Categorisation of Enterprise Resource Planning (ERP) Failures: An Opportunity for Formal Methods in Computing

S. K. Senaya, J. A. van der Poll and M. A. Schoeman

***Abstract*—**Enterprise Resource Planning (ERP) systems are increasingly being used worldwide to align organisations' business processes and optimise their activities for value-proposition. However, due to the complexity of ERPs, they are often characterised with system failure resulting in huge loss of investments organisations have made in the acquisition and deployment of such systems. In this paper the components of a modern ERP are identified and reasons for such failures are elicited through a comprehensive analysis of the literature, together with a case study from the researchers' work environments. ERP failures are categorised and the techniques of formal methods (FMs) are discussed. An example of the utility of formal specification using Z, is illustrated in the context of the case study. Future work in this area is indicated as the development of an ERP failures framework, followed by a framework to address the said ERP failures.

***Keywords*—**Categorisation, ERP failures, SDLC, Formal methods (FMs), Formal specification.

I. Introduction

In many organisations, departments or units work to attain certain goals and objectives, some of which are not congruent with those of the organisation. This may be due to the absence of a common system to harmonise their operational activities. Such challenges are arguably best solved through the introduction of Enterprise Resource Planning (ERP) systems [1], [2]. Numerous organisations have reverted to using ERP systems to optimise business processes and work efficiency for creating a strategic and competitive advantage [3], [4].

ERP systems incorporate data from all functional areas within the organisation into a single unit [5]. An ERP facilitates data and information sharing across the various departments to enable information flow between all business units, and to manage connections to stakeholders of the organisation [6].

ERPs are considered to be the spine of an organisation because their success or failure affect the goal of the organisation [7]. ERP systems are important tools for creating a competitive advantage in the knowledge-based business [8], yet, they are complex in nature and faced with the possibility of failure, consequently they need to be carefully designed [9].

Many software engineering companies assert that their ERP systems are agile and customisable to the business processes of any organisation [10]. However, an organisation's business processes are usually specific to their activities, which need to be critically considered and factored into the development or re-engineering of any systems. This, therefore, renders such agile-based assertions as inaccurate.

In this paper we investigate the components and life-cycle development of an ERP system and elicit some of the common failures of ERPs. These failures are subsequently categorised and an example of one such failure in terms of an industry case is investigated in terms of a formal-methods (FMs) approach.

The layout of the paper is: Following this introduction we state our research questions in Section II. Components of a modern ERP system are presented in Section III. Corresponding SDLC life cycle aspects applicable to ERPs are presented in Section IV. Various ERP failures from the literature are presented and categorised in Section V. The discussion on ERP failures is augmented in Section VI through an industry case. Owing to their potential use, FMs are presented in Section VII, followed by a brief illustration of their utility in Section VIII. A synthesis of ERP failures appears in Section IX, followed by conclusions and future work in Section X.

II. Research Question (RQs)

The RQs addressed in this paper are:

1. What are the components of a modern ERP system?
2. What are the standard ERP life cycles (ERPLCs)?
3. What are the causes and effects of ERP system failure in organisations?
4. How may the use of FMs facilitate correct ERP functionality?

III. ERP Components

Reference [11] asserts that most modern ERP systems comprise of eight (8) system components: Human Resources (HR); Inventory; Sales and Marketing; Procurement; Finance and Accounting; Customer Relationship Management (CRM); Engineering/Production; and Supply Chain Management (SCM). Fig 1 shows how the functional components of an ERP system are linked [12].

Manuscript received September 10, 2019.

S K. Senaya is a PhD Candidate in the School of Computing, Unisa, South Africa, Johannesburg.

J A. van der Poll is with the Graduate School of Business Leadership (SBL), Unisa, South Africa, Midrand.

M A. Schoeman is with the School of Computing, Unisa, South Africa, Johannesburg.

https://doi.org/10.17758/EARES8.EAP1119287

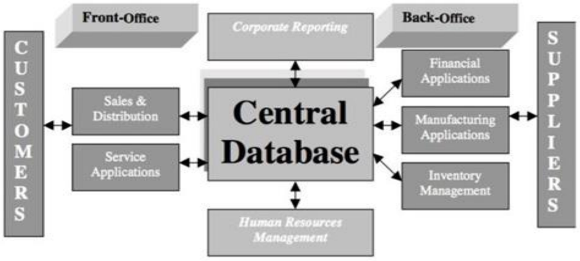

Fig. 1. ERP System Components Linking Functions [12].

The conceptual diagram in Fig. 1 indicates an ERP system has characteristics and features that integrate all activities within the organisation through a common *database* that connects all departments and allows for BI (Business Intelligence) throughout the organisation. Both *back and front offices* processes are linked directly to the central database, allowing for synchronisation of data from all functional units. The Suppliers and Customers are (indirectly) linked to the same database through the *Manufacturing Application* and *Sales and Distribution/Service Applications* modules respectively, offering end-to-end capabilities to the system. Note that owing to the said integration, any system/function error in any module of the system could affect the central database.

