AKADÉMIAI KIADÓ

UNIVERSITY of DEBRECEN

International Review of Applied Sciences and Engineering

DOI: 10.1556/1848.2024.00909

ORIGINAL RESEARCH PAPER

Kanban method in digital data processing

Mária Csernoch[1], Edit Gizella Szűcs[2*] and Domicián Máté[2,3]

[1] Department of Computer Science, Faculty of Informatics, University of Debrecen, Debrecen, Hungary

[2] Department of Engineering Management and Enterprise, Faculty of Engineering, University of Debrecen, Debrecen, Hungary

[3] DHET-NRF Sarchi Entrepreneurship Education, Department of Business Management, University of Johannesburg, Johannesburg, South Africa

Received: August 29, 2024 • Accepted: November 2, 2024

ABSTRACT

Current digital education operates as a push-system, despite the content and subject characteristics suggesting a need for a pull-system. One of the features of the push-system is a tool-centered approach, where the focus on tools, including both hardware and software, is the belief that it is possible to build knowledge inventory. The consequences of the push digital education approaches are that (1) the fundamental concepts of Computer Science are not being transferred, (2) the lack of the development of supporting methods, (3) and education does not seem to be interested in revealing the root causes and to be open for fundamental changes. This paper proposes the extension of the theory of the industrial pull systems to present as a potential solution to increase the effectiveness and efficiency of digital education and reduce or eliminate data processing inefficiencies generated by undereducated but misled end-users. In addition to theoretical discussions, it also delves into the detailed analysis, design, implementation, and testing of a real-world data processing problem. The presented problem, its analysis, solution, and accompanying discussion reveal how one of the tools (kanban) of lean production can be adapted to support the Just-in-Time philosophy in digital education, focusing on end-user programming.

KEYWORDS

lean, kanban, digital data processing, spreadsheet-programming, pull-system digital education, Just-in-Time

1. INTRODUCTION

In the ever-evolving fields of industry and services, the revolutionary approach of lean production is rapidly gaining widespread acceptance. It is founded on the principle of respecting and serving people. The two mainstays of this approach are Just-in-time production and autonomation, which involves the prevention and elimination of errors with human intervention when required (autonomation) [1–6] (Fig. 1).

Digital education is still characterized by push systems, where the focus is on tools, including hardware and software tools. The widespread adoption of tools gives the impression that we are making progress both in digitization and digitalization. However, Ohno [1] and Wolfram [7] have argued that the tools do not necessarily solve the problems at hand but rather they can be misleading. This phenomenon is occurring in digital education, as it is in the contemporary digital world. Many people are under the misconception that it is enough to buy better and better tools and download more and more applications.

> *"Where we in the West would look immediately for some magic automatic miracle like computer integrated manufacturing (CIM), robotics, or advanced manufacturing techniques, the Japanese are simply reducing wastes. …"* [1]

> *"…when major new machinery comes along, it's rather disorientating." …"With no effective, general education in computational thinking, most people can easily be misled, and they are."* [7]

*Corresponding author.
E-mail: edit@eng.unideb.hu

AKJournals

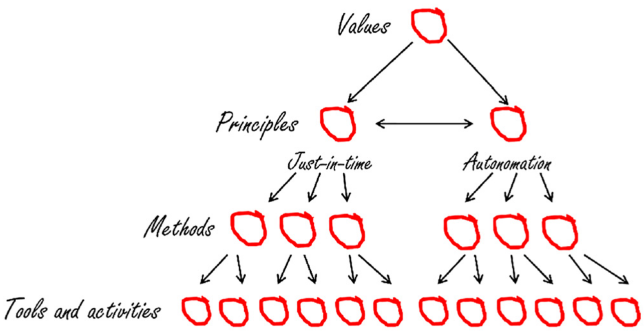


Fig. 1. The tree structure of the Toyota Production System (TPS)

In academic settings, the objective of teaching software programs and their user interfaces is to facilitate the construction of knowledge inventories that can be accessed when required, thereby enabling individuals to retrieve the necessary knowledge items to complete tasks (problems) [8]. Inventory is already one of the eight wastes in the industry [1, 4], but in an industrial environment, warehousing is a solvable problem. In contrast to the physical warehouse metaphor, the brain cannot store a large amount of unrelated data. The widely accepted school and training practices primarily apply the following approaches instead of focusing on problem-solving [9].

- Trying to provide educational institutes – schools, classrooms, students, teachers – and workplaces with as many tools and software as possible.
- Supports the publication of educational sources – printed and/or online textbooks, user manuals, built-in and external help tools, and online tutorials – that focus on devices – including hardware and software.
- Tests and quizzes focus on the recall of what is written in the tool-centered books and on the use of the tools, typically without content or with fictional, demotivating content [8].

In push digital education systems, building knowledge inventories can be the key to success in exams, pass DigComp tests based on unreliable self-reporting, and seem to make everyone happy. Schools go even further in self-delusion by projecting glitzy, shiny online tests that are supposed to motivate children. However, reality still needs to prove the effectiveness of these tools and methods. The reality is that tool-centered digital education needs to effectively cultivate the problem-solving and computational thinking skills and abilities of educators and students. Over time, the initial enthusiasm and novelty of such an approach may diminish, leading to a resurgence of the perceived monotony of traditional learning. What remains is the same: the erroneous digital artifacts (raw and edited data) whose creation and modification generate huge waste; the losses of human and computer resources.

1.1. Teacher beliefs

Chen et al. [10], in the Meaning System Model, summarize science teachers' beliefs on the nature of science and scientific knowledge (Fig. 2). The model also shows how teachers' belief affects students' knowledge and achievements, while the goal would be to lead students to Output 3.

In the context of digital education, both teachers and students tend to prioritize the attainment of Outputs 1 and 2. Today, many open-minded but digitally underqualified teachers – primarily "folk-teachers [11] – take their students to Output 4. Achieving Output 3 is an uncommon feat for students. It necessitates the comprehensive

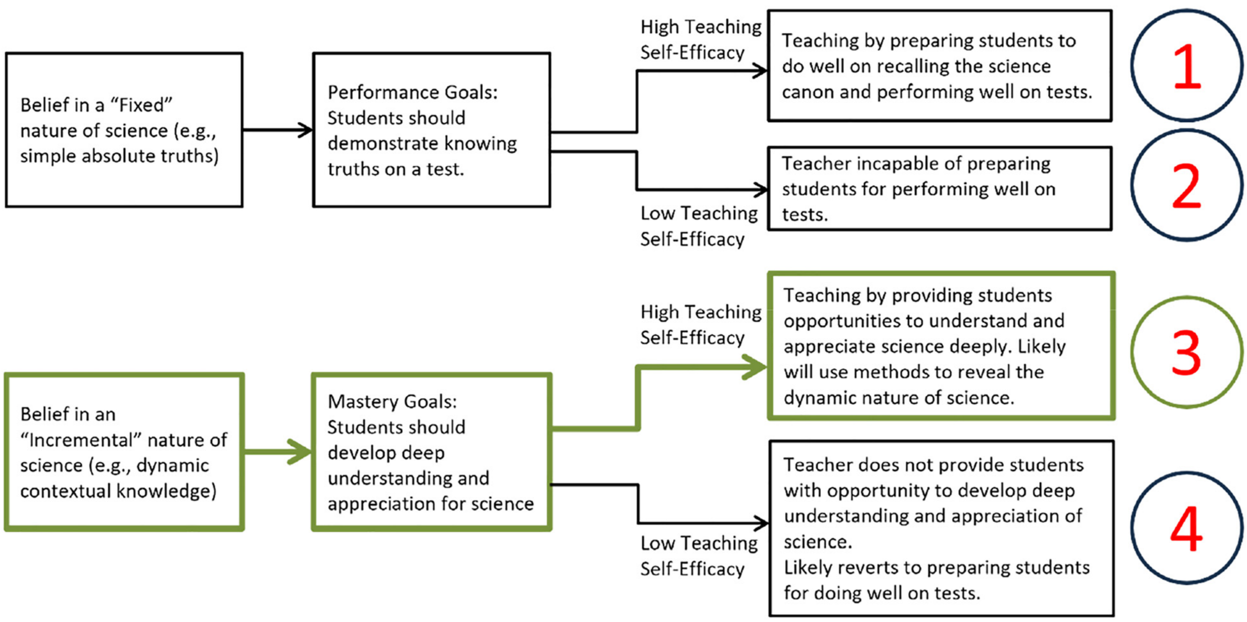


Fig. 2. Teachers' beliefs on the nature of science and scientific knowledge

integration of Technological Pedagogical Content Knowledge (TPCK or TPACK with TPCK plus Arts) into educational practices [12, 13], wherein computer-related knowledge implies the fundamental of Computer Science (CS) [14–19].

1.2. Born digital

Certified, serious computer science competitions (this category excludes competitions that are administered by teachers and schools targeting Outputs 1, 2 and 4 presented in Fig. 2) clearly show how tools influence the development of students' relation to computer science and how their computational thinking and digital problem-solving skills develop [20, 21]. It is also found that the expansion in the number of devices did not result in a proportional increase in the number of students and schools engaged in these competitions.

The notion that children are born with digital devices and possess a comprehensive understanding of their functionality has become a widely accepted premise. Some scholars posit that these children, having been born into the digital age, take its characteristics for granted. Therefore, they require no education or development. They assert that these children already possess a cognitive style that differs from that of the digital immigrants [22, 23]. Competitions and other scientific results [24–26] prove that digital natives are not born with CS knowledge and IT skills but with tools. Hence, the fundamentals of Computer Science (CS) [14–19] should be taught in a very conscious way to all ages.

1.3. Efficiency matrix

The current educational system in Hungary conceptualizes digital education as a means of imparting knowledge in the shortest possible time [27]. It should be emphasized that other countries are similarly afflicted with digital education issues on a global scale [28, 29]. However, in characterizing digital education, we must recognize the facts and achievements of specific subfields of CS. The objective of these courses is not to provide comprehensive coverage of all aspects of the subject matter but rather to focus on a specific subfield or area of specialization. This approach has the disadvantage of failing to encompass other subfields of CS. Based on the efficiency matrix (Fig. 3), the ideal circumstances would be to reach 'The perfect state' quarter. The 'Wasteland' is the most typical area, as well as the 'Efficient island' in specialized CS areas [30–32]. This approach also shows that efficient education of a subfield does not imply the efficient operation of the whole system – for example, programmers are usually very poor end-users, and CS ignores end-user computing [15, 30, 33, 34] practiced by billions of people on a daily basis, and especially CS-based end-user computing [30–41].

