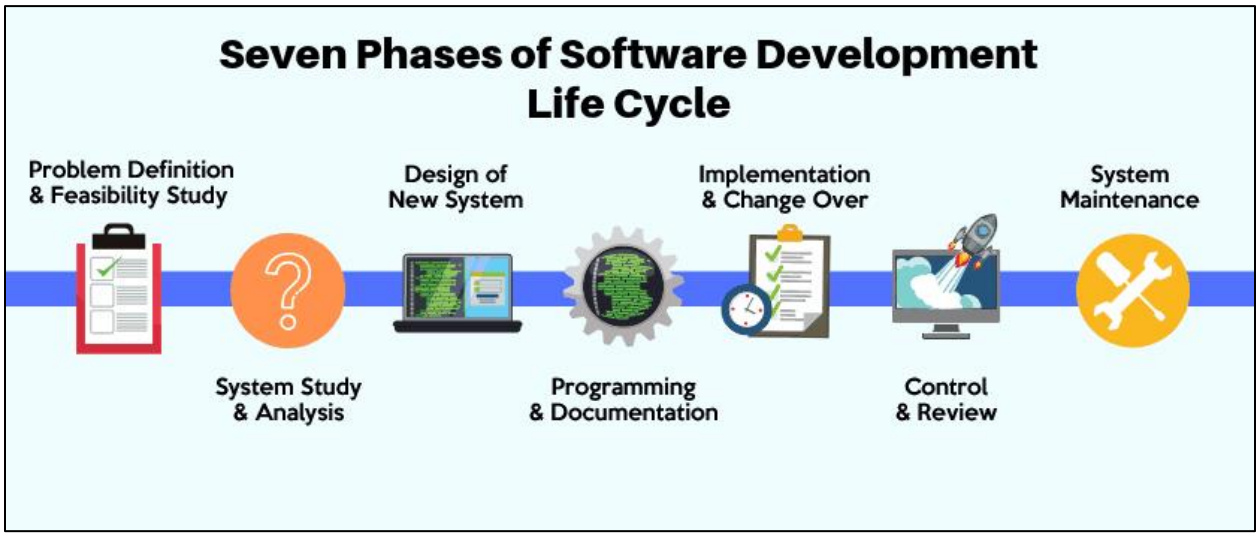
System Life Cycle

The System Life Cycle is a series of stages that are worked through during the development of a new information system. A lot of time and money can be wasted if a system is developed that doesn't work properly or do exactly what is required of it.

A new system is much more likely to be successful if it is carefully planned and developed through the following 7 stages:



1. Problem Definition and Feasibility Study

The very first part of the SLC is to define the problem. The system analyst must determine why a new system is required. After all, if there isn't a problem to start with, why would an organisation incur huge costs to develop a new one?

Once the system analyst is convinced that there is a problem which could be solved with a new IT system they have to determine whether it is feasible to actually go ahead and develop the system. Some of the questions that will need to be answered are:

- Cost – How much would the new system cost to develop?
- Budget - Would there be enough money available in the budget to develop the system?
- Time - How long would it take to make the system from start to finish?
- Hardware - Would new hardware be needed to run the system? How much would that cost?
- Training - What would the training implications be once the system had been developed?
- Technical feasibility - After finding out what is required is it technically possible to create the system

If the decision is made that it is worth developing a new system, the SLC will progress onto the next stage, Investigation and Analysis. If management decide to stick with the current system, the SLC will stop here.

2. System Study and Analysis

During the analysis stage systems analysts investigate the existing system in detail to identify exactly what the problems are with the existing system. Systems analysts will use a variety of fact-finding methods to gather information for example:

- Observations: The analyst will observe users actually using the system. They will probably follow a complete process from start to finish and note down every interaction that happens. This is very time consuming
- Questionnaires: Questionnaires enable the analyst to obtain the views of a large number of staff/ users. Questionnaires are also easier to analyse than face-to-face interviews but the trade-off is that they don't give as much detail.
- Interviews: The analyst will interview selected staff who use the current system in order to get a detailed overview of how things work. They will want to know what the main problems are and whether users have any suggestions on how to improve the way things work.
- Hands-on approach: analyst goes to work with other workers for a short time. The most effective way to understand the system but also the most time consuming.
- Examining documents: the analyst will go through the various documents to gather more information on how the current system is operating

3. Design of New System

All the information obtained in the previous stage is carefully examined and analysed to determine the requirements for the new system. Different possible solutions are identified and evaluated. Finally, the best solution is identified and designed. A design specification is produced containing a number of diagrams and other information such as:

- Inputs and Outputs
- Data storage and file organisation
- User interface: screen layouts, buttons, error messages
- Backup and recovery procedures
- Security procedures

The main diagrams produced include:

- Flowcharts
- Data-flow diagrams
- Structure Charts

4. Programming and Documentation

Once the design stage has been completed the software developers can begin to write the code and actually develop the new system. At this point it is very important that the program is well defined, designed, written and tested. During this stage some of the following are developed:

- the tables and data structures
- validation routines
- data capture forms
- data input forms
- automated processing routines i.e. macros
- queries
- the user interface i.e. screen, buttons, help messages
- printing outputs

Testing

Once the system has been developed it must be tested to ensure that it is working as expected. This stage involves:

