

Overview of the Application of Computer Aided Tools in Information Systems Development

Ayaz Kh. Mohammed¹, Abdülmalik L. Ahmed², Dindar I. Saeed³, Sardar O. Ramadhan⁴, Nashwan M. Salih⁵

¹ Computer System Department, Ararat Technical Private Institute, Kurdistan Region - Iraq

² Department of Computer Science, Kano University of Science and Technology, 713281 Wudil, Nigeria

³ Employee in Scientific Journal University of Zakho, University of Zakho, Kurdistan Region - Iraq

⁴ Department of Computer Science, University of Zakho, Kurdistan Region - Iraq

⁵ Department Information Technology, Duhok Polytechnic University, Kurdistan Region - Iraq

ABSTRACT

Developing Information Systems (IS) using computer-aided tools (CATs) yields immense benefits realized by following a set of activities and techniques. Apart from revolutionizing IS development, another key benefit of computer-aided approach remains the improvement of the overall IS development process. While integrated CATs could be used throughout the phases of IS development life-cycle, others are only meant serve specific phases. The present paper provides a short review on the concepts in IS Development using CATs. Furthermore, the paper highlights the most prominent CATs that are widely used and how these tools fit into different phases of IS development. It also describes how using CATs paved the way to the recent trends in user-centred and end-user IS development approaches.

Keywords: Information systems, Information systems development, Methodologies, Computer-aided tools.

1. INTRODUCTION

Developing Information Systems (IS) might be considered using a number of tools, approaches, and techniques. Options include structured programming, computer-aided tools (CATs), object-oriented implementation, cross-platform development, integrated development environments, and structured walkthroughs [1]. Just like any other option, when using computer-aided tools, set of activities are needed to develop the required IS solution [2] and basically, the solution could be constructed internally using the traditional or alternative methodologies or externally acquired as a pre-set solution that would be configured to fit required specifications [3]. As there is no unique methodology that would suit all IS development projects, any computer-aided methodology to be employed must specify the set of roles that CATs would play in that specific context [2].

Computer-aided tools are software products used in automating specific tasks involved in IS development. They are software support for the methodologies employed in designing a wide spectrum of IS. When integrated, computer-aided tools function beyond a mere software production, they support enterprise strategic planning, IS strategic planning, project planning, systems development, documentation and maintenance [4]. Upon classifying computer-aided tool(s) based on the activities of IS development

lifecycle they supported, there will be tools used in the conceptual phases of IS development (Planning, Analysis, and Design), those used in the construction phases (Development, Implementation, Testing, and Maintenance), and integrated computer-aided tools that could serve the entire IS development phases [5]–[7].

The overall aim of developing an information system using computer-aided tools is to improve the productivity and quality of the resulting systems by assisting the developer throughout the different phases of the development. These phases range from gathering functional and non-functional requirements to the design and implementation of the information system by considering all the relevant technical and operational features [4]. It worth noting that, advances in computer-aided tools simplify the user-centred approach to IS development and enables end-user IS development [8]. This paper studied the concepts of IS development and highlights prominent computer-aided tools used and best practices on how they are used. The paper is structured into four sections including the introductory section. Specifically, the second section highlights the related literature. Third section presents and discussed the prominent computer-aided tools used in IS development with future research directions. Lastly, the study is concluded in the fourth section.

2. RELATED LITERATURE SEARCH

This section highlights literary contributions that will help in understanding the past, present, and future studies on computer-aided tools and IS development. The section begins with a chronological description of IS development methodologies. Furthermore, justifications on exemplary methodologies for computer-aided tools based on scholastic perspectives were provided.

Aveson and Fitzgerald [9] made a historical overview of IS development methodologies from 1976 to 2006. The authors gave a clear insight into future trends and highlights on best practices in IS development using computer-aided tools. Prior to that, they described four chronological eras, which are pre-methodology, early methodology, methodology, and post methodology. *Pre-methodology* was characterized by crude development practices in which the dominant 'methodology' was rule-of-thumb and solely based on the experience of the programmer. This led to poor control and management of projects. In the *early methodology* era (the early 1980s), there was a growing admiration for the role of system analysts and more integrated IS solutions [9]. These reflections led to the development of the waterfall model which faced serious drawbacks on the way it was used [10]. In the *methodology era*, to curtail the limitations of the traditional waterfall model, numerous techniques and tools exemplifying the traditional waterfall model were derived. Remarkable techniques include entity-relationship modelling, normalization, data flow diagramming, structure diagrams, and entity life cycles. Tools include project management software, drawing tools, data dictionary software, systems repositories, and other computer-aided tools [11].

