még a címe is majdnem megegyezik. „Az oldalakon keresztül végrehajtott *közvetett szkripthívás* (XSS) során a támadók elérik, hogy egy kártékony szkript fusson le az áldozat gépén" [4]. Az *SQL befecskendezés* során a „hacker olyan speciális kódot, karakterláncot ad meg az online beviteli felületen (…), ami a szerveroldali feldolgozás során eltéríti az adatbázislekérést és az adatbázis olyan részeihez enged (akár módosítási) hozzáférést, amihez biztosított körülmények között amúgy nem lenne hozzáférése" [1].

A *Buffer Overflow (BoF) támadás*ok az űrlapok nem megfelelően validált beviteli mezőit veszik célba; több adatot tárolnak, mint amennyit a buffer fel képes dolgozni, és ezáltal adatvesztést, memória-hozzáférési hibát okoznak.

Ha a cél nem a bejutás, hanem a rombolás, arra is rengeteg eszköz áll a támadó rendelkezésére. A *levélbomba* esetén a célpont részére számtalan levelet küldenek, ezáltal megbénítva a levelezőprogramját. A módszer használhatatlanná teszi a fiókot. A *szolgáltatásmegtagadással járó (DoS) támadásnál* „a hacker hamis címekről folyamatosan adatcsomagokkal bombázza a szervert, és amikor az válaszol egyre, a támadó még több csomagot küld rá. Ettől fokozatosan lelassul a rendszer működése, rosszabb esetben pedig teljes összeomlásához vezet" [5]. A túlterhelés következtében a célpont szolgáltatásmegtagadásra kényszerül, és nem képes adatcsomagokat fogadni.

Jelszavak feltörése

A technológiai jellegű támadások elsődleges célja a jelszó megszerzése. A root jelszó birtokában az adatlopás, erőforrások illetéktelen használata, de még akár a tényleges károkozás is gyerekjáték. A hacker segédeszközök jó része *szótár alapú (dictionary) támadás*ra is képes. Ezek a támadások előre összeállított szótárlistákat használnak. Ezek a támadó eszközök általában nem csak az adott szavakra, hanem azok permutációira is képesek rákeresni. A *jelszó-hash-ek feltörésé*re szintén számos eszköz létezik. Minden egyes jelszóhoz előre meghatározott szabály alapján hash készül. Ezeket már jóval tovább tartana visszafejteni, ezért a hackerek az ún. szivárvány táblákat használják. Ezek hatalmas méretű szöveges állományok, amelyek minden jelszóhoz tartalmazzák a hozzá generált hash-kódot [10]. Ha a fenti módszerek csődöt mondanának is, az ún. *Brute Force (nyers erő) támadás*hoz is léteznek alkalmazások, amik betűk, számok és a legtöbb szimbólum használatával szisztematikusan kipróbálják az összes létező kombinációt. A zárolás elkerülésére a támadók olyan programot használnak, ami törés közben folyamatosan váltogatja az eléréshez használt proxy szervert.

Malware-ek és egyéb kártevők

Azokat a kártékony programokat vagy programrészleteket, amelyek működés szempontjából váratlan hatást fejtenek ki, összefoglaló néven logikai bombának nevezzük. „A logikai bombák valamilyen jelre (…) indulnak be, addig tünetmentesen lapulnak. Végső soron a vírusok, programférgek, trójai és backdoor programok és a rootkitek is a logikai bombák közé sorolhatók" [11]. A digitális kártevőkre elterjedt másik kifejezés az angol malware szó, amely a rosszindulatú programkódok gyűjtőneve.

A *vírus* a kibertérben leggyakoribb kártevő. Elsődleges ismertetőjegye a szaporodás; önnön programkódját lemásolva bejuttatja azt más gazdatestekbe. Az évtized negatív szenzációja volt a *zsarolóvírus*ok elterjedése, amik „állományokat titkosítanak a célgépen, amelyek feloldásáért a tulajdonosnak fizetnie kell, ez megakadályozhatja a hozzáférést bizonyos adatokhoz", sőt gyakran „az egész rendszer működését ellehetetlenítheti" [6]. Gyakori jelenség a *féreg* (worm), a vírus közvetlen leszármazottja. Különlegessége, hogy beavatkozás nélkül, főleg a levelező-programokat felhasználva terjed, és a csatolmányok megnyitásakor továbbítja magát a címtárban szereplő összes címre. A *trójai faló* elnevezésű programot a felhasználó maga engedi be a rendszerbe. A trójai lényege, hogy mindig valami olyat tesz, amit az őt rejtő alkalmazás funkciói alapján nem várnánk tőle; ellopja a személyes adatainkat, másolja és módosítja az állományokat és engedélyeket. A trójai egyik speciális felhasználási módja az *erőforrás-lopás*. Ilyenkor a felhasználó csak annyit érzékel, hogy indokolatlanul lelassult a gépe, például ha a trójai a gépet a napjainkban népszerű kriptovaluták bányászatára használja.

Az ipari kémkedés agyafúrt trükkje a kémprogramok használata. Az ún. *spyware*-ek célja az áldozat által megtekintett tartalom ellopása, továbbítása a támadó felé, aki ezáltal bizalmas és személyes adatokhoz jut. Ezen kártevők egy módosult változata a *keylogger*, ami a felhasználó által leütött billentyűket rögzíti, és küldi el a hackernek. Ezek a programok nem látszanak a futó processzek listájában sem [1].

A kéretlen programok nem rombolnak, nem lopnak erőforrást és adatokat, de zavaróak lehetnek, és alkalmasak arra, hogy kártékony alkalmazások feltelepítésére vegyék rá a kiszemeltet. A legismertebb ilyen programcsalád az *adware*, amely kéretlen reklámokat jelenít meg a felhasználó számára. Általában ingyenes programok „mellékhatásaként" települ fel. Szintén e a csoportba tartoznak a *spam*ek, vagyis a kéretlen e-mailek is. Ilyenkor a felhasználó postafiókjába ritkán vagy gyakran reklámot (rosszabb esetben kártevőt) tartalmazó, kéretlen üzenetek érkeznek. A fertőzések egyik fő oka a kéretlen csatolmányok megnyitása. A spam-ek egy speciális fajtája az úgynevezett megtévesztő vagy *adathalász e-mail*, a scam, amely egy hivatalos szerv vagy szolgáltató nevében próbál személyes adathoz jutni. A *megtévesztő programok* egy valójában nem létező fertőzésre, rendszerhibára hívják fel a figyelmet, és felkínálnak egy alkalmazást, amely segít megtisztítani a „fertőzött" rendszert. Ami így valóban fertőzött lesz…

A *betörő készlet* (rootkit) egy olyan program, ami egy adott operációs rendszeren próbál rendszergazdai jogosultságot elérni. Ha ez sikerül, a támadó parancsokat adhat ki, vagy irányíthatja azt a felhasználó tudta nélkül. „A rootkitek használóikat olyan szintű jogosultsághoz juttatják, melyekhez egyébként nem lenne hozzáférésük" [4]. A *hátsó kapu* (backdoor) egy olyan programrészlet, amely „bizonyos kiválasztott személyek részére illetéktelen hozzáférést enged a programhoz, a géphez, vagy az azokon kezelt adatokhoz" [4]. Mindezt természetesen a felhasználó tudta nélkül. Gyakran a fejlesztők is hátsó kapukat helyeznek el termékeikben, ezzel megkönnyítve a későbbi hibajavítást.

Kiberfenyegetettség vállalati környezetben

A Covid-19 járvány egyik közvetett hatásaként a jelenléti munkavégzést a cégek többségénél felváltotta a home office intézménye. Ez azonban rossz hatással volt a cégek információs biztonsági szintjére, elsősorban amiatt, hogy a dolgozók jelentős része otthoni eszközökkel kapcsolódott a céges hálózatra. „A járványhelyzetet jellemző bizonytalanságot kihasználva pedig a kiberbűnözők egyre több támadást indítottak. 2020-ban már minden percben 76 új kártevővariáns jelent meg". A tavalyi évet tekintve a legtöbb támadást a nagyvállalati környezetben is jelentős károkat okozó Emotet, a Qbot és az Urelast trójai okozták [15] – ugyanakkor azt is látni kell, hogy a vállalati környezetből kiszakadó munkavállalók biztonságtudatossága csökken, a hanyagság pedig megkönnyíti a kibertámadók dolgát.

Nem túlzás tehát kijelenteni, hogy az ember a kibertérben a legveszélyesebb, ugyanakkor a legveszélyeztetettebb szereplő is egyben. A támadásokat emberek hajtják végre, de a „legmagasabb biztonsági kockázatot mindig is a humán faktor jelenti. Ez alatt a biztonsági szempontú tervezési, kivitelezési hibáktól elindulva, a be nem tartott belső biztonsági szabályzat, az üzleti titkok és minősített információk helytelen kezelésén, az élőerős őrzés-védelem szakképzetlen állományán keresztül, a munkavállalók nem megfelelően biztonságtudatos magatartásán át számos humán kockázati lehetőséget érthetünk" [16]. Eltérő szinten és mértékben ugyan, de a behatolás veszélye minden gazdasági szervezetet érint. A megtévesztések sikerében közrejátszik a tény, miszerint a dolgozók nagy része a legalapvetőbb biztonsági előírásokkal sincs tisztában.

Az esetek nagy számában a céges policy-k hiányosak, nem személyre szabottak. A látszólag kisebb jelentőségű területeken dolgozók személyében óriási rés tátong az IT biztonság pajzsán. Ők általában a biztonságkritikus helyzetek felismerésére sincsenek kiképezve. Amint azt Deák is leszögezi: „az ilyen típusú támadások elleni hatékony védekezés egyik legfontosabb része a biztonságtudatosság fejlesztése az egyéneknél és szervezeteknél egyaránt" [3]. A kockázatot leginkább az jelenti, hogy a munkavállalók számtalan bizalmas információval rendelkeznek, amelyet készségesen meg is osztanak egymással. A social engineer a korábban már vázolt jellemvonások mellett a munkahelyhez, beosztáshoz kötődő körülményeket is számításba veszi. Ha valakinek kiszámítható a napirendje, ugyanazt a monoton rutinmunkát végzi, nehezebben ismeri fel a biztonságot veszélyeztető kéréseket.