The details presented align with Scherer et al. [39], who emphasized the complexity of integrating technology into education and the variability in technological applications across schools [28, 29, 39–41]. Our research group demonstrated that one of the factors contributing to this

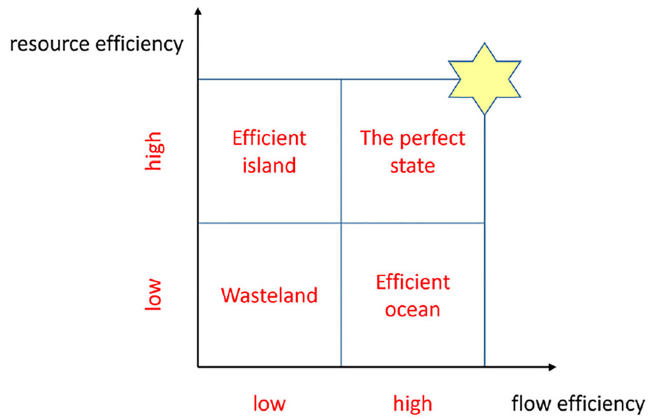


Fig. 3. Efficiency matrix

discrepancy is the previously referenced neglect of CS-based end-user computing. Many countries distinguish between Informatics and Digital Literacy [28, 29], and between Information Technology and Technology Literacy [40], leading to diverse frameworks. DigCompEdu [42] takes it a step further by defining digital resources that exclude data requiring analysis. Education systems focused on developing digital competence often prioritize tool-focused concepts [31, 41].

In addition, DigCompEdu only addresses Types 1 and 3 (Trouble-shooting and Target condition) [9], out of the four types of problems relevant to education, by stating that "[d]igital problem solving integrates learning activities, assignments, and assessments that demand learners to identify and solve technical problems, or to apply technological knowledge to new situations creatively" [42]. This definition overlooks Types 2 and 4, which involve identifying root causes and fostering innovation, respectively [9].

In this paper, we propose an approach that combines computer science and programming with end-user processes. This approach meets the requirements of both Informatics and Digital Literacy and accommodates a wide range of problems [9] from the outset of digital education. Our approach equips billions of end-users, not just informatics professionals, with the knowledge, skills, and abilities to carry out digital processes effectively and efficiently, as well as to create and modify digital artifacts. Additionally, it provides opportunities to assess and comprehend AI solutions and furnish both human and AI systems with clear data [38, 43–46].

The paper details an approach and a tool adapted from lean systems that have proven effective and efficient in industry and services [1–6]. However, digital education does not seem to be influenced by the philosophy, principles, methods, and tools of lean production. The paper argues that by embracing the philosophy and principles of lean production, processes and tools can be developed to serve all participants in the education system. It introduces a method and tool for use in classroom education and discusses the cognitive load involved in the problem-solving processes [47–49], as well as how fast and slow thinking [50] can be effectively and efficiently activated. The target condition of the selected problem is detailed in the following sections. The overarching goal is to establish a lean digital education system to reduce and eliminate waste [1–6] generated in current end-user processing [25, 26, 30, 38, 51–59] to support sustainable digital data management [55, 58, 59]. While this ambitious goal is beyond the scope of the paper, it allows for the introduction of one of the methods in their arsenal [60–66], which is in complete accordance with the kaizen philosophy [67–70] widely accepted in lean production [1–6] and aligns with the guidance required to find the balance in cognitive load [47–50].

The paper emphasizes the importance of aligning new educational theories with modern digital practices. It can be achieved by improving learner engagement through the application of lean production system principles, which have proven to be more effective in industry than other approaches. Additionally, the paper focuses on the importance of equipping end-users with essential digital skills, particularly emphasizing the development of computational thinking skills. It also discusses how the application of lean principles from industrial practices can offer valuable insights into structuring digital education for enhanced adaptability and responsiveness.

The study utilizes lean manufacturing principles, specifically the kanban methodology, to improve digital education processes. It demonstrates how these concepts can enhance educational approaches by addressing inefficiencies in digital education and offering solutions for challenges associated with push digital education approaches. The focus is primarily on tools, including both hardware and software. Additionally, the paper details effective data processing techniques in the field, targeting end-users and professionals in end-user roles. By integrating theoretical and practical aspects of Informatics, Computer Science provides a comprehensive examination, intention, and execution of real data processing issues, laying the groundwork for future applied engineering research and industrial practice.

2. MATERIALS AND METHODS: PUSH AND PULL DATA PROCESSING

One of the most popular data processing tools used by end-users is spreadsheet [34–36], including MS Excel. This paper compares push and pull digital learning-teaching approaches through a spreadsheet data processing problem. A push spreadsheet educational system is a decent one if it can present as many interface tools, commands and functions as possible, not infrequently including new features [37]. As many as possible also implies that many tools are detailed, including inappropriate or unnecessary. For instance, in a spreadsheet framework, the incorrect application of formatting includes the manual alignment of data, as this is an operation that is performed automatically by Excel once the data is recognized. If the alignment is manually changed, end-users lose the information that the alignment carries (left alignment indicates a text type, right alignment a number type). The alignment of the data can be considered as a simple andon – a simple signaling system to call attention to error [5] – that end-users would have to learn to read and use the information provided by it (Fig. 4).

2.1. Example: common failure in push systems

One of the most significant challenges associated with MS Windows and Excel is their inability to handle multiple languages effectively. This issue, which is particularly prevalent in Central European languages, can lead to significant data loss in numerous worksheets, regardless of the source of the data [38]. Based on the content of Fig. 5, we expect integer numbers in the **Uploads**, **Subs** and **Video Views** fields, as they store the number of uploads, subscriptions and views.

Figure 6 shows that the automatic data type recognition and the associated visual display (alignment) clearly indicate whether a number or a text type is stored. Accordingly, cells aligned to the left contain text, while those aligned to the right contain numbers. The discrepancy in alignment between the numerical data and the surrounding elements often needs to be clarified for end-users, who tend to align all the data uniformly to the center or to the right to facilitate comprehension. However, the visual One solution is to add 1 to the original values. The **Uploads1** and **VideoViews1** fields of Fig. 6 show the result of the additions. The result of the addition is an error value for text, while for real numbers, the integer part is incremented by 1. In the process of analyzing the data, it is important to be aware that integers have been lost for all values containing a comma (cells D34, D36, F6, F100 and F101). The most significant issue

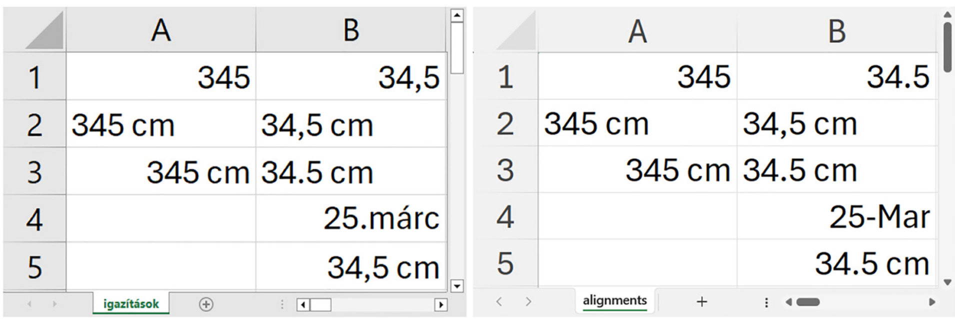


Fig. 4. The result of automatic data recognition in a Central European (Hungarian) and an English Excel: number (A1, A3, B1, B5), text (A2, B2, B3), date (B4)

Rank	Grade	Username	Uploads	Subs	Video Views
1st	B	Boutchoo!	289	548K	317,309,180
2nd	B	AndorraWorld	544	5.05K	63,365,715
3rd	B	Geminis tv	762	283K	239,840,399
4th	B-	Dance with aashi	153	5.66K	2,048,547
5th	B-	Trap - Freestyles & Noticias Del G...	269	6.83K	962,440
33rd	B-	E E	1,891	1.43K	1,617,122
34th	B-	Motovudu	146	23.9K	3,163,998
35th	B-	Impariamo italiano	1,828	15K	10,950,095
99th	B-	KAOTIC COC! TU MEJOR CANAL DE COC ...	11	959	727,284
100th	B-	vloger sumanta	233	668	106,936

Fig. 5. Andorra's YouTuber webtable on the Social Blade website[1] (Records 1–5, 33–35, and 99–100)

	A	B	C	D	E	F	G	H
1	Rank	Grade	Username	Uploads	Subs	VideoViews	Uploads1	VideoViews1
2	1st	B	Boutchoo!	289	548K	317,309,180	290	#VALUE!
3	2nd	B	AndorraWorld	544	5.05K	63,365,715	545	#VALUE!
4	3rd	B	Geminis tv	762	283K	239,840,399	763	#VALUE!
5	4th	B-	Dance with aashi	153	5.66K	2,048,547	154	#VALUE!
6	5th	B-	Trap - Freestyles & Noticias Del G...	269	6.83K	962,44	270	963,44
34	33rd	B-	E E	1,891	1.43K	1,617,122	2,891	#VALUE!
35	34th	B-	Motovudu	146	23.9K	3,163,998	147	#VALUE!
36	35th	B-	Impariamo italiano	1,828	15K	10,950,095	2,828	#VALUE!
100	99th	B-	KAOTIC COC! TU MEJOR CANAL DE CC	11	959	727,284	12	728,284
101	100th	B-	vloger sumanta	233	668	106,936	234	107,936

Andorra

Fig. 6. Andorra's YouTuber data table was incorrectly converted from the Social Blade website (Records 1–5, 33–35, and 99–100)

pertains to the fifth-ranked YouTuber (F6), as a zero has been omitted from the number during the automatic type conversion (Figs 5 and 6).

The 5W technique can be an invaluable tool in the field of data analysis [4–6] (Tables 1 and 2). The conversion of integers to both text and real numbers can be traced back to the comma. The reason for the comma is that, on the original website, the comma is a thousand-separator in English spelling, and the full stop is a decimal character. In the Hungarian (Central European) spelling, the decimal character is the comma (Fig. 6). If a single comma is present within the data set, Excel converts the integer to a real number.