Current era is terms as the *post-methodology* era; a more stable environment than in any time since the early days of IS development methodologies, some IS practitioners are seeing computer-aided approaches as a panacea to the problems of traditional development approaches [9], [12]. This has led some people to the rejection of methodologies in general. With advancement in integrated computer-aided approaches, the main concern is on making distinctions between inadequate methodologies and poor application and use of methodologies in IS development [2]. Nowadays, integrated computer-aided tools have streamlined the effort in the application of methodologies. Developers need to only understand their clients and went ahead with building systems that they feel, based on their skills and experience, are right. Thus, the emergent IS development approach is analogous to that of a *pre-*

methodology era or even more extreme. This is because the whole IS development process could be shouldered by the programmers or even the end-users [8], [9], [13].

3. THE TOOLS

Computer-aided tools for IS development are software components supporting a specific task in the development of an information system [5]. They simplify the IS development process, increase development speed, increase accuracy, and improve IS product quality.

3.1 Benefits of Using Computer-Aided Tools

Computer-aided tools are used in IS development because they are time-saving, they enrich graphical techniques and data flow, they allow optimum use of available information, they also provide enhanced creation and manipulation of documentation [14]. The most prevalent motives behind using computer-aided tools in IS development as gathered from [5], [14]–[19] are:

Computer-aided tools increase speed via automation by reducing the time to complete many tasks, especially those involving diagramming and associated specifications.

Computer-aided tools increase accuracy by providing an ongoing debugging and error checking which is vital in early defect removal and even when the system gets larger.

By automating and improving most of the tasks, the overall systems quality can be improved. Thus, reducing maintenance cost.

They enable comprehensive, concise and continues documentation.

They enable programming in the hands of non-programmers particularly those that understand the logic of the program, object-oriented approaches, and client-server basics.

They also allow greater user participation, which can lead to better acceptance of the new system.

However, using computer-aided tools can produce initial prototypes that are more expensive to build and maintain. Because these tools require a more extensive and accurate definition of user needs and requirements which are difficult to customize [13], [17].

3.2 Classification of Computer-Aided Tools

In most literary documents computer-aided tools are classified based on the phases of IS development lifecycle that a tool supports [14], [20] and the level of features integrated into the tool [8]. The most paramount feature found in integrated computer-aided tools for IS development is the central repository which serves as a data dictionary. Apart from backup, recovery, and documentation capabilities that could be achieved, the central repository equally serves as a source of common, integrated, and consistent information [14].

In order not to overemphasize the scope of computer-aided tools in IS development, Berdonosov and Redkolis [5] singled it out of computer-aided workbenches and computer-aided environments; computer-aided tools are software components supporting a specific task in the development of an information system. Non-integrated computer-aided tools served as either editing tools, programming tools,

verification and validation tools, configuration management tools, and project management tools. In other words, non-integrated computer-aided tools are for either Analysis and Design, Coding support, Export and Import, Project Management and Prototyping, Configuration management and Modelling, Querying and report generation, or Information Security etc. [16], [19]–[21].

3.3 Exemplary Computer-Aided Tools

By categorising computer-aided tool(s) based on the activities of IS development lifecycle they support; we have those that support the conceptual phases (Planning, Analysis, and Design), those that support the construction phases (Development, Implementation, Testing, and Maintenance), and the integrated computer-aided tools that usually support the entire IS development phases [5]. Table 1 highlighted categories of computer-aided tools used in IS development as gathered from [2], [19]–[21] and vendors' websites:

