Introduction to Health Informatics
Information Systems Development Methods

Information Systems (IS) development methods

By Robin Beaumont
e-mail: robin@organplayers.co.uk
Wednesday, 11 January 2012

Contents

1. Learning outcomes check list for the session ................................................................. 2
2. Introduction......................................................................................................................... 3
3. Traditional method.............................................................................................................. 3
   3.1 Problems with the traditional waterfall approach ............................................................. 5
4. Prototyping......................................................................................................................... 6
   4.1 Incremental (vertical) prototyping ................................................................................... 6
   4.2 Evolutionary (horizontal) prototyping .............................................................................. 7
   4.3 Problems with evolutionary prototyping ......................................................................... 9
5. Agile Model-Driven Development..................................................................................... 10
6. Outsourcing: Utopia or Dystopia .................................................................................... 11
7. Multiple Choice Questions............................................................................................... 12
8. Summary.......................................................................................................................... 13
9. References......................................................................................................................... 13
# 1. Learning outcomes check list for the session

This document aims to provide you with a number of skills (the 'be able to's' below) and relevant information (the 'know what's' below). Details are listed below. After you have completed the session you should come back to these points ticking off those with which you feel happy.

<table>
<thead>
<tr>
<th>Learning outcome</th>
<th>Tick box</th>
</tr>
</thead>
<tbody>
<tr>
<td>Be able to describe traditional waterfall development method</td>
<td>□</td>
</tr>
<tr>
<td>Explain the advantages and disadvantages of the traditional waterfall approach</td>
<td>□</td>
</tr>
<tr>
<td>Be able to describe the two main types of prototyping</td>
<td>□</td>
</tr>
<tr>
<td>Be able to describe the similarly between prototyping and the audit cycle</td>
<td>□</td>
</tr>
<tr>
<td>Explain the advantages and disadvantages of the prototyping approach</td>
<td>□</td>
</tr>
<tr>
<td>Be able to describe the more radical newer development paradigms such as XP and Agile.</td>
<td>□</td>
</tr>
<tr>
<td>Consider the advantages and disadvantages of Outsourcing for systems development and maintenance in the Healthcare sector</td>
<td>□</td>
</tr>
</tbody>
</table>
2. Introduction

This section describes several, often conflicting, approaches taken to developing (possibly computerised) information systems (IS).

3. Traditional method

The traditional method of developing computer Information Systems (ISs) is often referred to as the waterfall approach. The key stages are shown below. An analyst (=modeller) would come along and possibly convert a narrative document written by yourself into a system design document (called amongst other things a system specification or software requirements specification SRS). Basically this would involve describing how a system (software and hardware) would need to be set-up to fulfil your requirements. You may have specified your requirements in the form of a list of functions you would want the IS to perform called a Functional Specification. The functional specification may adopt a three tier level of compliance:

- The function must be met
- The function should be met. Indicates that the function must be met unless a waiver can be agreed upon
- The function is optional. Indicates that this would be preferred but is not mandatory.

She / he would make copious use of various diagrams such as flow diagrams (the precursor to dynamic modelling described elsewhere), entity diagrams (the precursor to object models) and a data dictionary. By way of a technique known as structured analysis and design. These documents would then be refined further and eventually translated into a particular programming language (COBOL, BASIC, FORTRAN etc.) to instruct the computer how to behave to imitate the specification. Once the system was up and running a Post Implementation Review (PIR) would be carried out. And that was the end of it!

It is important to note that once the analysis process was completed there would be no further contact between the system developers and the purchaser until they received the system.
The above approach has a clear number of stages each with a deliverable. The progression from one stage to another is in one direction, always forward and never backwards. Finally there is a clear completion stage in the process. This system worked fine when:

- The task was relatively simple (e.g. computerising an invoicing system)
- The analyst could understand the situation easily (i.e. did not require domain specific knowledge)
- Expectations were low
- The system did not need to continually adapt

The above description of how software may be developed is often called **a software development life cycle**. The method described above is often referred to as the **waterfall lifecycle** because of its uni-directional nature. The whole development process of analysis and design including or excluding software development, depending upon your viewpoint, is often referred to as **systems modelling**. This will be discussed in much greater detail in subsequent sections.

Just in case you are thinking that there are only a few methods of developing systems you should note that James Martin 1990 (book 3; p. 450) describes in detail eight categories of software lifecycle, including all the analysis stages where appropriate.

**Exercise 1.**
From a clinicians or end users perspective list some of the advantages and disadvantages of the above waterfall lifecycle.

