FACTFILE: GCE DIGITAL TECHNOLOGY

Rewarding Learning

AS1: APPROACHES TO SYSTEMS DEVELOPMENT

Alternative development approaches and Software projects

Learning Outcomes

Students should be able to:

- describe the main features of different approaches to systems development: the waterfall model, Rapid Application Development (RAD) and Agile;
- evaluate different approaches to systems development: the waterfall model, RAD and Agile;

Software projects

Students should be able to:

- describe the main elements in the organisation of a software project: resources, constraints and risks;
- explain the main features of project management for a software project;
- evaluate the use of project management tools: Gantt charts and critical path analysis;
- describe the importance of version management during systems development;

Content in Implementation Fact File

Alternative development approaches

✓ Approaches to system development
  - Waterfall model
  - Rapid Application Development (RAD)
  - Agile

✓ Evaluating approaches to systems development

Software projects

✓ Elements involved in organising software projects

✓ Main features of project management

✓ Evaluating project management tools
  - Gantt charts
  - Critical path analysis

✓ Version Management

✓ Questions

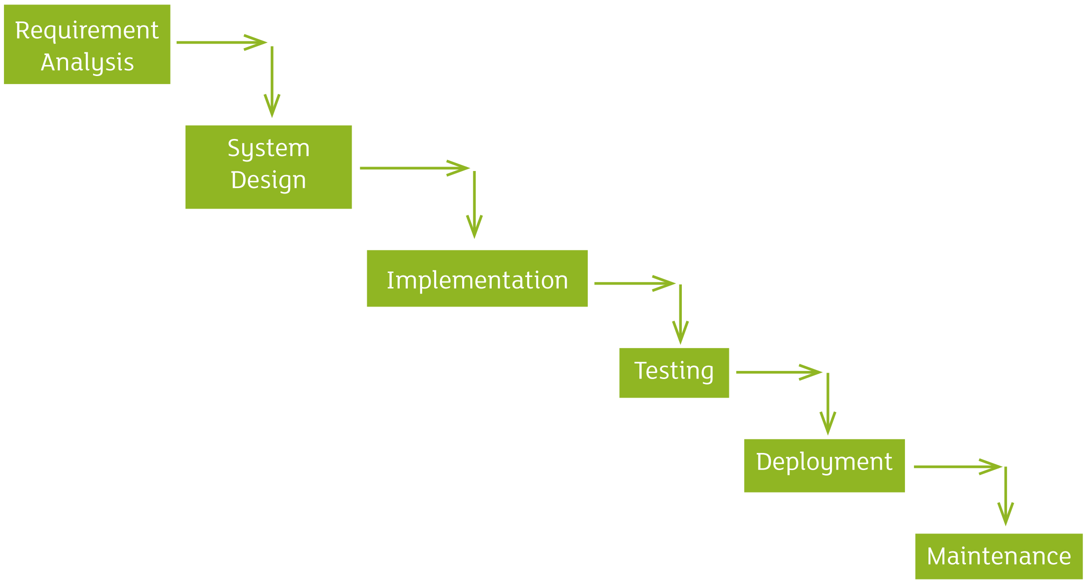
Alternative development approaches

Approaches to system development

A range of methods associated with system development have developed over the years as one methodology may not necessarily be suitable for all types of tasks. Some of these methodologies include, the waterfall model, Rapid Application Development and agile development. Each approach has its own set of strengths and weaknesses

The waterfall model
The waterfall model represents an abstract description of the systems life cycle where each stage of development flows on from the previous stage, i.e. each stage of development must be completed before the next one can begin. At the end of each stage a review takes place to ensure the project is on the correct path. Following each end of phase review a decision will be made as to whether the project should continue or be discarded. No stages of the waterfall overlap and testing can only take place when implementation is complete.

Overview of the waterfall model



Rapid Application Development (RAD)

Rapid Application Development is a type of incremental model, which means the whole project is subdivided into a series of builds, each of which undergoes its own separate lifecycle, passing through a requirements, design implementation and testing phases. Each of the components of the system are developed in parallel, produced within a given time frame and then assembled to produce a working prototype which is delivered to the user for feedback.

