Proceedings of the Federated Conference on Computer Science and Information Systems pp. 1257–1262
DOI: 10.15439/2017F273
ISSN 2300-5963 ACSIS, Vol. 11

User-Centered Design Case Study: Ribbon Interface Development for Point of Sale Software

Zdzisław Sroczyński
Institute of Mathematics
Silesian University of Technology
23 Kaszubska Str.
44-100 Gliwice, Poland
Email: zdzislaw.sroczynski@polsl.pl

***Abstract*—The article is devoted to user-centered design (UCD) applied to the development process of the point of sale software. The influence of UCD methodology on the whole project's progress and its results is described alongside with exemplary user interface designs. In particular, there is a ribbon menu elaborated with the results of the user experience evaluation.**

I. Introduction

THE software supporting point of sale (POS) operations is a common branch in computer industry. Although there are many solutions on the market, the variable users' requirements and changing law regulations still open the new opportunities to develop dedicated POS applications.

The operations at the retail desk take place frequently through all the working shift, therefore the human-computer interaction (HCI) for the POS software should be optimized to reduce time and guarantee the comfort of usage and reliability during repetitive activities. This reason makes the research in this area especially interesting and worth thorough experiments, regardless of potential organizational difficulties.

II. State of the Art

There is a growing interest in human factors in contemporary computing. User experience and human-computer interaction issues are vital parts of almost every software project. Moreover, the agile methodologies focusing at the user are becoming more and more popular, with user-centered design in the lead.

The user interface of popular office applications has evolved for a long time, from terminal mode, keyboard shortcuts, textual menu, dedicated graphical menus until pull-down menus and adaptable toolbars [1], which became a part of modern operating systems. The unified graphical user interface (GUI) improved the user experience especially in terms of learnability. Next GUI improvements introduced sets of standard controls and consistent look-and-feel, especially when it comes to mobile operating systems with touch screens.

Nowadays, the most progressive desktop user interface for the set of options and controls is the ribbon menu, introduced for the first time by Microsoft Corporation in Office 2007 suite. Despite the legal controversies, this kind of interface proved to be efficient and becomes common. It is a good example of implementation of Fitts's law [2] into computer GUI, giving the comfort of usage even for the people with lesser computer literacy.

There are continuous works on user interface design described in literature [3][4][5][6]. The research in the field of modern ribbon-based user interfaces is widely elaborated in [7][8][9][10][11]. Authors of [12][13] discuss some interesting applications of ribbon menus, while [14] gives a review of sophisticated interfaces of medical devices. Examples for alternative, adaptable interfaces and interactions designed to support disabled persons are given in [15][16][17][18] and [19]. Novel methods of interaction design for multimedia applications and computer games are discussed in [20] and [21].

The ribbon interface was strongly supported while its introduction in the MS Office suite, even with the use of gamification. Therefore Microsoft game "Ribbon Hero" is mentioned by many publications in that field [22][23][24].

The general philosophy behind the User-centered design (UCD) term is involving users in the design process of the computer system. Users' participation level can vary. It can be limited to consulting, observations and testing. On the other hand, the users can be intensively involved throughout every stage of the development as actual partners. UCD clearly suites and complements the other agile methodologies, being probably the most general framework incorporating human-computer interface and user experience factors into the software development process.

User-centered design is the subject of many research projects, from theory [25][26], through formal [27], up to real life examples [28][29]. The topics regarding evaluation of the user experience are covered by [30], [31] and [32].

III. Motivation and Methods

The presented research work is motivated by the author's observations during a real-world development process of the point of sale (POS) software. While the POS applications are quite common, there is a significant specialization in this kind of software, and due to diversity in business operations, the dedicated solutions happen very often. The incorporation of the user-centered design paradigm is natural in this case, as the users' needs can vary strongly.

This case study includes all the development stages of the POS software and presents some valuable insights from the software engineering point of view, because of the long period of monitoring, internal author's involvement in the development team and wide commitment of the actual users. It is worth noting, that the total timespan of the described development cycles is wider than ten years, covers several versions of the IDE tools and includes surveys from several dozen of employees. This way it illustrated the changes in the computer industry and human perception for one of the common software categories.