[1]Social Blade: Top 100 YouTubers in Andorra sorted by SB Rank. https://socialblade.com/youtube/top/country/ad. Date of download: 04/12/2023.

Conversely, if the data set contains two or more commas, the data will be converted to text (Fig. 6) (Table 1). The following error [38] is prevalent yet regrettably overlooked within educational contexts despite its frequency. Overall, errors in Excel spreadsheets are due to end-users needing more knowledge and negligent data management practices. The example demonstrated the importation of data into Excel but did not convey the requisite knowledge for data analysis or the generation of clean, valuable data. Consequently, the lack of CS basics is obvious. In the near future, one of the most significant challenges facing AI [45, 46] will be the processing of erroneous data, a consequence of the absence of adequate data cleansing [44]. It would be beneficial for educational institutions to prioritize data management and computer science (CS) fundamentals relevant to data management at the earliest opportunity [17].

Table 1. 5 whys (5W) to reveal why integers are converted to texts

Why are some integers converted to texts (F2:F6 and F34:F36)?	The presence of two commas within the numbers should be noted.
Why cannot Excel numbers have two commas (they were fine on the website)?	The website is in English, the comma is used as a thousand separator character, following English spelling rules.
Why didn't Excel automatically switch from English to Hungarian?	Excel cannot do this automatically; it cannot check the content of the data.
Why didn't the end-user check the correctness of the data before opening it?	The end-user does not know the difference between languages.
Why didn't the user prevent automatic data recognition?	The end-user needs to learn the different data types and Excel's data type recognition algorithm.

Table 2. 5W to decide why we lost a 6-digit integer and got instead a decimal fraction containing a 3-digit integer part and a 2-digit fraction part

Why is the result of the addition 963,44?	We added 1 to 962,44.
Why did the integer 962,440 become the real number 962,44?	Excel finds the closing 0 redundant and truncates the number.
Why does Excel find 0 redundant?	The final 0 of a decimal fraction is usually not presented.
Why did the original whole number become a decimal fraction?	The comma in the English web page is a thousand separator, while in Hungarian Excel, it is a decimal character.
Why is the comma left as a separator in the numbers?	During the conversion, it was not deleted by the user.
Why didn't the end-user delete the commas?	The end-user does not know the difference between languages.
Why didn't the user prevent automatic data recognition?	The end-user needs to learn the difference between data types and Excel's data type recognition algorithm.

The result of the correct data handling is shown in Fig. 7, where integers have been stored in the **Uploads** and **VideoViews** fields. However, the **Subs** field still contains text in those cases where the order of magnitude is specified by a letter.

The **Subs** field can contain three types of data (Fig. 8):

- integer,
- an integer or real number ending in K,
- integer or real number ending in M.

The next task (problem) is to convert the orders of magnitude K and M into numbers. The operation should be conducted within the context of a programming environment. In the present framework, Microsoft Excel represents a viable and straightforward option, wherein the coding is performed using the simplified functional programming language of Excel [37, 60, 64, 65, 76–83].

2.2. Pull system spreadsheeting

In spreadsheet tutorials and course books, the focus is on the tools: the graphical interface, commands, special features, and listing functions. In the following section, the requisite knowledge items for the utilization of spreadsheet functions are calculated. It should be noted that the parameters for the other functionalities are analogous. Excel offers nearly 600 functions, which is a huge number in itself. However, more than knowing the names of 600 functions is required to use them. Overall, the information listed below is needed to use (call) a function. Thus, 6 × 600 data are needed to use spreadsheet functions safely. However, it is impossible to learn such an amount of unrelated data; it cannot be stored in long-term memory [47]. In order to work effectively and efficiently with spreadsheet data, the information to be processed must be stored in a systematic format [47] and fast thinking is applied [50]. If schemata are not available, error-prone, slow thinking is activated [33, 50].

- name
- semantics
- arguments

	A	B	C	D	E	F
1	Rank	Grade	Username	Uploads	Subs	VideoViews
2	1st	B	Boutchoo!	289	548K	317309180
3	2nd	B	AndorraWorld	544	5,05K	63365715
4	3rd	B	Geminis tv	762	283K	239840399
5	4th	B-	Dance with aashi	153	5,66K	2048547
6	5th	B-	Trap - Freestyles & Noticias Del G...	269	6,83K	962440
34	33rd	B-	E E	1891	1,43K	1617122
35	34th	B-	Motovudu	146	23,9K	3163998
36	35th	B-	Impariamo italiano	1828	15K	10950095
100	99th	B-	KAOTIC COC! TU MEJOR CANAL DE COC ...	11	959	727284
101	100th	B-	vloger sumanta	233	668	106936

Andorra

Fig. 7. Andorra's YouTuber data table was correctly converted from the Social Blade website (Records 1–5, 33–35, and 99–100)

Rank	Grade	Username	Uploads	Subs	Video Views
41st	B	Matyas	1,067	526K	542,311,236
42nd	B	Videómánia	809	1.24M	611,384,243
43rd	B	JátékNet.hu	190	361	6,651,524
44th	B	Fireknight99	204	1.15M	657,369,520
45th	B	Not A Gamer	4	9.59K	6,174,753
46th	B	Artur & Tommy	26	19.1K	6,275,425

Fig. 8. Hungary's youtuber webtable on the Social Blade website[2] (Records 41–46)

- order of arguments
- range of arguments
- returned value (set of values)

While the majority of textbooks are selective in their content, they often present students with an overwhelming volume of data that may be challenging to retain fully. The current Grades 8–11 Digital Literacy textbook [71–74] in Hungary lists 66 functions (most of them are mentioned once = hapax legomena) associated with operators, reference types, diagrams, and other commands and features.

2.3. Pull system spreadsheeting

Similar to production and services [1–6], the Pull System Spreadsheet does not provide an exhaustive introduction to the full range of available tools. Instead, it focuses on a limited set of tasks (problems) and introduces only those tools that are necessary to solve the specific problem at hand. This approach includes data analysis, building algorithms, implementation, and testing [75]. In terms of the types of problem-solving, pull systems typically deal with Type 3 problems [9], whereas Type 1 and 2 problems may also appear as subtasks. Students classified as Type 4 are highly innovative and talented but require an educational approach that extends beyond the capabilities of traditional classroom settings to engage and challenge them effectively.

The steps of problem-solving listed above are in complete accordance with the widely accepted PDCA cycle in pull production systems [9], and the concept-based approach of Polya [75] proved effective and efficient primarily in teaching mathematics. In Polya's system, the analysis involves understanding the task/problem, interpreting the available data, interpreting the output, and exploring whether the expected output can be obtained from the available inputs. Accordingly, in education, the PDCA can be complemented by an introductory step (Understand, Plan, Do, Check, Act: UPDCA) to emphasize the importance of data analyses. Similar models are presented by many educators, and among them Wolfram's [7] and Schneider's [26] are most related to our concern. One such pull system spreadsheeting approach is Sprego (Spreadsheet Lego) [60], whose effectiveness has been demonstrated in a number of different educational settings [61–66, 76].

The given traits signify a training program for pull systems conducted within a Sprego environment.

- Sprego is strictly based on real data. Data sets are selected that match the age, background knowledge, and interests of the students, in complete accordance with TPACK [12, 13, 39 60–62].
- The file management is preferred and supported [39, 77] compared to typing data.
- The format, orderliness, and cleanliness of the data file to be processed depend on the background knowledge of the students and the objectives and tasks of the lesson [39, 60].
- The basic set of Sprego contains 12 functions [60].
- Only those tools are introduced and/or practiced which are required to solve the task [43, 60].
- Tasks are built on each other, giving students the opportunity to practice, and a limited amount of new knowledge items, considering the cognitive load, are introduced [47–50, 75].
- Ever improving: finding, analyzing, and comparing several possible solutions to a given problem is one of the first principles of Sprego in accordance with TPS [1–6].
- Handling errors plays a crucial role. The interpretation, analysis and handling of errors and error values are in accordance with jidoka [1–6]. Automation and autonomation work hand in hand.
- Discussing and practicing the basic concepts of CS related to the task.

In essence, Sprego [60] represents a simplified functional programming approach [35–37, 78–82]. From the perspective of end users, the course is beneficial for those who lack proficiency in high-level programming languages. From a pedagogical standpoint, it offers a foundation in database management and more advanced programming techniques. In this environment, the fundamentals of data management and programming can be learned [78], and schemata (algorithms) can be stored in long-term memory [47–50, 75].

The Sprego functions (Table 3) constitute around 2% of the total number of Excel functions. However, they can be

[2]Social Blade: Top 100 YouTubers in Hungary sorted by SB Rank. https://socialblade.com/youtube/top/country/hu. Downloaded on 2023. 12. 04.

Table 3. Sprego functions

Sprego Text	Sprego Number	Sprego Pro
LEFT()	SUM()	IF()
RIGHT()	AVERAGE()	ISERROR()
LEN()	MIN()	MATCH()
SEARCH()	MAX()	INDEX()

utilized to address the majority of issues pertinent to the domains of primary and general secondary education, with an emphasis on data management and programming. The Sprego set is open and can be extended when building a formula from its functions is impractical. Sprego functions can be nested within each other to create multilevel complex functions, which are useful for solving complex problems. With this solution, in complete accordance with Papert [14] and Wolfram [7], we are fully aligned with the ideas of computer science to teach higher-level mathematics. Taking all this into account, most of the functions introduced in textbooks are redundant since using the programming theorems [84], and Sprego functions (Table 3) [60], formulas for complex problems can be constructed easily.

3. RESULTS: CONVERTING SUBS DATA INTO NUMBERS

A synthesis of the data provided by the YouTubers discussed herein leads to the conclusion that in the **Subs** field, the letters K (kilo $= 10^3 = 1000^1$) and M (mega $= 10^6 = 1000^2$) indicate the order of magnitude of the numbers (Figs 5–9). In order to obtain the number of subscribers as an integer, it is necessary to convert these letters into a number. The following section will present the details of one potential solution, demonstrating the way a pull system can be integrated with a kanban system [1–6]. The expression kanban is adapted from TPS. "At Toyota every step of every manufacturing process has the equivalent of a gas gauge built in, (called kanban), to signal to the previous step when its parts need to be replenished. This creates pull which continues cascading backwards to the beginning of the manufacturing cycle." [5].