The stages of RAD are:

Business modelling–here a complete analysis of the business is carried out to determine the flow of information

Data modelling–where data objects are identified and defined and the relations between the data objects are established

Process modelling – process descriptions for making any kind of addition or modification to a data object or the retrieval of a data object are identified at this stage

Application generation – the actual system is developed using automated tools to generate code for the final application.

Testing and turnover – prototypes are tested at each iteration so the overall testing time is reduced in RAD however the application interface and the data flow between the components must be tested thoroughly.

Overview of RAD

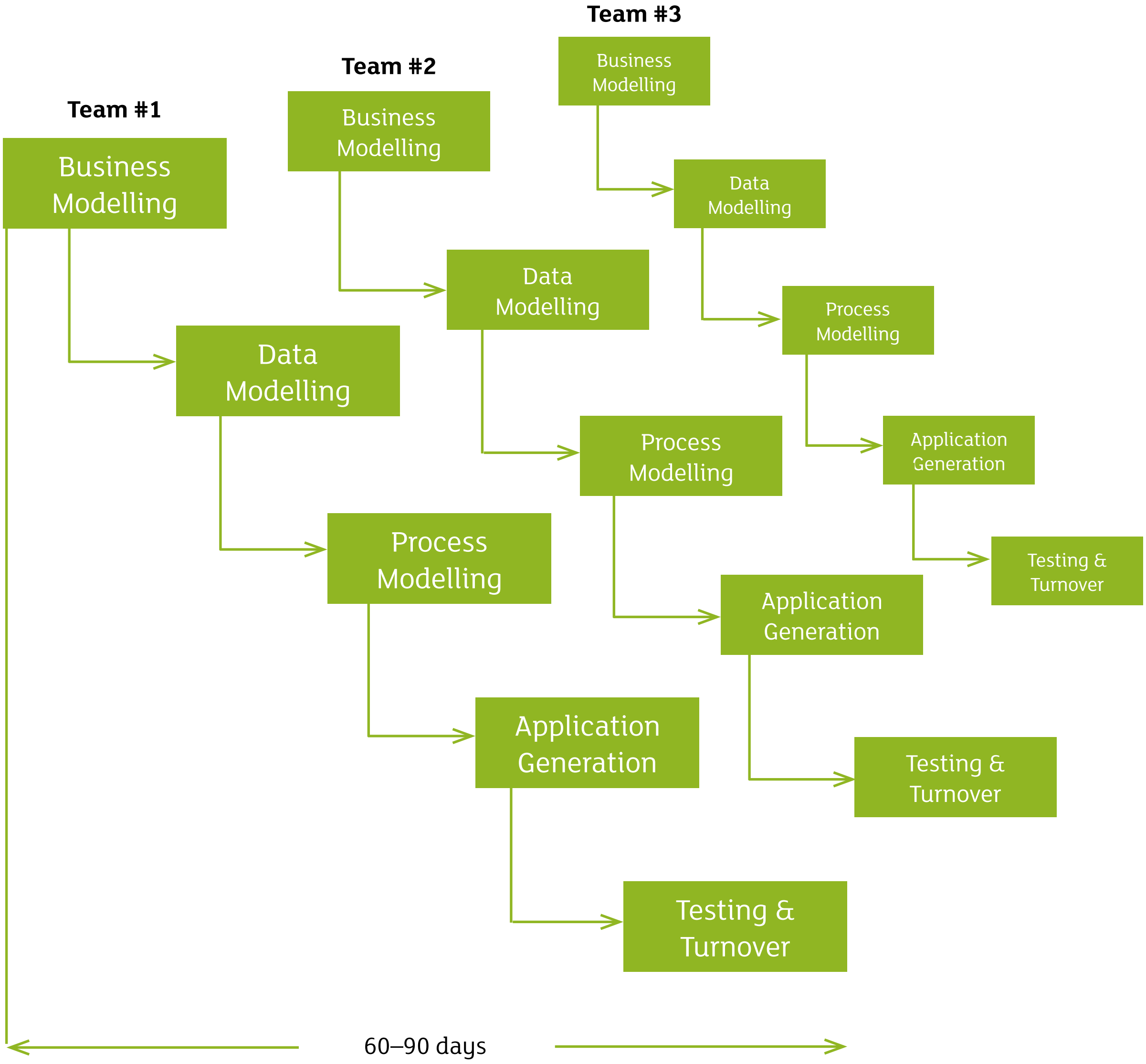


Figure 15–RAD Model

Agile

The agile development model represents a combination of the iterative and incremental models. The main aim of the agile method is the rapid delivery of a working product in response to customer needs. Agile methods divide the process into small incremental tasks which are provided in iterations which last a number of weeks. During each iteration, a series of teams will simultaneously work on planning, requirements analysis, coding, and testing (both unit testing and acceptance testing).

