

12 Software Development – Part 1

Name: _____ Class: _____ Date: _____

Vocabulary

Write each word three times.

English	Write it 3 times (English)		
development	_____	_____	_____
life cycle	_____	_____	_____
requirements	_____	_____	_____
waterfall	_____	_____	_____
agile	_____	_____	_____
prototype	_____	_____	_____
syntax error	_____	_____	_____
logic error	_____	_____	_____
normal data	_____	_____	_____
boundary data	_____	_____	_____
maintenance	_____	_____	_____

Recall

At IGCSE you wrote programs and fixed the mistakes in them. Bring this with you:

- A program rarely works perfectly the first time.
- Testing means trying the program to see if it behaves correctly.

At AS we look at how larger software is planned, tested and looked after.

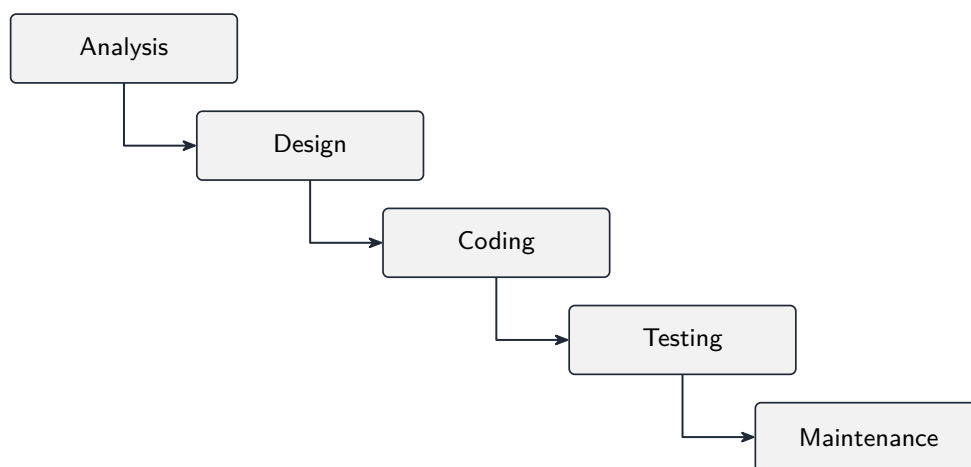
New ideas

Read these first. Each idea has a short worked example. Then do the Practice.

■ 1 The development life cycle

Software is built in stages called the **development life cycle** : find the **requirements** , design, write the code, test, then maintain it. Two common models:

- **waterfall** : each stage is finished before the next begins, in order;
- **agile** : build in small repeated cycles, often making a quick **prototype** to show the user early and get feedback.



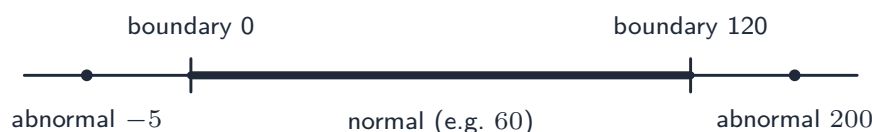
Worked example. A team builds a booking app the agile way: a basic prototype in two weeks, shown to the client, then improved in short cycles – instead of waiting months for one big waterfall release.

■ 2 Testing

Programs can contain errors:

- a **syntax error** breaks the language's rules, so the program will not run;
- a **logic error** lets the program run but gives the wrong answer.

Test with different kinds of data: **normal data** (typical values that should be accepted), **abnormal data** (wrong values that should be rejected), and **boundary data** (values right at the edge of what is allowed).

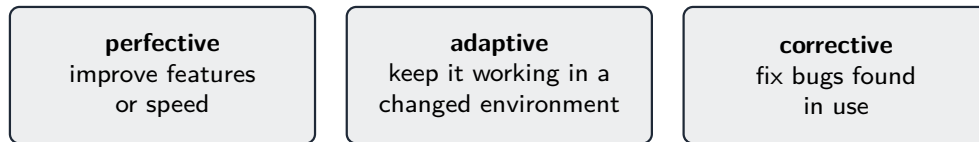


Worked example. A field accepts ages 0–120: normal data 25 (accept), boundary data

0 and 120 (the edges), abnormal data -3 or 200 (reject).

■ 3 Maintenance

After release, software still needs **maintenance** – fixing bugs that appear, adapting it to new hardware or rules, and adding improvements.



Worked example. After release: fixing a crash is corrective; making it run on a new phone is adaptive; adding a dark-mode option is perfective.

Watch out: a logic error is the dangerous one – the program *runs*, so it looks fine, yet quietly gives wrong answers. A syntax error at least stops the program.

Practice

12.1 Name three stages of the development life cycle. [3]

12.2 State one difference between the waterfall and agile models. [2]

12.3 State the difference between a syntax error and a logic error. [2]

12.4 A program accepts ages from 0 to 120. Give one piece of boundary data and one piece of abnormal data. [2]

12.5 State what is meant by software maintenance. [1]

12.6 Explain *why* boundary data is especially good at finding errors in a program. [2]

12.7 Challenge. A program should accept marks from 1 to 100. Give one normal, one boundary and one abnormal test value, and say what each one checks. [3]