3.1. Kanban diagram for calculating the subs values as integer

The approximate value of subscribers can be calculated based on the following kanban diagram, which illustrates the stages of the subscriber lifecycle (Fig. 10). The figure shows a possible solution, but as mentioned in the previous section, several further solutions can be accepted.

The objective is to print the Subs values as numbers (integer). To achieve this, it is first necessary to determine whether the Subs value contains a letter (K or M). In the case that it does not, the original value may be written out. In the instance that a letter is present within the original Subs value, the subsequent Subs value, represented by a number,

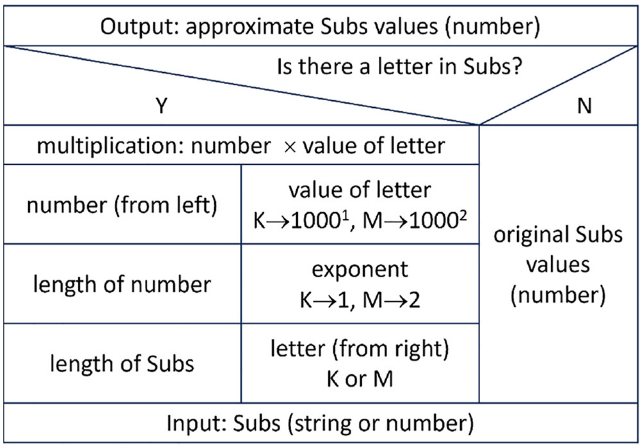
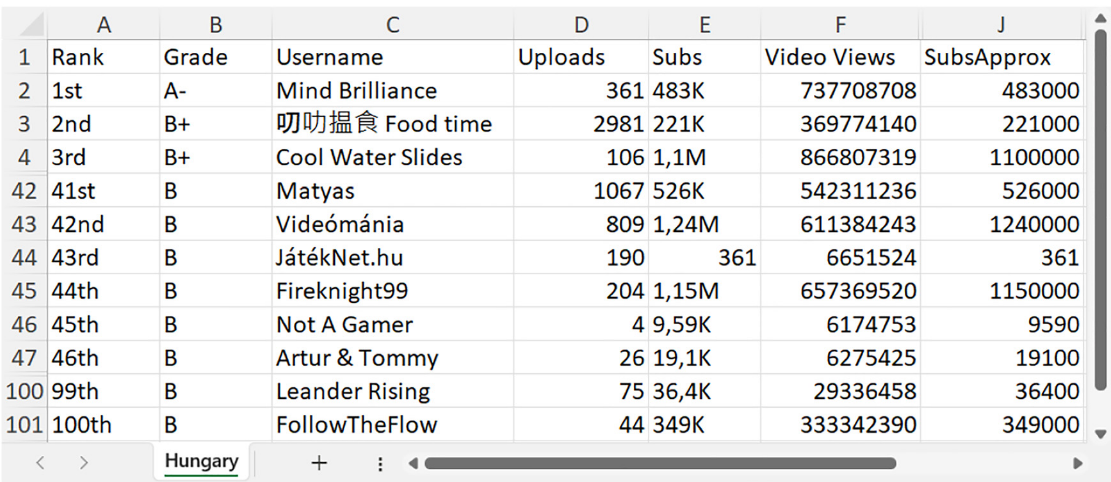


Fig. 10. One possible kanban diagram for calculating the approximate **Subs** values. From the top to the bottom, the output should be a number. To fulfill this requirement, we must know whether there is a letter in the original Subs until we reach the input values at the bottom



	A	B	C	D	E	F	J
1	Rank	Grade	Username	Uploads	Subs	Video Views	SubsApprox
2	1st	A-	Mind Brilliance	361	483K	737708708	483000
3	2nd	B+	叨叻揾食 Food time	2981	221K	369774140	221000
4	3rd	B+	Cool Water Slides	106	1,1M	866807319	1100000
42	41st	B	Matyas	1067	526K	542311236	526000
43	42nd	B	Videómánia	809	1,24M	611384243	1240000
44	43rd	B	JátékNet.hu	190	361	6651524	361
45	44th	B	Fireknight99	204	1,15M	657369520	1150000
46	45th	B	Not A Gamer	4	9,59K	6174753	9590
47	46th	B	Artur & Tommy	26	19,1K	6275425	19100
100	99th	B	Leander Rising	75	36,4K	29336458	36400
101	100th	B	FollowTheFlow	44	349K	333342390	349000

Hungary

Fig. 9. Hungary's YouTuber data table with the approximation of the text and the integers values of **Subs**, presented in Field **SubsApprox** (Records 1–3, 41–46, and 99–100)

is derived by multiplying the number located to the left by the value that replaces the letter to transcribe the number located on the left-hand side of the original string.

In order to ascertain the requisite number of characters required for its transcription, it is first necessary to ascertain its length. To calculate the length, it is first necessary to ascertain the length of the original string. It can be demonstrated that the length of the number is one less than the length of the original string.

In order to calculate the value of the letters, it is necessary to determine the exponent of 1000 that the letter represents. To determine a value raised to a power, understanding the exponent is essential. In order to compute the exponent, the base value must be identified. The letter is derived by extracting a single character from the right side of the original string.

Our objective is to express the approximate Subs values as numerical figures. Keeping this goal in mind, we can ascertain from the kanban which previous values are necessary to complete each step. The kanban diagram may also indicate that the available input is adequate to provide an approximate value for the number of subscribers. Additionally, it is important to understand that the letters K and M represent powers of 10 and/or 1000.

3.2. Design

In a digital environment, planning primarily means writing an algorithm. Algorithms can take many different forms; one of the most common of them is the block diagram (flowchart[3]). Three such algorithms are shown in Figs 11 and 12.

Figure 11 shows the two algorithms that can be used to calculate the two factors of multiplication. In both cases, we define a function, namely NUMBER(string Subs) and LETTER(string Subs). These functions perform two of the subtasks. Both functions receive as a parameter the string Subs. In the LETTER(string Subs) function, we need to specify the corresponding value to the letters K and M which are 1 and 2, respectively.

The output (returned value) of the NUMBER(string Subs) function is a number string (a string that contains only number characters), while the LETTER(string Subs) function

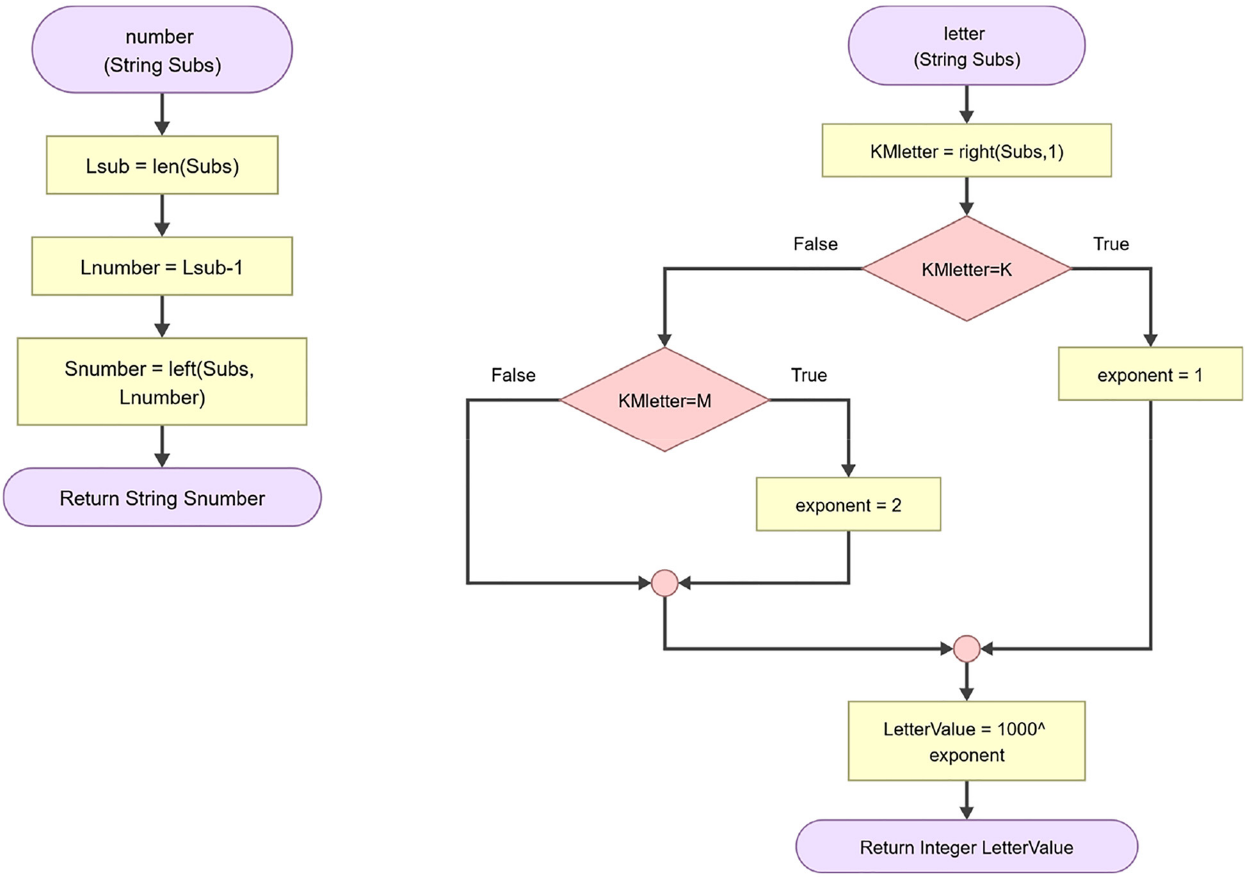


Fig. 11. Algorithm for separating the numbers (left) and the letters and converting letters into numbers (right)

[3]Flowcharts are created by Flowgorithm.