Evaluating approaches to system development

Approach to system development	Advantages	Disadvantages
Waterfall model	• Rigid model with specific outcomes expected at end of each phase. Makes it easy to understand and apply • Works well for projects where the user requirements have been clearly identified • No overlapping of phases so easy to implement	• Once testing begins amendments to system requirements cannot be made easily • No software produced until late in the development process so high level of risk involved
Rapid Application Development (RAD)	• Reduction in development time • Components reviewed at an early stage • Client feedback encouraged • Components can be reused	• Not suited to all systems • Expensive approach due to use of automated code generation and modelling tools
Agile	• Increased customer satisfaction due to fast delivery of software solution • System is adapted regularly to meet change in user requirements • All parties work closely on development	• Limited documentation which may mean future maintenance or review could be difficult • Project can easily lose focus if customer unclear of own requirements • Only experienced programmers suited to this type of development

Software projects

Elements involved in organising software projects

When organising any software project a number of elements need to be considered. The main elements to be considered by any project manager include:

Resources – In the development of a computer system these can be considered more specifically as capital, the development environment and personnel such as, system developers, project managers, system analysts, stakeholders and the development environment (including hardware and software). When considering the organisation of any computer project the organisation of personnel includes ensuring the availability of appropriately trained staff and the efficient deployment of appropriate staff across the project elements.

Constraints – constraints in project management refer to restrictions which are placed on the limitations of the project. These may be in terms of the scope of the project (i.e. limitations placed on the areas to be implemented as part of the project), schedule and cost. The scope of the project will identify the project goals and defines the boundaries of the project, while the schedule will identify a series of deadlines to be met across the project, including the deadline for project completion. Cost is the most obvious constraint and identifies the maximum spend available to the project.

Risks – risk management is an important part of organising a software project. Some of the main risks to be considered include:–

- User involvement – not involving the end user at the planning stage can lead to additional or changing requirements at later stage in the project. This in turn can lead to an increase in costs and time delays

- Scheduling flaws – inaccurate scheduling of tasks and inefficient deployment of staff can lead to the entire project falling behind deadline

- Inflation of requirements – ineffective consideration of user requirements from the outset can lead to more and more new features being identified as the project unfolds, this too can threaten the project timeline

- Specification breakdown – this can happen for many reasons, for example if the development team did not understand the specification or the user did not understand how their requirements were tied to the specification.

- Staff turnover – the loss of key personnel can lead to project delays

- Technology – consideration needs to be given to the familiarity of the project staff with the working platform, do they have prior experience with the hardware / software, are the tasks expected from the hardware reasonable?

Main features of project management

Software project management is the management of any project where a piece of software is being planned and implemented. Many project managers use specialised software to support them in the process of project management. Key features to consider when managing the development and delivery of a piece of software include:–

Monitoring – project monitoring is a key part of project management. It is important that all key activities are monitored so the project manager is constantly aware of progress and able to apply timely interventions in times of difficulty. Project management software can be used to aid the monitoring process. Progress reports from team managers can help provide the project manager with an overview of the project although alternative approaches including regular meetings or diaries can also be used. Based on their findings from such reports the project manager may use that information to help them keep the project on schedule.

Scheduling – late delivery of projects can be costly and as a results the project manager needs to ensure that key deadlines are met across the scope of any project. Key factors to consider when scheduling tasks across teams of staff include consideration of interdependency of tasks, task sequencing and parallelism, i.e. which tasks depend on the outcome of any other tasks, what order must the tasks be completed in and which tasks can be completed concurrently. Unrealistic deadlines (perhaps established by the client) can be destructive to project progress. Other destructive influences which may impact upon project progression may include among, other things, changing customer requirements which are not them reflected in the schedule and unforeseen human or technical difficulties that were not foreseen.

