

RISKS OF USER-DEVELOPED APPLICATION IN SMALL BUSINESS

ZULKHAIRI DAHALIN
Faculty of Information Technology
Universiti Utara Malaysia

ABSTRACT

This paper discusses the risks of developing computerised business applications by end-users particularly in the small business environment. As today's end-users are becoming more and more sophisticated, coupled with the proliferation of Information Technology (IT) that has brought the computerisation of business activities within reach of many small firms, understanding the benefits and risks of user-developed applications would contribute towards the small firm's effectiveness in IT adoption. This is particularly relevant in today's managing business where businesses, including the small firms, in a volatile environment will have to compete not only locally but also globally, and IT is seen to be an enabler that can help small firms to increase their competitiveness. Relevant literature on the benefits and risks of user-developed applications were sought and summarised in this paper. Findings were also based on a case study investigation of small firms with no formal IT function where observations were made on the end-user developers who were given the responsibility to develop the firm's computerised applications. Whilst the benefits have been enormously highlighted and at times overshadowed the risks, due attention is given to examining the risks so as to provide a more balanced report and a precautionary measure for end-user developers. Risks were categorised according to organisational and individual risks following the application development stages of planning, analysis, design and implementation.

ABSTRAK

Artikel ini membincangkan tentang risiko yang boleh di hadapi dalam pembangunan aplikasi perniagaan yang dilakukan oleh pengguna akhir terutama dalam persekitaran perniagaan kecil. Pada masa kini pengetahuan dan kemahiran teknologi maklumat pengguna akhir menjadi semakin canggih. Peningkatan penggunaan teknologi maklumat yang semakin meluas membolehkan syarikat-syarikat kecil melaksanakan komputerisasi terhadap

aktiviti-aktiviti perniagaan mereka. Dengan memahami faedah dan risiko pembangunan aplikasi oleh pengguna akhir dapat menyumbang ke arah keberkesanan penggunaan teknologi maklumat oleh syarikat-syarikat kecil. Keadaan ini menjadi lebih relevan dalam pengurusan perniagaan masa kini yang bersifat tidak menentu. Perniagaan termasuk syarikat-syarikat kecil terpaksa bersaing bukan sahaja di dalam negeri malah juga secara global di mana teknologi maklumat dilihat sebagai pemangkin yang dapat membantu syarikat-syarikat kecil ini meningkatkan daya persaingan masing-masing. Artikel ini juga merumuskan sorotan karya beberapa artikel yang berkaitan dengan faedah dan risiko pembangunan aplikasi oleh pengguna akhir. Hasil kajian berasaskan kajian kes juga dibincangkan dengan memberi tumpuan terhadap syarikat-syarikat kecil yang tidak mempunyai fungsi teknologi maklumat secara formal. Pemerhatian juga dilakukan terhadap pengguna akhir yang terlibat dalam pembangunan aplikasi komputer untuk syarikat. Risiko pembangunan aplikasi oleh pengguna akhir ini dikaji mengikut kategori risiko pada tahap organisasi dan individu berasaskan peringkat-peringkat pembangunan aplikasi yang bermula daripada peringkat perancangan, analisis, reka bentuk, dan di akhiri dengan peringkat perlaksanaan.

INTRODUCTION

The benefits and risks of user-developed applications have been well documented (Amoroso & Cheney, 1992; Guimaraes & Igbaria, 1996; Alavi & Weiss, 1986; Frank, 1988; Cale, 1994; Edberg & Bowman, 1996; Floyd, Walls & Marr, 1995; Panko, 1998). Many of these studies contended that the risks of user-developed applications could be disastrous. However, the fact that user-developed applications have been pervasive and continues to grow in small businesses tells much about the benefits of user-developed applications rather than the risks. This phenomenon could be attributed to several reasons. Firstly, the risks could be unrealistic and over-zealously reported involving isolated cases. Secondly, firms have taken the necessary precautionary steps to avoid the risks in light of these studies. Thirdly, these risks could act as a 'time-bomb' waiting to explode and even if they do explode they may go unnoticed in firms that are ignorant of IT. Lastly, but not least, firms may realise that the many benefits of user-developed applications may outweigh the risks and therefore they are willing to take the chance. This is particularly true given that the hiring of IT professionals could be difficult for small firms due to financial reasons as well as limited supply of IT human resource. Zulkhairi, Juhary and Mohd-Zukime (2004) found that there is a low level of quality IT usage among the SMI's in the industrial regions in Malaysia and there is a low level of IT skills and knowledge among the SMI workers (Hussein & Zulkhairi, 2004). No matter what the reasons are, recognising the risks of user-developed applications are necessary as today's small