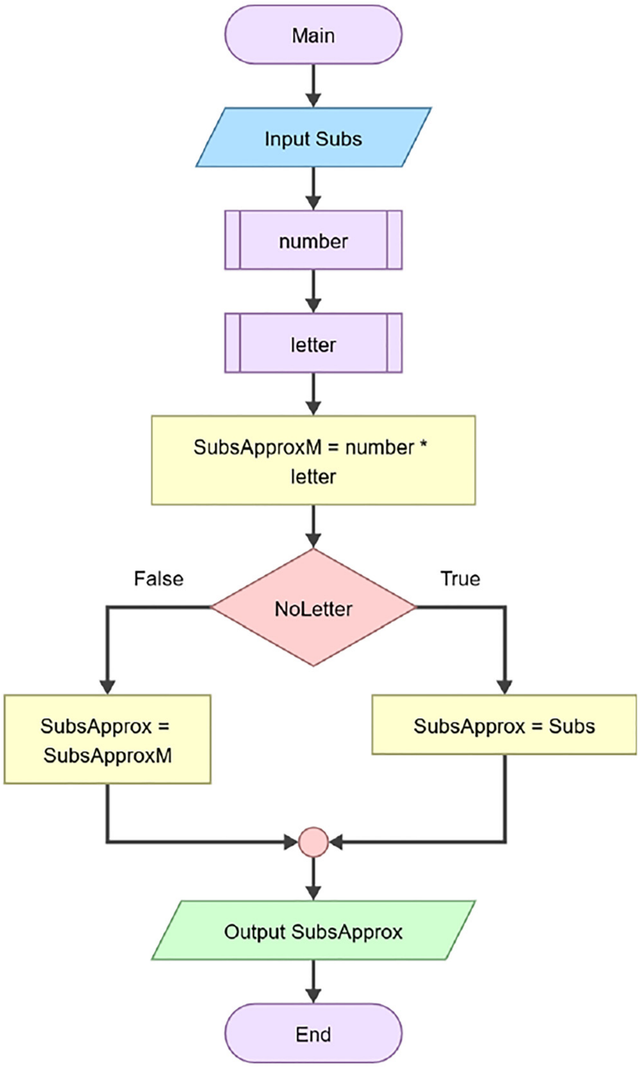


Fig. 12. The algorithm of the MAIN() function that calculates the approximate subs and checks their correctness

returns an integer. Why is the string not already converted to a number in the NUMBER(string Subs) function? While it is possible to perform this operation in Excel, it is unnecessary because the software interprets a string that resembles a number as such and performs the multiplication operation on it. It affords us the opportunity to bypass one stage within the process.

In the algorithm of the LETTER(string Subs) function, there are two consecutive questions to be answered. The classical implementation of this is carried out with two nested IF() functions. In Excel, the problem can be solved much simpler, with a linear search. We use this solution for the implementation, where the MATCH() function is called instead of the nested IF() functions.

The MAIN() function is the third algorithm (Fig. 12), which performs three tasks

- calls the functions NUMBER(string Subs) and LETTER(string Subs),
- multiplies the value returned by the two functions, and
- decides whether to write out the original Subs or the value calculated by multiplication.

3.3. Coding

The implementation is achieved through the invocation of Excel functions and the utilization of mathematical operators. The implementation is distinguished by the fact that the approximate substitutions for the 100 YouTubers are not calculated individually (by copying the initial formula), but rather using 100-element vectors employing array formulas. The advantage of this solution is that a single formula is created, which performs the same operation for all elements of the vector, similar to the loops known from programming (the index runs from 2 to 101 in the formulas). The vector solution is demonstrably safer than the widespread practice of copying. Array formulas are a lean tool in Excel that has been available since the software's inception, yet they have been largely unused, except for a few enthusiastic Excel programmers [37, 60, 79–82]. The introduction of dynamic array might increase the popularity of array formulas and the concept of loop in Excel [83]. The introduction of Sprego [60–66, 76] has revitalized array formulas, and there is potential for a major advancement with dynamic array formulas, currently a standard feature in Excel. Approximate substitutions can be computed using multilevel functions or individually, employing auxiliary cells and vectors. In this approach, each step is executed independently, generating a new field for each output. This method requires more space but offers greater transparency.

3.4. Writing out number strings

The steps for writing out the number strings are presented in Formulas Lsub, Lnumber, and Snumber, while the outputs in Fig. 13 in Fields **Lsub**, **Lnumber** and S**number** (Figs 10 and 13).

Lsub	G2:G101	{=LEN(E2:E101)}

In Field **Lsub** the length of the original Subs values is computed to determine the character count of these data. The number of characters includes the digits and number characters, the decimal comma and the K or M letters at the end of the string, which means a thousand or a million. The LEN() function does not distinguish between number and string data types; in both cases, it gives output, which is an integer.

Lnumber	H2:H101	{=LEN(G2:G101)-1}

From the length of **Subs**, the length of the numbers can be calculated (**Lnumber**), which is one less than the total length (**LSubs**). In this step, we ignore the fact that there are values that do not contain a letter.

	A	C	E	G	H	I
1	Rank	Username	Subs	Lsub	Lnumber	Snumber
2	1st	Mind Brilliance	483K	4	3	483
3	2nd	叨叻揾食 Food time	221K	4	3	221
4	3rd	Cool Water Slides	1,1M	4	3	1,1
42	41st	Matyas	526K	4	3	526
43	42nd	Videómánia	1,24M	5	4	1,24
44	43rd	JátékNet.hu	361	3	2	36
45	44th	Fireknight99	1,15M	5	4	1,15
46	45th	Not A Gamer	9,59K	5	4	9,59
47	46th	Artur & Tommy	19,1K	5	4	19,1
100	99th	Leander Rising	36,4K	5	4	36,4
101	100th	FollowTheFlow	349K	4	3	349

number

Fig. 13. The steps of writing out the numbers as a string data type in Field **Snumber** are based on the algorithm of the NUMBER() function

Snumber	I2:I101	{=LEFT(E2:E101,H2:H101)}

In the third step, we write out **Lnumber** characters from the left side of **Subs** to obtain the **Snumber** values with the LEFT() function, which has two arguments: **Subs** and **Lnumber**. The data type of the output is a string.

3.5. Writing out the letters that indicate the order of magnitude

The next step is to write out the letters (K or M) that indicate the order size. It is important to note that the number strings and the letters can be written interchangeably, as shown in the kanban diagram and the algorithms (Figs 10–12). According to the algorithm, we calculate the **KMletter**, the **exponent**, and the **LetterValue** fields, while K and M in Cells N2 and N3 are stored to serve the linear search to get the exponents (Figs 10 and 14).

KMletter	J2:J101	{=RIGHT(E2:E101)}

The first step is to write out the letter (K or M) to calculate its value. The letter is positioned at the end of the **Subs** string (on the right side of the string) and is 1 character long. Accordingly, we can perform the printout by calling the RIGHT() function. The function only needs to have one argument, which is **Subs**. The second argument of the RIGHT() function can be omitted since the number of characters to display is 1, which is the default value. Using the default value, we can use a shorter form of the function.

exponent	K2:K101	{=MATCH(J2:J101,N2:N3,0)}

After the writing of the letter, the powers of K and M over 1000 are calculated. The algorithm suggests the use of nested IF() functions (Fig. 11), but Excel has a simpler solution, the linear search algorithm. In order for the algorithm to work in an efficient way, we add the two possible values in an auxiliary vector (Fig. 14). It implies that the function returns 1 for the letter K and 2 for the letter M. These numbers give the exponent of 1000. If there are no

	A	C	E	G	H	I	J	K
1	Rank	Username	Subs	KMletter	exponent	LetterValue		
2	1st	Mind Brilliance	483K	K	1	1000		K
3	2nd	叨叻揾食 Food time	221K	K	1	1000		M
4	3rd	Cool Water Slides	1,1M	M	2	1000000		
42	41st	Matyas	526K	K	1	1000		
43	42nd	Videómánia	1,24M	M	2	1000000		
44	43rd	JátékNet.hu	361	1	#N/A	#N/A		
45	44th	Fireknight99	1,15M	M	2	1000000		
46	45th	Not A Gamer	9,59K	K	1	1000		
47	46th	Artur & Tommy	19,1K	K	1	1000		
100	99th	Leander Rising	36,4K	K	1	1000		
101	100th	FollowTheFlow	349K	K	1	1000		

letter

Fig. 14. The steps of calculating the value of the letters K and M are based on the algorithm of the LETTER() function

matches, an error value is obtained, which serves as andon in this situation (Fig. 14: Cell K44).

LetterValue	L2:L101	{=POWER(1000,K2:K101)}

By converting the letters K and M to numbers, we can now calculate their exact values, which can be done simply with the POWER() function. The output of the function is 1,000, 1,000,000 or an error value, depending on the input of the value stored in **exponent** (Fig. 14).

3.6. Writing out the approximate subs values

The values calculated in the previous two steps are required for the evaluation of the **SubsApprox** values. First, the **number** and the **letter** values are multiplied. The result is an integer or an error value presented in **SubsApproxM** (Fig. 15). In accordance with the algorithm, the subsequent step entails determining whether the letters in question were successfully identified. In Excel, this is achieved by ascertaining whether the linear search yielded an error value.

SubsApproxM	M2:M101	{=I2:I101*L2:L101}

The Field **SubsApproxM** stores the results of the multiplication of Fields **letter** and **number**. It provides an approximate value for the **Subs** field and an error value in cases where the value of the **Subs** field is a number. It should be noted that neither K nor M is present at the end of the original data.

NoLetter	N2:N101	{=ISERROR(K2:K101)}

The next step is to ask whether the linear search resulted in an error value, which can be carried out with ISERROR() function. The answer to this yes/no question is TRUE or FALSE, depending on whether the original Subs field contained a number or a string, respectively.

SubsApprox	O2:O101	{=IF(N2:N101,E2:E101,M2:M101)}

In the final stage of the process, a decision is made regarding the content to be included in the two distinct scenarios (**SubsApprox**). If the linear search resulted in an error value, then there is no letter in the original **Subs** value, in which case the original **Subs** number is written out. If the linear search found a letter K or M, then the result of the multiplication is written out from Field **SubsApproxM**. The output is either the original number less than 1000 or an approximation of the original **Subs** value as a number.

3.7. Discussion

The preceding sections presented a methodology for approximating the value of **Subs**. According to the original problem, the analyses revealed that there might be several different solutions, algorithms, and implementations. First, the kanban diagram was set up to reveal whether the expected output can be reached from the input which arrived from a webtable (Fig. 8). In the conversion process [43], the thousand separator characters were removed from the numbers to match the requirements of both the English and the European languages. The output of the conversion process was a 1st normal form table with 6 fields and 100 records.