Budget – from the outset, budget is a major factor in any software project. Clients will want to know how much the solution is going to cost, and the project manager will need to ensure at all times across the budget that the project costs are not escalating out of control. Budget or available finance places a constraint on the project and will need to take into consideration factors such as man hours, software and hardware requirements, and licensing issues. Where a project runs the risk of running behind schedule, this will invariably have an impact on the cost. A project manager is expected to work within the constraints of the available budget.

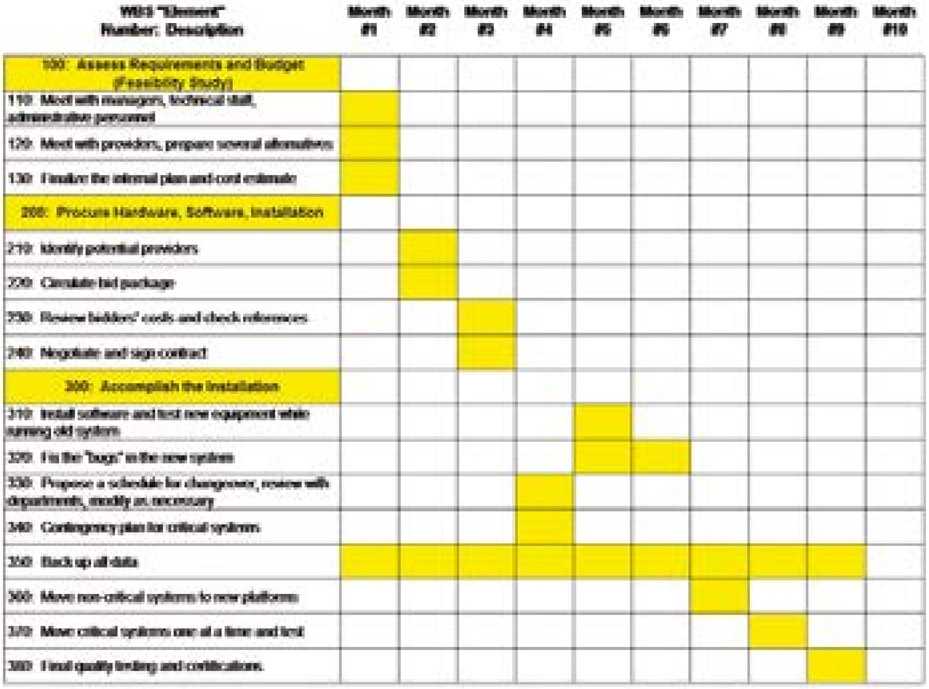
Resources – in any project resources are considered to be the people, equipment, facilities and anything else necessary for the successful completion of the project. Effective scheduling and optimum deployment of those resources is necessary if the successful completion of the project is to be guaranteed. Resources should be allocated based on the priority assigned to each of the project components.

Evaluating project management tools

A variety of tools are available to support project managers in their management of resources and other features of the project. Some of these tools include Gantt charts and Critical path analysis:-

Gantt charts – a gantt chart is a type of bar chart used to show the start and finish dates of the various elements of a project. The gantt chart shows all activities forming part of the overall project plotted against time.

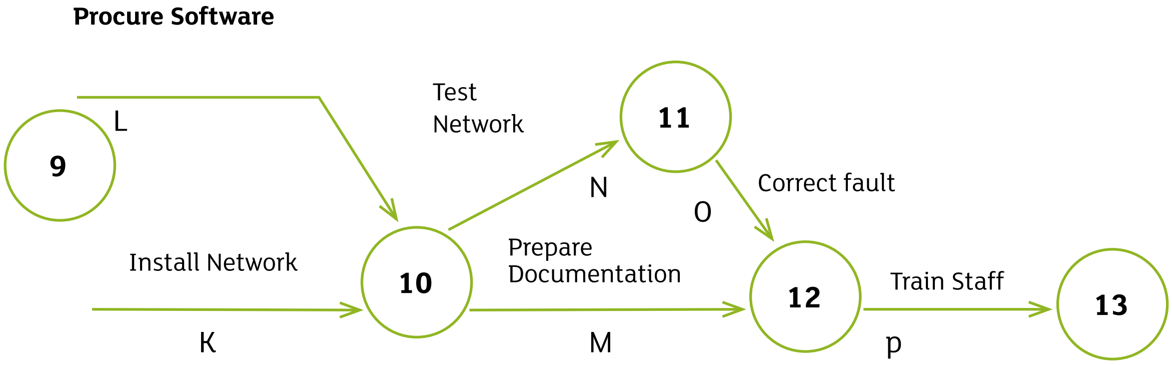
See below for an example of a gantt chart