És „bárki, aki belső ismeretekkel rendelkezik egy adott cégről, veszélyessé válhat. Azon cégek számára, amelyek fájlokban és adatbázisokban tartják az alkalmazottak személyes információit, a kockázat sokkal nagyobb" [9]. Az adatbázisok feltörése, az ebből eredő károkozás, adatlopás, a szemétbe dobott

érzékeny információk mind-mind a céges politikák hiányosságaira vezethetőek vissza. Az alábbiakban egy olyan, általam kidolgozott kibervédelmi modell ismertetése következik, amely IT biztonsági szakemberekkel történő hosszas egyeztetések és ellenőrzési folyamatok során készült, és a vállalati szintű kibervédelem minden lehetséges szintjére kiterjed.

Kibervédelem vállalati szinten

„A hackerek elleni védekezésünk első vonala a leggyengébb, és ez nem más, mint maguk a felhasználók és az általuk választott *jelszavak*" [10]. Az operációs rendszerbe való bejutás, a hálózati hozzáférés, az adatbázisok és FTP-k használata mind-mind autentikációt igényel. Éppen ezért a jelszavak megszerzése, visszafejtése a támadók egyik leggyakoribb módszere. A CyberArk 2021-es felmérése szerint a távolról dolgozók 84%-a újra felhasználja a korábban lejárt jelszavait, és a betörések szignifikáns része korábban megszerzett jelszavak felhasználásával történik. A jelszavak sebezhetőségének mértéke a felhasználó hozzáállásának és a vállalati előírásoknak a függvénye. A személyre szabott jelszavak kis utánajárással könnyedén kitalálhatóak. Sokan használnak végtelenül triviális jelszavakat. A külön e célra létrehozott listában megtalálható szavakból képzett jelszavak dictionary-módszerrel rövid idő alatt megfejthetők, a rövid jelszavak pedig hamar áldozatul esnek egy brute force-támadásnak. A hanyagság pszichológiai háttere abban rejlik, hogy „az emberek nagy többségével még nem történt jelszószerzésből adódó visszaélés. Ezért úgy gondolják, hogy ez csak rémisztgetés, ezért nem is veszik komolyan" [16].

A *vállalati politika* olyan előírások gyűjteménye, melyek „útmutatást adnak az alkalmazottaknak az információt védő magatartáshoz, ezen túlmenően potenciális biztonsági fenyegetések elleni fellépéshez szükséges hatékony eszközök kialakításának alapvető építőeleme" [9]. Fontos, hogy soha ne adaptáljuk másik szervezettől, mindig cégre szabott legyen. A vonatkozó politikáknak képesnek kell lenniük megelőzni és felismerni a biztonságot veszélyeztető eseményeket, és megfelelően reagálni rájuk. Ahol csak lehet, ember helyett technológiát használjunk, és embert annak ellenőrzésére. A vállalaton belül a legnagyobb kár a fontos információk elégtelen védelméből fakad. Ennek kivédésre a policy része kell, hogy legyen a jelszavakra vonatkozó előírások meghatározása. Ezen előírásoknak tartalmazniuk kell a területre való fizikai bejutás kritériumait is, külsősre és dolgozóra egyaránt. A vonatkozó politikák elengedhetetlen része a már rendszeren belüli felhasználók tevékenységi körének meghatározása. A kockázatok csökkentése érdekében gyakran jelölnek ki ún. DMZ-t a belső hálózat és Internet közé, ahova csak jogosult felhasználók ellenőrzött munkaállomásai és alkalmazásai nyerhetnek belépést. Fontos követelmény, hogy a jogosultakon kívül senki ne telepíthessen programot, és a rendszer kívülről szigorú előírások teljesítése révén legyen csak elérhető.

Az adatosztályozási irányelvek kiemelt jelentőséggel bírnak, mivel erősítik az információ- és adatbiztonságot. A policy-ban le kell fektetni azt a követendő eljárást, ami ahhoz szükséges, hogy egy másik fél részére biztonságosan információt adhassunk át. Ennek ki kell terjednie a számítógép-rendszer és a szerverek adataira is. Fel kell készülni a jelenlegi és volt alkalmazottak támadási kísérleteire is. Az elbocsátott alkalmazott hozzáférését, mailcímét töröljük, kulcsait vegyük el, kártyáját semmisítsük meg, formaruháját hozza vissza. Az őt ismerő kollégákat tájékoztassuk az elbocsátás tényéről.

A *fizikai infrastruktúra* is a kibertér szerves része, így annak védelméről is gondoskodnunk kell. „Az őrzés-védelmet, mint a vagyonvédelem egyik szakterületét további három részre szükséges bontani: mechanikai védelemre, elektronikus rendszerekre, és élőerős őrzés-védelemre. Ezeknek a részeknek mind megvannak a sajátos gyenge pontjaik, mivel tökéletes védelem, mint olyan, nem létezik" [16].

Erősen biztonságkritikus helyeken és szituációkban célszerű többlépéses azonosítást alkalmazni. Természetesen a fizikai védelem alapja mindenkor az élőerő, a beléptetésért felelős humán személyzet – a biztonsági őr, a portás, a recepciós, az ügyeletes. Nekik minden részletre kiterjedő munkaköri leírást kell meghatározni. Ügyeljünk arra is, hogy a létesítménybe való fizikai bejutás még ne jelentse azt, hogy ezután az idegen szabadon közlekedhet. Vendégek számára kísérőt kell kijelölni, és figyeljünk oda rá, hogy ki hol jogosult tartózkodni. Érdemes előírni a belépési napló vagy portanapló vezetését is. „A modern vagyonvédelmi megoldások közé tartoznak az olyan zárszerkezetek, melyek nem kulccsal, hanem

más információ tartalmú eszközzel, mint például RFID, vagy biometrikus érzékelővel vannak ellátva" [16], és gyakori a formaruha vagy kitűző viselése is. A behatolók fizikai távoltartását szolgálják a zárak, kerítések, sorompók és őrzött parkolók is. Manapság nem ritka a behatolás-érzékelő rendszerek használata (mozgásérzékelő, riasztó, normál, infra- vagy hőkamerák), sőt biztonságkritikus szervezeteknél hang- vagy biometrikus azonosítás kell a riasztás elkerüléséhez

2. ábra A kibervédelem konceptuális modellje (saját szerkesztés)

.

Egy nagyvállalatnál kiemelkedően védett területnek számítanak például az adattárolás eszközei, storage-ek, szerverek, hálózati aktív elemek, netkapcsolattal rendelkező eszközök és munkaállomások. „A támadók leggyakrabban a hálózati aktív eszközöket próbálják átállítani azok kezelőfelületén keresztül. A *külső védelmet* általában tűzfalakkal (firewall) biztosítják. Ez egy szűrő, amelynek vissza kell tartania a kívülállókat, különösen a hackereket attól, hogy egy szerverre vagy egy hálózatba betörjenek" [1]. A tűzfal dolga, hogy kiszűrje a rosszindulatú adatcsomagokat, kéréseket. A szoftveres védelem nélkülözhetetlen eleme a vírusirtó. Erre a célra számtalan program áll rendelkezésre, céges környezetben a többgépes licensz, és a plusz funkciók miatt mindenképpen fizetős megoldást érdemes választani. A jó vírusirtó legfontosabb ismérvei: (1) a vírusdefiníciós adatbázis gyakori frissítése; (2) nem lassítja a gép működését; (3) központilag menedzselhető; (4) jó minőségű, 24/7 online és/vagy telefonos segítségnyújtás; (5) gyors, magas hatékonyságú keresés; (6) e-mail csatolmányok, letöltések, gyanús kódok real-time vizsgálata és (7) ártalmas (adathalász, bejelentett) webhelyek tiltása. A mobil eszközök (mobiltelefonok, laptopok, tabletek) komoly biztonsági kihívást jelentenek, különösen, ha hozzáférésük van a bizalmas információkat tároló vállalati hálózathoz. Ezeket az eszközöket mindenképpen jelszóval kell védeni, az adatforgalmat titkosítani, és biztonsági programokat kell telepíteni az adatlopások megelőzése érdekében. A vezeték nélküli (Wi-Fi) hálózatnak mindig rejtettnek kell lennie, erős titkosítással ellátva, és arra is ügyelni kell, hogy az aktív eszközökre ne lehessen alapértelmezett jelszóval bejutni. Vendégek számára tartsunk fent céges hálózattól elkülönített Wi-Fi elérést!

Az intranet a cég *belső hálózata*; jellemzően csak az alkalmazottak férnek hozzá. Fontos, hogy a különböző minősítésű információkhoz különböző jogosultságokkal lehessen hozzáférni. A böngészés és levelezés biztonsága webszerver használatával szavatolható. Ez egy olyan szoftver, amely weblapok, html oldalak és az azokon tárolt multimédiás tartalmak kezelését végzi, és továbbítja azokat az internetes keresők felé. A cég online arca és az intranet belső kiszolgálója is egyben, ezért védelmére kiemelt figyelmet kell fordítani. A webszerver védelméhez ajánlott a távoli elérés tiltása, a lehetséges bevitelek szűrése és a megfelelő jogosultság-menedzsment, továbbá ajánlott az összes lehetséges frissítéssel ellátott, biztonságos operációs rendszer használata, a felesleges kiegészítők mellőzésével. A logok rendszeres ellenőrzése és a periodikus, automatizált biztonságos mentés alkalmazása növeli a rendszer biztonságát. A belső hálózathoz való hozzáférés szabályozásának egyik fontos eszköze a MAC címek szűrése. Ha megtörtént a belépés, a naplófájlokból nyomon követhető a jogosult felhasználó tevékenysége. Külsős partnerek általában csak VPN-en keresztül férhetnek a céges hálózathoz, biztonsági ellenőrzés teljesítése után. A sebezhetőség felmérésének folyamata a fenyegetések azonosítását és a kockázatok felmérését jelenti automatizált tesztelési eszközök segítségével. Az eredményeket sebezhetőségi felmérés-jelentésben összegzik a döntéshozók számára. A felmérésben az adott vállalatnak aktív szerepet kell vállalnia. Fontos megérteni az üzleti folyamatokat, pontosan megjelölni az azokat kiszolgáló adatokat és alkalmazásokat, valamint az azokat futtató és tároló hardverelemeket. A hardvereket összekötő hálózati infrastruktúra feltérképezése csak ez után következhet. Végül szükséges a gyengeségek feltárására irányuló átvilágításokat (vulnerability scan) végezni, aminek eredményeit üzleti környezetben is értékelni kell.