businesses readily accept the challenges of globalisation by making use of IT to increase their business competitiveness. This is evident as recent findings have shown that small firms are now going international with the help of IT, especially the Internet, to enable e-commerce (Westhead, Wright & Ucbasaran, 2002).

BENEFITS OF USER-DEVELOPED APPLICATIONS

As reported by Amoroso and Cheney (1992), the benefits of user-developed applications are not only specific to the end-users. The benefits also go to the management and the Information Systems (IS) staff as well as to the IS department. Many of these benefits are similar and cited in other studies as well (Guimaraes & Igbaria, 1996; Alavi & Weiss, 1986). Eventhough these studies were based on user-developed applications that appear in large organisations, lessons could be learnt for small businesses as well. Whilst benefits to the IS organisation, such as minimised application development backlog, decrease in IS spending, improved programmer productivity, and improved IS and end-user relations (Amoroso & Cheney, 1992; Guimaraes & Igbaria, 1996) are typical to the traditional large firm setup, other benefits may also be applicable to small businesses. These include better decision-making, improved end-user computer literacy, more satisfied end-users, faster response to information requests, direct end-user control of data and information, and increased in end-user productivity. Additional benefits include overcoming the shortage of IT professionals (Alavi & Weiss, 1986), and encouraging innovation with less IS bureaucracy.

Within the specificity of small businesses, Raymond (1987) described several benefits of user-developed applications which, in addition to some of the above, include lower IT investment, less dependent on vendors for application development, lessen the need for hiring IT personnel, and as an alternative to buying commercial packages. Furthermore, as the development efforts do not require additional financial and human resources and the fact that today's software is becoming more user-friendly, user-developed applications seem to be more appropriate for small businesses (Raymond & Bergeron, 1992). In addition, due to the assimilation of IT in most tertiary and training institutions, firms are more likely to hire graduates with higher computer literacy than they were in the past, making user-developed applications in small businesses more practical.

User-developed application can be seen to be a "natural" solution to the limitations faced by small businesses with respect to the effective

utilisation of their IT resources. It is the nature of small businesses to be “poorly” resourced, and since user-developed applications do not require the presence of internal IT expertise, users with the necessary development skills would be capable of developing applications appropriate to their needs with no support from the IT professionals. Furthermore due to the specificity of small businesses with lack of IT expertise, the gulf between IT specialists and user-developers that may be present in large organisations is no longer an issue in small businesses. With end-user developers, there is no need to hire dedicated programmers and analysts to develop applications and make effective use of IT resources. As user-developed applications evolved within the firm, users become less dependent on third parties and they will be in a better position to evaluate alternatives to IT adoption such as outsourcing, acquiring ready-made packages, or developing their own application. User-developed applications could give added flexibility to small firms to adopt IT as it allows more choices for IT adoption resulting in a more flexible approach to the effective use of IT.

RISKS OF USER-DEVELOPED APPLICATIONS

Alavi and Weiss (1986) made a comprehensive and widely cited study of user-developed application risks. The authors identified potential organisational risks associated with user-developed applications in different stages of the end-user development life cycle and suggested suitable control mechanisms to manage these risks. In the analysis stage, the risks identified include no proper analysis on acquisition of end-user tools and lack of analysis on application requirement. This would result in ineffective investment in end-user tools, incompatible software, risks of security and integrity of data, not enough time spent on problem diagnosis and identifying systems requirements, and solving the wrong problem. This finding is also supported by Brancheau and Brown (1993), whom found user-developed applications tend to promote incompatible end-user applications, threats to data security, integrity and privacy due to lack of development methods and knowledge of data management, and ineffective use of financial resources when investing in end-user tools, hardware and software. Janvrin and Morrison (1996) agreed that user-developed applications will encourage applications to be developed without regards to any requirements analysis and design, and does not follow any structured methods resulting in more errors and less user confidence. The authors also found that end-user developers tend to spend very little time in planning, and problem definitions and diagnosis.