1. Setting up the system so that it matches the design specification.
2. Testing carried out using the plan to make sure that all the parts of the system work correctly with normal, extreme and erroneous data:
    - Normal test data is used to check that a system can handle the sort of data that would be expected during day-to-day use
    - Extreme test data is used to check that a system can cope with data that lies on the boundaries of what is acceptable
    - Erroneous (or exceptional) test data is used to check that a system can identify data that is wrong and reject it

Documentation

It is also at this stage that the software produced is documented. Normally the following manuals are produced:

- User
- Technical
- Program

User's Documentation:

This documentation has to be of a non-technical nature so that the **non-specialist user** can use the software without problems. A user documentation explains how to run and use the software. It includes:

- instructions on how to load and run the system;
- detailed instructions on how to operate each part of the system such as inputs and outputs required at each stage;
- Error messages, their meaning and how to deal with them.
- Where to get more help, such as telephone support lines and on-line tutorials.

Technical Documentation:

This describes how the system works. It is written for computer professionals such as **technicians** and **operators**. This manual should include details on

- a description of what the system is designed to do;
- how to install the system;
- hardware requirements of the system (including peripherals and their configuration);
- maintenance that should be carried out;
- how to solve certain problems that may arise;

Program Documentation:

This is a complete description of the software to be used by **programmers** when maintenance of the software is required. It should include

- system design specifications
- system flowcharts
- data-flow diagrams
- the format of input data and outputs produced
- the program listings (a printout of the whole program)
- data structures used
- screen layouts and user interface designs
- the test plan

5. Implementation and Change Over

The system has been developed and tested. It is working correctly and doing everything that was agreed during the design stage. The business is waiting in eager anticipation for the new system to be handed over to them.A key decision is which method of the three different methods of installation will be chosen. These are:

- Direct (Straight)
- Parallel
- Phased

Direct method

This is where the company literally switches off the old system and switches on the new one. This is probably the most straightforward method but is also probably the riskiest.

Advantages	Disadvantages
New system available to everyone in company immediately	Most risky method - if something goes wrong, there is nothing to fall back on. Have to wait while the problem is fixed.
Often the cheapest method of installation	Have to transfer all of the data to the new one before the old one can be switched off
Don't need to keep duplicate sets of data	There will be a period of time where no system is available because can't have old one working while new one is being switched on
	There will be a period of upheaval while the system is brand new and staff are finding their way around it

Parallel method

This is a more popular method than the previous one. With a parallel changeover the organisation runs both the old and new system in parallel for a time. Once the organisation is sure that the new system is working properly and that staff are ready to begin using it they will make the decision to completely change over. During a quiet period, perhaps during the night or at a weekend, the data is fully transferred from the old system which is then shut down.

Advantages	Disadvantages
Less risky than the direct method. If the new system fails, the old system is still up-to-date	Time consuming as data has to be entered onto both systems
Less stress for staff as they still have the security of the old system	One system can become out of sync. with the other.
Staff can take their time to learn to use the new system	Maintaining duplicate sets of data can lead to errors
	Extra cost of running and maintaining two systems

Phased method

This is where the old system is still active but parts of the new system or modules are brought online, for example, perhaps just the data entry screens and the printing modules are made available but the 'back end' of the system remains the same. Once any problems are ironed out with the new modules then extra modules will be introduced. Effectively the installation happens in small chunks.

Advantages	Disadvantages
Less risk of the whole system going wrong, if something happens, it will only affect that specific part.	This method of installation can take a long period of time
Staff are introduced to the changes in small stages	As parts of the system are used, users ask for changes which then hold up the installation of the next phase
	It might be difficult to integrate the old and the new systems

6. Control and Review

The installation stage is over: The system is up and running. After the new system has been running for a few weeks or months the system is revisited to identify any modifications that may need to be made. Once all the staff are fully trained and bugs have been ironed out, the analyst looks at the overall project and considers how things went.

Review involves all the key players in the project. These include

- Original client
- Project manager
- System Analyst
- Designers
- Developers
- Testers
- Support staff such as trainers
- A selection of end-users

During the review stage, two key questions are considered:

- Does the finished solution meet its requirements?
- Does it solve the problem?

It is important that failures and particular successes should be discussed.

7. Maintenance

This phase continues for the lifetime of the system. The technical and programming documentation is essential to support the maintenance stage.

There are three kinds of maintenance needed:

- Corrective maintenance
- Adaptive maintenance
- Perfective maintenance

Corrective maintenance

This is where problems are identified with the system after installation. Perhaps an item on the template isn't printing out correctly or maybe one of the on-screen buttons isn't linking to the correct form. A fault report is raised and the developers fix the problem. Corrective maintenance can also involve fixing hardware faults or replacing equipment as necessary.

Adaptive maintenance

This type of maintenance often occurs as a result of external influences or strategic changes within the company. For example, the Government recently changed the VAT rate from 17.5% to 20%. This change would have meant that many organisations had to make alterations to their systems.

Perfective maintenance

The system has been in place and running fine for a while. However, over time, the end user will often find tweaks or minor improvements which could be made to improve the way the system works. Perhaps a slightly different screen or data input form. These tweaks are not major enough to prompt a complete new system, so the maintenance team adapt the system to suit.

It is important to be aware that while the system remains in use the maintenance stage will be ongoing.