Az IAM (Identify and Access Management) rendszer egy olyan keretrendszert biztosít az üzleti folyamatokhoz, amely lehetővé teszi az elektronikus és digitális belépési jogosultság menedzselését és nyomon követését. Ennek segítségével tudja a vállalat szabályozni a belső és kritikus információkhoz való hozzáférés szintjeit, valamint meghatározni a felhasználók jogosultságait a vállalat információrendszerén belül. Természetesen az IAM magában foglalja a naplózási, összegzési és kimutatás-készítési folyamatok támogatását is. Minden vállalkozásnál fel kell készülni az esetlegesen megtörtént behatolások okozta adatvesztések esetén történő helyreállításra. Ilyen prevenciós eljárás lehet a biztonsági mentések használata, a hardver és szoftver redundancia, a megerősített rendelkezésre állás, az adatvisszaállítás, a virtuális gépek, az ún. hideg és meleg tartalék és így tovább.

Egy komplex védelem részét képezik a behatolást észlelő és ellenük védő rendszerek. „Feladatuk, hogy az internet irányából jövő adatokat vizsgálják, valamint a gyanús elemeket blokkolják. Ha a támadó hozzáfér ezekhez a beállításokhoz, ő átírja a szabályokat, amik szerint dolgozik az IDS vagy az IPS, akkor gyakorlatilag már azt csinál, amit szeretne” [16]. A támadások kivédése érdekében a *munkaállomásokat* naprakész állapotban kell tartani. Ajánlott a legfrissebb biztonsági szoftvereket, rendszer- és böngésző frissítéseket telepíteni a digitális kártevők és egyéb online támadások kivédése érdekében. A vírusirtó adatbázisa mindig naprakészre legyen frissítve, és minden rendszerszintű változás (bővítés, patch-ek telepítése, új programok) esetén kötelező ellenőrzés kell. A tartalomszűrés azon alapul, hogy „minden beállított forgalom, amely áthalad a tűzfalon, ellenőrzésre kerül. Ha ezen az ellenőrzésen megfelelt, akkor a tűzfal elküldi az internetes forgalmat a CVP szervernek, amely vírus és tartalomszűrést végez” [11]. A visszaküldött/módosított tartalomhoz férhet hozzá a felhasználó. Megvalósítható a webes tartalom, a levelezés és az FTP adatforgalom szűrése is. A tartalomszűrés sok esetben belső cenzúraként is működhet; tilthatók vagy engedélyezhetők bizonyos oldalak. A munkaállomások védelmének első lépése lehet a Bitlocker meghajtó titkosítás, de védhetjük jelszóval a megosztott tartalmakat (könyvtárakat, meghajtókat) is. Gyakori a smartkártyás azonosítás kötelezettsége is.

A *szoftvereket és fájlokat* szintén védeni kell; az adatbázisok tartalma rengeteg információval bír a social engineereknek. „A bizalmas fájlok megvédhetők megfelelő hozzáférési ellenőrzések alkalmazásával, hogy csak az arra jogosultak tudják megnyitni. Egyes operációs rendszerekben olyan ellenőrző vezérlők vannak, amelyeket beállíthatunk bizonyos események naplózására” [10]. A cégek legfontosabb

adatbázisai általában üzletileg kritikus alkalmazói szoftvereken keresztül hozzáférhetőek, amik gyors elérést igényelnek. Itt szintén meg kell határozni azt az optimumot, amely egyszerre szavatolja a biztonságot és a használhatóságot. Az operációs rendszert és az alkalmazói szoftvereket is ajánlott rendszeresen frissíteni. A kódban maradt hibák kijavítására patcheket adnak ki, ezzel a már említett biztonsági réseket, exploitokat szűntetik meg. Különösen támadhatóak az ActiveX vezérlők, ajánlott kikapcsolni, és csak egyedileg engedélyezni a futtatásukat a webböngészőben. A frissítéseket (a vírusirtóét is!) kompetens személy (pl. a rendszergazda) frissítse, ne bízzuk a felhasználóra!

Az *adatbiztonság* elérésének kulcsa az adatok integritásának és hozzáférhetőségének biztosítása. Ehhez az kell, hogy az összes munkaállomás adatait rendszeresen mentsük le, különösen a kritikus adatokat, mint például a Word és Excel fájlokat, a könyvelés és a HR adatait, a prezentációkat és minden egyéb fontos dokumentumot. Nagyvállalati környezetben célszerű automatikus mentést alkalmazni a megfelelő célszoftverekkel. Az adatosztályozás fontos lépése azon információk elkülönítése, amelyek a vállalat szemszögéből szenzitív, bizalmas adatok lehetnek. „A belső információk nem titkosnak minősített információk, ennek ellenére mégis az adott szervezet dolgozóira vonatkoznak, éppen ezért nem javasolt megosztani idegenekkel" [3]. Az adatosztályozási előírások azok szenzitív jellege alapján külön kategóriákra osztják a belső dokumentumokat, egyúttal megjelölve azt is, hogy milyen rang, szint szükséges a hozzáféréshez. Például az adatosztályozás egy lehetséges módja: (1) *nyilvános információk*, amik semmilyen korlátozás alá nem esnek; (2) *belső*, nem nyilvánosság elé tárható *információk*; (3) *bizalmas, szenzitív adatok*, melyek csak előre meghatározott, feljogosított körben elérhetők és (4) *titkos*, a szervezet céljait, piaci helyzetét veszélyeztető, szigorú elbírálás és adatkezelés alá eső *információk*. Ez az adatkezelés terjedjen ki a mailben, telefonon történő egyeztetésekre és továbbított fájlokra is! A diszkrecionális hozzáférés kezelésének alapja a felhasználók, jogosultságok azonosítása. A jogosultsági szintek és szerepkörök a védelmi tartományba (esetleg közös munkacsoportba) történő belépéskor kerülnek meghatározásra. A már nem kellő, kidobásra ítélt adatokat tartalmazó dokumentumok kezelését szintén szabályozni kell. A kukabúvárkodás elleni leghatékonyabb módszer a papírok teljes megsemmisítése. Fontos, hogy a szabályozás kiterjedjen az adathordozókra is. A komplett eszközöket (szerverek, számítógépek, laptopok, tabletek), ha javíthatatlanul meghibásodnak, arra szakosodott céggel kell elvitetni – ezt egyébként nem csak céges előírás, de környezetvédelmi szabályozás is előírja.

Összefoglalás, konklúzió

Napjainkra a közösség terek, az internet, az okoseszközök, azaz összességében a kibertér alapjaiban változtatta meg mindennapi életünket. A kibertérben színre lépő aktorok, valamint a kiszolgáló és beágyazott támogató elemek közötti komplex kapcsolat egy rendkívül bonyolult közeghez vezet, ami nagy fenyegetést jelent éntudatunkra, magánéletükre, adatainkra és munkakörnyezetünkre is. Ezek a fenyegetések jelentkezhetnek humán, fizikai, logikai vagy rendszerszinten, maga a támadás pedig történhet fizikai, információs vagy tudati dimenzióban is. A támadást elkövetők potenciális köre széles spektrumot fed le a gondatlan alkalmazottaktól a hackereken át a szervezett terrorista csoportokig. A kibertámadóknak számtalan változatos eszköz áll rendelkezésére az egyén vagy környezete támadására; a támadó használhatja a pszichológiai manipuláció eszköztárát, különböző technológiai megoldásokat, támaszkodhat digitális kártevőkre, vagy próbálkozhat jelszavak megszerzésével. Az elterjedt technológiai alapú biztonsági megoldások önmagukban vállalati szinten sem elegendőek a fenyegetések kivédésre. Egy-egy elkövetett támadás esetén a reagálás önmagában általában nem elég, hiszen a kár már megtörtént – prevencióra, jól kiépített védelemre van szükség. Az is egyértelmű, hogy az okozott károk mérsékelhetők (vagy megelőzhetők) egy jól kiépített biztonsági rendszerrel. Ennek okán szükséges egy olyan átfogó kibervédelmi modell kidolgozása, amely a sebezhető területek teljes spektrumát lefedi, és amelyből kiindulva hatékony és jól lehatárolható védelmi szintek különböztethetők meg. Az utóbbi években, évtizedekben a támadások gyakorisága és az okozott kár mértéke emelkedő tendenciát mutat. Mivel a kibertámadások gazdasági hatásai a teljes vállalati szférában a vállalkozások prosperitását

fenyegető meghatározó tényezőként jelentkezek, így érdemes lenne nagyobb hangsúlyt fektetni a védekezés és megelőzés módszereire, hiszen a támadók mindig egy lépéssel a védelem előtt járnak.

Felhasznált irodalom

[1] Bluebird, Kazári, Cs. (2003). Hacker, cracker, warez. Computer Panoráma, Budapest.

[2] Csepeli, Gy., Prazsák, G. (2012). Információs társadalom 2.0. ELTE, Budapest.

[3] Deák, V. (2017). A social engineering humán alapú támadási technikái. Online. http://biztonsagpolitika.hu/rovat/publikaciok-2017. Letöltve: 2018.11.20.

[4] Fehér, K. (2016). Kezdő hackerek kézikönyve. BBS-INFO Kiadó, Budapest.

[5] Flamminch, M. (2002). Hackerek – Vázlat a magyar hacker szubkultúráról. Online. http://www.mediakutato.hu/cikk/2002_03_osz/01_hackerek. Letöltve: 2018.11.17.

[6] Horváth, A., Kiss, F., Benkő, Zs., Szanyi, I. (2016). A szoftver sérülékenységek kihasználási módozatai in. IT és hálózati sérülékenységek társadalmi-gazdasági hatásai, INFOTA, p. 65.

[7] Joint Publication 1-02 Department of Defense Dictionary of Military and Associated Terms. Online. https://fas.org/irp/doddir/dod/jp1_02.pdf, p. 57 . Letöltve: 2018.02.05.

[8] Kelemen, R., Pataki, M. (2015): A kibertámadások nemzetközi jogi értékelése in. Katonai Jogi és Hadijogi Szemle, 1/2015, 57. o.

[9] Mitnick, K., Simon, W. (2003). A legendás hacker – A megtévesztés művészete. Perfact kiadó.

[10] Mitnick, K., Simon, W. (2006). A legendás hacker 2– A behatolás művészete. Perfact kiadó.

[11] Muha, L. (2008). Az informatikai biztonság egy lehetséges rendszertana. Bolyai Szemle, 17 (4).

[12] Nielsen, S. (2012). Pursuing Security in Cyberspace. Foreign Policy Research Institute, Philadelphia.

[13] Oroszi Eszter (2008). Social Engineering: Az emberi erőforrás, mint az információbiztonság kritikus tényezője. Corvinus Egyetem, Budapest.