The above discussion provides an answer to our **RQ1**.

IV. ERP Life Cycle

From references [13], [14], and [15] we can synthesise the following ERPLC (ERP Life Cycle):

- Organised planning
- System/Platform selection
- Installation
- Data conversion and loading
- Procedure development
- User training
- Testing and validation
- Transition to new system (from old one)
- Project completion (miscellaneous back-end activities)

The implicit linear life cycle above shows strong similarities with the familiar Waterfall SDLC. Despite criticisms of the waterfall model, recent work stills supports Waterfall for the software development part of an ERP [16].

Reference [17] advances a model for the SDLC that supports object-orientation for component-based system development like ERPs (refer Fig. 2).

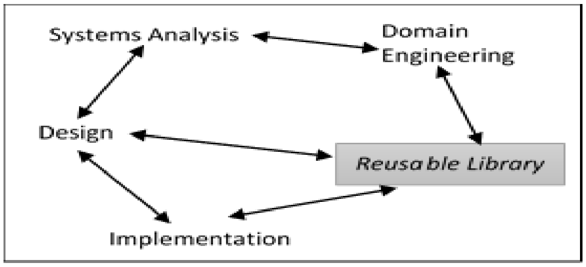

Fig. 2. Component-Based Software Development Model [17]

The above discussion gives an answer to our **RQ2**.

Organisations are often faced with challenges with ERP system components because most software developers do not properly specify the processes in the system [18]. Therefore, since an ERP includes many organisational components, closer analysis may well reveal its higher-level life cycle to be different (more complex) from the traditional SDLCs (e.g. Waterfall) or the later OO (object-oriented) approaches.

As discussed further on in this paper, ERPs often suffer failure, and to alleviate these at a higher level, developers may employ techniques such as hyper-heuristics in an intelligent system design during the specification phase [19]. The researchers furthermore argue that the use of formal methods (FMs) at selective levels of ERP development could further alleviate failure (refer sections VII and VIII).

V. Causes And Effects Of ERP System Failure

In recent times, organisations are progressively focusing on factors that lead to ERP system failures for which few or no models (currently) exist to resolve the challenge [20].

The Standish Group indicted in 2014 that 31.1% of projects will be cancelled before completion, and 52.7% of projects will cost 189% more than the original budget [21]. Similarly, [22] indicated some major flaws associated with software failure in their study in addition to those often indicated by the Standish Group in their Chaos report on application software development. The Standish Group attributed software defects as an important reason for software failure [23] in their report titled: “Worst Case of Malpractice Ever”.

Naturally, the failure of an ERP has a negative effect on an organisation’s productivity which could lead to financial challenges [24]. Consequently, in the following sections we identify, following a thematic approach various causes of ERP failures in the literature. We start with more general, and mostly non-functional aspects and move to more specific, functional aspects.

A. *Complexity of ERP Systems*

Owing to their multi-component structure, Information System applications such as ERPs are often confronted with system challenges; and are problematic to develop, customise, or modify [25], [26]. From the ERP failure cases presented by [27] and [28], it could be deduced that, the sheer complexity of the system, resulting from software modifications to satisfy changes in the organisation’s business process caused its failure.

In an attempt to combat ERP complexity, early calls were made by [29] for a model to explain the complexity of an ERP system in *basic terms*. To this end the schema calculus of the Z specification language (Section VII) could assist in adhering to basic terms. Object-oriented versions, e.g. Object-Z could also be useful to divide an ERP system in more manageable parts.

B. *Human Resources (HR)*

The research in [30] indicate that system deficiency or failure could arise when (amongst other) company strategy and human resource capabilities are not correctly addressed. Reference

[31] reports a lack of customer interaction and incorrect expectations as important reasons for ERP failure. While interaction with customers may not be strictly an (internal) HR aspect, these nevertheless involve humans in the process.

Ultimately, the success of a software project and therefore an ERP is dependent on the skill sets of the systems developers. In certain dimensions this may be classified as an HR aspect. In fact, [18] lists the unavailability of individuals with the correct technical skills within an organisation to support the functionalities of the system as a major cause of ERP system failure. Reference [32] concurs that one of the major reasons for ERP failure is the people (individuals) involved.

In line with the above observation, we postulate that individuals with the necessary FMs skills may facilitate ERP success, albeit at a detailed level.

C. Project Management (PM)

Inadequate (software) project management practices have also been cited as reasons for ERP failure [33]. In support, [8] attributes poor software project management processes as being the major cause of ERP system failure. Reference [34] presents a simplified project management maturity model for developing countries, and it is plausible that simpler models may address the PM complexities of ERP development in (e.g.) developing countries.

D. Business Processes (BPs)

One of the major reasons behind the failure of ERP systems stems from the system not being correctly aligned with the business processes of the organisation [35] because often systems that do not meet the company's specific requirements are chosen [12].

Amongst other reasons, [31] and [32] also attribute ERP failure to incorrect businesses processes.