Amoroso and Cheney (1992) found “solving the wrong problem” to be one of the risks associated with user-developed applications. This is supported by Panko (1998), whom suggested that omission errors caused by misdiagnosis of problems have the highest rate of being undetected and “is the most dangerous” of errors. The author further suggested that only few organisations have policies on end-user development that caused end-user (spreadsheet) developers to disregard the use of rigorous development disciplines. This lack of development methods and techniques seems to pose consistent risks in user-developed applications and were also echoed in other studies (Amoroso & Cheney, 1992; Edberg & Bowman, 1996; Floyd *et al.*, 1995; Guimaraes & Igbaria, 1996).

In the design stage, among the organisational risks identified by Alavi and Weiss (1986) were no documentation, lack of user tests, lack of quality in applications, development time not spent wisely, and redundant efforts on similar development tasks. Other studies also found unreliable systems due to lack of quality assurance (QA) procedures (Brancheau & Brown, 1993; Floyd *et al.*, 1995), no reviews and validation on developed applications (Janvrin & Morrison, 1996)), and lack of documentation and testing (Cale, 1994; Edberg & Bowman, 1996). Another risk in the design stage may include frequent changes in the application design due to lack of development standards and procedures (Floyd *et al.*, 1995).

Alavi and Weiss (1986) categorised risks identified in the implementation stage in terms of the operational and maintenance of end-user applications. In the operation of end-user applications, among the risks include threats to the security and integrity of data, and putting additional burden on the corporate computing resources. Frank (1988) found that the user department lacked control of file backup, off-site storage and physical access to sensitive files with lack of control against fire hazards to PC files as the most serious risk. Other related findings include tendency of application development to create pockets of isolated personalised information systems across departments as opposed to an organisational information systems (Brancheau & Brown, 1993). In addition, end-user developed applications tend to be in conflict with the overall organisational objectives, and tend to withhold valuable information from the knowledge of the organisation and at the same time, retain data which is of little value (Floyd *et al.*, 1995).

For the maintenance of end-user applications, failure to document and test changes, and failure to upgrade applications are the two main risks identified in the Alavi and Weiss study. In addition, organisations could

also incur loss of investment in “personal” applications when the individual end-user that develops the applications leaves the organisation (Floyd *et al.*, 1995). Other implementation risks identified include poorly maintained applications, corruptible corporate data, strained IS-user relationships, information overloading, and lack of integration between end-user and corporate applications (Guimaraes & Igbaria, 1996).

In studying individual end-user developer risks, Panko and Sprague (1998) found that all studies of spreadsheet that they encountered, many are not error-free with the number of errors not acceptable in practice. The authors report between 20% to 40% of all spreadsheet models contain errors and found omission errors to be the most common with more than half of all errors. Logic errors were also numerous but mechanical errors were significantly less. The authors suggested only systematic code inspections in groups/teams are able to reduce development errors and minimise individual risks to an acceptable level. However, this study has only examined spreadsheet application development. To be more effective, studies should also examine risks in other end-user developed applications using non-spreadsheet development tools such as word processing, databases, application generators, web development and other fourth-generation languages.

Though none of these studies examined the risks of user-developed applications within the specificity of small business, it can still be assumed that most of the risks mentioned are also applicable to the small business environment. Nevertheless it would be in the interest of IS management in small businesses to undertake separate studies on end-user development risks and outcomes within the specificity of the small business. This paper attempts to present the outcome of a case study of two small firms involved in such user-developed applications.

THE CASE STUDY APPROACH

Information systems has been recognised as multi-disciplinary (Brancheau & Brown, 1993) and the multi-faceted nature of IS research has made case study research an appropriate research strategy (Cavaye, 1996; Remenyi & Williams, 1996). Furthermore, studying contemporary events in their natural settings is more appropriate in a qualitative research method, where case study research has emerged as a suitable candidate (Yin, 1994; Mason, 1996). Since IS is multi-faceted, therefore it is appropriate to employ the case study approach as a research strategy for this study.