[14] Papp, Z. I. (2018). A kiberterrorizmus módszerei, lehetséges eszközei és az ezek ellen történő védekezés alternatívái című doktori értekezése, Nemzeti Közszolgálati Egyetem, Budapest.

[15] PC Pult Online: Jelentősen nőtt a kibertámadások száma. Online. https://pcpult.hu/it-tech/security/jelentosen-nott-a-kibertamadasok-szama.html. Letölte: 2021.08.16.

[16] Sörös, T., Váczi, D. (2012). Social engineering a biztonságtechnika tükrében, XXXI. Országos Tudományos Diákköri Konferencia Had- és Rendészettudományi Szekció, Budapest.

[17] Világgazdaság.hu: A pandémia alatt megszaporodtak a pénzlopásos támadások. Online. https://www.vg.hu/cegvilag/2021/07/a-pandemia-alatt-megszaporodtak-a-penzlopasos-tamadasok. Letöltve: 2021.08.16.

Németh Richárd kiváló minősítésű gazdaságinformatikus diplomát szerzett a Széchenyi István Egyetemen. Egyetemi tanulmányi mellett elvégezte a Java-programozói és webfejlesztői képzéseket is. Közel húsz éve dolgozik az informatika-szakmában illetve az ehhez kapcsolódó szakterületeken. Jelenleg egy, a régió vezető járműgyártójánál informatikai háttértámogatást biztosító cégnél dolgozik és IT-biztonsági területen szakfeladatokat lát el. 2021 őszétől a Széchenyi István Egyetemen tanít, és MSc hallgatóként folytatja tanulmányait. Immár második éve foglalkozik nyelvi lektorálással egy MTA „A" kategóriás folyóiratnál, valamint szerkesztője egy jogi-informatikai kutatócsoport kiadványainak. Különböző tudományos kiadványokban számos publikációja jelent meg, illetve áll megjelenés alatt. Kutatási tevékenysége szerteágazó, a vállalati kiberbiztonság, a kibertérsebezhetőségek témájától egészen a 3D-nyomtatás, a gamifikáció, valamint az adatvizualizáció vizsgálatáig terjed

Gépi tanuló rendszerek audit-kihívásai

BARTA GERGŐ
IT biztonsági vezető és mesterséges intelligencia szakértő
eMail: gbarta@deloittece.com

ABSTRACT

A gépi tanuló rendszerekkel kapcsolatos kutatások száma és az intelligens rendszerek ipari és üzleti felhasználása az előző évtizedben jelentősen megnövekedett, mely köszönhető nagyrészt a technológiai környezet megváltozásának. Gépi tanuló rendszerek alkalmasak többek között, az elérhető adatokból való tudás kinyerésére, predikcióra, nyelvi elemek feldolgozására, képek és videók elemzésére stb. ezáltal magas színvonalon képesek üzleti feladatok automatizálására, amennyiben általánosító képességük meghaladja az elvárt metrikákat. A gépi tanuláson alapuló alkalmazások fejlesztése eltérő paradigmán alapszik, összehasonlítva a tradicionális szoftverfejlesztési megközelítéssel, ezért az implementációból származó kockázatok felmérése is új szemléletváltozást igényel, melyeket informatikai kontrollokkal szükséges mitigálni. A belső informatikai kontrollok hatékonyságának megbizonyosodásáért felelős funkcionális szakterület az IT audit osztály, melynek hatóköre a gépi tanuló rendszerek kontrolljainak ellenőrzésére, fejlesztési folyamatainak felülvizsgálatára és a rendszerbevezetésre is ki kell terjednie a közeljövőben, ezért az auditoroknak naprakész tudással kell rendelkezniük az intelligens szoftverek üzemeltetésének világában. A cikk a gépi tanuló rendszerek fejlesztésének általános folyamatát elemzi, és a releváns szakirodalom feldolgozásával azon lehetséges mérföldköveket keresi, melyben elengedhetetlen az IT audit bevonása, ugyanakkor, a nemrégiben megjelent kutatási eredmények és tapasztalatok alapján az IT audit által szükséges elvégzendő teszteket tárgyalja.

Bevezetés

A mesterséges intelligencián alapú rendszerfejlesztések piaca világszinten a Statista [1] adatai alapján 2016-ban meghaladta a 3 milliárd dollárt, a becslések szerint 2017-ben a 4 milliárd dollárt, 2018-ban a 7 milliárd dollárt, 2019-ben pedig, az előrejelzések 11 milliárd dollárt prediktálnak. Vitathatatlan tény, hogy a mesterséges intelligencián[2] és gépi tanuló eljárásokon alapuló alkalmazások megjelenése rohamos tempóban nő, mivel a módszertanok kifejlesztéséhez és a kutatáshoz szükséges feltételek adottá váltak. Gépi tanuló rendszerek fejlesztéséhez egyik legnagyobb mértékben hozzájáruló technológiai környezet a Big Data, mely többek között, nagy mennyiségű adat tárolását és feldolgozását teszi lehetővé, ezzel támogatva a tanuló algoritmusok adatigényét. Másik feltétel a magas hardver kapacitás elérhetősége, a grafikus kártyák fejlődésével a számításigényes algoritmusok futási és tesztelési ideje lerövidült, teret hagyva a folyamatos kísérletezésnek, melyre korábban nem volt lehetőség.

Az intelligens szoftverek számos iparágban bíztató eredményekkel szolgálnak, egyidőben új kihívásokat és kockázatokat jelentenek a technológiákat alkalmazó vállalatok részére, mivel fejlesztési logikájuk és belső szerkezetük merőben eltérnek a tradicio-

[2] A mesterséges intelligencia és a gépi tanulás definíciója gyakorta az akadémiai körökön kívül nem válik szét, itt mesterséges intelligencián Peter és Russel [2] klasszikus definícióját érti a szerző, mely „*olyan intelligens ágensek tervezése és építése, amelyek képesek a külső környezet érzékelésére, és olyan döntéseket és cselekvéseket hoznak, mely befolyásolja ezt a környezetet.*" A definíció megengedi, hogy a mesterséges intelligencia fogalomkörébe belefoglaljuk a tradicionális szabályalapú rendszereket is, azonban a továbbiakban szándékosan a gépi tanulás szó lesz használva az intelligens alkalmazások tekintetében, kiemelve, hogy ezen rendszerek esetében nem szükséges az explicit szabályalkotás a döntéshozatalhoz.

nális szoftverfejlesztés alapkoncepcióitól. A gépi tanulás, egy olyfajta mérnöki megközelítés, mely a tradicionális szoftverfejlesztést túlszárnyalva, nem explicit előre definiált szabályok mentén támogatja a döntéshozást (pl. „if statements"), mint pl. a 80-as és 90-es évek szakértői rendszerei, hanem az adatokban meghúzódó mintákat hivatott automatikusan felismerni, majd matematikailag ezt leképezni. Andrew Ng a Google Brain egykori vezetője nagyon egyszerűen úgy fogalmaz, hogy a gépi tanulás nem más, mint a rendelkezésreálló attribútumokhoz (x-ekhez) egy tanulási függvényen keresztül a célváltozókat (y-okat) rendelni [3]. Számos gépi tanuló rendszer egyik legnagyobb hátránya a „fekete doboz" jellege, melyben ez a hozzárendelés megtörténik. Ez azt jelenti, hogy ezen algoritmusok által automatizáltan megalkotott szabályok az emberi megértés számára jelenleg nem vezethetők le világosan. Számos iparágban pl. gyógyszergyártás az algoritmusok átláthatósága és azok döntési képességeinek levezethetősége jelentős. Fekete doboz algoritmusra példa a 2000-es évek elején sok kutatási eredménnyel büszkélkedhető SVM (Support Vector Machine) alkalmazások és a most, főleg képi- és hangfeldolgozásban alkalmazott neurális hálók. Ezen módszerek további kihívások elé állítják az IT auditorokat a modell kiértékelését megnehezítve.

Evidens, hogy a gépi tanuló rendszerek ellenőrzésére is audit eljárásokat kell kidolgozni, hogy a vezetés megbizonyosodjon, hogy az alkalmazásfejlesztés és üzemeltetés kontrollált körülmények között zajlik, és, hogy a rendszerimplementáció az üzleti javakat és célkitűzéseket szolgálja a hatályos jogszabályok és belső politikák betartásával. A tanulmány a következőkben a témában megjelent kutatási eredményeket és publikációkat ismerteti, majd kitér az informatikai audit célkitűzésének rendszerére. Soron követően a gépi tanuló rendszerek általános fejlesztési folyamata, majd a felmerült lehetséges audit eljárások kerülnek kifejtésre, melyek a bemutatott szakirodalmi példák és tapasztalatok alapján lettek tanulságként levonva.

Kapcsolódó munkák

A témában relatív kevés kutatói munka érhető el, mivel a tudományos publikációk jelentős része vagy a modellfejlesztésre és a minél magasabb teljesítményre, vagy a gépi tanuló rendszerek üzleti felhasználására fókuszál, mely azt jelenti, hogy a kutatások jelenleg jobban érdekeltek a minél magasabb teljesítményt ígérő robotok megalkotásában, mint azok felügyeletében és ellenőrzésében, ahogy ez Andrews et al. [4] kutatásából is alátámasztható. Mindazonáltal, a gépi tanuló rendszerek audit kérdései és kihívásai megmozgatták az audit területen tevékenykedő szakértőket, az ISACA (Information Systems Audit and Control Association) szervezet több publikációt is közzétett, melyben a felmerült problémákról értekezik.

Az egyik érdekes gondolat, hogy mivel a munkaerőpiac nem képes megfelelő számú szakembert biztosítani a területre, és viszonylag komplex az intelligens rendszerek fejlesztése, ezért várhatóan számos projekt vállalaton kívül harmadik felek által lesz leszállítva, mely a beszállítói auditok mértékét fogja növelni fókuszálva az intelligens rendszerfejlesztés technológiai környezetére és kontrolljaira [5]. Továbbá, az ISACA álláspontja, hogy az IT auditoroknak nem feltétlenül kell rendelkezniük a szükséges modellezési szaktudással, minél inkább a mögöttes irányítási keretrendszert kell ellenőrizni, hogy az megfelel-e az iparági jógyakorlatoknak, jogszabályi elvárásoknak, üzleti célkitűzéseknek és a szervezet által kiadott COBIT 2019 (Control Objectives for Information and Related Technologies) keretrendszert ajánlja, mint iránymutatás a mesterséges intelligencián alapuló alkalmazások auditálásához [5].