In the initial stage of the development of POS software, the general assumptions for the project were defined through experts' brainstorming and surveying potential users during face-to-face interviews and panel discussions. The experts – IT development team and customer's management staff – drew conclusions from the review of the out-of-the-box ready POS software. Examined software packages were in general too complex and did not cooperate well with external hardware, especially fiscal printers (popular in Poland model Vento) – there was no option to include salesman identification in the receipt. The other inconvenience forced the operator to close a shift and begin another with every change of person at the desk, which did to fit into manner of work in the customer's retail network, taking too much time. In the end, the stock operations were too complex for the small retail shops, forcing to create shipping and delivery notes, as well as queue priorities (LIFO/FIFO) and variable prices for the articles.

Therefore the need of software targeting small and medium-sized retail companies, with limited financial resources, operating by users with average computer skills was formulated. Issues related to this category of family business often determine applicable technical solutions [33][34].

On the other hand, the potential users – salesmen – formulated the following, comprehensive list of additional requirements and remarks:

- informative, simple graphical interface and set of operations,
- fast processing, especially while scrolling data in the grid, containing a few thousands of articles,
- focus at article search and filtering, taking into account many attributes,
- fast login and switching of users – there may be several salesmen on one shift in the same time,
- easy operations helping to avoid errors, so only one open receipt at the time, all discounts per receipt,
- fast access to the stock level for the article, editable at any time without dedicated shipping/delivery notes,
- possibility to use negative stock levels, to support mid-shift deliveries,
- adaptation to the formal Polish law regulations, fully Polish interface,
- proper, fast and stable communication with fiscal printers,
- convenient reports, fiscal and statistical, giving informative results about the efforts of particular employees,
- barcodes printing on the ordinary printer and self-adhesive paper sheets,
- browser of archive receipts with filtering,
- cash payments and withdrawals support,
- different payment methods: cash, credit/debit card, gift cards.

It is clear, that partially the recommendations from the experts were parallel to the users' remarks.

These to sets of requirements were elaborated by the development team with the usability in mind, taking into account the main attributes of proper user experience, distinguished in [35]: efficiency, satisfaction, learnability, memorizability and faultlessness. Consequently almost all requirements were incorporated in the very first version of the POS software, which had to be prepared in the very short time – about two weeks – due to deadlines set by the customer (small retail network).

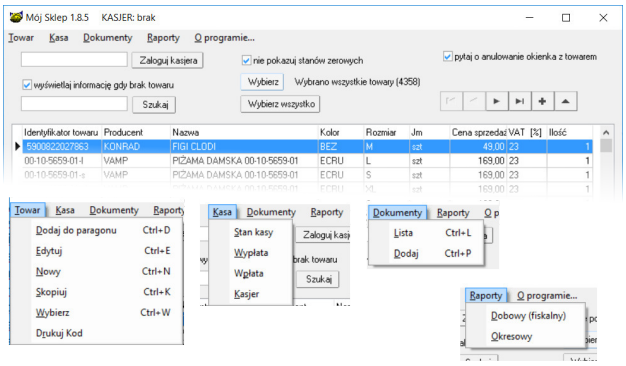

Fig. 1. The user interface of the first version of POS software with classic pull-down menu. Expanded menu options are shown in the subpictures.

The POS software is based on a data grid, overlapping almost the whole screen (see Fig. 1). For the fastest possible operations it was designed as desktop Windows application using local database. This way it matched requirements for low budget (no need for server hardware and software) and allowed easy integration with SDK for the fiscal printers. The development IDE was Embarcadero RAD Studio, generating pure win32 applications, with excellent database connectivity components and known for solid backwards compatibility. This choice profited in the future, when subsequent version of POS software appeared without struggle for adapting changed APIs. Nowadays, thanks to multiplatform capabilities introduced meanwhile to RAD Studio compilers, there is a possibility to port the software to different desktop operating systems: macOS and Linux without great effort, or use it as a base for mobile applications running at Android or iOS [36].

There were three main software versions developed during the further development of the POS application basing on UCD methodology. They are thoroughly elaborated in the following section.