**Exercise 2.**
From a managers perspective list some of the advantages and disadvantages of the above waterfall lifecycle.
3.1 Problems with the traditional waterfall approach

Problems with the above traditional method became increasingly acute as more complex software was developed for areas which had not traditionally been suitable for computer systems. Originally computer systems mimicked clerical operations such as invoicing and bookkeeping, with clearly defined procedural tasks and relatively simple data. Very different from the process of a patient moving through a hospital stay or a doctor caring for a long term patient with renal failure.

Specific problems associated with the traditional approach included:

- Rigidity
- Lack of input from various user perspectives
- Difficulty in revisiting previous stages as a result of subsequent lessons learnt
- Knowledge and control firmly in the hands of the Analysts ('experts').

The traditional waterfall approach incorporated a specific method of specifying the IS known as structured analysis and design which presented its own problems:

- Structured Analysis and Design created an unnatural division between the data and process aspects of the IS. Consequently the proposed system was often inadequately specified.
- The unreal division of describing data and process aspects was difficult for people to understand.

This approach rested on two central beliefs, firstly that it was possible to elucidate the true requirements from the stakeholders. However both the stakeholders and the analysts were equally unsuitable to specify the requirements. The stakeholders, where usually the commissioners of the project, frequently understood little of the realities of what would make a successful implementation. Similarly the analysts lacked domain expertise (i.e. in the Healthcare arena possessed no health care knowledge). The second belief is that requirements are stable and can be adequately captured within the unidirectional framework.

Along with these problematic beliefs there existed (and still exists) a overly optimistic belief that requirements could be specified using the various analysis techniques around. A famous software book *The mythical Man-Month*, by Brooks (1975) breaks the seal on the mythical waterfall approach to systems design:

"In most projects the first system built is barely usable. It may be too slow, too big, awkward to use, or all of the three. There is no alternative but to start again, smarting but smarter, and built a redesigned version in which these problems are solved....

The management question is, therefore, is not whether to build a pilot system and throw it away. You will do that. The only question is whether to plan in advance to build a throwaway, or to promise to deliver the throwaway to the customer. Seen this way, the answer is much clearer."

Quoted in Beyond Programming Bruce Blum OUP 1996 p.253

While many different solutions to these problems have been offered we will concentrate on two, Prototyping and participative design methods. Prototyping will be considered in this section in depth whereas participative design methods in less detail as they will be discussed latter.

Exercise 3.

Suggest some solutions to the problems associated with the traditional waterfall approach.
4. Prototyping

Although there are several varieties of prototyping we will only consider two:

- Incremental
- Evolutionary

4.1 Incremental (vertical) prototyping

Incremental prototyping is a method of developing a computer system by a process of delivering a small but complete part of the system each time. Imagine building a kitchen by getting a new unit each month. Each prototype therefore provides more facilities than the previous version. By way of an example consider a maternity system. The system developers may decide upon the following delivery:

May '94  Patient Registration/ discharge facilities
Sept. '94  Basic reporting facilities
March '95 Clinical module
July '95 Clinical / audit report facilities
Nov. '95  Full reporting facilities and Casemix

Incremental prototyping therefore works by delivering a completely finished small part of the system each time until a complete system eventually emerges.

Vertical (incremental) Prototyping - Nielsen 1993

 Nielsen 1993 (p94) describes this type of prototyping as vertical prototyping. You can see why by looking at the diagram below. The depth (height) represents the functionality of the IS and the width represents the breath of the system. Therefore an incremental prototype delivery represents a vertical slice.

This is in direct contrast to the process of evolutionary prototyping described below.
4.2 Evolutionary (horizontal) prototyping

In this process the system designers create right from the start a mini system of the bare essential facilities required. Taking the same example as above:

May '94 Basic Patient Registration/ discharge, clinical and reporting facilities
Sept. '94 Evaluation and review of requirements
March '95 Second attempt
July '95 Evaluation and review of requirements
Nov. '95 Formal development of final system

Evolutionary prototyping is therefore the process of developing a computer system by a process of gradual refinement. Each refinement of the system contains a system specification and software development phase. In contrast to both the traditional waterfall approach and incremental prototyping, which required everyone to get everything right the first time this approach allows participants to reflect on lessons learned from the previous cycle(s). It is usual to go through three such cycles of gradual refinement. However there is nothing stopping a process of continual evolution which is often the case in many systems.