Andrew Clark [6] a CRISP-DM (Cross Industry Standard Process for Data Mining) keretrendszert javasolja, mely gépi tanuló rendszerek fejlesztésére testre szabható és iparági gyakorlatokat tartalmaz az adatfeldolgozási, modellezési és modellértékelési folyamatokban. Clark kiemeli, hogy a keretrendszert az auditokhoz módosítani szükséges, hogy az célozottabban támogassa az ellenőrzési folyamatokat, de az hatékonyan alkalmazható az adatelőkészítési fázisban. Ribeiro et al. [7] a Local Interpretable Model-Agnostic Explanations (LIME) keretrendszert dolgozták ki, melyet közvetlenül nem auditoroknak szántak, de a modell segítséget nyújt a gépi tanuló rendszerek jobb és érthetőbb interpretálásában, melyet az auditorok a modellértékelésnél felhasználhatnak az alkalmazott gépi tanuló rendszerek megértése végett.

Információbiztonság és informatikai audit

A vállalati információrendszerek bizalmas és üzletileg kritikus információt kezelnek, ezért az üzleti rendszer sérthetetlenségének és bizalmasságának biztosítása kiemelt szerepet ölel fel a szervezetek életében. Erre világítanak rá jelen korunk szabványai is, mint pl. az ISO 27001-es szabvány család, a fentebb említett COBIT 2019 , NIST 800-53, vagy ay ISF kiadvány, a „The Standard of Good Practice for Information Security 2018". A szabályozói környezet Magyarországon és az Európai Unióban is magas hangsúlyt fektet az információbiztonsági előírások megfogalmazására és azok betartatására.

Országunkban kiemelendő, a pénzügyi szektor a legjobban szabályozott ágazat, melyet hazánkban a Magyar Nemzeti Bank kötelező érvénnyel rendszeresen ellenőriz. Információ-biztonságot érintő ajánlatok és rendeletek hazánkban többek között, az MNB 7/2017. (VII.5.) számú ajánlása az informatikai rendszer védelméről, MNB 2/2017. (I.12.) számú ajánlása a közösségi és publikus felhőszolgáltatások igénybevételéről, MNB 19/2017. (VII.19) az elektronikus hírközlő eszközök auditjáról, 2011. évi CXII. törvény az információs önrendelkezési jogról és az információszabadságról stb. illetve, az Európai Parlament és Tanács 2018. május 25-től hatályba léptette az EU 2016/679 Általános Adatvédelmi Rendeletét (GDPR), melynek hatóköre kiterjed az Európai Unióban és az Európai Gazdasági Térségben működő minden egyes szervezetre megcélozva a személyes adatokat feldolgozó rendszerek biztonsági előírását és biztonságos üzemeltetését.

A biztonság fenntartásáért az információbiztonsági szervezeti egység felelős a szervezeti felépítésben. Az információbiztonsági szervezet feladata az információrendszerek, a hálózat és egyéb fizikai és logikai eszközök védelmi intézkedéseinek meghatározása, betartatása és folyamatos nyomon követése. Az információbiztonsági osztálynak függetlenül külön kell működnie más szervezeti funkcionális egységektől, még az informatikai üzemeltetési osztálytól is, kikényszerítve a négyszem-elvet. Az információbiztonsági szintet informatikai és szervezeti kontrollok implementálásával lehet növelni, melyek szükségességét, erősségét és rendszerességét a szervezeti informatikai kockázatelemzés által lehet érvényre juttatni.

Mindazonáltal, az implementált informatikai kontrollok megléte még nem szavatolja a hibamentes működést, a vezetésnek rendszeresen meg kell győződnie arról, hogy a mitigáló intézkedések hatékonyan működnek, nem lehetséges azok megkerülése, illetve kijátszása. Erre egy gyakorlati példa az alkalmazások jogosultságkezelését megkerülendő hibajavításra (éles programfejlesztésre) szolgáló jogokkal történő visszaélés, melyekkel az alkalmazás szintű biztonsági logikát felül lehet írni, mely, ha a naplófájlok írási jogkörrel való hozzáférésével párosul, akkor egy rosszindulatú felhasználó vagy támadó képes csalások kivitelezésére a rendszerben, úgy, hogy saját elektronikus lábnyomait törölni képes, mely információbiztonsági eseményhez, incidenshez vezet. Információbiztonságot érintő incidensek alapvetően három különböző forrásból érkezhetnek:

- külső, a szervezettől kívülálló csoportoktól, személyektől, melyek lehetnek kiberbűnözők, versenytárak ipari kémei, hobby hackerek, szélsőséges esetben terroristák;
- belső, a szervezet belső dolgozói, akik szándékosan vagy véletlen bizalmas információt osztanak meg a külvilággal;
- természeti események, melyek az információ elérhetőségét és rendelkezésre állását veszélyeztetik pl. árvíz, mely megkárosítja a szerverparkot.

A belső informatikai kontrollok hatékonyságának feltárására szolgáló funkcionális terület megnevezése az informatikai audit csoport, melyet olyan üzleti és informatikai szakmai tudással rendelkező szakemberek alkotnak, melyek elsődleges célja a belső informatikai kontrollkörnyezet ellenőrzése és folyamatos tesztelése, annak meggyőződésére, hogy az informatikai folyamatok teljes mértékben a szervezetek üzleti célkitűzéseit követik a lehető legnagyobb információbiztonsági szint mellett. Az informatikai auditok célja alapvetően a vezetés számára bizonyosságot nyújtani, hogy a belső üzleti folyamatokat támogató IT és információbiztonságot érintő kontrollok megfelelően lettek kialakítva, implementálva és hatékonyan funkcionálnak.

A belső kialakított mechanizmusoknak garanciát kell biztosítaniuk, hogy:

- az informatikai rendszerek helyesen és pontosan dolgozzák fel az üzleti adatokat;
- a szervezet munkatársai és külső partnerei betartják a szabályzatokat, procedúrákat, irányelveket, melyeket a vezetés lefektetett és jóváhagyott;
- a belső eljárások szavatolják, hogy bármilyen katasztrófahelyzet esetén az adatok bármikor elérhetők legyenek, integritásuk és bizalmasságuk fenntartható, az üzletmenetfolytonosság biztosított [8].

Az informatikai audittal foglalkozó belső funkcionális szervezeti egységnek függetlenül kell működnie a többi szervezeti egységtől és közvetlen a felső vezetésnek kell jelentenie, ha bármi problémát észlel a belső kontrollkörnyezettel kapcsolatban. A kontrollok hatékony működésének megbizonyosodására az informatikai audit teszteket végez egy előre meghatározott audit terv alapján, mely során mintavételezési eljárással a kontrollok gyakoriságának megfelelően próbál meggyőződni a kiválasztott kontrollok megfelelő üzemeltetéséről alátámasztó evidenciák elemzése által. Az informatikai audit nem kizárólag az információbiztonságot érintő intézkedések ellenőrzését végzi, hanem bármi olyan a vezetőség vagy külső szabályozó által támasztott követelményt ellenőrizhet, ami a belső informatikai működéssel kapcsolatba hozható, mint pl. a belső projektek időzítésének és költségeinek nyomon követését, jogszabályi megfelelősségét stb. Ezért az audit hatóköre kiterjedhet más területekre is, és feltételezhetően az IT audit lesz az a funkcionális csoport, mely a gépi tanulás és mesterséges intelligencia alapú rendszerek megfelelőségét is ellenőrizni fogja, ezért kulcsszerepet fognak betölteni az intelligens rendszerek fejlesztésének, implementálásának és nyomon követésének folyamataiban is, mely, mivel más paradigmán alapul, ezért újfajta tudást követel meg az auditorok részéről. Kiemelendő, az IT audit sem tud abszolút bizonyosságot szerezni a kontrollok megfelelő működéséről, mivel kvázi lehetetlen az összes minta ellenőrzése, kizárólag részleges garanciát tud adni arról, hogy a szervezet manuális és automatikus kontrolljai a vezetőség által elvártan operálnak.

Gép tanuló rendszerek fejlesztése

A kérdés jogosan felmerülhet, hogy miért is lehet szükséges a tradicionális szoftvermérnöki munkát kiváltani gépi tanuló eljárásokkal. Tételezzük fel, hogy a jövőben be szeretnék vezetni az arcalapú fizetést az áruházakban. Ahhoz, hogy ez lehetségessé váljon, szükséges egy olyan szoftver mely kétséget kizáróan képes azonosítani a fizetni szándékozót, mely azt jelenti, hogy minden egyes személyt, aki személyes adataival (arckép, bankszámlaszám stb.) hozzájárult a szolgáltatáshoz, azt egyértelműen meghatározzon. Ha a szoftver az arcképet előre definiált szabályok alapján azonosítaná, képzeljük el, hogy hány elágazás megírására lenne szükség ahhoz, hogy ez teljesüljön. Természetesen, a szerző legjobb tudomása szerint, a mai nap még nem létezik olyan tudományosan publikált kutatói munka, mely arcképeket 100%-os pontossággal klasszifikálni tudna, azonban a példa kedvéért érdemes elgondolkozni a problémán. A gépi tanuló eljárások (és itt főleg a mély neurális hálókról van szó) pixelenként értelmezik a képeket, melyben minden egyes pixel egy dimenziónak feleltethető meg. A pixelek egymásutániságát, elhelyezkedését és kialakított mintázatát értelmezi a tanuló eljárás, majd megfelelő mennyiségű tanítópont után képes döntést hozni, hogy adott kép kit is ábrázol. A problémát hangsúlyozva, tehát ha minden egyes osztályra külön szabályokat kellene manuálisan létrehozni, akkor a szoftverkód minden bizonyossággal több milliárd soros lenne, ráadásul az emberi arc változásainak a nyomon követése fenntarthatatlanná és karbantarthatatlanná tenné az alkalmazást (ami ugyancsak problémaként előjöhet a gépi tanuló rendszerek esetében is).

A gépi tanulás, tehát, a mesterséges intelligencia egyik alterülete, melynek célja a rendelkezésre álló adathalmazban meghúzódó mintázatok felismerése, megtanulása, majd a mintázatok alapján új adatrekordok csoportosítása [9]. A gépi tanulás Raschka megfogalmazása alapján, „a számítógépek felruházása a tanulás képességével” [10]. Sokkal technikaibb definíciót ad meg Russel és Norvig [2]: „A tanulás alapgondolata az, hogy a megfigyeléseket ne csak az ágens jelenlegi cselekvéseinek kialakítására használjuk, hanem arra is, hogy javítsuk a cselekvésre való jövőbeli képességeit”.

