Higher Still
Notes
www.hsn.uk.net

Advanced Higher Computing

HSN55100
Unit 1 Outcome 1

Contents

Systems Analysis and Design

The purpose of this phase of software development is simply to identify the existing problem – to specify the purpose of a new system. By clearly identifying the problem, the suitability of the solution will be improved.

There are three parts to the process:

Feasibility Study

This part is generally the responsibility of the project leader, who works for the consultants (larger organisations may have special departments which deal with feasibility studies, in which case the project leader would be a member of their own staff).

The purpose of the feasibility study is to determine whether or not a new system is possible and worthwhile. There are four categories of feasibility which are considered:

Technical Feasibility
The needs of the proposed system must be attainable using existing technology (unless the development of new technology is the aim).

Economic Feasibility
It is common for clients to require the system to 'pay for itself'; where savings made over a period of years as a result of the system will cover the cost of implementing it. Often, clients will see the system as so imperative that they are less concerned about the cost.

Legal Feasibility
The new system must comply with any relevant laws – for example, the implementation of a new database will have to follow the Data Protection Act in the UK.

Schedule Feasibility
If a client requires a solution to be implemented in six months' time, the feasibility of any proposal will depend on its ability to meet this demand – clearly, a system which will take a year to develop would be of no use to the client.

The product of the feasibility study is a report for the client. Based on the conclusion of the study, the client must decide whether or not to commit to further analysis. They are under no obligation to proceed, even if the report recommends doing so.

The advantage of the feasibility study is that it is cheaper and quicker than a full analysis, and will allow the client to feel they are making an informed decision about the project.

System Investigation

If the client commits to a System Investigation, the consultants will appoint a Systems Analyst, or a team of Systems Analysts, who will study the problem and determine what should be done to solve it. The skill of systems analysis is to be able to see the whole system in as much detail as possible, without being too concerned by the acute details. The System Investigation does not solve the problem; it considers what must be done to solve the problem.

There are three aspects of the System Investigation:

Analysis of the Existing System

For problems based on an existing system, the analysts will study its inputs, processes and outputs to gain a deeper understanding of the client's needs. This can be done in a number of ways:

Interviews

These should identify any problems staff have with the existing system.

Questionnaires

If there are a large number of staff, interviews would not be practical – in such instances, questionnaires will be used instead.

Observation and Task Analysis

By watching staff doing their work, an analyst can identify problems with the system which the solution should aim to remove.

Document Analysis

Forms and reports from the existing system will help to identify data flows.

Background Information

Knowledge about the field in which the client operates will improve the analyst's understanding of the system.

Specialist Knowledge

A basic understanding of the processes which the system assists helps the analyst see the requirements of the solution.

Modelling of the New System

The analyst will make sense of the information they have gathered, and form an understanding of the overall shape of the system. There are two main methods used to do this:

Data Flow Modelling

This method looks at how data flows through the system, as it is changed from its input form to the output.

Documentation would include data flow diagrams and data dictionaries.

Object Modelling

Objects are the 'things' which make up a system, and this method looks at how the objects interact – these are the system's operations.

Documentation would include an object hierarchy and a use network.

Producing the Operational Requirements Specification

This document is a legally binding contract with the client, and will often be revised several times as the client helps to refine the model of the system they want.

Operational Requirements Specification

The end product of a System Investigation is a Formal Operational Requirements Specification, which is the final version, agreed upon by both consultant and client. This document will offer mutual protection; the client is guaranteed that the system they are paying for will perform as they wish, and the consultants know exactly what problem they are being asked to solve. Any additional functionality required by the client will be counted as maintenance, and will cost them extra.

Clearly, it is important for both parties that the document is as unambiguous as possible, so much time will be spent making it clear and precise. There are four parts to the specification:

Functional Specification
The functionality required of the system.

Physical Specification
The hardware needed by the system.

Data Requirements
The storage capacity required of the new system.

System Prospectus
Including the development schedule and details of the user documentation and training which will be required.

This document marks the end of the 'investigative' phase of the project – after it has been agreed upon, the project moves to the 'development' phase, the Software Development Process. At this stage, programmers are appointed by the consultants.

The Software Development Process

Analysis of Operational Requirements

The ORS is studied to gain an understanding of the problem to be solved. In complex projects, the overall structure of the solution may be modelled to make it easier to understand.

As well as this, the development will be organised in terms of the personnel required and the schedule to follow. The overall aim of this stage is to produce a System Specification.

System Specification

A System Specification is a document, produced by the analyst(s) in charge of the development, which describes the system which will be developed. The information in this document is usually structured as follows:

Introduction

Scope and Purpose of the Document
The aim of the system specification is very simply defined.

Objectives
The primary aims of the solution are identified here.

Constraints
Any limitations or requirements which have been identified are stated here, such as the capacity required of the new system, and the budget for the development.

Hardware Specification
The minimum hardware requirements are identified, keeping within any budget constraints.

The specification of a new computer system should include details of the processor and main memory, as well as backing storage, peripherals and consumables.

