8.4 Development and testing

Once the design stage is completed, it is then necessary to create the system and fully test it. This section considers some of the development stages and testing strategies which are often adopted by systems analysts.

Development stages

If the system contains files (e.g. a database) then the file structure needs to be finalised at this stage (e.g. what type of data is being stored in each field, length of each field, which field will be the key field, how the data files will be linked, etc.). Once the file structure has been determined, it is then created and fully tested to make sure it is robust when the system actually goes live.

Since it is important that the correct data is stored in files, there are certain techniques that need to be adopted to make sure the data populating the file/s and database/s is at least of the right type and that it conforms to certain rules. Validation routines and verification methods (discussed in Section 8.3) are used to ensure this happens. Again, these routines have to be fully tested to ensure they do trap unwanted data but also to make sure any data transferred from a paper-based system to an electronic system has been done accurately.

Any system being developed will have some form of user interface. The types of hardware were chosen in the design stage. How these are used to interface with the final system now needs to be identified, for example how the screens (and any other input devices) will be used to collect the data and the way the output will be presented. If specialist hardware is needed (e.g. for people with disabilities), then it will be necessary to finalise how these devices are used with the system when it is implemented. This will be followed by thorough testing to ensure the user screens are user friendly and that the correct output is associated with the inputs to the system.

Testing strategies

Testing of each module needs to be done to ensure each one functions correctly on its own. Once the development of each module is completed, the whole system needs to be tested (i.e. all modules functioning together). Even though each individual module may work satisfactorily, when they are all put together there may be data clashes, incompatibility and memory issues, etc.

All of this may lead to a need to improve the input and output methods, file and database structures, validation and verification methods, etc. Then the system will need to be fully tested again. It is a very time-consuming process but the system has to be as perfect as possible before it goes live.

Testing will use many different types of data, which will fall into one of three categories: normal, extreme or abnormal. Let us suppose one of the fields in a database is the date and this must be in the form dd/mm/yyyy, where each element of the date must be numeric:

- **Normal**: this is data which is acceptable/valid and has an expected (known) outcome, e.g. the month can be *any* whole number in the range 1 to 12.
- **Extreme**: this is data at the limits of acceptability/validity, e.g. the month can be either of the two end values i.e. 1 *or* 12.
- **Abnormal**: this is data outside the limits of acceptability/validity and should be rejected or cause an error message. For example, all the following values are not allowed as inputs for the month:
  - negative numbers (e.g. –1, –15)
  - any value greater than 12 (e.g. 32, 45)
  - letters or other non-numeric data (e.g. July)
  - non-integer values (e.g. 3.5, 10.75).