Ebben a definícióban megjelenik a jövő, azaz a predikció megnevezése, miszerint a gépi tanulás egy olyan automatizált folyamat, mely hozzájárul a jövőre vonatkozó sejtések előrejelzéséhez. Dua és Du [11] szerint a gépi tanulás „egy tudományos modell építése, mely képes a jelenlegi adatokból tudást képezni". Hastie et al. [12] egyszerűen „az adatokból való tanulás" mikéntjeként definiálja a fogalmat.

A gépi tanulás alábontását a szakirodalom többékevésbé egységesen kezeli. Hastie et al. [12] a gépi tanulást felügyelt és felügyelet nélküli tanulásként csoportosítja, míg pl. Raschka [10] egy harmadik kategóriát is külön kiemel, ami a megerősítéses tanulás. A felügyelt tanulás egy meglévő adathalmaz, azaz historikusan összegyűjtött információ alapján mintázatkeresési eljárások összessége, mely a mintázat feltárását követően hozzájárul egy adott célváltozó értékének előrejelzéséhez [13].

A felügyelt tanulásra példa a pénzügyi intézeteknél is alkalmazott profilozó eljárások, melyek alapján kategorizálásra kerül, hogy adott potenciális ügyfél milyen mértékben lesz képes a felvett hitelt törleszteni. A célváltozó ebben az esetben a kockázat egy 0-tól 1-ig terjedő skálán a pénzvisszafizetés lehetséges valószínűsége, mely, ha 1-hez közeli akkor a potenciális ügyfél kockázatosnak minősíthető, és a pénzintézet rendelkezhet úgy, hogy nem folyósít hitelt, míg 0-hoz közeli érték alacsony kockázatnak tudható be, tehát a pénzintézet folyósít hitelt az igénylőnek. A historikus adatok ebben az esetben lehetnek az igénylő neme, kora, lakcíme, havi nettó jövedelme stb.

Egy másik korszerű alkalmazási területe a felügyelt tanulásnak az önvezető autók által alkalmazott automatizmusok. Historikus adatok alapján a gépi eljárás képes felismerni az önvezető autó előtt, mellett, hátul közlekedő más járműveket és eszerint pozícionálni önmagát, megelőzve a baleseteket. Értelmezi a közlekedési táblákat, észreveszi a gyalogosokat, tehát a cél, alkalmazkodni a külső környezethez [14]. A gépi tanuló eljárások másik kategóriája a felügyelet nélküli tanulás. Az ebbe a csoportba tartozó eljárások a begyűjtött adatokban hasonlóan mintázatot keresnek, azonban a cél nem egy célváltozó becslése, hanem a mintázat alapján az adatok egyes kategóriákba való csoportosítása.

A klaszterező eljárások tipikusan felügyelet nélküli eljárások. A marketingkutatásban gyakorta használatosak a piaci szegmentációk feltárása, így azok elemzése és kiértékelése a felügyelet nélküli tanulási eljárások egyik gyakori alkalmazása. A megerősítéses tanulásban a rendszer feladata, hogy képes legyen a környezetéhez alkalmazkodni és optimális stratégiát válasszon egy adott költségfüggvény minimalizálása, vagy jutalomfüggvény maximalizálása érdekében [2]. Az automatizált sakkjáték a megerősítéses tanuláson alapuló gépi tanuló rendszer egy klasszikus példája.

A gépi tanuló rendszerek fejlesztése az információrendszerek fejlesztéséhez hasonlóan több fázisra bontható, mely fázisok tervezése azért indokolt, hogy a lehetséges döntési pontok (pl. adattranszformáció, modellválasztás, metrika választás) a modell objektíven mért jóságához a lehető legnagyobb mértékben hozzájáruljon. A gépi tanuló eljárások fejlesztése esetén, összegezve, célszerű az általános alkalmazás- és információrendszer fejlesztési módszerekkel magas szinten párhuzamot vonni, azonban a szerző álláspontja szerint, ezt ki kell terjeszteni a tudományos modellek tervezésének és fejlesztésének módszertanával. Gépi tanuló rendszerek esetén is igaz az állítás, hogy az alkalmazást körültekintően, bevált iparági gyakorlatok és kutatási eredmények által igazolt módszertanok mentén érdemes fejleszteni. Szepesné [15] az információrendszerek fejlesztését 3 részre osztja. Első szakaszban, melyet a szerző előszakasznak nevez, történik a kiindulási helyzet elemzése, a feladat megfogalmazása, a költségelemzés, illetve az előnyök definiálása, tehát az információrendszer célkitűzésének meghatározása. A második szakasz a fejlesztés szakasza, melyben kivitelezésre kerül az adatbázis struktúra leírása, az alrendszerekre való bontás, a specifikációk véglegesítése és a rendszerkomponensek együttműködésének biztosítása. Az utolsó szakaszban, a felhasználói szakaszban veszi kezdetét az alkalmazás használata, karbantartása, felülvizsgálata és optimalizálása. Kovács [16] ajánlást fogalmaz meg a tudományos szimuláció és modellépítéssel kapcsolatban. Elsőként a modellalkotás célját kell meghatározni, majd a vizsgálandó rendszert kell definiálni, amit a modellkalibráció követ, azaz az inputként szolgáló adathalmazt kell a modellre szabni. Az utolsó lépés a validáció, az ellenőrzés szakasza, hogy a modell elérte-e az előzetesen definiált célját.

A felhasznált szakirodalmak áttekintésének segítségével az alábbiakban leírtak szerint érdemes megtervezni és fejleszteni a gépi tanuló alkalmazásokat, melyben figyelmet kell szentelni az iparági gyakorlatok és a tudományos céllal történő modellalkotás elvárásainak. A gépi tanuló alkalmazások fejlesztésének az első fázisa a gépi tanuló rendszer célkitűzésének definiálása (üzleti probléma leírása és üzleti specifikáció megfogalmazása). Ebben a szakaszban történik az adott üzleti probléma kiértékelése, és hogy az új alkalmazás milyen módon lesz képes az üzleti problémát megoldani, elhárítani, karbantartani. A célkitűzés meghatározásakor a felhasználni kívánt adatvagyont, annak beszerzési kritériumait, a függő és független változókat is meg kell határozni, majd még az adatgyűjtés költséges fázisa előtt kiértékelni tesztadatokkal, hogy valóban az alkalmazás tervezeti szinten úgy fog működni, ahogy azt annak felhasználói elvárják. Amennyiben a specifikáció megfelelő minőségű eredményeket produkál, úgy a következő fázis az adatok beszerzése. Adatokat lehetséges különböző forrásokból összegyűjteni, mely lehet az adatok vásárlása harmadik személytől, az adatok munkahelyen belüli kollekciója, internetes letöltése stb. Az adatok begyűjtése után az adatok transzformációjára van szükség.
Mivel az alkalmazni kívánt adathalmaz gyakorta nem áll megfelelő formában rendelkezésre, így azok tisztítása, transzformációja, tehát előfeldolgozása indokolt. Azon felül, hogy egy adott gépi modell teljesítményét is képes fokozni, a magas minőségű adattranszformáció ellenálóbb lehet külső támadások ellen, ahol az alkalmazás funkcionalitásának megkárosítása a támadó célja [17].

A gépi tanuló eljárások adatfeldolgozó szakaszát követően, a fejlesztési szakasz veszi kezdetét. A fejlesztési szakaszban kiválasztásra kerülnek az alkalmazni kívánt modellek, majd előre felállított jóságkritériumok mentén azok kiértékelése annak érdekében, hogy a teljesítményt mutatószámokkal ki lehessen fejezni. A jóság kritériumokra előzetesen metrikákat kell építeni, hogy objektivitása biztosítható legyen. A gépi tanuló módszerek jelentős része nagy mennyiségű paraméterrel rendelkezik, ezért a jóság metrikák legelső kiértékelése után szükségszerű kísérletet végezni, azaz „tuningolni” a modellek paramétereit az optimális teljesítmény elérése érdekében.

A modell építése és optimalizálása, valamint a jóságmetrikák megfelelő teljesítményének elfogadása után a gépi tanuló rendszer éleskörnyezetbe való állítása következik, melyben a rendszer üzemszerű működése kezdetét veszi. Összefoglalva és leegyszerűsítve a leírtakat, a gépi tanuló rendszerek általános fejlesztési folyamata 4 jól elkülöníthető részre tagolódik, amennyiben elvonatkoztatunk a kezdeti üzleti probléma specifikálásától, mely alapvetően általánosan igaznak tekinthető bármilyen szoftverfejlesztési folyamatra:

- Input adatok feldolgozása
- Modellezés
- Rendszer kiértékelése
- Üzemeltetés

A továbbiakban ezen meghatározott és általánosított fejlesztési folyamat mentén kerül részletezésre az IT audit bevonásának lehetőségei és a tipikusan az IT audit által elvégzendő feladatok meghatározása.

Input adatok feldolgozása

A gépi tanuló rendszerek adatokból táplálkoznak, ezért a helyes konklúzió leszűrése érdekében elengedhetetlen, hogy az input adatok megfelelő minőségben és mennyiségben rendelkezésre álljanak, továbbá, hogy az adatok között meghúzódó összefüggések hozzájáruljanak a modell predikciós erejéhez és ezáltal magas teljesítményű általánosító képességgel rendelkezzenek.

Ribeiro et al. [7] egy képfeldolgozó alkalmazást fejlesztettek, melynek a célja Husky kutyák és farkasok különválasztása és klasszifikálása volt. Az algoritmus magas pontossággal képes volt felismerni és megkülönböztetni az állatokat, azonban új, az algoritmus számára ismeretlen képek esetén az kudarcot vallott. Elemzésükből kiderült, hogy a rendszer csak olyan képekkel lett betanítva, melyen a farkasokat ábrázoló képeken hó is szerepelt, így a klasszifikáció sikertelen volt, amint egy olyan képet dolgozott fel, melyen Husky szerepelt egy havas tájon.

Számos kutatói munka javasolja, többek között [10][18]0, hogy bizonyos algoritmusok esetén az input adatokat transzformálni kell pl. standardizálni vagy normalizálni, hogy a gépi tanuló eljárás által használt költségfüggvényt idő- és költséghatékonyan

optimalizálni lehessen. Neurális hálók esetén például gradiens eljárást alkalmazva, az ugyanazon a skálán lévő input adatok lehetővé teszik, hogy a rendszer gyorsabban konvergáljon. Hasonló az eset klaszterezési eljárásoknál, ahol a különböző mérési skálán lévő adatok torzíthatják a végeredményt pl. az „év" és „fizetés" attribútumok esetén a „fizetés"-nek lesz nagyobb szerepe a különböző klaszterek létrehozásánál. Raschka [10] kiemeli, hogy nem minden algoritmus esetén szükséges előzetes adattranszformáció, pl. döntési fáknál az eredeti adatokkal is hatékonyan működik az algoritmus, sőt, így lehetséges értékes információt is szerezni a belső logikáról, hogy mely változók voltak, melyek legjobban hozzájárultak egy-egy predikcióhoz és milyen logika mentén történt a klasszifikáció.

