1. Introduction

There are a number of approaches that traditionally have been used to analyse and design information systems (IS) projects. Boehm (1988) reviews some of the older approaches, such as the 'code-and-fix' model, where a code is written and problems are fixed. This approach was problematic when it came to implementation, since fixing was often expensive.

In this topic, we will look at the different types of systems analysis and design methods, including the Waterfall model prototyping and rapid application development (RAD).

Objectives: Systems Analysis and Design Methods

Upon completion of this topic, you should be able to

- describe the different system development methodologies and their suitability in different project settings
- use Rapid Application Development (RAD) to quickly develop systems in an iterative manner

2. The Waterfall Model

The Waterfall model recognises feedback loops to minimise rework between stages of software development. It also incorporates prototyping as a means to more thoroughly understand new applications. The Waterfall model is so called because each step follows its predecessor in sequence.

View the presentation to learn more about the Waterfall model.

What is it?

The Waterfall model consists of the following stages, each of which can revert to the prior stage if attempts at validation uncover problems. The stages listed are for a software life cycle product. Variations in the stage labels are used for different types of projects such as acquisition of software, implementation of a vendor system or other kinds of projects.
What are the advantages?

The Waterfall model has the advantages of encouraging planning before design, and decomposes system development into sub-goals with milestones corresponding to completion of intermediate products. This allows project managers to more accurately track project progress, and provides project structure.

What are the disadvantages?

In the original Waterfall model, problems accumulated over stages and were not noticed until project completion, resulting in very expensive code. User needs were often not met, resulting in rejection of products after they were built. Therefore, feedback loops were added, along with prototyping to catch problems early. The Waterfall model does not allow rapid response to the pervasiveness of change in IS projects. The orderly sequence of activities in the Waterfall model does not accommodate new developments. Some systems, especially those involving higher levels of uncertainty and with less investment at stake, are often designed and built using more flexible development methods, such as rapid prototyping, object-oriented process or rapid application development.

3. Prototyping

Click on the link below to learn more about the Prototyping approach.

Prototyping approach

Alternate Text

Prototyping approach

When dealing with systems, which involve beneficial features that are both difficult to predict and price, the systems development approach has proven ineffective. The hard, clear dollar benefits are rarely sufficient to justify adopting the system. Users very often do not know what benefits or features a system will provide until they see it in operation. The Prototyping approach involves building a small-scale, mock-up system that allows the user to try it out. The user could then ask for modifications based upon a better idea of what the system could do. Prototyping is a much less thoroughly planned approach, but is often appropriate for applications with low investment and low structure. This can result in much lower development cost and time, especially when there are many uncertainties about what the system should consist of. Mountain Fuel Supply Inc. is a national supplier of fuel. They have decided to automate the process to keep track of their delivery and sales. The system would also help them keep track of the containers shipped, containers available for refill, damaged in transit or scrapped, and new containers that needed to be purchased. Their delivery system is region-wide and is divided into seven regions, coded as A to G. Before they can automate their entire system, they decide to first automate the delivery process in region C.

Which of the following four options do you think would be the benefits of automating the process in region C and then implementing it in all the regions?

- It results in lower risk to the project overall since there are uncertainties
about the features of the system
  - It will lead to a better understanding of users' needs and requirements before the whole project for all the regions is implemented
  - It will significantly reduce the amount of testing that needs to be done
  - As a pilot system, it can be used as a basis for project estimation for the entire system
  - It will enable the entire system to be delivered quicker by 20% to 40%
  - It is effective in getting users' reactions with respect to the look-and-feel of the user interface
  - 1,2,4,6

Prototyping is very useful in leading to greater understanding in project requirements definition. It also has been found to improve design effectiveness, because users are directly integrated into the design process in a manner that they clearly understand. However, some problems have been encountered in its implementation. Prototyping is not appropriate for all types of system development. But for smaller scale systems, prototyping is a very effective means of demonstrating what proposed systems would be able to do, and is particularly effective in getting users' reactions with respect to the look-and-feel of the user interface.

Davis (1992) describes two distinctly different applications of prototyping - throwaway and evolutionary. The labels are accurately descriptive.

Click each tab to learn more about these two applications.

**Prototyping Approaches**

**Throwaway prototype**

This prototype system is built as quickly as possible with the intent of learning about the implications of specific requirements. Once the throwaway prototype is built and used experimentally, it is discarded. The developer, then, writes a more complete software requirements specification. Throwaway prototypes work best in isolation to verify relatively small parts of complex problems.

**Evolutionary prototype**

This prototype system is part of the development process and is built with quality in mind, to include software requirements specification, design documentation and thorough testing. Evolutionary prototype systems are used to determine requirements, which have not been thought of. When the prototype is complete, the software-requirements specification is modified to incorporate what has been learned. The system is redesigned, recoded and retested. Evolutionary prototypes work best when critical functions are well understood except for specific items.

Prototyping is the process of developing a small working model of a program component or system with the intent of seeing what it can do. Thus, it is a learning device, especially appropriate when users are not absolutely sure what they want in a system. Prototyping can, therefore, be used in conjunction with the Waterfall approach.
Prototyping is the process of developing a small working model of a program component or system with the intent of seeing what it can do. Thus, it is a learning device, especially appropriate when users are not absolutely sure what they want in a system. Prototyping can, therefore, be used in conjunction with the Waterfall approach.

4. Rapid Application Development

Prototyping is also a common feature of rapid application development (RAD). In RAD, the overall idea is to develop applications as quickly as possible. Applications are developed in an iterative manner, where parts of the applications are delivered bit by bit rather than all at once. *Time-boxing* is commonly employed, whereby the development team agrees to deliver some part of the application within a certain period of time. In RAD, there is always a live version of the application running, which gradually evolves and is enhanced through various iterations. RAD typically involves small project teams that have direct contact with end-users.