WBS "Element" Number: Description	Month #1	Month #2	Month #3	Month #4	Month #5	Month #6	Month #7	Month #8	Month #9	Month #10
100: Assess Requirements and Budget (Feasibility Study)										
110: Meet with managers, technical staff, administrative personnel	■									
120: Meet with providers, prepare several alternatives	■									
130: Finalize the internal plan and cost estimate	■									
200: Procure Hardware, Software, Installation										
210: Identify potential providers		■								
220: Circulate bid package		■								
230: Review bidders' costs and check references			■							
240: Negotiate and sign contract			■							
300: Accomplish the Installation										
310: Install software and test new equipment while running old system					■					
320: Fix the "bugs" in the new system					■	■				
330: Propose a schedule for changeover, review with departments, modify as necessary				■						
340: Contingency plan for critical systems				■						
350: Back up all data	■	■	■	■	■	■	■	■	■	
360: Move non-critical systems to new platforms							■			
370: Move critical systems one at a time and test								■		
380: Final quality testing and certifications									■	

Critical path analysis (CPA) – this is a useful tool that allows project managers to analyse project activities so as to minimise delays and to ensure that any conflicts are delayed. Critical Path analysis helps managers calculate the minimum time needed to complete a project. CPA allows the manager to construct a model of the project where all the tasks needed to complete the project are listed, time taken for each activity will be listed and any dependencies among the activities will also be listed.

See below for an example of critical path analysis



With Activity Dependency Table

	Activity	Duration Days	Preceding Activities
K	Install network	15	D, E, I, J
L	Procure Software	28	A
M	Prepare documentation	15	K, L
N	Test network	10	K, L
O	Correct fault	6	N
P	Train users	10	M, O

(*diagram from pg 237 of BCS dictionary*)

	Benefits	Drawbacks
Gantt Chart	• Help eliminate idle time in project • Provides clarity – allows all elements of project to be displayed on a single page • Can provide motivation for team members – focus is on individual task deadlines • Aids time management	• May be misleading – the size of the bar does not reflect the amount of work associated with the task • Can become complex for larger projects with many components
Critical Path Analysis	• Dependencies displayed visually • Supports efficient allocation of resources and hence costs • Helps make impact of schedule revisions more visible	• Can be difficult to develop for larger projects with a large number of activities

Version management

Version management is the monitoring of changes to a system during the development process and during its operational life. All changes made to the source is tracked, along with who made the change, why, and any enhancements made; this supports the tracking of any changes to the software and also means all changes can be reversed if necessary. This also helps ensure that everyone involved (developers and users) are working on the same systems.

Questions

1 The waterfall model, Rapid Application Development and agile development are commonly used models for project development.

a. Identify the main features of the water fall model. [2]

b. Identify the main features of Rapid Application Development. [2]

c. Identify the main features of agile development. [2]

....

....

....

....

....

....

....

....

2 Describe the five stages of Rapid Application Development. [5]

....

....

....

....

....

....

....

3 Identify the main advantages and disadvantages of agile development methods. [4]

....

....

....

....

....

....

....

....

4 Rapid Application Development can include greater input from the end user.

a. Describe how the end user can become involved during the RAD process. [2]

b. Identify the advantages and disadvantages of a high level of user input. [4]

....

....

....

....

....

....

....

....

....

5 Risk management is a key element of software development. Identify the main risks a project manager should consider during the development process. [4]

....

....

....

....

....

....

....

....

....

6 Gantt charts and CPA can be used to support project management.

a Describe the main features of a gantt chart. [3]

b. Describe the main features of CPA. [3]

....

....

....

....

....

....

....

....

7 Gantt charts and CPA can be used to support project management.

a. Identify the advantages and disadvantages of using gantt charts. [4]

b. Identify the advantages and disadvantages of using CPA. [4]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

Bibliography

BCS Academy Glossary Working Party, 2013, *BCS Glossary of Computing and ICT*, 13th Edition, Swindon, BCS Learning and Development Ltd