Two firms were used as respondents from a list of a prior survey (Dahalin & Golder, 1998) consisting of respondents representing different levels of sophistication in terms of IT adoption. Both firms were chosen with no formal IT function and no IT professionals. Both were categorised as small manufacturing firms involved in the steel-making industry, where one firm is involved in making steel tubes for the heat transfer industry and the other is in the business of making cold drawn seamless steel tube products. Data collection and analysis were done on these firms and findings from the cases are presented in the next section.

The primary data gathering technique used was through semi-structured interviews whilst other sources of evidence used include company documents and observations. A total of 15 interviews were carried out in both firms involving the end-user developers, the firms' senior management team and other end-users at the supervisory level. Each interview lasted between one to three hours for a duration of three months. A further three months were spent for data transcription and reviews of the interview scripts, and another two months for analysis and preparation of reports.

ANALYSIS OF FINDINGS

A pattern-matching strategy proposed by Yin (1994) was used to compare the data as an empirically based pattern of evidence. This is to ensure the findings are based on strong internal validity of the cases. A cross-case analysis is presented to compare data from both firms where patterns could be established. This would enable similar or contrasting results be obtained, consistent with Yin's suggestion on literal replication and theoretical replication (Yin, 1994). A cross-case summary of the firms' characteristics is presented in Table 1 to reiterate some of the main findings already discussed previously. Obvious similarities and differences can be seen in the three categories of the firms' basic profile, IT adoption and end-user developed applications.

Some of the data presented above provide support for previous studies on IT adoption in small business, whilst other evidence may suggest the contrary. The capability of small firms to develop their own applications confirms earlier studies (Raymond, 1985; Lees, 1987). The lack of IT support is typical of many small businesses (Doukidis, Yap & Raman 1996; Raymond, 1990). A strong CEO support implies an increase in IT adoption (Thong *et al.*, 1996). This is supported by the pattern of evidence on the number of PCs and the number of users as well as the number of systems and IT products in one firm in relation

to the other. In addition, data on end-user developed applications showed that in the absence of experience in application development, training can be an alternative to encourage end-users to develop applications as suggested by Raymond (1988). Conversely, this also implies that in the absence of training in application development, on-the-job experience can also become an alternative to encourage end-user application development as demonstrated by Firm A.

Table 1
Summary of Firms' Characteristics

Characteristics	Firm A	Firm B
1. Basic Profile <ul style="list-style-type: none"> • Ownership • Industry • Age • Size (staff) • Sales (million) 	<ul style="list-style-type: none"> • Family • Manufacturing • 10 years • 45 • 6.0 	<ul style="list-style-type: none"> • Family • Manufacturing • 70 years • 49 • 4.5
2. IT Adoption <ul style="list-style-type: none"> • Years using IT • No. of PCs • No. of Users • IT Support • No. of systems/IT products • No. of applications • CEO Support 	<ul style="list-style-type: none"> • 3 • 18 • 18 • Nil • 10 • 13 • Strong 	<ul style="list-style-type: none"> • 15 • 6 • 9 • Nil • 3 • 13 • Weak
3. End-User Development <ul style="list-style-type: none"> • End-User Typology • Applications developed • Source of end-user support • Experience in development • Planning for development • Development Policy • Development methods • Development responsibility • Documentation • User tests • Security/backup procedure 	<ul style="list-style-type: none"> • 1 – End-user developer • 3 – Command level EU • 12 • Self-supported • 4 years • Not documented • Informal • Not standardized • Part time • Minimal (scratch) • Nil • Yes, informal 	<ul style="list-style-type: none"> • 1 – End-user developer • 1 – Command level EU • 9 • vendor/training • 7 years • Nil • Nil • Not standardized • Part time • Nil • Nil • Nil

Contrasting results can also be seen based on the data presented. There is no evidence to suggest that the firms are experiencing difficulty on developing applications due to having users with low level of computer literacy, as suggested by Montazemi (1988). However, difficulties have

been experienced in other areas such as time constraint in the development, limited human resources, and firm's profitability in relation to application development. Both firms have never suggested the lack of technical knowledge as a form of difficulty or stumbling block for application development.