IV. Application Development Cycle

The programmers team developed three main versions of the POS application, having of course many subversions with minor improvements. They were developed in subsequent iterations and in general the system was immediately upgraded to the newest version available. The first version had a classic MS Windows pull-down menu and the set of functionality needed to perform sale operations. This version has been used for a long time on rather budget computers with small monitor screens. There was no need to customize font sizes in the data grid and only essential fiscal reports were available.

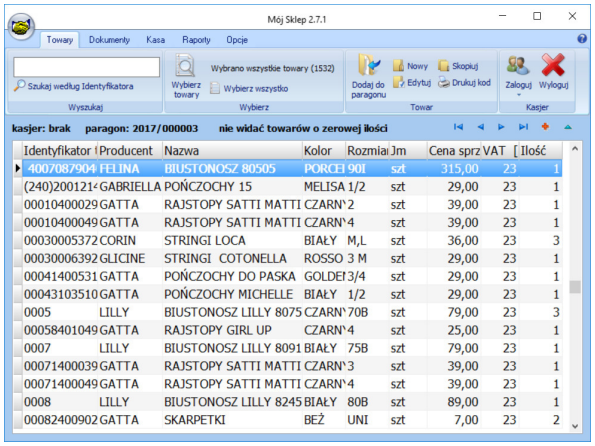

Fig. 2. POS application with a menu mimicking Office 2007 MS Ribbon, after reorganization of options and introduction of icons. More important options acquired bigger icons.

The second main version of the GUI for POS application, shown in Fig. 2, had a MS Ribbon menu (introduced in MS Office 2007) with three colour themes analogous to standard settings in Microsoft Windows (Luna, Obsidian, Silver). This ribbon had skeuomorphic look-and-feel with the characteristic yellow focus. It also mimicked the most confusing behaviour of genuine MS Ribbon: hiding the ribbons after double click on the menu option. In this state the ribbon menu is losing one of the main advantages in comparison to common pull-down menu, as the controls are not visible without extra click needed to unfold the ribbon. The introduction of the ribbon interface was partially inspired by some of the surveyed users, showing interest in a "modern" look of the software. The second reason was the management's need to simplify the software and reduction of the duration and costs of new employee trainings.

The novel ribbon-based menu interface in this version of POS software forced some refactorings in the internal structure of the application. The TAction component was used to put the event handlers in order, making the software somewhat more compatible with MVC (model-view-controller) paradigm. Although this improvement had nothing to do with users' opinions or influence, it significantly helped to support the ribbon menu and modern look-and-feel in the next software version.

Microsoft Corporation decided to force developers to "sign" a special licence for the usage of MS Ribbon control. The licence concerned not the internals of the software component (actually included in the operating system since Vista version), but the overall graphical design, look-and-feel and user experience. This way independent software vendors were put in rather troublesome conditions, possibly violating Microsoft's licence even when providing their own implementation of the ribbon control. In fact, the ribbon is of course very similar to tabs, and moreover – analogous solutions were available and used in many applications before Office 2007. The only (but important) difference is the consistent GUI proposed and promoted by Microsoft. Consequently, the usage of ribbon interface became less common, than it could be without these controversies. Eventually, to avoid legal issues, the support for the ribbon interface in Embarcadero RAD Studio was ceased, what had an impact on the development of our POS application and its third version.

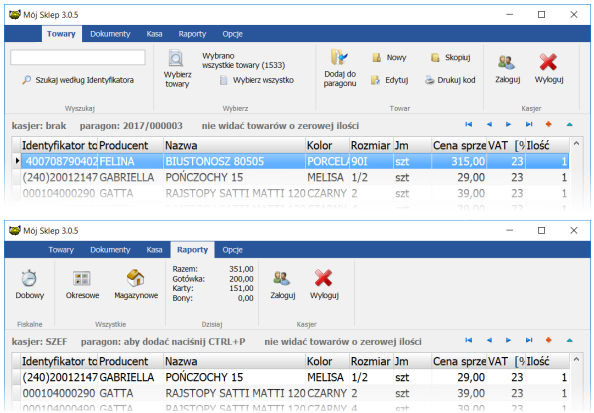

Fig. 3. GUI of the most modern looking version of the POS software, following Office 2016 flat design. Two different ribbon tabs are shown with the exemplary options and info labels.