TABLE 1
Exemplary computer-aided tools

Category	Description	Few Examples
Diagramming tools	Diagram tools are used to represent components of the information system, a graphical representation of the structure, data and control flow among various system components.	Dia, LibreOffice Draw, Diagram Designer, PlantUML, Draw.io, Flow Chart Maker and other tools for diagramming creative modern flowcharts.
Tools for Process Modelling	Process Modelling tools serve in creating models of the software process. The deliverables of these tools are used by managers in modifying models in line with the predefined software requirement or choosing the right process model.	EPF Composer, Tina and Snoopy for Petri nets
Tools for Project Management	Project planning, costing as well as effort estimation, resource planning, and project scheduling are supported by Project Management Tools. They also help in real-time cross-platform sharing and the storing of project information.	Microsoft Project, Creative Pro Office, Basecamp, Trac Project
Tools for Documentation	Documentation tools are used before, during, and after IS development project. They assist in generating training manual, system manual, installation manuals, reference manual etc. for both end-users and technical users.	Adobe RoboHelp, GitHub Pages, DrExplain and Doxygen for documentation
Tools for Analysis	These tools are used to automatically check for any inconsistency, gathering requirements, spotting data redundancies or erroneous omissions, and also an inaccuracy in the diagrams.	Accompa, Visible Analyst for total analysis, CaseComplete for requirement analysis, Accept 360
Tools for Design	These tools serve in designing the system modules base on the system's block structure. They also offer detail on interconnections between modules and within each module.	ArgoUML, Eclipse UML2 Tools, Animated Software Design
Tools for Configuration Management	These tools are used in version control, revision management, change control management, and baseline configuration management. They enable automatic version management, tracking, and release management	Rudder, Fossil, Accu REV, Git, Vagrant, SmartFrog, Juju
Tools for Programming	Programming tools involve programming environments such as the Integrated Development Environment (IDE), simulation tools, and inbuilt modules library. These tools	Netbeans IDE, Cscope to search code in C, Eclipse

	help with a comprehensive way of building software product and comprises features for simulation and testing.	
Tools for Prototyping	A simulated version of a system could be created using these tools that provide preliminary look as well as feel of the actual/final product and simulates a few aspects of it.	Mockup Builder, Serena prototype composer
Tools for Web Development	Web design tools assist in designing web pages with all component elements such as forms, script, text, graphics and so on. They provide a live preview of the system as well.	Dreamweaver, Fontello, Foundation 3, Adobe Edge Inspect, Brackets
Tools for Quality Assurance	These tools serve in monitoring the product engineering process and adopted methods to create the software product so as to ensure conformity with the organization's quality standards. They consist of tools for change control tools, configuration, and software testing tools.	SoapTest, TestingWhiz, AppsWatch, Selenium, JMeter, TestComplete, Watir, Ranorex
Tools for Maintenance	Tools for modifying and maintaining the deployed software. Automatic error reporting mechanism and logging, automatic error ticket generation as well as root cause Analysis are among common computer-aided tools used by software organizations in order to maintain the phases of the IS development lifecycle	HP Quality Centre, Bugzilla for defect tracking,

Diagramming tools		•		
Tools for Design		•		
Tools for Maintenance			•	
Tools for Process Modelling	•	•		
Tools for Analysis	•	•		
Tools for Configuration Management		•	•	
Tools for Quality Assurance			•	•
Tools for Programming			•	•
Tools for Prototyping	•			•
Tools for Web Development		•	•	•
Tools for Project Management	•	•	•	•
Tools for Documentation	•	•	•	•
	Specification	Design	Implementation	Verification and Validation

Fig 1. Activity-based classification of CASE tools

Despite the constant evolution of IS development approaches in the last few decades, the focus of computer-aided approach remains the improvement of the overall IS development process. As stated earlier, computer-aided tools are mainly utilized in order to reduce the duration and cost of development projects. The tools also improve the quality of the developed IS. Developers preferred computer-aided tools for their usability and enhanced productivity; making the system development task more enjoyable [5], [17].

Challenges include time spend on training the developer on how to work with the tools and selecting the most suitable tool for a particular context. Most of the tools are not free and need upgrades. Therefore, project budget if not for large-scale might not justify the expenditure [17].

3.4 Selecting the Most Suitable Computer-Aided Tool

One of the most important organization-wide best practice involves the careful selection of tool suitable for a particular IS development. This is where a well-documented development process will play a vital role when selecting a computer-aided tool, and providing documented training stages as a key factor in determining the scope and quality of training offered by the vendor. At the initial implementation phase, vendor support can be the most crucial activity in using a computer-aided tool [5], [21]. Some development projects might pose specific needs with the varying level of customization that might be critical or otherwise, especially if an off-the-shelf software/tool does not suit the needs of the development project. It is paramount to consider the compatibility of the deployed tool with existing components and interfacing with other important software components (e.g. operating system software). Delivering the value for money is the bottom line expected from any computer-aided tool deployed. Therefore, both qualitative and quantitative costs must be considered in order to assess the expected value of a tool [1]. However, despite recent advances in the application of CAT in other sciences [22], [23] and advances in IS development with artificial intelligence [24], [25],