Reference [36] reports that critical ERPs fail to create value not because they could not process the numerous instructions given by the user, but because they are unable to maintain the rule bases. This is significant since [37, p 97] regards both CRM and SCM (Supply Chain Management) applications to be mission critical.

Following the above (more) general reasons for ERP failure, we now turn to more specific aspects, albeit still high level.

E. Software Model Alignment

As may be expected, software challenges are at the heart of ERP failure. ERP systems fail due to inadequate procedures that software engineering companies follow when creating these complex systems [38]. This could also be deduced from the study by [39] on the "FoxMeyer Drugs" ERP failure – the failure is attributed to the adoption of inadequate software model alignment during business expansion. Likewise, [9] aligns inappropriate development techniques followed in the design stage of the system development life cycle leading to financial losses and bankruptcy of affected organisations.

As mentioned above, the use of the Waterfall approach in building ERP software components is recently still being proposed by researchers, e.g. [16]. Yet, in line with (more) current developments we suggest the use of an agile-based methodology combined with a lightweight formal-methods approach at the heart of the critical parts of the ERP engine. To this end, Fig. 3 indicates a generic agile approach with pockets of FMs during each design sprint or iteration (Section VII elaborates on FMs aspects).

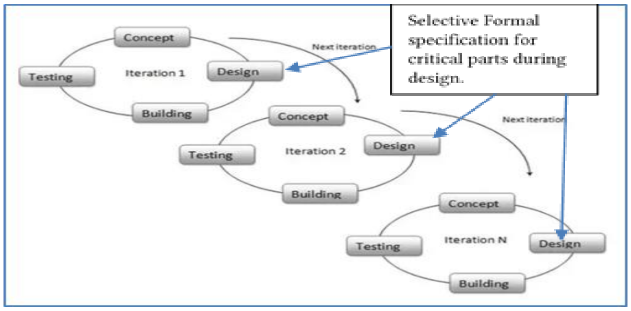

Fig. 3. Agile Software Methodology (Adapted from https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm).

F. Partial ERP Module Integration

Fig. 1 indicates a central database to be underlying an ERP system. More presently, however, the trend seems to favour the use of a Data Warehouse (DW) in which a number of operational databases are integrated, together with BI (Business Intelligence) applications on top of it [40]. We get:

$$\mathit{Amalgamate}: DB1 \times DB2 \times \ldots \times DBn \nrightarrow DW \tag{1}$$

Where each DBi, $1 \leq i \leq n$ represents an operational database and DW the resultant data warehouse.

Yet, such integration of databases into a common data-warehouse domain, may initially lead to failure [41]. Failure of correctly integrating legacy systems with a new ERP, or running mixed standalone systems with a partial ERP may lead to inconsistencies and incompatibilities [18].

In support of the above, [42] enumerates some well-known ERP system failure cases:

- The case of Hershey's mistake which resulted from the linking of their supply chain application to the CRM component of their Systems Application Product (SAP) ERP; and
- The Nike's Supply Chain challenge which occurred after an update of the company's supply chain system with a new demand-planning software solution.

G. Hidden Information

Often during systems upgrade or migration to a new system some code and/or data becomes hidden. It is also possible that certain modifications to the system affect other parts that are not immediately clear to the developers. Consequently, ERPs may fail owing to hidden information having a negative effect on system functions [12], [24].

Hidden information is related to traceability identified below. Consequently, FMs proofs could identify and alleviate the challenge accordingly. Note how the hidden information problem also links with the partial ERP module integration challenge discussed before.

H. Traceability

Traceability of ERP elements links to many aspects introduced above. Yet, it is given special mention by [43] in that the correct linking of elements from the system model and the source code to provide traceability of data captured through the software development process will facilitate the production of reliable ERPs.

Directly related to traceability is the discharging of proof obligations (POs) resulting from a formal specification. A good example is given in [44] in the context of Spivey's *BirthdayBook* example.

I. Reliability

Our last failure element (reliability) stems from general observations.

ERP reliability, or the lack thereof may be classified as an overarching, non-functional concern (requirement) that governs all aspects of ERPs. From an interpretive study of an ERP failure implementation case (in which FoxMeyer purchased two different systems and used a third company to integrate and implement the two systems) in [39], it can be deduced that the cause of the ERP system failure was due to system unreliability.

Reliability is classified as a non-functional requirement and similar to defining the availability of web services [45], one could develop a formal (Z-like) definition for ERP reliability.

The above concludes our discussion of ERP failure elements from qualitative literature, giving a partial answer to **RQ3**.

Next, we discuss an ERP failure in the researchers' work environment.

VI. A Case Of ERP System Failure

An industry case of a *Finance and Accounting System* of a modern ERP considers students' payment and registration functions to illustrate possible causes of failure, focusing on the critical function areas with respect to the system processes.

The system process functions of a student payment and registration system is shown in Fig. 4 and analysed thereafter.