The third, developed recently version of POS terminal software introduced novel visual appearance based on MS Office 2016 GUI guidelines as a consequence of development according to current trends. This time consulting users suggested to stick to one colour scheme, so the blue one was chosen. In this version for the first time the non-standard component from third-party software vendor was used, as standard RAD Studio development environment does not provide modern looking ribbon components due to licensing problems described above. An introduction of Almediadev BusinessSkinForm suite provided flat Office 2016 ribbon (Fig. 3) and the other controls, but it also required another set of source code refactorings.

These circumstances indicate that legal issues can affect software engineering development process: slow it down or even enforce extra costs.

Here are some other improvements introduced according to users' suggestions and cooperation during the UCD-driven development process:

- direct preview of daily income on the ribbon,
- coloured discount warning,
- simplified user logging – fast selection of the user,
- user logging from every ribbon tab,

- font size customization and other HDPI screen improvements,
- alternate row colours in the grid,
- planned availability for the article – ordering simplification,
- inventory report.

There are somewhat complex improvements in the list, as well as minor visual changes. Anyway, they help the users in more efficient work and suite perfectly to their actual needs.

The ribbon interface was in fact introduced experimentally only at the moment, when the appropriate components in the development IDE became available. Because the users signalled the interest in similar solutions before, there was a positive reception of this novelty. Most of the users pointed out the high visual compatibility with the other modern applications (i.e. Microsoft Office). Some more objective factors are of course worth noting also, as for example better visibility of controls, shorter learning path and better memorizability. The standard ribbon interface is less adaptable than classic pull-down menus, as the ribbon tabs are in general still in the same sequence with exactly the same set of controls. This approach can be inconvenient in very complex, huge and multifunctional applications (which miss the place in the ribbon for enormous count of options). Although, when it comes to properly designed software with well defined functionalities, this method meets users' expectations.

This kind of unification becomes especially valuable, when the user model for the software is variable, because of different professional experience of the users. In this case the fixed GUI is acceptable for power users and simultaneously easy to learn for beginners.

V. Evaluation of the User Experience

There were 25 users involved into the UCD process: full-time employees, management staff and some interns. The level of commitment clearly differed depending on the particular job position. Computer skills of the users were in general similar, as majority of them knew the basics of Windows operating system and popular Office applications from the school. For two senior employees computerized POS was a complete novelty and they were significantly against it. We observed classic difficulties, as for example tendency to learn exact sequences of keystrokes without monitoring the system response. Sometimes these less experienced users were just ignoring the messages, and moreover – were not even able to remember the general meaning of the messages.

Except these problems at the very beginning of the development of POS system, all the users were successfully using the application, the barcode reader and the fiscal printer. Although not all users were employed for the whole time of the development process, the remarks from them were valuable and useful for the rest of the crew.

The first version of the application had just 5 users working with very budget desktop computers (Pentium II class), next the shops network increased, eventually reaching 20 users. Nowadays, the application is utilized on very wide set of

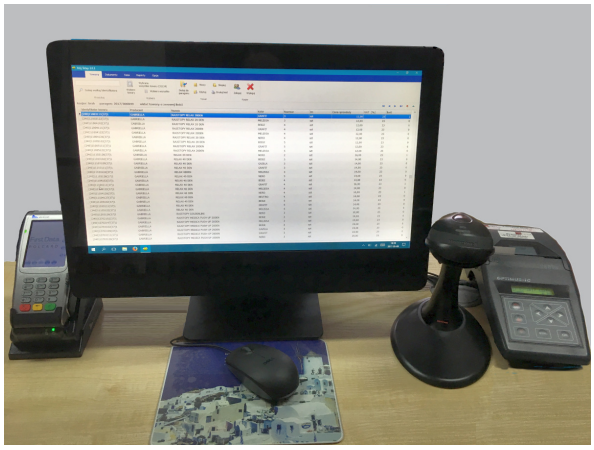

Fig. 4. POS application during user experience tests at the complete workplace, equipped with credit card terminal, Dell all-in-one computer running Windows 10, barcode scanner and fiscal printer Vento.

computer systems, from common desktop ones, up to modern all-in-one machines with touch screens (as Dell OptiPlex 3030 – see Fig. 4) or laptops (as hybrid Lenovo Flex). Direct connection with fiscal printer forces usage of MS Windows machines, but touch screens are sort of game changer here, because younger salesmen are very familiar with this technology and use it intuitively. This way the user experience gap between classic desktop applications and mobile world narrows.