development with the focus of overall improvement of the development process. Ideally, an integrated computer-aided tool is a suite of assorted tools that enable efficient and accurate system development through all phases of the IS development. Fine-tuning of existing tools is one of the key factors for the increased user acceptability, due to the perceived usability issues. In this study we reviewed the key concepts in IS, and highlights the most prominent CATs that are widely used and how these tools fit into different phases of IS development. We also hinted the efficacy of CATs in user-centred and end-user IS development. Accordingly, the study findings have shown that future research directions on using computer-aided tools in IS development will likely pay attention to further automation of the early IS lifecycle phases; to elicit requirements by interpreting natural language used in writing requirement document. Language independent IS development is another trend that would play a vital role in the evolution of computer-aided tools. Another trend is on human-centred computer-aided tool(s) development; to improve the usability of the current tools so as to fill the gap between user mental model of the integrated tools and the way it can be used. Furthermore, future web-based computer-aided tools in alignment with the trends in cloud computing will as well face an enamours advancement. Nonetheless, as the usability of computer-aided tools in an organization solely depends on the proper use of the tools and selection of the right tool for the right job, future IS development team must first decide if the scale and/or needs of a project justify the costs associated with the computer-aided tool(s) to be used and must then select the right tool from the hundreds available. Proper upskill, future IS development teams can utilize integrated computer-aided tool(s) in automating all the activities involved in the entire IS development phases, producing a quality IS that will require less maintenance.

4. CONCLUSION

Computer-aided tools used in IS development are software products that automate/support part of, or the entire IS development process. Using computer-aided tools is an efficient way of supporting IS