In accordance with the kanban diagram, an algorithm was established comprising a MAIN() function and two subordinate functions., NUMBER() and LETTER(). Based on these algorithms, the implementation of the problem was carried out with the LEN(), LEFT(), RIGHT(), MATCH(), ISERROR(), and IF() Sprego functions and one supplementary function, POWER(). The POWER() function is typically employed for solving maths problems. However, as the illustration demonstrates, it may also be utilized for text manipulation. Given its relatively limited prevalence, it is not a particularly notable function and, therefore, merits no designation as a Sprego function.

Considering the output values, it is important to note that only values less than 1000 give the exact results; in all

	A	C	E	I	L	M	N	O
1	Rank	Username	Subs	number	letter	SubsApproxM	NoLetter	SubsApprox
2	1st	Mind Brilliance	483K	483	1000	483000	FALSE	483000
3	2nd	叨叻搵食 Food time	221K	221	1000	221000	FALSE	221000
4	3rd	Cool Water Slides	1,1M	1,1	1000000	1100000	FALSE	1100000
42	41st	Matyas	526K	526	1000	526000	FALSE	526000
43	42nd	Videómánia	1,24M	1,24	1000000	1240000	FALSE	1240000
44	43rd	JátékNet.hu	361	36	#N/A	#N/A	TRUE	361
45	44th	Fireknight99	1,15M	1,15	1000000	1150000	FALSE	1150000
46	45th	Not A Gamer	9,59K	9,59	1000	9590	FALSE	9590
47	46th	Artur & Tommy	19,1K	19,1	1000	19100	FALSE	19100
100	99th	Leander Rising	36,4K	36,4	1000	36400	FALSE	36400
101	100th	FollowTheFlow	349K	349	1000	349000	FALSE	349000

SubsApprox

Fig. 15. The steps of calculating the approximate Subs (**SubsApprox**) values are based on the algorithm of the MAIN() function. The input values arrive from the **number** and **letter** fields

other cases, an approximate value is the solution. It can be explained by the fact that only approximate values are given on the original web page, where the letters K and M represent the order of magnitude.

In the coding process, each step is presented in a separate field. However, using multilevel functions can reduce the number of fields. As previously stated, one of the benefits of multilevel functions is that they facilitate the integration and application of advanced mathematical principles in the context of practical problem-solving. The fundamental tenets of spreadsheet design advocate the utilization of built-in formulas [85, 86]. Nevertheless, as any rigorous programming course will teach students, it is vital to strike a balance between the level of complexity in a code and its readability and ease of maintenance.

Building multilevel functions requires end-users (students) to use the input and output values of functions and formulas. From this experience, end-users can gain insight into the transfer of values between functions, the role of enclosing functions (for example, the use of parentheses), and the knowledge that will prove beneficial when they call functions from diverse libraries and/or write, declare, and define their own functions. Furthermore, most of the Excel functions are n-ary functions with various data types as D_f and R_f, which knowledge and practice further serve in building up high-level mathematical concepts through problem-solving. Conversely, the reading, comprehension, decoding and explanation of multilevel functions is an invaluable aid in the development of students' programming abilities and skills.

Formula 1 – Formula 3 present three possible multilevel formulas to calculate the number part, the letter values, and the approximate values of Subs. After each step, the output values are checked to see whether they are correct. If any errors can be detected the process is stopped and not continued until the error is corrected. In the event of the emergence of irremediable errors, recourse is had to the algorithms or the kanban to identify alternative solutions. In the presented formulas, the red characters indicate the extension of a previously completed formula. The role of this process is of great consequence. Initially, the most inside formula is created. It is then subjected to rigorous scrutiny for accuracy. Once validated, the formula is then expanded in accordance with the specifications of the external function or operator.

Formula 1. Calculating the number part of the Subs values with a multilevel formula

Snumber	{=LEN(E2:E101)} {=LEN(E2:E101)-1} {=LEFT(E2:E101,LEN(E2:E101)-1)}

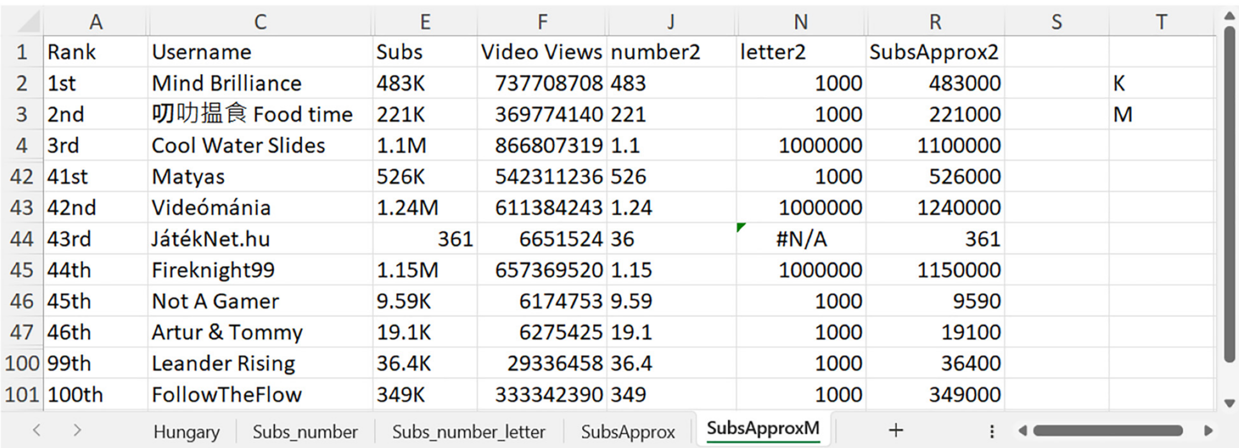
Formula 2. Calculating the letter values of the Subs values with a multilevel formula

LetterValue	{=RIGHT(E2:E101)} {=MATCH(RIGHT(E2:E101),S2:S3,0)} {=POWER(1000,MATCH (RIGHT(E2:E101),S2:S3,0))}

Formula 3. Calculating the approximate Subs values with a multilevel formula

SubsApprox	{=J2:J101*N2:N101} {=ISERROR(J2:J101*N2:N101)} {=IF(ISERROR(J2:J101*N2:N101), E2:E101,Q2:Q101)}

As previously outlined, the thousand separators and the decimal character represent a significant challenge in the data cleansing process. Deleting the thousand separator characters solves one problem. However, the decimal character is more complicated. The spreadsheet screenshots presented above are normalized to Central European Excel, which implies that during the conversion process, the decimal character is changed to a comma. The different languages are not compatible with each other. Consequently, autonomation (automation with a human touch) [1–6] plays a crucial role in getting the right database [38]. The English version of the cleaned data is presented in Fig. 16.



	A	C	E	F	J	N	R	S	T
1	Rank	Username	Subs	Video Views	number2	letter2	SubsApprox2		
2	1st	Mind Brilliance	483K	737708708	483	1000	483000		K
3	2nd	叨叨揾食 Food time	221K	369774140	221	1000	221000		M
4	3rd	Cool Water Slides	1.1M	866807319	1.1	1000000	1100000		
42	41st	Matyas	526K	542311236	526	1000	526000		
43	42nd	Videómánia	1.24M	611384243	1.24	1000000	1240000		
44	43rd	JátékNet.hu	361	6651524	36	#N/A	361		
45	44th	Fireknight99	1.15M	657369520	1.15	1000000	1150000		
46	45th	Not A Gamer	9.59K	6174753	9.59	1000	9590		
47	46th	Artur & Tommy	19.1K	6275425	19.1	1000	19100		
100	99th	Leander Rising	36.4K	29336458	36.4	1000	36400		
101	100th	FollowTheFlow	349K	333342390	349	1000	349000		

Fig. 16. The English version of the cleaned data and the outputs of calculating the approximate Subs without using additional fields for the partial results. Fields **numberMulti**, **letterMulti**, and **SubsApproxMulti** hold the multilevel functions

A similar problem is faced with separator characters of the CSV files, especially if the CSV extension is associated with Excel. The comma does not serve the European languages, so it is replaced with the semi-column. However, the semi-column does not serve the English spreadsheets. In the case of CSV files, the separator characters must be handled by the end-user (autonomation) to make sure that it is correctly transferred.

4. CONCLUSIONS

The present study introduces a computer science-focused end-user teaching methodology with a pull system, which follows a theoretical grounding and discusses the detailed solution of a spreadsheet problem. Prevailing yet ineffective push system approaches to end-user data processing. The focus of push system approaches is on the tools – including both hardware and software – where the primary objective of training is the demonstration of the capabilities of these devices and interfaces. The push system approaches adopted by schools, training systems, textbooks, and teachers primarily ignore humans, the principles of learning processes, the Cognitive Load Theory [47–49], how fast and slow thinking operates [50], in general, the problems that students (later end-users) should solve. The result is a proliferation of erroneous documents that are shared and distributed with inadequate scrutiny, thus perpetuating the cycle of errors at an accelerated pace [33–38, 51–59].

One of the consequences of incorrect document management is that the creation and modification processes generate enormous losses. The losses caused by these processes are still underdeveloped for various reasons. One of the prevailing challenges in the field is the need for more consensus on how to effectively manage, train, assist, motivate and develop end-users. It includes not only professional IT specialists in their role as end-users but also teachers and trainers [87–89]. The lack of a unified approach to end-user management is a significant obstacle to progress in this area. A second reason, to some extent a corollary of the previous one, is the lack of objective measurement tools for evaluating the effectiveness of documents and end-user activities. Some progress has already been made with Excel documents [33–38], where the consequences of errors can be immediately apparent.

Nevertheless, there needs to be more advancement in the processing of natural language digital texts, including text documents, presentations, and web pages. The creation of digital texts through bricolage results in significant losses in terms of both human and machine resources [51–59, 87–89]. It is, therefore, essential to quantify these losses and implement strategies to reduce them.