UCD was involved at every stage of the development, from the first general project, upto the newest ribbon-based application. The main methods of evaluation for the overall user experience were face-to-face surveys and observations of users' behaviour in real world POS installations. This way a very strong relation between development team and final users was built, which is distinctive for agile methodologies. While all these methods were rather informal, the resulting software become a stable and reliable solution supporting POS operations.

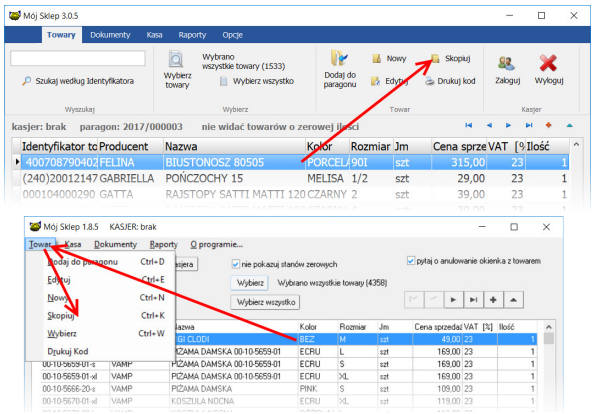

Fig. 5. The advantage of ribbon menu against ordinary pull-down one. For some controls mouse track can be two times shorter and there is only one click needed.

The users of the final product were also surveyed by more formal paper questionnaires. We asked about the overall satisfaction from the usage of the system (Q1), does the ribbon interface have a general advantage against common menu (Q2), is the ribbon interface "Office 2016"-alike more readable than "Office 2007" one (Q3), can the actions with ribbon interface be performed faster (Q4), does the ribbon interface help to memorize the recipes for common operations (Q5), is the POS software more comfortable than the others you know (Q6)? All the questions had the scale from 1 (strong disagreement) to 5 (strong agreement).

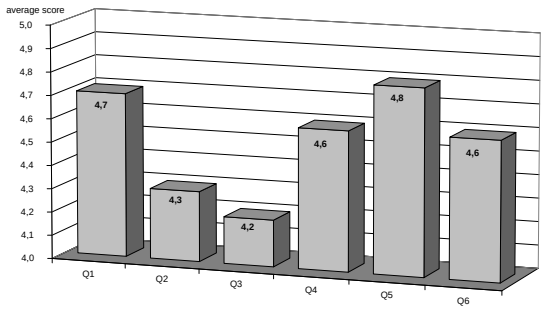

Fig. 6. Average scores for particular questions from the questionnaire about the latest version of the POS software with ribbon menu.

Average response rate about 4.5 (compare Fig. 6) confirms that UCD approach was the right choice. It helped to develop the software which fits very well to users' preferences. There were some users with a bit more conservative approach, sceptical about the novelties and their opinions proved to be decisive for slightly lower ratings in questions 2 and 3. People accustomed to proven solutions are naturally less inclined to accept and appreciate significant changes. On the other hand, the results for the question about memorizability (Q5) point out, that objective indicators for ribbon menu are much better than superficial opinions about it.

Incremental development process took less resources than classic waterfall model, although the time needed was probably longer. The usage of stable and backwards compatible software toolset seems to be another essential factor in the agile UCD. When the subsequent iterations were taking their time, there was no pressure on extra effort involved with the maintenance of the IDE, compiler and software libraries. Instead of that, the novel possibilities, as for example ribbon interface, appeared and were ready to implement. This way the POS project went from the very basic menu driven application to modern looking one. User experience of the project gained much from that solution without extensive costs.

The initial experiments shown, that in general the ribbon interface optimizes the effort of the user. The reduction of mouse movement and number of clicks is significant, which increases the reliability of actions and helps in faster work (Fig. 5). The extra factor concluded from these experiments is the size of controls and necessity of thorough project for layout of every ribbon tab, in order to profit from Fitts's law as much as possible.