Bizonyos algoritmusok pl. konvolúciós neurális hálók magas adatigénnyel rendelkeznek összehasonlítva egyéb gépi tanuló eljárásokat pl. SVM, mint, ahogy erre Andrew Ng [3] is rávilágít. Bár jelen kutatói munkákból nem derül ki egyértelműen, és nem is lehet meghatározni, hogy egy algoritmus milyen adatmennyiség mellett működik hatékonyan, olyan leíró adatok, mint a rendelkezésre álló változók száma, az algoritmus komplexitása (pl. neurális hálók esetén az architektúra nagysága, rejtett rétegek és neuronok száma) irányadó lehet az adatigénnyel kapcsolatban. Ba és Caruana [20] empirikus kísérletet végeztek el, és azt találták, hogy még neurális hálók esetén is lehetséges magas performancia elérése, jelen magas adatigényesség összefüggésbe hozható a mély architektúrákkal és azok tanítási eljárásaikkal. Azt feltételezik, hogy létrehozható olyan alacsony adatigényű algoritmus, melyhez kapcsolódó tanító eljárás magasabb pontossággal képes döntéseket hozni, tehát nem az adatban keresendő a probléma, hanem magában a tanító procedúrában.

A GDPR megjelenése óta a személyes adatok feldolgozása központi kérdéssé vált. Bár nem csak a gépi tanuló rendszerek esetén releváns, kiemelten fontos, hogy bármilyen adatfeldolgozás esetén szükséges hozzájárulást szerezni természetes személyektől az adatok tárolása érdekében, továbbá, hogy az adatfeldolgozás mögött megjelenjen az adatminimalizálás és jogos adatfeldolgozás érvényre juttatása. Az adatvédelmi törvények gátat szabhatnak az intelligens rendszerek felhasználásában, amennyiben az adatok felismerhetetlensége nem biztosított, továbbá az érzékeny adatok pl. vallás, politikai nézet, szexuális hovatartozás stb. további kockázatok jelenthet, mely kulturális rasszizmushoz, emberek közötti megkülönböztetéshez vezethet.

Számos kutatás pl. [3][10]0 az adatokat több részletre osztja még mielőtt bármilyen adatfeldolgozás kezdetét venné. A célja az adathalmaz felosztásnak, hogy a gépi tanuló rendszert a későbbiekben külön, még a rendszer által nem látott adatokon lehessen validálni és tesztelni. A felosztás általában 70%-10%-20%-os arányban javasolt (tréning adat, validációs adat, teszt adat). A cél, hogy a rendszert függetlenül lehessen ellenőrizni a teszt adatokon, a paramétereket, pedig tuningolni a validációs halmazon.

Mivel a gépi tanuló rendszerek alkalmazása számos iparágban és funkcionális területen képes automatizálására, ezért az auditornak mindenképp az adott szakterület vezetőjével szükséges egyeztetnie és megértenie a problémát, ahhoz, hogy megbizonyosodjon a rendszer helyes működőképességéről. A példák alapján tisztán látszik, hogy az input adatok validálása kritikus, és az auditornak az alábbi feladatokat kell ellátni:

- Meggyőződni, hogy az input adatok megfelelően transzformálva lettek, tesztelnie kell a transzformációs logikát, megvizsgálni, hogy a transzformált adatok alkalmasak a használt gépi tanuló algoritmusok általi feldolgozásra. Szükség esetén reperformálhatja az adattranszformációt és összehasonlíthatja az eredményeket.
- Megértenie az üzleti problémát, és ésszerű biztosítékot kell szereznie, hogy az adatok megfelelően képesek reprezentálni az adott szakterületi problémát.
- Meggyőződni, hogy az adatok megfelelő és hiteles forrásból származnak.
- Benchmark adatok alapján megbizonyosodni, hogy a későbbi adatfeldolgozáshoz megfelelő mennyiségű adat rendelkezésre áll.
- Meggyőződni, hogy a beszerzett adatok nem tartalmaznak másokra nézve sértő vagy diszkrimináló attribútumokat, az adatgyűjtés legális volt, adott személyekről gyűjtött adatok esetén megtörtént az adathozzájárulás.

Modellezés

A statisztikai adatfeldolgozás legnagyobb előnye, hogy képes magyarázatot adni, hogy mi, miért és hogy történik, míg bizonyos gépi tanuló algoritmusok „fekete doboz"-ként funkcionálnak, így az ok-okozati összefüggések felderítése ellehetetlenül. Ilyen algoritmus pl. a neurális háló, vagy az SVM. Fekete doboz algoritmusok esetén az algoritmus teljesítőképessége mérhető, azonban, hogy miért történt egy specifikus döntés, az nem interpretálható. Pl. jelenleg nem értjük, hogy a neurális hálók egyes neuronjai milyen mértékben járulnak hozzá a végleges output meghatározáshoz. A felmerülő probléma, hogy efféle alkalmazások tekintetében, mégis milyen eszközöket alkalmazhat az auditor, ha bizonyosságot szeretne szerezni, hogy az alkalmazások támogatják az üzleti célokat? Az ISACA [5] szerint, a válasz nem az algoritmusok belső struktúrájának felderítésében keresendő, hanem maga az egész rendszerfejlesztési és alkalmazási folyamatban.

Wang és Kosinski [21] kísérletet végeztek, melyben képek klasszifikációját végezték el, mely emberek arcait ábrázolta. A kísérlet arra összpontosított, hogy milyen mértékben lehetséges megállapítani egy adott személyről az ő szexuális beállítottságát. A rendszer 91%-os pontosságot ért el férfiak, míg 83%-os pontosságot ért el nők esetében. A kísérletet kiterjesztették és valós személyeknek kellett eldönteni ugyanazt, mint a felhasznált gépi tanuló rendszer, az ő teljesítményük 61% volt férfiak, 54% nők esetében. A felhasznált algoritmus neurális hálók volt.

A kutatás több jelentős dologra is felhívja a figyelmet. Elsősorban, mivel az algoritmus fekete doboz, ezért a döntési logika nem értelmezhető, annak csak a pontossága ellenőrizhető, ami azt jelenti, hogy valóban lehetséges szignifikáns döntést hozni az emberi arc alapján a felhasznált célszemélyek szexuális hovatartozására vonatkozóan. Másodsorban elgondolkodtató, hogy ha nem értjük ezen algoritmusok belső működését, akkor mi akadályozhat meg egy algoritmust egy üzleti környezetben, hogy az ne az illetékes jogszabályok kikerülése mellett legyen képes egy üzleti folyamatot automatizálni, vagy az implementáló szervezet reputációnak veszteségére negatívan diszkrimináció árán teremtsen „értéket" a vállalatnak.

Továbbá, minden gépi tanuló algoritmus rendelkezik előnyökkel és hátrányokkal, ezért feltétlenül fontos, hogy legyen egy kezdeti felvetés az adathalmaz birtokában, hogy mi az elvárt eljárások teljesítőképessége és melyik lesz majd preferált. Gyakorta több algoritmus is felhasználásra kerül, mivel jelenleg nincs olyan módszer, mely explicit megmondaná, hogy melyik eljárás fog adott adathalmazon tökéletesen működni, ezért empirikus kísérletekre van szükség, vagy aggregált módszerek alkalmazására, mely együttesen veszi figyelembe az algoritmusok eredményeit 0.

Modellezés kapcsán felmerülhet, hogy a forráskód teljesen a nulláról lett megírva vagy bármilyen sztenderd könyvtár alkalmazva volt. Jelenleg számos keretrendszer elérhető pl. Keras, TensorFlow, Scikit-Learn stb. melyek éleskörnyezetben való alkalmazása licenszdíjjal járhat.

Az auditornak a modellezéssel kapcsolatban meg kell győződnie, hogy:

- A kiválasztott modellek alkalmasak az üzleti probléma megoldására.
- Fekete doboz algoritmus alkalmazása esetén az eredmények megfelelően interpretálhatók és minden olyan kockázat eliminálva van, mely nem-megfelelősséghez, vagy etikai kérdéshez vezethet.
- Az alkalmazott keretrendszer jogosan került felhasználásra beleértve a licensszel járó költségeket.
- A fejlesztés bizonyos verziói megfelelő verziókezelő rendszerben kerültek letárolásra.
- A forráskód kizárólag a vezetés által jóváhagyott személyek részére elérhető, mindenféle módosítás rögzített és felülvizsgált.

Rendszerkiértékelés

Az implementáció előtti lépés a rendszerkiértékelés, melyben az implementáló szervezet megbizonyosodik, hogy a gépi tanuló rendszer elérte az előre meghatározott követelményrendszert (pl. jobban teljesít, mint egy ember) és megfelelő általánosító képességgel rendelkezik, azaz új problémák esetén is képes elfogadható döntést hozni. A pontosság (mint mérőszám) számos esetben egy jól bevált metrika, azonban annak értelmezése korántsem egyértelmű.

A 95%-os pontosság sok esetben elfogadható, azonban, ha a felhasználni kívánt adathalmaz, mint pl. az UCI machine learning repository-ban megtalálható csődelőrejelzéssel kapcsolatos adatbázis [22], egy adott megoldandó problémára relatív kevés negatív példát képes felsorakoztatni, akkor a 95% kiindulóteljesítmény is elégtelen lehet. Az említett adathalmaz kizárólag 2091 csődbement szervezetre tartalmaz adatot, mely összehasonlítva 41314 olyan vállalattal, mely az adatgyűjtés időszakában üzemelt, az 4.8% és 95.2% arány, tehát, ha egy algoritmus pontosságát nézzük egy előre elkerített tesztadaton, akkor az adatfeldolgozást követően legalább 95%-os pontossággal fog a rendszer előrejelezni.

Andrew Ng [3] különböző mérőszámok mentén értékeli a gépi tanuló rendszerek teljesítőképességét. Először is azt javasolja, hogy szükséges meghatározni egy elvárt pontosságot (ami nem feltétlen 100%). Amennyiben a tréningadatokon nyújtott teljesítmény elmarad az elvárt pontosságtól, akkor azt „elkerülhető bias"-nak nevezi, mely több adat beszerzésével és az algoritmus finomhangolásával csökkenthető.