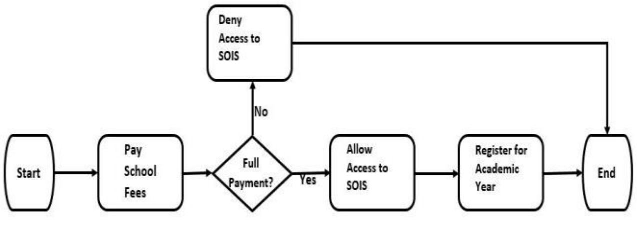

Fig. 4. Student Payment and Registration System Process (synthesized by researchers).

The registration process starts with customers, in this case students at a university who receive an invoice (also known as a bill of payment) from the university at the start of an academic year and then effect payment in full or part into the University's official bank account. A daily update of student payment transactions is imported into the University's financial system to credit all respective student accounts with amounts paid.

The payment of fees has implications for registration in that students are allowed access by the Student Online Information System (SOIS) to register for the academic year only if they made full payment per the tuition- and finance policy of the university. The finance and accounting systems update the SOIS to allow or deny student registration based on the amount configured for the student as full payment.

Following a decision by the university authorities, the system was re-engineered to accommodate those students who were financially challenged and were unable to make a full payment upon registration. However, an ERP failure occurred following the re-engineering: The system allowed students with partial or no payment to be fully registered, access all academic resources, and continue with their studies, while it denied access to students who made a full payment upon registration.

The failure resulted from the SOIS being unable to correctly update the relevant operational database(s) in the data warehouse following the re-engineering. Plausibly, this problem could be a result of data duplication arising from multiple relational databases across functional units, or an example of partial/hidden information discussed above.

Following the above industry case of an ERP failure, we arrive at a more complete answer to **RQ3**.

The use of Formal Methods (FMs) mentioned above has been advocated in the construction of reliable software, e.g. [46] and [19] have enumerated the approach of formal-method techniques in their work as a way of developing reliable systems for mission-critical systems which may also apply to ERPs in such environments. Consequently, we present further on an attempt at a formal specification of the above SOIS registration-and-payment system, but first a brief overview of some aspects of FMs.

VII. Formal Methods (FMs) IN ERP Software Construction

In our context the use of FMs embodies the use of Discrete Mathematics and logic for the construction of provable, or at least highly-dependable software [44], [46] – [48]. Reference [49] concludes that the use of FMs offers the best approach towards constructing reliable software. Owing to the numerous possibilities for ERP failure indicated before, it is plausible that the use of FMs may be useful in the building of these systems, not only at the (lower) software level, but also higher up in packaging the various ERP components (c.f. Fig. 1).

Using a FMs approach, a specifier usually starts out by developing a formal specification of the system to be developed. Such specification defines the desired properties of the resultant system [50]. A formal specification also serves as a reliable way for examining processes, satisfying requirements and testing results, as well as writing instructional documents for software systems [44]. Reference [51] asserts that FM techniques help to construct a formal procedure and they show how these may positively impact the development of critical systems.

An important aspect of a FMs approach is to identify and discharge proof obligations (POs) arising from the specification, e.g. show the specification, and therefore the resultant system, will have certain desirable properties (i.e. "testing" of results), and equally important, show that undesirable properties are absent from the specification. These are equally relevant to ERPs, since one would strive to achieve correctness of critical systems such as ERPs [37].

According to [52], researchers over the past decades have worked extensively on various (some of these FMs) theories, for example; Theory of Reasoned Action (TRA), Task-technology Fit Model and the IS Success Model to solving ERP system challenges at the design phase. The Formal Concept Analysis (FCA) practice also aids modelling data entities and their relations, as well as analysing the intent and extent of the data [53]. The data are organised into parts. They include formal concept abstractions of what is understood by humans to give a clear interpretation of the system at the specification phase [54].

The formal specification language Z [44], [50] is a lucrative FMs option for abstract system specification. Z is based on 1^{st} -order logic and a strongly-typed fragment of Zermelo-Fraenkel (ZF) set theory [55]. An example of a Z specification is given in relation to the case of ERP failure, namely the SOIS finance and accounting subsystem described above.

Despite the value of FMs to produce correct and dependable systems [56], especially in mission/safety-critical applications, we note that the adoption of FMs techniques in practice amongst software engineers is not encouraging [57]. FMs usage embodies a steep learning curve in mastering the underlying mathematics and logic. Yet, the researchers argue it is no harder or easier than to master one's first programming language.

VIII. Specifying The Sois Registration System

The industry case – Student registration system – above presents an interesting opportunity for specifying a sequence of operations, namely, the registration of students, together with no payment yet, or registration together with payment in full.

The Z specification language is modular in the sense that individual operations may be specified individually, and subsequently combined through a schema calculus to obtain larger operations (schemas).