VI. Conclusions

UCD introduced into the development process of the POS software was a key factor of success. It helped to design an interface that users desired, suitable for the necessary activities, but not complicated. This way the quality of user experience increased with every novel version of the software, eventually reaching the modern and effective ribbon menu form.

All three versions of the software profited from the UCD approach, although the scope of the improvements was variable: from the very tiny details to significant rearrangements of the whole user interface. The cooperation between the developers and the users of the system was fluent and agile, as UCD model did not force any artificial restrictions and time frames.

The survey of users' opinions about the software system designed this way leads to interesting conclusions. First, the users profit from easiness and memorizability of the ribbon interface. On the other hand, some users have of course doubts when it comes to fundamental changes in UX.

In general, the evaluation of the satisfaction and experiments regarding some objective characteristics confirmed usefulness of the user-centered design methodology. Further research steps can be focused on statistical analysis of user behaviour with the use of actiontracking [37] and optimization of the controls' placement and sizing.

References

[1] I. Crk, "Predictive pointing in cascading pull-down menus," in *Information and Computer Technology (GOCICT), 2015 Annual Global Online Conference on*. IEEE, 2015. doi: 10.1109/GOCICT.2015.17 pp. 41–45. [Online]. Available: https://doi.org/10.1109/GOCICT.2015.17

[2] C. E. Wright and F. Lee, "Issues related to HCI application of Fitts's law," *Human–Computer Interaction*, vol. 28, no. 6, pp. 548–578, 2013. doi: 10.1080/07370024.2013.803873. [Online]. Available: http://dx.doi.org/10.1080/07370024.2013.803873

[3] P. Weichbroth and M. Sikorski, "User interface prototyping. techniques, methods and tools," *Studia Ekonomiczne*, vol. 234, pp. 184–198, 2015.

[4] K. Z. Gajos, D. S. Weld, and J. O. Wobbrock, "Decision-theoretic user interface generation," in *AAAI*, vol. 8, 2008, pp. 1532–1536. [Online]. Available: http://dl.acm.org/citation.cfm?id=1620270.1620326

[5] J. Scarr, A. Cockburn, C. Gutwin, A. Bunt, and J. E. Cechanowicz, "The usability of commandmaps in realistic tasks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2014. doi: 10.1145/2556288.2556976 pp. 2241–2250. [Online]. Available: https://doi.org/10.1145/2556288.2556976

[6] P. Bachmann, "Patterns for internationalization and cross-cultural usability," in *Proceedings of the 20th Conference on Pattern Languages of Programs*. The Hillside Group, 2013, p. 19. [Online]. Available: http://dl.acm.org/citation.cfm?id=2725692

[7] L. Colazzo, A. Molinari, and S. Tomasini, "Is new necessarily good? testing usability of the new office 2007 user interface," in *Proceedings of EdMedia: World Conference on Educational Media and Technology 2008*, 2008. ISBN 978-1-880094-65-5 pp. 1371–1379.

[8] M. Dostál, "User acceptance of the Microsoft ribbon user interface," in *Proceedings of the 9th WSEAS international conference on Data networks, communications, computers*. World Scientific and Engineering Academy and Society (WSEAS), 2010, pp. 143–149. [Online]. Available: http://dl.acm.org/citation.cfm?id=1948805.1948832

[9] A. Darejeh and D. Singh, "Increasing Microsoft Office usability for middle-aged and elder users with less computer literacy," *Journal of Industrial and Intelligent Information Vol*, vol. 2, no. 1, 2014. doi: 10.12720/jiii.2.1.56-62

[10] ——, "An investigation on ribbon interface design guidelines for people with less computer literacy," *Computer Standards & Interfaces*, vol. 36, no. 5, pp. 808–820, 2014. doi: 10.1016/j.csi.2014.01.006. [Online]. Available: http://dx.doi.org/10.1016/j.csi.2014.01.006

[11] D.-M. Petrosanu and A. Pîrjan, "Solutions for developing and extending rich graphical user interfaces for office applications," *Journal of Information Systems & Operations Management*, p. 1, 2015. [Online]. Available: https://ideas.repec.org/a/rau/journl/v9y2015i1p157-167.html

[12] N. U. Khan and J.-C. Lee, "Development of a music score editor based on musicxml," *Journal of the Korea Society of Computer and Information*, vol. 19, no. 2, pp. 77–90, 2014. doi: 10.9708/jksci.2014.19.2.077. [Online]. Available: https://doi.org/10.9708/jksci.2014.19.2.077

[13] J. Kadlec, D. P. Ames, and J. Nelson, "User interface design considerations for a time-space GIS," in *2012 International Congress on Environmental Modelling and Software Proceedings*, 2012.