The fundamental premise of the proposed pull system is that it does not aspire to impart a comprehensive understanding of Excel, encompassing an exhaustive array of interface tools, commands, and functions. Instead, it seeks to identify algorithm- and programming-centric solutions to address specific tasks. The focus is on the problems for which it offers solutions that are in accordance with the fundamentals of CS. Consequently, only those tools are introduced that are necessary to solve the problem and the approach allows space for practices of previously introduced methods and tools to build schemata. The knowledge acquired through this pull system can be applied to solve additional problems in end-user environments effectively and efficiently. Furthermore, it provides students with the necessary foundation for pursuing advanced computer science studies by focusing on the transfer of fundamental CS principles.

Conflict of interest: Edit Gizella Szűcs is a member of the Editorial Board of the journal, therefore they did not take part in the review process in any capacity and the submission was handled by a different member of the editorial board. The submission was subject to the same process as any other manuscript and editorial board membership had no influence on editorial consideration and the final decision.

REFERENCES

[1] T. Ohno, *Toyota Production System: Beyond Large-Scale Production*. New York: Productivity Press, 1988.

[2] J. F. Krafcik, “Triumph of the lean production system,” *Sloan Manage. Rev.*, vol. 30, no. 1, pp. 41–52, 1988.

[3] N. Modig and P. Åhlström, *This is Lean. Resolving the Efficiency Paradox*. Stockholm: Rheologica Publishing, 2018.

[4] J. P. Womack and D. T. Jones, *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*, Free Press, 2003.

[5] J. K. Liker, *The Toyota Way: 14 Management Principles from the World’s Greatest Manufacturer*. New York, NY: McGraw-Hill, 2004.

[6] M. Rother, *Toyota Kata: Managing People for Improvement, Adaptiveness, and Superior Results*. New York, NY: McGraw Hill, 2010.

[7] C. Wolfram, *The Math(s) FIX: An Education Blueprint for the AI Age*. Wolfram Media, Inc., 2020.

[8] M. Csernoch, T. Nagy, and J. Csernoch, “Computer cooking vs. Problem-solving,” *Teach. Maths. Comput. Sci.*, vol. 22, no. 1, pp. 35–58, 2024. https://doi.org/10.5485/tmcs.2024.13454. Accessed: Aug. 20, 2024.

[9] A. Smalley, *Four Types of Problems: From Reactive Troubleshooting to Creative Innovation*, Lean Enterprise Institute, Inc, 2018.

[10] J. Chen, D. Morris, and N. Mansour, “Science teachers’ beliefs: perceptions of efficacy and the nature of scientific knowledge and knowing,” *Int. Handbook Res. Teachers’ Beliefs*, pp. 382–98, 2014. https://doi.org/10.4324/9780203108437-31.

[11] R. Lister, “After the gold rush: toward sustainable scholarship in computing,” *Proc. Tenth Conf. Australas. Comput. Educ.*, vol. 78, pp. 3–17, 2008.

[12] P. Mishra and M. Koehler, “Technological pedagogical content knowledge: a framework for teacher knowledge,” *Teach. Coll. Rec. Voice Scholarship Educ.*, vol. 108, no. 6, pp. 1017–54, 2006. https://doi.org/10.1177/016146810610800610.

[13] M. Koehler, "TPACK explained," *TPACK.ORG*. http://matt-koehler.com/tpack2/tpack-explained/. Accessed: Jun. 20, 2024.
[14] S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*. New York, NY: Basic Books, 1980.
[15] E. Soloway, "Should we teach students to program?" *Commun. ACM*, vol. 36, no. 10, pp. 21–4, 1993. https://doi.org/10.1145/163430.164061.
[16] M. Guzdial and E. Soloway, "Teaching the Nintendo generation to program," *Commun. ACM*, vol. 45, no. 4, pp. 17–21, Apr. 2002. https://doi.org/10.1145/505248.505261.
[17] J. M. Wing, "Computational thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33–5, 2006. https://doi.org/10.1145/1118178.1118215.
[18] P. J. Denning, "The science in computer science," *Commun. ACM*, vol. 56, no. 5, pp. 35–8, May 2013. https://doi.org/10.1145/2447976.2447988.
[19] M. Lodi and S. Martini, "Computational thinking, between Papert and wing," *Sci. Educ.*, vol. 30, no. 4, pp. 883–908, Apr. 2021. https://doi.org/10.1007/s11191-021-00202-5.
[20] NJSZT: Nemes Tihamér National Applied Studies Competition, 2023. http://tehetseg.inf.elte.hu/nemesa/index.html. Accessed: Jun. 20, 2024.
[21] NJSZT: Nemes Tihamér Nemes International Programming Competition. http://nemes.inf.elte.hu/. Accessed: Jun. 20, 2024.
[22] M. Prensky, "Digital natives, digital immigrants," *On the Horizon*, vol. 9, no. 5, pp. 1–6, 2001. https://doi.org/10.1108/10748120110424816. Accessed: Jun. 20, 2024.
[23] M. Prensky, "Digital natives, digital immigrants part 2: do they really think differently?" *On the Horizon*, vol. 9, no. 6, pp. 1–6, 2001. https://doi.org/10.1108/10748120110424843. Accessed: Jun. 20, 2024.
[24] P. A. Kirschner and P. De Bruyckere, "The myths of the digital native and the multitasker," *Teach. Teach. Educ.*, vol. 67, pp. 135–42, 2017. https://doi.org/10.1016/j.tate.2017.06.001. Accessed: Jun. 20, 2024.
[25] G. Stoner, "Accounting students' it application skills over 10 years," *Account. Educ.*, vol. 18, no. 1, pp. 7–31, Feb. 2009. https://doi.org/10.1080/09639280802532224.
[26] K. N. Schneider, L. L. Becker, and G. G. Berg, "Beyond the mechanics of spreadsheets: using design instruction to address spreadsheet errors," *Account. Educ.*, vol. 26, no. 2, pp. 127–43, 2017. https://doi.org/10.1080/09639284.2016.1274912.
[27] Amendment of Government Decree No 110/2012 (VI. 4.) on the publication, introduction and application of the National Core Curriculum. 5/2020 (I. 31.) Korm. Rendelet.
[28] J. Vahrenhold, Ed., *Informatics Education in Europe: Are We All in the Same Boat? Report by The Committee on European Computing Education (CECE)*. New York, NY: Association for Computing Machinery, 2017. Accessed: Jun. 20, 2024.
[29] W. Gander, Ed., Informatics Education: Europe Cannot Afford to Miss the Boat Report of the Joint Informatics Europe & ACM Europe Working Group on Informatics Education 2013. Accessed: Jun. 20, 2024.
[30] Department for Education, "National curriculum in England: Computing programmes of study," *GOV.UK*. https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study. Accessed: Aug. 24, 2024.
[31] L. Mannila, "Digitally competent schools: teacher expectations when introducing digital competence in Finnish basic education," *Seminar.net*, vol. 14, no. 2, pp. 201–15, 2018. https://doi.org/10.7577/seminar.2980.
[32] S. Sentance and A. Csizmadia, "Computing in the curriculum: challenges and strategies from a teacher's perspective," *Educ. Inf. Tech.*, vol. 22, no. 2, pp. 469–95, 2016. https://doi.org/10.1007/s10639-016-9482-0.
[33] R. R. Panko, "The Cognitive Science of spreadsheet errors: why thinking is bad," in *2013 46th Hawaii International Conference on System Sciences*, 2013. https://doi.org/10.1109/hicss.2013.513.
[34] L. Raković, "A Framework for managing spreadsheet-based end-user applications," *Int. J. Manage. Decis. Making*, vol. 18, no. 1, pp. 76–92, 2019. https://doi.org/10.1504/ijmdm.2019.096695.
[35] S. P. Jones, A. Blackwell, and M. Burnett, "A user-centred approach to functions in Excel," in *Proceedings of the Eighth ACM SIGPLAN International Conference on Functional Programming*, 2003. https://doi.org/10.1145/944705.944721.
[36] T. J. McGill and J. E. Klobas, "The role of spreadsheet knowledge in user-developed application success," *Decis. Support Syst.*, vol. 39, no. 3, pp. 355–69, 2005. https://doi.org/10.1016/j.dss.2004.01.002.
[37] J. Walkenbach, *Excel 2016 Bible*. John Wiley & Sons, 2015.
[38] T. Rattenbury, J. Hellerstein, J. Heer, S. Kandel, and C. Carreras, *Principles of Data Wrangling: Practical Techniques for Data Preparation*. Beijing etc.: O' Reilly, 2017.
[39] R. Scherer, F. Siddiq, and J. Tondeur, "The technology acceptance model (TAM): a meta-analytic structural equation modeling approach to explaining teachers' adoption of digital technology in Education," *Comput. Educ.*, vol. 128, pp. 13–35, 2019. https://doi.org/10.1016/j.compedu.2018.09.009.
[40] C. Wilson, L. A. Sudol, C. Stephenson, and M. Stehlik, *Running on Empty: The Failure to Teach K–12 Computer Science in the Digital Age*. New York, NY: The Association for Computing Machinery, The Computer Science Teachers Association, 2010.
[41] B. Engen Ed., "Digital natives: digitally competent?," in *Society for Information Technology & Teacher Education International Conference*. Jacksonville, Florida: Society for Information Technology & Teacher Education International Conference, 2014, pp. 1–9.
[42] C. Redecker, "European framework for the digital competence of educators," in *DigCompEdu*, Y. Punie, Ed., Luxembourg: Publications Office of the European Union, 2017. Accessed: Jun. 20, 2024.
[43] K. Sebestyén, G. Csapó, and M. Csernoch, "The effectiveness of the webtable-datatable conversion approach," *Ann. Mathematicae Informaticae*, vol. 56, pp. 109–21, 2023. https://doi.org/10.33039/ami.2022.12.010.
[44] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. Paritosh, and L. Aroyo, "Everyone wants to do the model work, not the data work': data Cascades in high-stakes AI," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM Digital Library, 2021, pp. 1–15. Available: https://doi.org/10.1145/3411764.3445518.
[45] D. Kahneman, O. Sibony, and C. R. Sunstein, *Noise: A Flaw in Human Judgement*. London, UK: William Collins, 2022.
[46] O. Hatamleh and G. Tilesch, *BetweenBrains; Taking Back Our AI Future*, GTPublishDrive, 2020.
[47] J. Sweller, P. Ayres, and S. Kalyuga, "Cognitive load theory in perspective," *Cogn. Load Theor.*, pp. 237–42, 2011. https://doi.org/10.1007/978-1-4419-8126-4_18.