Among the organisational risks associated with user-developed applications identified in the planning stage of the case study firms include lack of formal planning in application development and lack of strategy for implementation as presented in Table 1. Though there was evidence of an informal policy on user-developed application in one of the firms, the decision to develop an application is usually short term and depends on the availability of the end-user developer. In the other firm, applications were developed as and when the developer moved to different departments. This resulted in contention among the departments trying to win over the only expert available. Individual risks appear to be the lack of planning knowledge and lack of formal management training.

In the analysis stage, the organisational risks identified include lack of analysis on user requirements, lack of development methods and techniques, and threats to security and integrity of data. However, there is evidence of proper analysis on acquisition of development tools and ensuring compatibility of all hardware and software used. In user requirements, there was evidence that suggested that application development was done without consulting other users as the user-developer thought it unnecessary to seek input due to the developer's own knowledge and experience in the business itself far surpassing the other users. However, this resulted in less user satisfaction with the applications. Lack of formal training in development methodology has also contributed to this, as well as the problem of discovering errors too late in the development stage. In individual risks, as well as the lack of formal training by the end-user developer in one of the case study firms, substantial amount of time was spent on performing diagnosis and trial and error during application development. The part-time nature of the development work has also made the whole development task more challenging and difficult.

A consistent risk found in the design stage is the lack of documentation on the applications developed. A common excuse was the lack of time as the developer has to either catch up with the backlog in his formal work, or had to move on to developing other urgent applications. Another organisational risk in the design stage is the development of similar applications used by different departments within the same firm. In the case study an example is the existence of three separate

stock control applications, two keeping track of the same parts used by the Sales and Production departments, and one for material control application used by the Production department. There was evidence of user tests being done before implementation but that was used more as a platform for user training than for validation and QA procedures. One individual risk identified in the design stage is the frequent changes in the design due to the lack of standard design method. This is exemplified in one of the case firms where the Order Processing system went through three major revisions in three years. The third revision was an oversight on the cascading problem in the design of the database due to failure to perform appropriate relational normalisation. This could be seen as an “omission error” as suggested by Panko and Sprague (1998), in the earlier discussion.

In the implementation stage, the organisational risks identified include, lack of control over security of the development tools and backup of files. Though these are also risks associated with general IS issues, they are also user development risks as it involves operations of user-developed applications. One of the case study firms relates an incident where its PCs located in one of the departments were stolen and because there were no backups, applications used by the department had to be developed again from scratch. The situation got even worse when users from another department were affected when their PCs were needed to re-develop the stolen application. There was also individual risk in the implementation stage where an application developed by an end-user was not formally recognised but continued to be used by several other end-users in the department. Ignoring such talent would not benefit the firm and there could even be potential loss of investment in such “personal” applications when the end-user that develops the application leaves the organisation.

CONCLUSION

Whilst there are many benefits associated with user-developed applications, understanding the risks is a first step towards developing a successful user-developed application. This paper highlights the many possible risks a small firm may encounter in its endeavour to develop its own application, with the aim to avoid such negative occurrence so as the development risks could be minimised. Only through carefully managing these risks can small firms reap the full benefits of user-developed applications and be in a position to utilise IT for their competitive advantage. From the case study investigation, it seems reasonable to suggest that end-user application development in small firms should be properly planned and not left to the individual

initiative. Top management should be aware of their internal talents and be more receptive and supportive of the development efforts. Instead of putting constraints on user-developed applications, the management should allow this trend to pervade across the firm and provide the necessary tools, training, policy and incentives that would encourage this innovative and development culture.