Two definitions of the above processes (Registered and payment pending; or Registered and paid in full) in Z are given below (the use of a "′" denotes an after state of a component):

$$\begin{array}{|l}
\multicolumn{1}{l}{\_Register\_\_\_\_\_\_\_\_\_\_} \\
\Delta\ Students \\
s? : Student \\
\hline
s? \notin registered \\
registered' = registered \cup \{ s? \} \\
payed' = payed \\
\hline
\end{array}$$

$$\begin{array}{|l}
\multicolumn{1}{l}{\_Pay\_\_\_\_\_\_\_\_\_\_} \\
\Delta\ Students \\
s? : Student \\
amount? : \mathbb{R} \\
\hline
s? \in registered \\
payed' = payed \cup \{ s? \mapsto amount? \} \\
registered' = registered \\
\hline
\end{array}$$

In the interest of space, the state schema *Students* where the components *registered* and *payed* (amongst others) would be defined, is not shown. Also, basic Z does not have real numbers (ℝ), but we abstract away from such detail.

Registering a student and accepting full payment could be modelled as a combined operation using Z schema calculus:

$Register\_and\_Pay \mathrel{\widehat{=}} Register \wedge Pay$

The combined operation is:

$$\begin{array}{|l}
\multicolumn{1}{l}{Register\_and\_Pay\_\_\_\_\_\_\_\_\_\_} \\
\Delta\ Students \\
s? : Student \\
amount? : \mathbb{R} \\
\hline
s? \notin registered\ \wedge s? \in registered \\
payed' = payed\ \wedge (payed' = payed \cup \{ s? \mapsto amount? \}) \\
registered' = registered \wedge (registered' = registered \cup \{ s? \}) \\
\hline
\end{array}$$

The combined operation illustrates inconsistent behaviour, illustrating the value proposition in using formalisms in ERP software development, c.f. the SOIS case above. The above example is in fact one of the so-called "dark corners" of Z, at least as far as the schema calculus is concerned.

The above FMs presentation gives an answer to **RQ4**.

IX. Summary Of ERP Challenges

A number of challenges leading to ERP failure has been elicited above. These are summarised in Table 1 below, together with the suggestions for further investigation as to feasible solutions for each.

TABLE I: SUMMARY OF ERP CHALLENGES AND SOLUTIONS

Challenge	Literature	Solution
Complexity	[25]–[28]	Approach ERP development in an incremental and modular fashion. Z's schema calculus could assist with a *basic-terms* ERP. Else consider an OO version, e.g. Object Z.
Human Resources	[18]	Pay attention to the value to be added by FMs individuals.
Project Management	[8], [33]	Adopt simpler PM practices for ERP complexities – [34].
Business Processes	[12], [35], [36]	Adopt basic-terms ERP development (refer above).

https://doi.org/10.17758/EARES8.EAP1119287

Challenge	Literature	Solution
SDLC Alignment	[9], [38], [39]	Consider Agile with a light-weight FMs approach for the specification phase (c.f. [57]).
Traceability	[43]	State and discharge POs resulting from a formal spec [44].
Hidden Information	[12], [24]	State and discharge POs.
Partial ERPs (module integration)	[41], [42]	Avoid partial ERP systems. Link all components in one enterprise system.
Reliability	[39], [45]	Develop a formal specification for ERP reliability.

X. Conclusion

This paper defined a modern ERP as comprising of eight (8) major parts. Following a qualitative and thematic approach a number of challenges that often lead to ERP failure has been identified in the literature as well as a case study from the work environment of the researchers. Preliminary solutions to these challenges have been proposed as summarised in Table 1. Some of the challenges are functional in nature while others tend to have a non-functional character. The use of formal methods, especially during the specification phase has been suggested as a solution for some of the challenges. Central to these is the role of qualified individuals to take up these tasks.

Future work in this ERP arena may be pursued along a number of avenues: The challenges identified have to be synthesised into a "challenges" framework for further analyses, e.g. specialisations and instantiations of these have to be investigated. On the strength of such framework, a framework to address ERP challenges will have to be constructed. It is anticipated that the use of FMs will play a major role in the new framework. Following the initial development of the framework, it has to be validated through industry cases and surveys among practitioners to eventually arrive at a formal ERP model that could be used by ERP software engineers.

References

[1] E. Nazemi, M. J. Tarokh, and G. R. Djavanshir, "ERP: a literature survey," *Int. J. Adv. Manuf. Technol.*, vol. 61, no. 9–12, pp. 999–1018, Aug. 2012. https://doi.org/10.1007/s00170-011-3756-x

[2] S. Nizamani, K. Khoumbati, I. A. Ismaili, and S. Nizamani, "A conceptual framework for ERP evaluation in universities of Pakistan," *Sindh Univ. Res. J.*, vol. 45, no. 3, pp. 467–475, 2014.