[48] P. A. Kirschner, J. Sweller, and R. E. Clark, "Why minimal guidance during instruction does not work: an analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching," *Educ. Psychol.*, vol. 41, no. 2, pp. 75–86, 2006. https://doi.org/10.1207/s15326985ep4102_1.
[49] J. Sweller, P. A. Kirschner, and R. E. Clark, "Why minimally guided teaching techniques do not work: a reply to commentaries," *Educ. Psychol.*, vol. 42, no. 2, pp. 115–21, 2007. https://doi.org/10.1080/00461520701263426.
[50] D. Kahneman, *Thinking, Fast and Slow*. Penguin Books, 2011.
[51] J. M. Carroll and M. B. Rosson, *Paradox of the Active User*. San Jose etc.: IBM Thomas J. Watson Research Division, 1986.
[52] J. M. Carroll, *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. Cambridge, MA: MIT Press, 1987.
[53] M. Ben-Ari, "Bricolage Forever!" in *Psychology of Programming Interest Group*, 1999.
[54] M. Ben-Ari and T. Yeshno, "Conceptual models of software artifacts," *Interacting Comput.*, vol. 18, no. 6, pp. 1336–50, 2006. https://doi.org/10.1016/j.intcom.2006.03.005.
[55] K. Nagy and M. Csernoch, "Pre-testing erroneous text-based documents: logging end-user activities," *Front. Educ.*, vol. 7, Mar. 2023. https://doi.org/10.3389/feduc.2022.958635. Accessed: Apr. 02, 2024.
[56] S. G. Powell, K. R. Baker, and B. Lawson, "A critical review of the literature on spreadsheet errors," *Decis. Support Syst.*, vol. 46, no. 1, pp. 128–38, Dec. 2008. https://doi.org/10.1016/j.dss.2008.06.001.
[57] "Horror stories," European Spreadsheet Risk Interest Group, https://eusprig.org/research-info/horror-stories/. Accessed: Jul. 6, 2024.
[58] M. Csernoch, K. Nagy, and T. Nagy, "The entropy of digital texts—the mathematical background of correctness," *Entropy*, vol. 25, no. 2, p. 302, 2023. https://doi.org/10.3390/e25020302. Accessed: Apr. 02, 2024.
[59] M. Csernoch, T. Nagy, K. Nagy, J. Csernoch, and C. Hannusch, "Human-centered digital sustainability: handling enumerated lists in digital texts," *IEEE Access*, vol. 12, pp. 30544–61, 2024. https://doi.org/10.1109/access.2024.3369587. Accessed: Apr. 02, 2024.
[60] M. Csernoch, *Programming with Spreadsheet Functions; Sprego: Spreadsheet Programming with Only a Dozen Functions*. Budapest: Műszaki Könyvkiadó Kft., 2014.
[61] M. Csernoch and J. Csernoch, "How Beaufort, Neumann and Gates met? Subject integration with spreadsheeting," in *Proceedings of the EuSpRIG 2023 Conference "The Spreadsheet Crisis: Regaining Control,"*. London, UK: EuSpRIG, 2024, pp. 15–31. Accessed: Jun. 20, 2024.
[62] M. Csernoch and J. Csernoch, "Subject integration with spreadsheets. Ignoring education is the greatest risk ever," in *Proceedings of the EuSpRIG 2023 Conference*. London, UK: EuSpRIG, 2024. Accessed: Jun. 20, 2024.
[63] M. Csernoch, P. Biró, J. Máth, and K. Abari, "Testing algorithmic skills in traditional and non-traditional programming environments," *Inform. Educ.*, vol. 14, no. 2, pp. 175–97, 2015. https://doi.org/10.15388/infedu.2015.11.
[64] G. Csapó, M. Csernoch, and K. Abari, "Sprego: case study on the effectiveness of teaching spreadsheet management with schema construction," *Educ. Inf. Tech.*, vol. 25, no. 3, pp. 1585–605, 2019. https://doi.org/10.1007/s10639-019-10024-2.
[65] G. Csapó, K. Sebestyén, M. Csernoch, and K. Abari, "Case study: developing long-term knowledge with Sprego," *Educ. Inf. Tech.*, vol. 26, no. 1, pp. 965–82, 2020. https://doi.org/10.1007/s10639-020-10295-0.
[66] M. Csernoch, P. Biró, and J. Máth, "Developing computational thinking skills with algorithm-driven spreadsheeting," *IEEE Access*, vol. 9, pp. 153943–59, 2021. https://doi.org/10.1109/access.2021.3126757.
[67] T. Narusawa and J. Shook, *Kaizen Express: Fundamentals for Your Lean Journey*. Cambridge, MA: Lean Enterprise Institute, 2009.
[68] R. Maurer, *The Spirit of Kaizen: Creating Lasting Excellence One Small Step at a Time*. New York: McGraw-Hill, 2013.
[69] R. Maurer, *One Small Step Can Change Your Life: The Kaizen Way*. New York, NY: Workman Pub., 2014.
[70] I. Kato and A. Smalley, *Toyota Kaizen Methods: Six Steps to Improvement*. New York: Productivity Press, 2011.
[71] A. Abonyi-Tóth, C. Farkas, and P. Varga, *Digital Culture 8*. Budapest, Hungary: Oktatási Hivatal, 2023. https://www.tankonyvkatalogus.hu/storage/pdf/OH-DIG08TA__teljes.pdf. Accessed: Jun. 20, 2024.
[72] P. Varga, K. Jeneiné Horváth, Z. Reményi, C. Farkas, I. Takács, G. Siegler, and A. Abonyi-Tóth, *Digital Culture 9*. Budapest, Hungary. https://www.tankonyvkatalogus.hu/storage/pdf/OH-DIG09TA__teljes.pdf. Accessed: Jun. 20, 2024.
[73] A. Abonyi-Tóth, C. Farkas, Z. Reményi, and K. Jeneiné Horváth, *Digital Culture 10*. Budapest, Hungary: Oktatási Hivatal, 2021. https://www.tankonyvkatalogus.hu/storage/pdf/OH-DIG10TA__teljes.pdf. Accessed: Jun. 20, 2024.
[74] A. Abonyi-Tóth, C. Farkas, Z. Fodor, K. Jeneiné Horváth, Z. Reményi, G. G. Siegler, and P. Varga, *Digital Culture 11*. Budapest, Hungary. https://www.tankonyvkatalogus.hu/storage/pdf/OH-DIG11TA__teljes.pdf. Accessed: Jun. 20, 2024.
[75] G. Polya, *How to Solve It*. Princeton University Press, 1945.
[76] V. L. Takács and K. Bubnó, "Mathability in business education," *Acta Polytechnica Hungarica*, vol. 19, no. 1, pp. 9–29, 2022. https://doi.org/10.12700/aph.19.1.2022.19.2.
[77] M. Csernoch, "The stepchild of informatics education: file management," *Academia Lett.*, 2021. https://doi.org/10.20935/al2295. Accessed: Jun. 20, 2024.
[78] A. Sarkar, J. Borghouts, A. Iyer, S. Khullar, C. Canton, F. Hermans, A. Gordon, and J. Williams, "Spreadsheet use and programming experience: an exploratory survey," in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM Digital Library, Apr. 2020, pp. 1–10. Available: https://doi.org/10.1145/3334480.3382807.
[79] P. Sestoft, *Spreadsheet Technology*. Copenhagen: IT University of Copenhagen, 2011.
[80] P. Warren, "Learning to program: spreadsheets, scripting and HCI," in *Proceedings of the Sixth Australasian Conference on Computing Education*, vol. 30, 2004, pp. 327–33.
[81] J. Walkenbach, *Excel 2010 Bible*. Wiley, 2010.
[82] J. Walkenbach, *Excel 2013 Formulas*. John Wiley & Sons, 2013.
[83] Microsoft, "Dynamic array formulas and spilled array behavior," Microsoft Support. https://support.microsoft.com/en-us/office/dynamic-array-formulas-and-spilled-array-behavior-205c6b06-03ba-4151-89a1-87a7eb36e531. Accessed: Aug. 24, 2024.
[84] OH, *Four-digit Function Tables, Relationships and Data. Mathematics, Computer Science, Physics, Astronomy, Geography, Chemistry*. Budapest: Education Office, 2019.

[85] ICAEW, “Twenty principles for good spreadsheet practice – ICAEW,” ICAEW THOUGHT LEADERSHIP. https://www.icaew.com/-/media/corporate/files/technical/technology/excel-community/20-principles-of-good-spreadsheet-practice-2018.ashx. Accessed: Dec. 5, 2023.

[86] ICAEW Insights, “Updated: 20 principles for good spreadsheet practice,” ICAEW. https://www.icaew.com/insights/viewpoints-on-the-news/2024/may-2024/updated-20-principles-for-good-spreadsheet-practice. Accessed: Aug. 18, 2024.

[87] M. Csernoch and J. Csernoch, „How to set an example as an IT teacher,” In Hungarian: “Hogyan mutathatunk példát informatikatanárként,” in *INFODIDACT’2021 14. Informatika Szakmódszertani Konferencia*, P. Szlávi and L. Zsakó, Eds., Budapest: Webdidaktika Alapítvány, 2022, pp. 17–34. https://people.inf.elte.hu/szlavi/InfoDidact21/Manuscripts/CsMCsJ.pdf. Accessed: Jun. 20, 2024.

[88] G. Papp and M. Csernoch, “The correctness of PowerPoint presentation in different age groups.” In Hungarian: “PowerPoint prezentációk helyessége különböző korosztályokban,” in *INFODIDACT’2021 14. Informatika Szakmódszertani Konferencia*, P. Szlávi and L. Zsakó, Eds., Budapest: Webdidaktika Alapítvány, 2022, pp. 113–28. Accessed: Jun. 20, 2024.

[89] M. Csernoch and J. Csernoch, “Digital culture – prezentations – knowledge transfer.” In Hungarian: “Digitális kultúra – Prezentáció – Tudástranszfer,” *INFODIDACT’2021 14. Informatika Szakmódszertani Konferencia*, P. Szlávi and L. Zsakó, Eds., Budapest: Webdidaktika Alapítvány, 2022, pp. 17–34. Accessed: Jun. 20, 2024.