Unlike the Waterfall model, RAD is well-suited to projects where there are fast-changing business requirements. *Gantthead.com*, an online community for IT project managers, cautions against the use of RAD for all projects. The following table lists the suitability or unsuitability of RAD in projects.

<table>
<thead>
<tr>
<th>Sl.#</th>
<th>Project features</th>
<th>Suitable for RAD</th>
<th>Unsuitable for RAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Scope</td>
<td>The scope is focused and the business objectives are well defined and narrow.</td>
<td>The scope is broad and the business objectives are obscure or broad.</td>
</tr>
<tr>
<td>2</td>
<td>Data</td>
<td>The data for the project already exists (completely or in part). The project largely comprises analysis or reporting of the data.</td>
<td>The data is complex and voluminous. It must be analysed, designed and created within the scope of the project.</td>
</tr>
<tr>
<td>3</td>
<td>Decisions</td>
<td>The decisions can be made by a small number of people who are available and preferably co-located.</td>
<td>There are many people involved in the decisions on the project. The decision-makers are not available on a timely basis or they are geographically dispersed.</td>
</tr>
<tr>
<td>4</td>
<td>Team</td>
<td>The project team is small (preferably six people or less).</td>
<td>The project team is large or there are multiple teams whose work needs to be co-ordinated.</td>
</tr>
<tr>
<td>5</td>
<td>Technical</td>
<td>The technical architecture is</td>
<td>The technical</td>
</tr>
</tbody>
</table>
### Reading: Rapid Application Development

The following article aims to break the common belief that Rapid application development (RAD) approach is more suited for small, medium size applications. The article presents a RAD like approach to a large, complex project setting.


### 5. Exercise

Click the link for an exercise to identify the suitability of a project for RAD.

**Exercise: Rapid Application Development**

**Alternate Text**

**Exercise Alternate Text**

**Exercise**

Consider the descriptions of different projects. Read each project description and determine why the project may or may not be suitable for RAD.

**University portal**

Mexville University wants to develop an online portal that will enable students to check their academic performance and check class announcements. However, at the early project meetings, the main stakeholders for the project keep on injecting new ideas about what the university portal should do. The president of the university would like the university portal to go live in the next 6 months.

Which one of the following do you think is the reason to implement RAD for this project?

1. Use RAD because the key technology components are in place
Supply chain integration
Champion is a supplier of camera casing. Champion supplies Lilflux cameras, a specialist camera company. Both companies have been transferring supply orders using a well-established manual process for the last 5 years. However, they now wish to automate the transfer of supply orders.

Which one of the following do you think is the reason to implement RAD for this project?
1. Use RAD because the project team is likely to be small
2. Use RAD because the data requirements are fixed
3. Do not use RAD

The correct answer is
• option 2, Use RAD because the data requirements are fixed

Sales management system
Electran is a retailer of digital lifestyle products such as MP3 players and digital cameras. Electran has a sales team that spans the entire Asia-Pacific region. It wishes to build a sales management system that will provide its sales staff across the region with the latest information about its products, specifications and availability. The system will also assist sales managers in tracking sales orders and collating sales figures. This is a big project involving many stakeholders from different regions.

Which one of the following do you think is the reason to implement RAD for this project?
1. Use RAD because the technical requirements for the system are reasonable
2. Do not use RAD because the project team is large and the decision-making will be made by individuals who are geographically dispersed
3. Do not use RAD because the technical architecture is unclear

The correct answer is
• option 2, Do not use RAD because the project team is large and the decision-making will be made by individuals who are geographically dispersed

E-government services
The vehicle tax agency wishes to implement an e-government system that will enable citizens to pay their vehicle taxes online. The key stakeholder is a manager within the agency who has a good understanding about the manual process but is excited about what opportunities an online system would provide. He will be the key decision-maker regarding system functionality and look and feel. He expects that as government policies change in the future, the system should be flexible enough to be able to accommodate such changes.

Which one of the following do you think is the reason to implement RAD for this project?
1. Use RAD because the project team is small and decisions are being made by a small number of people
2. Use RAD because the project involves the analysis and reporting of data that
already exists
3. Do not use RAD because the business objectives are obscure

The correct answer is
- option 1, Use RAD because the project team is small and decisions are being made by a small number of people

6. Self-Assessment

Now, try the self-assessment questions to test your understanding of the topic. Click the following link to open the Self-Assessment in a new window. Self-Assessment

Q1. Which one of the following describes the Waterfall model?
1. It has four phases – system feasibility, product design, code and testing
2. It is based on an iterative model of software development
3. It does not generate a working product until the later stages
4. It is not widely used in the IS industry

Q2. Which one of the following statements refers to a Throwaway prototype?
1. It is used to learn about users' requirements
2. It is built with high quality from the start
3. It evolves to become the final product
4. It is an accurate reflection of the actual system

Q3. Which one of the following projects is suitable for rapid application development (RAD)?
1. The project team is small and the technical architecture of the new system is clearly defined
2. The business objectives of the project are broad in nature
3. Project stakeholders are geographically dispersed
4. The team is large and needs co-ordination with multiple units

7. Summary

This topic covered the following main points:
- To develop systems efficiently, a methodology needs to be applied. Organisations use a wide variety of methodologies.
- The most fundamental model is the Waterfall model, anticipating a one-time pass through the sequential activities of system feasibility analysis, requirements planning, design, coding, integration, system implementation, followed by operation and maintenance.
- Due to the creative nature of IS projects, however, these activities are rarely sequential and revert to prior steps when necessary. As problems are solved and new problems uncovered, there will usually be revision.

References