[3] A. Mustafa, Y. E. Serra, and K. A. Ekmekçi, "The effect of ERP Implementation CSFs on Business Performance : An Empirical Study on Users ' Perception," *Procedia - Soc. Behav. Sci.*, vol. 210, pp. 35–42, 2015. https://doi.org/10.1016/j.sbspro.2015.11.326

[4] P. Ju, H. Wei, and C. Tsai, "Model of Post-Implementation User Participation within ERP Advice Network," *Asia Pacific Manag. Rev.*, vol. 21, no. 2, pp. 92–101, 2016. https://doi.org/10.1016/j.apmrv.2015.11.001

[5] C. A. Rajan and R. Baral, "Adoption of ERP System: An Empirical Study of Factors Influencing the Usage of ERP and its Impact on End User," *IIMB Manag. Rev.*, vol. 27, no. 2, pp. 105–117, Jun. 2015. https://doi.org/10.1016/j.iimb.2015.04.008

[6] Y. E. Gelogo and H. Kim, "Mobile Integrated Enterprise Resource Planning System Architecture," *Int. J. Control Autom.*, vol. 7, no. 3, pp. 379–388, 2014. https://doi.org/10.14257/ijca.2014.7.3.36

[7] M. Y. Jamil and R. Qayyum, "Enterprise Resource Planning (ERP) Implementation in Pakistani Enterprises: Critical Success Factors and Challenges," *J. Manag. Res.*, vol. 2, no. 2, pp. 1–35, 2015.

[8] P. D. Deshmukh, G. T. Thampi, and V. R. Kalamkar, "Investigation of Quality Benefits of ERP Implementation in Indian SMEs," *Procedia Comput. Sci.*, vol. 49, no. 1, pp. 220–228, 2015. https://doi.org/10.1016/j.procs.2015.04.247

[9] E. Abu-Shanab, R. Abu-Shehab, and M. Khairallah, "Critical Success Factors for ERP Implementation: The Case of Jordan," *Int. Arab J. e-Technology*, vol. 4, no. 1, 2015.

[10] B. Aslan, M. Stevenson, and L. C. Hendry, "The applicability and impact of Enterprise Resource Planning (ERP) Systems: Results from a Mixed Method Study on Make-To-Order (MTO) Companies," *Comput. Ind.*, vol. 70, pp. 127–143, 2015. https://doi.org/10.1016/j.compind.2014.10.003

[11] G. Pravin, "Basic Modules of ERP System," *ESDS*, 2013. [Online]. Available: https://www.esds.co.in/blog/basic-modules-of-erp-system/#idc-container . [Accessed: 16-Oct-2017].

[12] R. Mazzawi, "Enterprise Resource Planning Implementation Failure : A Case Study from Jordan," *J. Bus. Adm. Manag. Sci. Res.*, vol. 3, no. 5, pp. 79–86, 2014.

[13] C. Weiss, M. Ko, J. Keckeis, and R. Friedemann, "Qualitative Analysis of Different ERP Evaluation Models," *Innov. Enterp. Inf. Syst.*, vol. 285, pp. 17–25, 2017. https://doi.org/10.1007/978-3-319-58801-8_2

[14] R. Rebello, "ERP Implementation Life Cycle - What is it?," 2017. [Online]. Available: https://www.acumatica.com/blog/erp-implementation-life-cycle/. [Accessed: 30-Apr-2019].

[15] 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.

[16] M. Kramer, "Best Practices in Systems Development Lifecycle: An Analyses Based on the Waterfall Model," *Rev. Bus. Financ. Stud.*, vol. 9, no. 1, pp. 77–84, 2018.

[17] L. F. Capretz, M. A. M. Capretz, and D. Li, "Component-Based Software Development," *27th IEEE Conf. Ind. Electron. Soc.*, 2001. https://doi.org/10.1007/978-1-4471-0719-4_53

[18] T. Barker and M. N. Frolick, "ERP Implementation failure: A Case Study," *Inf. Syst. Manag.*, vol. 20, no. 4, pp. 43–49, 2003. https://doi.org/10.1201/1078/43647.20.4.20030901/77292.7

[19] N. Pillay, "Intelligent System Design using Hyper-Heuristics," *South African Comput. J.*, no. 56, pp. 107–119, 2015. https://doi.org/10.18489/sacj.v56i1.268

[20] R. G. Saadé, H. Nijher, and M. Sharma, "Why ERP Implementations Fail – A Grounded Research Study," *Proc. informing Sci. Inf. Technol. Educ. Conf.*, vol. 62, pp. 191–200, 2017.

[21] T. Clancy, "The Standish Group Report Chaos - Project Smart", 2014.

[22] E. J. Laurenz and C. Verhoef, "The Rise and Fall of the Chaos Report Figures," 2010.

[23] H. Mulder, "Worst Case of Malpractice Ever," *The Standish Group and Antwerp Management School*, p. 14, 2015.

[24] A. S. Shatat, "Critical Success Factors in Enterprise Resource Planning (ERP) System Implementation : An Exploratory Study in Oman," *Electron. J. Inf. Syst. Eval.*, vol. 18, no. 1, pp. 36–45, 2015.