Software Specification

Software Development Environment
This defines the software required for the new system. This will generally consist of:

- Operating System (may be dependant on hardware)
- Software Environment
  The application(s) which will be used to solve the problem, including commercial applications, such as Microsoft Office, and a programming environment, such as Java.
- Other Software
  This is usually communications software, such as browsers or e-mail applications.

Data Description
Here, the data which will be handled by the system is discussed, and memory requirements are identified. The state of the data throughout the system is also detailed (ie whether it is stored on disc or held in memory).

Functional Description
At this stage, a high-level description of the system's functions is produced, generally using structure charts.

Project Issues

Projected Development Costs
The total cost of the development is estimated, using the costing of the required hardware and software.

Project Schedule
The time for the software to be developed, and the hardware installed, is estimated. Time for retraining staff may also be considered.

Software Design and Validation

The system is usually designed from the top down; the high level functional description from the System Investigation is refined using structure charts and pseudocode until the design is clear enough to code. Data flow in the system is also analysed, using Data Flow Diagrams, to see how the functions integrate. With large systems in particular, the development must be well organised to make sure that all the programmers produce code which will integrate with the rest of the system, and to ensure that the schedule can be followed.

The Operational Requirements Specification is used to check that the system being developed is what was asked for (verification) and what is needed by the client (validation).

Testing is also considered at this stage. Test data are devised for each module, with a specific reason and expected result for each test case (input). Dry runs are then conducted on the pseudocode to ensure the logic performs as desired.

Implementation and Testing

At this stage, the coding takes place. There are two ways the system can be coded: depth first or top down.

Depth First
With this method, the most difficult parts of the system are coded first, as they may take most time. Also, in systems which use a database, the database is often developed first to provide the rest of the system with realistic data for testing.

Top Down
The overall structure of the system is established first, using stubs (very simple modules which imitate the expected behaviour of the full functions). Once the whole system is in place, the stubs are replaced with fully functional modules.

Top Down Testing
Testing is conducted on the individual modules as they are coded to ensure the logic is correct. Since modules require a complete program in order to run, a small program called a driver (or test-harness) is created; this runs the module and supplies it with data.

Black-Box (Functional) Testing
Test cases are generated using the functional specification to ensure the system can provide the expected output for the required range of inputs. If the system can meet these needs, it is acceptable.

Black-box testing can be carried out by a separate team, who did not create the system, and who do not need to know how it works. The specification is used to create test cases, and any unexpected results are simply returned to the programming team for a bug-fix.

White-Box (Structural) Testing
This type of testing is also known as glass-box testing, as it relies on being able to see and understand the workings of the modules. Testers must know the programming language, since they construct test cases to check every logical path (generally, a programmer will check their code this way as they work). White-box testing is often used extensively when black-box testing returns an error.

Integration and Testing of Whole System

In larger systems, the modules are combined to form sub-systems within the main system, and their operation is then tested at this level. After this, the sub-systems are integrated, and the whole system is tested. Finally, 'acceptance testing' is conducted on site, where the client's new system is installed. Test cases for all these stages are created using methods such as black-box and white-box.

Documentation and Maintenance

User and technical documentation are produced, although this generally begins before the system is complete since it is a slow process. The resulting documentation is seldom concurrent with the final system, since the programmers may be making changes very late in the development which are not reflected in the documentation, as no extra time is available.

Maintenance begins immediately after acceptance testing, as corrective maintenance may be required. Enhancements added to the system later on are also classed as maintenance. This stage is the longest part of the process, but is usually overlooked.

Personnel

Client

The client is the organisation and its staff for whom the system is to be developed. Because the body of staff may be very large, their needs must be represented. Management will aim to do this in the initial stage, and then Systems Analysts gather detailed information more directly.

Management

These are leading members of staff from the client organisation, who are responsible for representing the organisation's needs. Important decisions are the responsibility of management – they can commission the feasibility study, and must then decide whether or not to commit. They must also negotiate the Operational Requirements Specification to see that they are getting what they need.

Consultants

This is the company appointed by the management of the client organisation to carry out the development. The consultants provide a Project Leader, Systems Analyst(s) and Programmers.

Project Leader

The Project Leader is the most senior member of the consultant's staff. They must liaise with management to determine the problem definition, and will carry out the Feasibility Study. They must oversee the whole development.

Systems Analyst

A Systems Analyst is a member of the consultants' staff who is assigned to the project to conduct a System Investigation. After this, they are responsible for the rest of the development. In reality, there is often more than one analyst working on a project, each reporting to the Project Leader. Analysts must be able to communicate clearly and effectively with the client in order to analyse the system fully. Many analysts also have some technical knowledge, to allow them to communicate with the Programmers who work for them.

Programmers

These are a large part of the consultants' staff, and are responsible for the implementation of the system in a programming language. They are under the management of the Systems Analyst. Their job involves working out the detailed logic and then implementation and testing (although detailed testing may be the responsibility of a dedicated team). Programmers also carry out maintenance on the software.