[14] M. Wichrowski, "Usability engineering in the prototyping process of software user interfaces for mobile medical ultrasound devices," *Computer Science*, vol. 16, 2015. doi: 10.7494/csci.2015.16.3.219. [Online]. Available: http://dx.doi.org/10.7494/csci.2015.16.3.219

[15] A. Bier and Z. Sroczyński, "Adaptive math-to-speech interface," in *Proceedings of the Mulitimedia, Interaction, Design and Innnovation*, ser. MIDI '15. New York, NY, USA: ACM, 2015. doi: 10.1145/2814464.2814471. ISBN 978-1-4503-3601-7 pp. 7:1–7:9. [Online]. Available: http://doi.acm.org/10.1145/2814464.2814471

[16] P. Kasprowski and K. Harezlak, "Using non-calibrated eye movement data to enhance human computer interfaces," in *Intelligent Decision Technologies*. Springer, 2015. doi: 10.1007/978-3-319-19857-6_31 pp. 347–356. [Online]. Available: https://doi.org/10.1007/978-3-319-19857-6_31

[17] R. Damaševičius, M. Vasiljevas, J. Šalkevičius, and M. Woźniak, "Human activity recognition in aal environments using random projections," *Computational and mathematical methods in medicine*, vol. 2016, 2016. doi: 10.1155/2016/4073584. [Online]. Available: http://dx.doi.org/10.1155/2016/4073584

[18] D. Połap and M. Woźniak, "Introduction to the model of the active assistance system for elder and disabled people," in *International Conference on Information and Software Technologies*. Springer, 2016. doi: 10.1007/978-3-319-46254-7_31 pp. 392–403. [Online]. Available: https://doi.org/10.1007/978-3-319-46254-7_31

[19] J. G. Schoeberlein and Y. Wang, "Usability evaluation of an accessible collaborative writing prototype for blind users," *Journal of Usability Studies*, vol. 10, no. 1, pp. 26–45, 2014. [Online]. Available: http://uxpajournal.org/usability-evaluation-of-an-accessible-collaborative-writing-prototype-for-blind-users/

[20] M. Wichrowski, D. Koržinek, and K. Szklanny, "Google glass development in practice: Ux design sprint workshops," in *Proceedings of the Mulitimedia, Interaction, Design and Innnovation*, ser. MIDI '15. New York, NY, USA: ACM, 2015. doi: 10.1145/2814464.2814475. ISBN 978-1-4503-3601-7 pp. 11:1–11:12. [Online]. Available: http://doi.acm.org/10.1145/2814464.2814475

[21] M. Woźniak, D. Połap, C. Napoli, and E. Tramontana, "Application of bio-inspired methods in distributed gaming systems," *Information Technology And Control*, vol. 46, no. 1, pp. 150–164, 2017. doi: 10.5755/j01.itc.46.1.13872. [Online]. Available: http://dx.doi.org/10.5755/j01.itc.46.1.13872

[22] M. Smoleń, "Gamification as creation of a social system," in *Gamification. Critical Approaches*. The Faculty of "Artes Liberales", University of Warsaw. Warsaw, 2015. ISBN 978-83-63636-44-9

[23] A. Darejeh and S. S. Salim, "Gamification solutions to enhance software user engagement - a systematic review," *International Journal of Human-Computer Interaction*, vol. 32, no. 8, pp. 613–642, 2016. doi: 10.1080/10447318.2016.1183330. [Online]. Available: http://dx.doi.org/10.1080/10447318.2016.1183330