[25] V. Mittelstädta, P. Braunera, M. Blumb, and M. Ziefle, "On the Visual Design of ERP Systems The – Role of Information Complexity, Presentation and Human Factors," *Procedia Manuf.*, vol. 3, pp. 448–455, 2015. https://doi.org/10.1016/j.promfg.2015.07.207

[26] G. C. Peng and J. M. B. Nunes, "Barriers to the Successful Exploitation of ERP systems in Chinese State-Owned Enterprises," *Int. J. Bus. Syst. Res.*, vol. 4, no. 5/6, pp. 596–620, 2010. https://doi.org/10.1504/IJBSR.2010.035077

[27] M. Babaei, Z. Gholami, and S. Altafi, "Payne," *Inf. Syst.*, vol. 54, pp. 2014–2016, 2015. https://doi.org/10.1016/j.is.2015.05.003

[28] Payne, "A Review of CRM Failures," in *A Review of CRM Failures*, 2007, pp. 23–52.

[29] C. Marnewick and L. Labuschagne, "A Conceptual Model for Enterprise Resource Planning (ERP)," *Inf. Manag. Comput. Secur.*, vol. 13, no. 2, pp. 144–155, Apr. 2005. https://doi.org/10.1108/09685220510589325

[30] C. Salloum and J. Ajaka, "CRM Failure to Apply Optimal Management Information Systems," *Arab Econ. Bus. J.*, vol. 8, no. 1–2, pp. 16–20, 2013. https://doi.org/10.1016/j.aebj.2013.11.002

[31] P. Zerbino, D. Aloini, R. Dulmin, and V. Mininno, "Framing ERP Success from an Information Systems Failure Perspective: A Measurement Endeavor," *J. Electron. Commer. Organ.*, vol. 15, no. 2, pp. 31–47, 2017. https://doi.org/10.4018/JECO.2017040103

[32] I. Friedrich, J. Sprenger, and M. H. Breitner, "CRM Evaluation an Approach for Selecting Suitable Software Packages," *Inf. Softw. Technol.*, no. 2000, 2009.

[33] A. H. Yousef, A. Gamal, A. Warda, and M. Mahmoud, "Software Projects Success Factors Identification Using Data Mining," *2006 Int. Conf. Comput. Eng. Syst. ICCES'06*, no. April 2015, pp. 447–453, 2006. https://doi.org/10.1109/ICCES.2006.320489

[34] A. Sukhoo, A Barnard, M. M. Eloff, and J. A. van der Poll, "Accommodating Soft Skills in Software Project Management," *Issues Informing Sci. Inf. Technol.*, vol. 2, pp. 691–703, 2005. https://doi.org/10.28945/860

[35] S. G. Yaseen, "Critical Factors Affecting Enterprise Resource Planning Implementation : An Explanatory Case Study," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 9, no. 4, pp. 359–363, 2009.

[36] P. Harmon, *The Scope and Evolution of Business Process Management*, no. May 2010. 2016. https://doi.org/10.1007/978-3-642-00416-2_3

[37] D. Chaffey, *Digital Business and e-Commerce Management*, 6th editio., vol. 410, no. 1. Pearson, 2015.

[38] N. Futekova and V. Monov, "Conceptual Framework for Evaluating the Effectiveness of the Implementation of Enterprise Resource Planning Systems in Small and Medium-Sized Enterprises," *Econ. Altern.*, no. 3, pp. 117–125, 2014.

[39] J. E. Scott, "The FoxMeyer Drugs' Bankruptcy: Was it a Failure of ERP?," *Am. Conf. Inf. Syst.*, pp. 223–225, 1999.

[40] M. Mohan, "Processing Data from Diverse Databases," *United States Patent -US8943087B2*, 2015. [Online]. Available: https://patents.google.com/patent/US8943087B2/en. [Accessed: 09-May-2019].

[41] S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, and D. Aumueller, "Triplify – Light-Weight Linked Data Publication from Relational Databases," *Track Semant. Web / Sess. Linked Data*, pp. 621–630, 2009. https://doi.org/10.1145/1526709.1526793

[42] T. Wailgum, "10 Famous ERP Disasters, Dustups and Disappointments," 2009. [Online]. Available: https://www.cio.co.nz/article/296723/10_famous_erp_disasters_dustups_disappointments/. [Accessed: 24-Aug-2018].

[43] A. Delater and B. Paech, "Traceability between System Model, Project Model and Source Code," *Inst. Comput. Sci. Univ. Heidelb.*, pp. 1–7, 2016.

[44] J. Spivey, *The Z Notation: A Reference Manual*, Second Edition, UK, 1992.

[45] M. Lall, J. A. van der Poll, and L. M. Venter, "Towards a Formal Definition of Availability of Web Services, SDIWC," in *International Conference on Computing, Networking and Digital Technologies*, 2012, pp. 273 – 284.

[46] J. A. van der Poll, "A Computing Medley on Program Verification, Specification and Automated Reasoning," *Sch. Comput. UNISA, Pretoria, South Africa*, p. 7, 2008.