REFERENCES

- Alavi, M. & Weiss, I. R. (1986). Managing the risks associated with end-user computing. *Journal of Management Information Systems*, 2(3), 5-20.
- Amoroso, D. L. & Cheney, P. H. (1992). Quality end-user developed applications: Some essential ingredients. *DATA BASE*, 23(1), 1-11.
- Brancheau, J. C. & Brown, C. V. (1993). The management of end-user computing: Status and directions. *ACM Computing Survey*, 25(4), 437-482.
- Cale, E. G., Jr. (1994). Quality issues for end-user developed software. *Journal of Systems Management*, 45(1), 36-39.
- Cavaye, A.L.M. (1996). Case study research: A multi-faceted research approach for IS. *Information Systems Journal*, 6, 227-242.
- Dahalin, Z. & Golder, P. A. (1998). *Information Technology adoption and end-user computing survey: Preliminary results*. Division of Electronic Engineering & Computer Science, Aston University: United Kingdom. February 1998.
- Doukidis, G. I., Lybareas, P., & Galliers, R. D. (1996). Information Systems Planning in small business: A stages of growth analysis. *Journal of Systems Software*, 33(2), 189-201.
- Edberg, D. T. & Bowman, B. J. (1996). User-developed applications: An empirical study of application quality and developer productivity. *Journal of MIS*, 13(1), 167-185.
- Floyd, B., Walls, J., & Marr, K. (1995). Managing spreadsheet model development. *Journal of Systems Management*, 46(5/6), 38-49.
- Frank, J. (1988). Quality control of personal computing. *Journal of Systems Management*, 39(12), 32-39.
- Guimaraes, T. & Igbaria, M. (1996). Exploring the relationship between EUC problems and success. *Information Resource Management Journal*, 9(2), 5-15.
- Hussein, S. M. & Zulkhairi, M. D. (2004). *Firm competitiveness and the challenges of globalization: Selected case studies from the IT industry in Malaysia*. Presented at the 8th International Conference on Global Business and Economics Development, 7-10 Jan. Guadalajara, Mexico.

- Janvrin, D. & Morrison, J. (1996). *Factors influencing risks and outcomes in end-user development*. In Proceedings of the 29th Annual Hawaii International Conference on System Sciences, 3-6 Jan., 346-355.
- Lees, J. D. (1987). Successful development of small business information systems. *Journal of Systems Management*, 38(9), 32-39.
- Mason, J. (1996). *Qualitative Research*. London: SAGE Publications, Inc.
- Montazemi, A. R. (1988). Factors affecting information satisfaction in the context of the small business environment. *MIS Quarterly*, 12(2), 239-256.
- Panko, R. R. (1998). What we know about spreadsheet errors. *Journal of End-User Computing: Special Issues on Scaling Up End-User Development*, 10(2), Spring, 15-21.
- Panko, R. R. & Sprague, R. H. (1998). Hitting the wall: errors in developing and code inspecting a 'simple' spreadsheet model. *Decision Support Systems*, 22(4), 337-353.
- Raymond, L. (1985). Organizational characteristics and MIS success in the context of small business. *MIS Quarterly*, 9(1), 37-52.
- Raymond, L. (1987). The presence of end-user computing in a small business: An exploratory investigation of its distinguishing organizational and information systems context. *Information & Operations Research*, 25(3), 198-213.
- Raymond, L. (1988). The impact of computer training on the attitudes and usage behavior of small business managers. *Journal of Small Business Management*, 26(3), 8-13.
- Raymond, L. (1990). End-user computing in the small business context: Foundations and directions for research. *Data Base*, 20(4), 20-26.
- Raymond, L. & Bergeron, F. (1992). Personal DSS success in small enterprises. *Information & Management*, 22(5), 301-308.
- Remenyi, D. & Williams, B. (1996). The nature of research: Qualitative or quantitative, narrative or paradigmatic?. *Information Systems Journal*, 6, 131-146.
- Thong, J. Y. L., Yap, C. S., & Raman, K. S. (1996). Top management support, external expertise and information systems implementation in small business. *Information Systems Research*, 7(2), 248-267.
- Westhead, P., Wright, M., & Ucbasaran, D. (2002). International market selection strategies selected by 'micro' and 'small' firms. *OMEGA The International Journal of Management Science*, 30(1), 51-68.
- Yin, R. K. (1994). *Case study research: Design and methods*, (2nd ed.) London: SAGE Publications, Inc.
- Zulhairi, M. D., Juhary, A., & Mohd-Zukime, M. J. (2004). *An assessment of k-workers among SMI's in Malaysia*. In the proceedings of the Knowledge Management International Conference & Exhibition (KMICE) 2004, 14-15 Feb. Penang.