[24] G. Barata, S. Gama, J. A. Jorge, and D. J. Gonçalves, "Relating gaming habits with student performance in a gamified learning experience," in *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play*. ACM, 2014. doi: 10.1145/2658537.2658692 pp. 17–25. [Online]. Available: https://doi.org/10.1145/2658537.2658692

[25] C. Abras, D. Maloney-Krichmar, and J. Preece, "User-centered design," *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications*, vol. 37, no. 4, pp. 445–456, 2004. [Online]. Available: citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.94.381

[26] E.-O. Baek, K. Cagiltay, E. Boling, and T. Frick, "User-centered design and development," *Handbook of research on educational communications and technology*, no. 1, pp. 660–668, 2008. doi: 10.4324/9780203880869.ch49. [Online]. Available: https://www.routledgehandbooks.com/doi/10.4324/9780203880869.ch49

[27] J. Earthy, B. Sherwood-Jones, and N. Bevan, "ISO standards for user centered design and the specification of usability," in *Usability in government systems: User experience design for citizens and public servants*. Elsevier, 2012. doi: 10.1016/B978-0-12-391063-9.00049-3. [Online]. Available: http://dx.doi.org/10.1016/B978-0-12-391063-9.00049-3

[28] D. Salah, R. Paige, and P. Cairns, "Integrating agile development processes and user centred design-a place for usability maturity models?" in *International Conference on Human-Centred Software Engineering*. Springer, 2014. doi: 10.1007/978-3-662-44811-3_7 pp. 108–125. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-44811-3_7

[29] J. Nelles, S. Kuz, A. Mertens, and C. M. Schlick, "Human-centered design of assistance systems for production planning and control: The role of the human in industry 4.0," in *Industrial Technology (ICIT), 2016 IEEE International Conference on*. IEEE, 2016. doi: 10.1109/ICIT.2016.7475093 pp. 2099–2104. [Online]. Available: https://doi.org/10.1109/ICIT.2016.7475093

[30] P. Lehane, "Mapping the user experience: Development of a validated instrument from the plans and scripts of the computer community of practice," *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments*, 2012. doi: 10.17011/ht/urn.201211203033. [Online]. Available: https://doi.org/10.17011/ht/urn.201211203033

[31] S. Rajagopalan, "Product personification: PARAG model to successful software product development," *International Journal of Managing Value and Supply Chains*, vol. 6, no. 1, p. 1, 2015. doi: 10.5121/ijmvsc.2015.6101. [Online]. Available: https://doi.org/10.5121/ijmvsc.2015.6101

[32] S. M. A. Shah, G. I. G. Al-Matroushi, and M. F. Qureshi, "Usability assessment of open source application," *International Journal of Advanced Research in Computer Science*, vol. 4, no. 2, 2013.

[33] J. Zukowska and Z. Sroczynski, "Gift cards authorization through GSM in a distributed trade network–case study," in *Internet–Technical Development and Applications*. Springer, 2009. doi: 10.1007/978-3-642-05019-0_12 pp. 101–108. [Online]. Available: https://doi.org/10.1007/978-3-642-05019-0_12

[34] M. Sikorski, "HCI and the economics of user experience," in *Maturing Usability: Quality in Software, Interaction and Value*, E. L.-C. Law, E. T. Hvannberg, and G. Cockton, Eds. London: Springer London, 2008. doi: 10.1007/978-1-84628-941-5_14. ISBN 978-1-84628-941-5 pp. 318–343. [Online]. Available: http://dx.doi.org/10.1007/978-1-84628-941-5_14

[35] J. Nielsen and R. Budiu, *Mobile Usability*. New Riders Press, 2012. ISBN 978-0-321-88448-0

[36] Z. Sroczyński, "Designing human-computer interaction for mobile devices with the FMX application platform," *Theoretical and Applied Informatics*, vol. 26, No 1-2, pp. 87–104, 2014. [Online]. Available: https://taai.iitis.pl/taai/article/view/388

[37] ——, "Actiontracking for multi-platform mobile applications," in *Computer Science On-line Conference*. Springer, 2017. doi: 10.1007/978-3-319-57141-6_37 pp. 339–348. [Online]. Available: https://doi.org/10.1007/978-3-319-57141-6_37