[47] J. A. van der Poll, "Formal Methods in software Development: A Road Less Travelled.," *South African Comput. J.*, no. 45, pp. 40–52, 2010. https://doi.org/10.18489/sacj.v45i0.33

[48] A. Adesina-Ojo, "Towards the Formalisation of Object-Oriented Methodologies," University of South Africa, Cambridge, 2011. https://doi.org/10.1145/2072221.2072252

[49] A. Roth, "Specification and Verification of Object-Oriented Software Components," Universität Karlsruhe (TH), 2006.

[50] J. Woodcock and J. Davies, *Using Z Specification, Refinement, and Proof*. Prentice Hall, 1996.

[51] J. A. van der Poll and P. Kotze, "Enhancing the Established Strategy for Constructing a Z Specification," *Sch. Comput. South Africa, 0003*, no. 35, pp. 118–131, 2005.

[52] C. Sorgenfrei, K. Ebner, S. Smolnik, and M. E. Jennex, "From Acceptance to Outcome: Towards an Integrative Framework for Information Technology Adoption," in *Association for Information Systems AIS Electronic Library (AISeL) ECIS 2014 Proceedings*, 2014, pp. 1–18.

[53] R. Cole and P. Eklund, "Analyzing an Email Collection Using Formal Concept Analysis," *Princ. Data Min. Knowl. Discov. Proc. Eur. Conf.*, pp. 309–315, 1999. https://doi.org/10.1007/978-3-540-48247-5_35

[54] A. Alsaig, A. Alsaig, and M. Mohammad, "EAI Endorsed Transactions Context-Based Project Management," *EAI Endorsed Trans. Context. Syst. Appl.*, vol. 4, no. 1, pp. 1–10, 2017. https://doi.org/10.4108/eai.6-7-2017.152902

[55] H. B. Enderton, *Elements of Set Theory*. Academic Press, Inc, 1977. https://doi.org/10.1016/S0049-237X(08)71114-5

[56] G. Jeremy and O. José Nuno, "Teaching Formal Methods," in *Second International Conference, TFM 2009 Eindhoven, The Netherlands, November 2-6, 2009 Proceedings*, 2009, Jeremy Gib., pp. 1–184.

[57] S. Gruner, "Formal Methods and Agile Methods , FM + AM ' 11," *South African Comput. J.*, vol. 6, no. 46, pp. 1–2, 2010 https://doi.org/10.1007/s11334-009-0101-8

Stephen K. Senaya is a PhD Candidate in Computer Science with School of Computing, University of South Africa (UNISA), South Africa. He obtained a B.Ed (Information Technology) from the University of Education Winneba (UEW), Ghana in 2009; Post Graduate certificate in Cyber Law from the Asian School of Cyber Laws, India in 2011; MSc (Information Technology) in 2012 from Mahatma Gandhi University (MGU), India; professional certificates in Hewlett Packard (HP) in 2006, Microsoft Systems Engineering (MCSE) in 2010, CompTIA Security+ in 2010, CompTIA Project+ in 2011 and Computer Emergency Response (CERT) in 2017.

He is currently an ICT Systems Analyst with the ICT Directorate of University of Health and Allied Science (UHAS), Ghana, and responsible for Enterprise Business Systems/Administrative Computing Unit of the University. Before joining UHAS in December 2014, he was an ICT Instructor/Systems Administrator in the Computer Science Department of Ho Technical University, Ghana and provided technical/ICT supports to the University. His research interest includes: specification, reasoning and formalisms of Business ICT systems, Business Intelligence, Internet Technologies, Cloud Computing and Cyber Security.

Prof JA van der Poll obtained a BSc from the University of Stellenbosch, South Africa in 1980 and a Hons BSc (Computer Science) in 1982, also from Stellenbosch. He obtained an MSc (Computer Science) from Unisa (University of South Africa) in 1987 and a PhD in Computer Science also from Unisa in 2001. From 1988 to 2013 he was employed by the School of Computing, Unisa. He became a full Professor in Computing in 2007 and moved to Unisa's Graduate School of Business Leadership (SBL) in Midrand in June 2013.

His research interests are in the construction of highly dependable software for Business ICTs, specifically the formal specification and subsequent reasoning of the properties of Business Intelligence applications, Data warehousing, the IOT, IT Governance, and aspects of the 4th Industrial Revolution. He is an NRF rated researcher, category C2 and supervised numerous Master's and Doctoral students to completion.

Dr MA Schoeman obtained a BSc degree from the University of Stellenbosch, Stellenbosch, South Africa in 1982, and a BSc Hons (Computer Science) in 1983, also from the University of Stellenbosch. She obtained a National Higher Diploma in Tertiary

Education from the Technikon Pretoria, Pretoria, South Africa in 1987, and an MSc in Computer Science from the University of South Africa, Pretoria, South Africa in 2006. She obtained her PhD in Computer Science in 2016 from the University of South Africa.

She is currently a senior lecturer in the School of Computing at the University of South Africa, South Africa on the Science campus in Florida, Johannesburg, South Africa. She was briefly employed as a programmer, before starting her teaching career at the Technikon Pretoria. She subsequently moved to the University of South Africa in 1999. Her research interests and publications include computing education, human-computer interaction, visualisation and mobile agents.