Nielsen 1993 (p94) describes this type of prototyping as **Horizontal prototyping**. You can see why by looking at the diagram below where the depth (height) represents the functionality of the IS and the width represents the breadth of the system. Therefore an incremental prototype represents a horizontal slice.

The diagram below shows the evolutionary software development lifecycle. Those of you who are familiar with the audit cycle will note immediate similarities, as will those of you who know the nursing process.
The diagram below demonstrates the above lifecycle with three iterations. This is based upon a project I was involved in called Prodigy, which was developing the prescribing aspect of GP computer systems.

Although this approach may seem radical it must be noted that the original article was published as long ago as 1988 by Boehm who described in depth such a process alongside examples of its use. Possibly this demonstrates just how far behind software development is in the health sector!

**Exercise 4.**

Read through the above article by Boehm at:


**Exercise 5.**

What additional resources may be required for the evolutionary prototyping approach over that of the traditional waterfall approach?
4.3 Problems with evolutionary prototyping

The main problems with evolutionary prototyping are due to poor management:

- Lack of defined milestones
- Lack of achievement - always putting off what would be in the present prototype until the next one
- Lack of proper evaluation
- Lack of clarity between a prototype and an implemented system
- Lack of continued commitment from users. This process requires a greater degree of sustained commitment from users for a longer time span than traditionally required. Users must be constantly informed as to what is going on and be completely aware of the expectations of the ‘prototypes’. This is discussed in far greater depth elsewhere.

Exercise 6.
Consider how some of the above problems associated with evolutionary prototyping might be solved.

A more detailed iterative prototyping development method which takes into account the above factors can be found in chapter 12 section ‘Getting the users involved’.
5. Agile Model-Driven Development

In the 1990's a radical backlash to the large prescriptive approaches to systems development emerged such as extreme Programming (XP) and Agile software processes including feature-driven development (FDD), Agile Model-Driven Development (AMDD) and Agile Modelling (AM).

Exercise 7.

To find out about AMDD go to: http://www.agilemodeling.com/essays/amdd.htm.

What are your views about the Agile requirements change management process described above.

Agile Modelling is not a prescriptive developmental method but offers guiding principles called, core practices which at present are:

1. Active Stakeholder Participation
2. Apply the Right Artifact(s)
3. Collective Ownership
4. Create Several Models in Parallel
5. Create Simple Content
6. Depict Models Simply
7. Display Models Publicly
8. Iterate to Another Artifact
9. Model in Small Increments
10. Model With Others
11. Prove it With Code
12. Single Source Information
13. Use the Simplest Tools

Most important is the Active Stakeholder participation, which Scott W. Ambler, one of the main proponents of AM has to say:

"Active Stakeholder Participation - An expansion of XP's On-Site Customer which describes the need to have on-site access to users that have the authority and ability to provide information pertaining to the system being built and to make pertinent and timely decisions regarding the requirements, and prioritization thereof. AM expands XP's On-Site Customer practice to have project stakeholders -- including direct users, their management, senior management, operations staff, and support (help desk) staff -- actively involved in the project. This includes making timely resourcing decisions by senior management, public and private support for the project by senior management, active participation of operations and support staff in the development of requirements and models pertaining to their respective areas. You can easily promote active stakeholder participation on your projects if you adopt inclusive modeling techniques."

Taken from: http://www.agilemodeling.com/practices.htm

One may argue that organisations are themselves in a software development maturation process where they move from the Traditional Waterfall through to Prototyping and finally mature by using one of the Agile approaches.

Obviously this viewpoint is very much up for discussion!

Exercise 8.

Spend some time reading about the other core practices: http://www.agilemodeling.com/practices.htm.

Where is your organisation in the software development maturation process?

Do you see Agile techniques as being the pinnacle of software development methods?

What is preventing your organisation from using such techniques?
6. Outsourcing: Utopia or Dystopia

In the context of Healthcare information Systems this is the development or maintaining, or more rarely the evaluation of a system by the use of an external company (sometimes nowadays in another country).

Exercise 9.


Outsourcing is seen as a good management technique and encouraged. However I personally hold the opposite viewpoint for several reasons:

- Invariably Outsourcing requires very clear 'statements of requirements' yet from the above sections it is clear that this is not possible and development should be a fluid and interactive process.

- Healthcare organisations lack IT and systems development skills the aim should be to development them not encourage dependency

- Healthcare organisations are unique - but often unmodified production industry techniques/solutions are forced upon them by outsourcing companies.