A tesztadatokon végzett eltérés az algoritmus „varianciája", mely magas érték esetén jelentheti, hogy az algoritmus túlilleszkedett a tréningadatra, memorizálta az összefüggéseket, vagy megtanulta a véletlen zajokat. Ng [3] kiemeli, hogy akkor megbízható egy algoritmus, amikor a bias és variancia értéke ugyanakkor alacsony, azonban általánosságban az egyik érték csökkentése, a másik érték növekedésében testesül meg. Az előzőeket szemléltetve, tételezzük fel, hogy a cél egy alkatrészeket minőségbiztosító robot elkészítése. Az emberi hiba értéke 10%, a vezetés pedig eldönti, hogy mivel a világ legjobb robotja is 5%-ban téved, ezért az elvárt teljesítőképességet ugyancsak 5%-ban határozza meg, ami ezesetben az „elkerülhetetlen bias".

Amennyiben a robot a tréningadatokat elemezve 7% (bias) hibát tévesztett, akkor az „elkerülhető bias" mértéke 2%. Ha a robot a tréningadaton mérve összesen 15% hibát vétett, akkor a „variancia" mértéke 8% (15% - 7%). A magas bias azt jelenti, hogy a rendszer alulilleszkedik, azaz nem sikerült az adatok közötti összefüggést megtalálnia. Mindezen metrikákon felül számos más jóságmérték is elképzelhető. Bizonyos esetekben az algoritmus pontossága másodlagos lehet, ha a magyarázó képessége által új érték teremthető, de gyakorta alkalmazott az igazságmátrix, és az F1 pontszám, mely a pontosság és precizitás egy másik mérőszáma. Az auditornak a modell kiértékelésével kapcsolatban bizonyosságot kell szereznie, hogy:

- A rendszer teljesítőképességének mérése megbízható és az üzleti céllal összhangban lévő mutatószámok alapján lett meghatározva. A bias és variance mértéke elfogadható. A rendszer nem illeszkedik túl a tesztadatokra, és nem is illeszkedik alul (mintavételezés).
- A rendszer új, még nem látott adat esetén is megfelel a vezetés által elvárt metrikáknak. Ebben az esetben az auditor is tesztelheti a rendszert, egy a modellalkotásban még nem felhasznált adathalmazon.
- A rendszer körültekintően tesztelve lett technikai szakértők és üzleti felhasználók által is, a tesztek dokumentálva lettek és a felelős üzleti terület azt elfogadta.

Nyomon követés

A nyomon követési fázis az implementáció utáni folyamatos rendszermonitorozás, amikor az alkalmazás már élesben üzemel. Online rendszerek esetén fontos, hogy a teljesítmény folyamatosan ellenőrizve legyen, mivel a rendszer, nagy valószínűséggel, további adatokat fog beépíteni és tanulási mechanizmusai révén feldolgozni. Az auditornak, kockázatokkal arányos módon, folyamatos teszteket kell végeznie, hogy meggyőződjön, hogy a gépi tanuló rendszer az éleskörnyezetben is az elvártaknak megfelel, nem történt olyan incidens, mely új hibát generált volna a rendszerben vagy annak környezetében, és a további adatfeldolgozás az előző pontokban leírtakkal összhangban történik.

Konklúzió

A cikk a szakirodalomból vett példák alapján összegezte azon a gépi tanuló rendszerek fejlesztési és üzemeltetési folyamatában szereplő kritikus pontokat, melyben IT audit szakértők bevonása indokolt lehet. Mivel a vállalati belső IT kontrollok hatékonysága semmilyen esetben sem maximálisan biztosított,

ezért azokat rendszeresen auditálni kell külső vagy belső IT auditorok által. A gépi tanuló alkalmazások fejlesztési folyamata eltér a tradicionális szoftverfejlesztési eljárásoktól, ezért az auditoroknak is képezniük kell önmagukat, hogy megértsék az intelligens rendszerek üzemeltetéséből adódó problémákat, és az audit tervekben szerepeltessék a biztonságot szavatoló kontrollokat. Mivel jelenleg nem létezik olyan egységes keretrendszer vagy szabvány, mely kitérne a gépi tanuló rendszerek kontrollkörnyezetére (tervezés, implementálás, ellenőrzés stb.), ezért javasolt azok mihamarabbi kialakítása, mely iránymutatásként szolgál a szervezetek részére. Javasolt továbbá a belső mesterséges intelligenciára vonatkozó stratégia és politika megalkotása is, mely vállalati szinten képes szabályozni az intelligens alkalmazások kialakításának és fenntartásának folyamatait, mind az adatfeldolgozási, modellezési, kiértékelési és nyomkövetési szakaszokban.

Referenciák

[1] Statista.com (2019): Revenues from the artificial intelligence (AI) market worldwide from 2016 to 2025 (in million U.S. dollars). Letölthető: https://www.statista.com/statistics/607716/worldwide-artificial-intelligence-market-revenues/

[2] Russel, J, Stuart és Norvig, Peter (2005): Artificial Intelligence: A Modern Approach. (3rd ed.). India: Pearson Education.

[3] Ng, Andrew (2018): Machine Learning Yearning. Technical Strategy for AI Engineers, In the Era of Deep Learning. (Draft ver.) USA: deeplearning.ai.

[4] Andrews, Whit., Sau, Moutusi., Dekate, Chirag., Mullen, Anthony., Brant, Kenneth., Revang, Magnus., és Plummer, Daryl (2017): Predicts 2018: Artificial Intelligence. Letölthető: https://www.gartner.com/document/3827163?ref=solrAll&refval=193910164&qid=780b332f7d9afba6f17865ea8b939339

[5] ISACA (2018): Audit and Assurance. Auditing Artifical Intelligence. Letölthető: http://www.isaca.org/Knowledge-Center/Research/Documents/Auditing-Artificial-Intelligence_res_eng_1218.pdf?regnum=

[6] Clark, Andrew (2018): The Machine Learning Audit—CRISP-DM Framework. Letölthető: https://www.isaca.org/Journal/archives/2018/Volume-1/Pages/the-machine-learning-audit-crisp-dm-framework.aspx?lipi=urn

[7] Ribeiro, Marco Tulio., Singh, Sameer., és Guestrin, Carlos. (2016). Why Should I Trust You? Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16.

[8] Molnár, Bálint és Kő, Andrea (2009): Információrendszerek auditálása. Az informatika és az információrendszerek ellenőrzési és irányítási módszerei. Corvinho Technology Transfer kft., Budapest.

[9] Vashisth, Shubhangi., Linden, Alexander., Idoine, Carlie., Hare, Jim., Sicular, Svetlana., Krensky, Peter., Shen, Nigel., Sallam, Rita (2017): Machine Learning: FAQ From Clients. Letölthető: https://www.gartner.com/document/3772097?ref=solrAll&refval=203325447&qid=e107afa5add2a6fdb686dd81

[10] Raschka, Sebastian (2015): Python Machine Learning. Packt Publishing, Birmingham.

[11] Dua, Sumeet és Du, Xian (2011): Data Mining and Machine Learning in Cybersecurity. Taylor and Francis Group, USA.

[12] Hastie, Trevor., Tibshirani, Rober., és Friedman, Jerome (2009): The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Second Edition. Springer Science, New York.

[13] Sagar, G. V. R. (2015): Modeling of Artificial Neural Networks using Evolutionary Algorithms. Lambert Academic Publishing, Germany.

[14] Maqueda, Ana, I., Loquercio, Antonio., Gallego, Guillermo,. Garcia, Narciso., Scaramuzza, Davide (2018): Event-based Vision meets Deep Learning on Steering Prediction for Self-driving Cars. Letölthető: https://arxiv.org/pdf/1804.01310.pdf

[15] Szepesné, Stiftinger, Mária (2010): Rendszertervezés 1. Az információrendszer fogalma, feladata, fejlesztése. Székesfehérvár, TÁMOP – 4.1.2-08/I/A-2009-0027, Nyugat-magyarországi Egyetem, Geoinformatikai Kar.

[16] Kovács, Zoltán (2009): Szimulációs eszközök és megoldások műszaki és gazdasági rendszerekben. II. „Innováció az egyetemi képzésben és kutatásban" Jubileumi Tudományos Konferencia. Balatonvilágos, előadásjegyzet.

[17] Bhagoji, Arjun, Nitin., Cullina, Daniel,. Sitawarin, Chawin., Mittal, Prateek (2017): Enhancing Robustness of Machine Learning Systems via Data Transformation. Letölthető: https://arxiv.org/pdf/1704.02654.pdf

[18] Rashid, Tariq (2016): Make Your Own Neural Network. Independently published.

[19] Barta, Gergő (2018): Implementing and Evaluating Different Machine Learning Algorithms to Predict User Localization by the Strength of User Devices' Wi-Fi Signal. Sefbis Journal XII./2018.
[20] Ba, Lei, Jimmy és Caruana, Rich (2014): Do Deep Nets Really Need to be Deep? Draft for NIPS 2014. Letölthető: https://arxiv.org/pdf/1312.6184v7.pdf
[21] Wang, Yilun és Kosinski, Michal (2017): Deep Neural Networks are more Accurate than Humans at Detecting Sexual Orientation from Facial Images. Letölthető: https://www.gsb.stanford.edu/sites/gsb/files/publication-pdf/wang_kosinski.pdf
[22] UCI Machine Learning Repository (2016): Polish companies bankruptcy dataset. Letölthető: https://archive.ics.uci.edu/ml/datasets/Polish+companies+bankruptcy+data

Barta Gergő doktori disszertációját „Mesterséges intelligencia módszerek alkalmazása az informatikai rendszerek biztonsági auditjában" témakörben írta a Magyar Agrár- és Élettudományi Egyetem doktorandusz hallgatójaként. Kutatási területe kiterjed a mesterséges intelligenciával ellátott szoftveres megoldások fejlesztésére, valamint az intelligens döntéstámogató rendszerek kockázatmenedzsmentjére. A Deloitte Zrt. Kockázatkezelési tanácsadás üzletágának menedzsere, főleg IT megfelelőségi, biztonsági és adatvédelmi projekteket irányít Magyarországon és nemzetközi szinten. Gergő nemzetközileg tanúsított információ-rendszer ellenőr, IT biztonsági vezető és mesterséges intelligencia szakértő. A My-X mesterséges intelligencia kutatócsoport állandó tagja.