REFERENCES

- [1] R. M. Stair and G. W. Reynolds, *Information systems essentials*, 5th ed. Australia: Course Technology Cengage Learning, 2012. Accessed: Apr. 07, 2018. [Online]. Available: https://books.google.com.cy/books/about/Information_Systems_Essentials.html?id=T2TsmgEACAAJ&redir_esc=y
- [2] P. Isaias and T. Issa, "Information System Development Life Cycle Models," in *High level models and methodologies for information systems*, New York: Springer Science + Business Media, 2015, pp. 21–40. doi: 10.1007/978-1-4614-9254-2.
- [3] N. Ahituv, S. Neumann, and M. Zviran, "A system development methodology for ERP systems," *J. Comput. Inf. Syst.*, vol. 42, no. 3, pp. 56–67, 2002, [Online]. Available: http://staf.cs.ui.ac.id/WebKuliah/TKSI/A_Systems_Development_Methodology_for_ERP_Systems.pdf
- [4] P. . Loucopoulos and V. Karakostas, "C.A.S.E. Technology Introduction," in *System Requirements Engineering*, New York, NY, USA: McGraw-Hill, Inc., 1995, pp. 208–232. doi: 10.1016/B978-0-444-52734-9.50009-8.
- [5] V. Berdonosov and E. Redkolis, "TRIZ-fractality of computer-aided software engineering systems," *Procedia Eng.*, vol. 9, pp. 199–213, 2011, doi: 10.1016/j.proeng.2011.03.112.
- [6] A. A. Lawan, A. S. Abdi, A. A. Abuhassan, and M. S. Khalid, "What is Difficult in Learning Programming Language Based on Problem-Solving Skills?," 2019. doi: 10.1109/ICOASE.2019.8723740.
- [7] A. Zakari, A. A. Lawan, and G. Bekaroo, "A Hybrid Three-Phased Approach in Requirement Elicitation," in *Lecture Notes in Electrical Engineering*, vol. 416, 2017, pp. 331–340. doi: 10.1007/978-3-319-52171-8_30.
- [8] H. Lieberman, F. Paternò, M. Klann, and V. Wulf, "End-User Development: An Emerging Paradigm," in *End User Development*, Dordrecht: Springer Netherlands, 2006, pp. 1–8. doi: 10.1007/1-4020-5386-X_1.
- [9] D. Aveson and G. Fitzgerald, "Methodologies for Developing Information Systems: A Historical Perspective," in *The Past and Future of Information Systems: 1976–2006 and Beyond*, Springer., vol. 214, J. . Avison, D.; Elliot, S.; Krogstie, J.; Pries-Heje, Ed. Boston: International Federation for Information Processing, 2006, pp. 27–38. doi: 10.1007/978-0-387-34732-5_3.
- [10] S. Balaji and M. M. Sundararajan, "Waterfall Vs V-Model Vs Agile : A Comparative Study on SDLC," *Int. J. Inf. Technol. Bus. Manag.*, vol. 2, no. 1, pp. 26–30, 2012, doi: 10.1.1.695.9278.
- [11] I. Sommerville, *Software Engineering*, 9th ed. United States of America: Pearson Education, Inc, 2011. doi: 10.1111/j.1365-2362.2005.01463.x.
- [12] M. Boahene and S. Bowles, "Information systems development methodologies: Are you being served," *Proc. 10th Australas. Conf. Inf. Syst.*, pp. 88–99, 1999.
- [13] S. B. Denerel and T. Birişçi, "Drawing Tools Used from Past to Present in the Architectural Design Process#," *J. Int. Environ. Appl. Sci.*, vol. 12, no. 3, pp. 230–237, 2017.
- [14] S. Cronholm, "Why CASE Tools in Information Systems Development?-an empirical study concerning motives for investing in CASE tools," *Proc. 18th Inf. Syst. Res. Semin. Scand. IRIS*, vol. 18, pp. 132–144, 1995, [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.34.853&rep=rep1&type=pdf>
- [15] D. Jankowski, "Computer-aided systems engineering methodology support and its effect on the output of structured analysis," *Empir. Softw. Eng.*, vol. 38, pp. 11–38, 1997, doi: 10.1023/A:1009790131937.
- [16] R. Sudarsan, S. J. . Fenves, R. D. . Sriram, and F. Wang, "A product information modeling framework for product lifecycle management," *CAD Comput. Aided Des.*, vol. 37, no. 13, pp. 1399–1411, 2005, doi: 10.1016/j.cad.2005.02.010.
- [17] E. P. E. Turban, R. K. Rainer, and Richard, "Information System Development," in *Introduction to Information Technology*, 1st ed., Wiley, 2000, p. 461. [Online]. Available: www.wiley.com/college/turban/0471073806/sc/ch14.pdf
- [18] A. Demirci, "Student Views on the Use of Computer Aided Graphic Design Courses Abstract," *e-Journal New World Sci. Acad.*, vol. 6, no. 4, pp. 472–484, 2011.
- [19] A. Ozkan and K. Yıldırım, "Comparison of Conventional and Computer-aided Drafting Methods from the View of Time and Drafting Quality," *Eurasian J. Educ. Res.*, no. 62, pp. 239–254, 2016, doi: 10.14689/ejer.2015.58.2.
- [20] A. Forward and T. C. Lethbridge, "The relevance

of software documentation, tools and technologies,” in Proceedings of the 2002 ACM symposium on Document engineering - DocEng '02, 2002, p. 26. doi: 10.1145/585058.585065.

[21] P. Vik, D. Luís, P. Guilherme, and J. Oliveira, “Layout and process optimisation: using computer-aided design (CAD) and simulation through an integrated systems design tool,” *Int. J. Simul. Process Model.*, vol. 9, no. 1/2, p. 46, 2014, doi: 10.1504/IJSPM.2014.061437.

[22] P. Pérez, J. A. Serrano, M. E. Martín, P. Daza, G. Huertas, and A. Yúfera, “A computer-aided design tool for biomedical OBT sensor tuning in cell-culture assays,” *Comput. Methods Programs Biomed.*, vol. 200, p. 105840, Mar. 2021, doi:

10.1016/J.CMPB.2020.105840.

[23] M. M. Fischer, H. J. Scholten, and D. Unwin, “Geographic information systems, spatial data analysis and spatial modelling: an introduction,” *Spat. Anal. Perspect. GIS*, pp. 3–20, Mar. 2019, doi: 10.1201/9780203739051-1.

[24] Q. H. Vuong et al., “Artificial intelligence vs. Natural stupidity: Evaluating ai readiness for the vietnamese medical information system,” *J. Clin. Med.*, vol. 8, no. 2, 2019, doi: 10.3390/jcm8020168.

[25] B. Tang et al., “AI-Aided Design of Novel Targeted Covalent Inhibitors against SARS-CoV-2,” *Biomolecules*, vol. 12, no. 6, 2022, doi: 10.3390/biom12060746.