- Empirically it have been shown that continual end user involvement is vital for systems success (see the section - Getting users involved in developing Information Systems http://www.robin-beaumont.co.uk/virtualclassroom/chap12/s4/des2.pdf)

Exercise 10.

I have presented a very negative picture of IT/Systems development outsourcing above, what positive points can you think of to counterbalance the argument?

Obviously the situation is not so clear cut, for example it would be ludicrous for each General Practitioner Surgery in the UK to develop their own Information system. However GP system development presents a paradox because one of the most successful GP systems (Emis) is owned and developed by GPs themselves!

One of the most important writers on outsourcing in the UK (with a global reputation) is Professor Leslie Willcocks at the London School of Economics (SLE) (http://www.lse.ac.uk/people/l.p.willcocks@lse.ac.uk/ ) His Book Intelligent Outsourcing (Butterworth 2004) is very interesting and also you can see the large number of papers he has written on outsourcing by following the links at http://personal.lse.ac.uk/willcock/ .

He also publishes a blog and if you are interested in the UK situation you should read his blog entitled, "Austerity will lead to further government outsourcing, but key lessons need to be learnt to avoid the mistakes of the past" at: http://eprints.lse.ac.uk/36278/1/blogs.lse.ac.uk-Austerity_will_lead_to_further_government_outsourcing_but_key_lessons_need_to_be Learnt_to_avoid_the_.pdf
7. Multiple Choice Questions

1. The traditional method of developing computer Information is often referred to as the . .
   a) Functional approach  
   b) Sociotechnical approach  
   c) Waterfall approach  
   d) Iterative approach  
   e) Prince approach

2. What is another name for an analyst?
   a) Domain expert  
   b) Modeller  
   c) Forecaster  
   d) Statistician  
   e) Architect

3. In a traditional Functional specification each function is classified into one of three levels of compliance, what are they (select three):
   a) Frequently met  
   b) Should be met  
   c) Can not be met  
   d) Must be met  
   e) Met at a future date  
   f) Optionally met  
   g) Only met on paper

4. Which of the following is NOT a characteristic of the Waterfall software development lifecycle:
   a) Provides a defined deliverable at each stage  
   b) Has build in feedback loops  
   c) Is suitable for relatively simple systems  
   d) Is suitable the modeller does not require much domain specific knowledge  
   e) The system is not continually evolving

5. What were the two main inadequacies with the Structured Analysis and Design Approach. What was one of them? (Select 1 option)
   1) Created problems when using it to model simple systems  
   2) Created a unreal division between modellers and programmers  
   3) Created a new programming language  
   4) Created problems with trying to understand the system  
   5) Created a unreal division between process and data

6. There are two principle types of prototyping, what are they (select two options):
   1) Paper  
   2) Incremental  
   3) Evolutionary  
   4) Simulation  
   5) Casemix  
   6) Rapid  
   7) Screen
7. Which of the following is not a recognised problem of evolutionary prototyping? (select one)
   a) Lack of proper evaluation
   b) Lack of defined milestones
   c) Lack of achievement
   d) Lack of clarity between a prototype and an implemented system
   e) Lack of initial commitment from users.

8. Within the field of HI Boehm in 1988 published a very influential paper concerning what (selection one option)
   a) Signal detection theory
   b) Software development
   c) Mythical man month
   d) Naser project management
   e) Software Case tools

9. Which of the following is a characteristic of Agile model techniques: (select one)
   a) Complex Case tool support
   b) Functional requirements
   c) Active Stakeholder Participation
   d) Controlled stakeholder involvement
   e) Clear demarcation of roles

10. Who is one of the main writers on IT outsourcing, with an international reputation on the topic (select one):
    a) Mary Kay
    b) Leslie Willcocks
    c) David Williams
    d) Phil McKay
    e) Robin Beaumont

8. Summary

This document introduced the various structured ways Information Systems (ISs) can be developed in contrast to the haphazard way that most systems seem to have evolved. The traditional waterfall approach was described and evaluated alongside the more modern alternative of prototyping, and the more radical agile techniques. An example of how evolutionary prototyping is being used in the health care arena was given. The concept of outsourcing was introduced.

9. References

The article about evolutionary prototyping:

A book describing the various approaches taken to software design over the years:
Blum Bruce 1996 Beyond programming. OUP. Oxford.


Chen, Yu-Che, Perry, James 2004 IT Outsourcing – A Primer for Healthcare Managers. BUSINESS BRIEFING : GLOBAL HEALTHCARE – ADVANCED MEDICAL TECHNOLOGIES