THE EASE OF ACCEPTABILITY OF RAD (RAPID APPLICATION DEVELOPMENT) BY A SYSTEM DEVELOPER

Leong Siok Jen

School of Computing and Information Technology, INTI College Subang Jaya, Selangor, Malaysia
(sjleong@inti.edu.my)

ABSTRACT

In order to remain competitive nowadays and to satisfy the ever-changing business requirements, it is crucial that systems are developed faster, at a higher quality and relatively lower cost. However, with the traditional Waterfall approach, application tends to be developed slowly. Therefore, Rapid Application Development (RAD) has been hailed as a better alternative than the Waterfall approach. Despite all these claims, there is little study done on whether this concept is easily acceptable by the system developer. This study is important because, even though its concept may be worthwhile, the success of using RAD depends on the organisations' system developers acceptability towards it. It is hoped that through this study, it would at least create some awareness for management intending to introduce RAD in the organisation.

INTRODUCTION TO INFORMATION SYSTEM DEVELOPMENT

The history of system development has been bogged down with many problems. A survey conducted by Price Waterhouse shows that delivering system on time and budget has been a perennial problem for IT executives (Martin, 1995). Issues of cost containment, integrating IT with business and meeting IS project deadlines have also never been out of the top three for the past fifteen years. Not only that, application backlogs have remained the most pressing problem since the earliest days of IT. Typically, in order to reduce application backlogs, management often employs more people on the project. However, there is a limit as to the number of programmers that could be assigned to one project. This is because as the number of developers is increased in one project, there will be a higher degree of management and communication difficulties.

Numerous reasons were put forward as to why application backlogs exist. Among them are, for example, system maintenance, the ever-increasing demand from users, inefficient use of tools and the ever changing business requirement. In view of the numerous problems involved in information system development, the structured system development approach was advocated to mitigate those problems. It is more commonly known as the Waterfall Model, because of the implication of information flowing downwards from one stage to the next. Basically it involves the stages of requirement gathering, system analysis, system design, coding and debugging, testing, maintenance and evaluation. It is still one of the more popular and widely-used frameworks for the system development process.

However, the Waterfall approach is not without its limitation, in which it sometimes tends to be inflexible to changes, causes projects to overdue, discourages active user involvement and promotes bureaucracy in the development process. In the light of this, the Rapid Application Development (RAD) approach was introduced by Martin to minimise the problems with the Waterfall approach.

RAPID APPLICATION DEVELOPMENT (RAD)

Numerous authors have provided their definition of RAD. Therefore, it would be difficult to produce one universally acceptable definition for RAD. Nevertheless, as the name implies the purpose of using Rapid Application Development (RAD) is to get systems developed more quickly, at a lower cost and at a higher quality than the traditional Waterfall approach. Furthermore, this approach embraces change as it recognises that business requirements usually change during the long development process.

Generally RAD has the following main components: -

a) complete and early user involvement in planning and design using Joint Application Development (JAD) and Joint Requirement Planning (JRP).This planning phase is characterised by structured meetings and intensive workshops where full participation by users, technical and business specialists is essential.

b) the use of prototypes to show users as early as possible skeleton systems that will eventually develop into the complete system. A prototype is normally built using specialist tools and techniques to speed up the process of requirement analysis and design of human computer interfaces. Examples of popular PC-based tools and techniques used for prototyping include Microsoft Access (for database development); Visual Basic (for various application integration and development); and Delphi (for database application integration).

c) the use of iterative development whereby a predetermined number of iterations with prototype is tested and amended by end-users.

d) timeboxing whereby clear time limits are set for each part of the RAD development. It involves finding out what can be produced within a timescale of around 30-120 days.

e) uses a small team of 6-7 people whereby the team members are given empowerment. The members of these teams are generalists instead of specialists as in the Waterfall model.

f) use of automated tools such as Integrated Computer Aided Software Engineering tools (I-CASE tools). It is a tool that automates the entire system development lifecycle, which covers activities such as system planning, analysis, design and coding.

g) Use of object oriented techniques to provide access to pre-designed and pre-built software applications (or objects). This technique is popular nowadays as the growth of graphical interfaces, internet browser technology and the increasing use of client-server technology have stimulated an interest in object oriented system development (OOSD). The advantage of object-oriented system design lies in the reusability of computer code, which lead to faster development of a system. The most popular object-oriented programming languages used nowadays are C++ and Java

RAD is basically implemented in 3 stages