Computing GCSE – 2.3

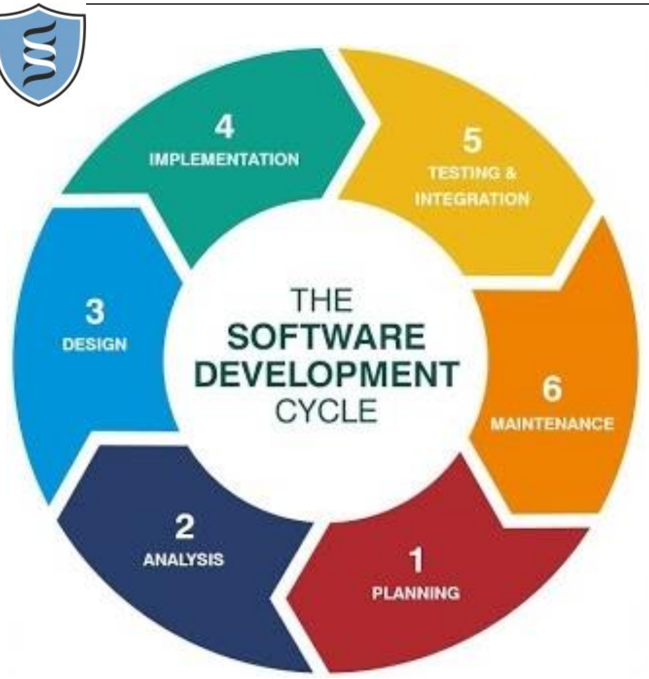
J276/02 – Producing Robust Programs

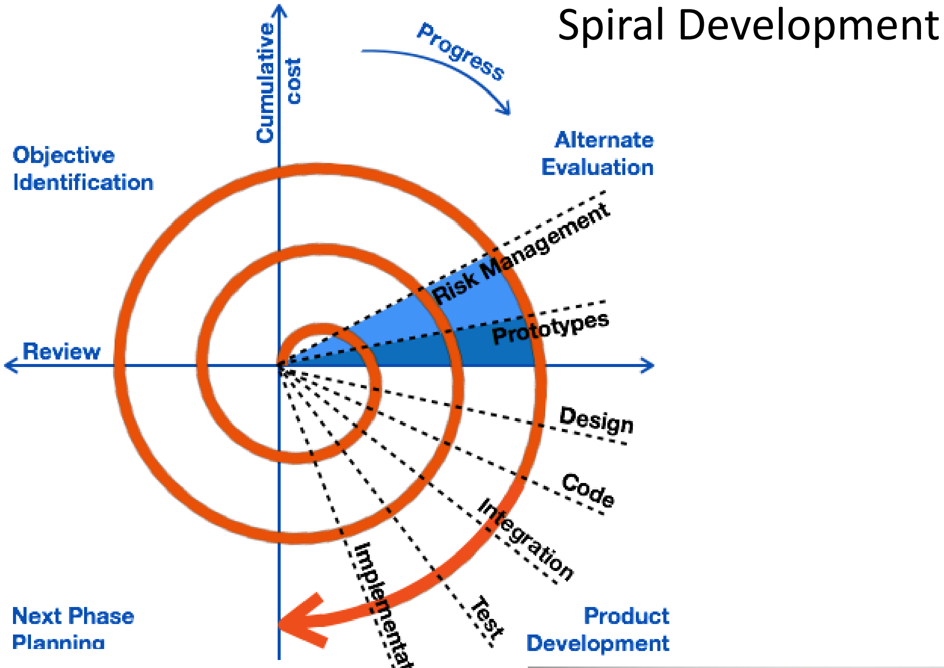
KEY VOCABULARY	
Defensive design	Planning a program from the very beginning to prevent accidental or purposeful misuse
Input sanitization	Removing erroneous data from a system prior to processing
Data validation	Ensuring all data is in the correct format prior to processing
Contingency planning	Having built in checks and outcomes based on what happens when things go wrong
Anticipating misuse	Building programs which do not allow a user to deliberately break the system
Authentication	Having different levels of user, and preventing everyday users from being able to significantly change a system
Maintainability	Building software which is modular to enable sections to be updated and replaced without having to write the whole program again from scratch
Code comments	Annotating code so that the person maintaining or working with your code in the future is able to understand your thought process
Indentation	Making code more readable by laying it out in a manner that keeps sections of code separate
Iterative testing	Step by step testing to ensure that small sections of the code work, before new parts are added and then retested. Important to allow *traceback* to find what caused any errors
Terminal testing	Significant testing done once a program is complete under a range of conditions and on multiple hardware – often called *Alpha Testing*
Beta Testing	Making a small release of the software to a group of tech-literate enthusiasts to broaden the usage-testing and get lots of feedback prior to full release.
syntax error	An error in the typing of the code. Missing punctuation, spacing etc
Test data	Data chosen to test the program. Testers use a specific range of data

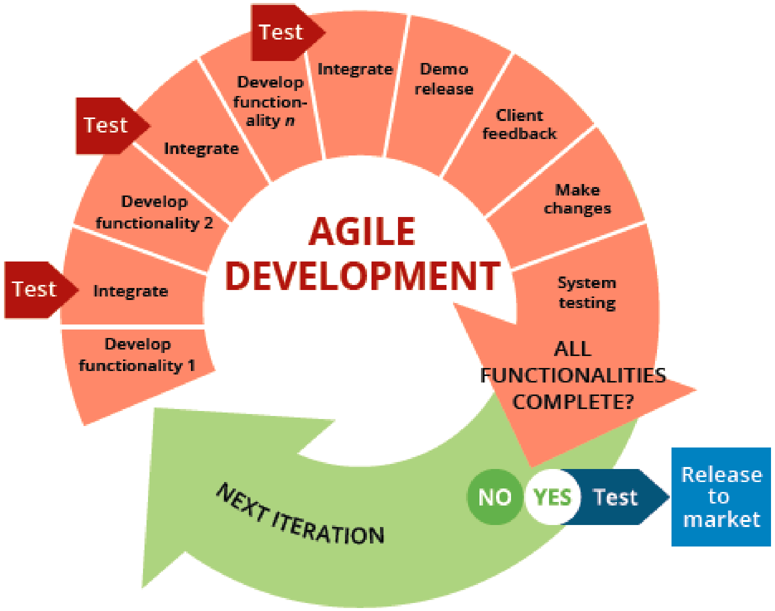
TESTING DATA	
Data Range	The data that will be used to check the code works correctly
Valid Data	Obvious data which should definitely pass
Valid Extreme	Unusual data – the highest and lowest data – on the very edge of what should pass
Invalid Extreme	Data, of correct type, which is on the very edge of what should fail
Invalid Data	Data, of the correct type, that should definitely fail
Erroneous Data	Data that is the wrong type and should fail
Expected Outcome	The data the code should output if it is running correctly

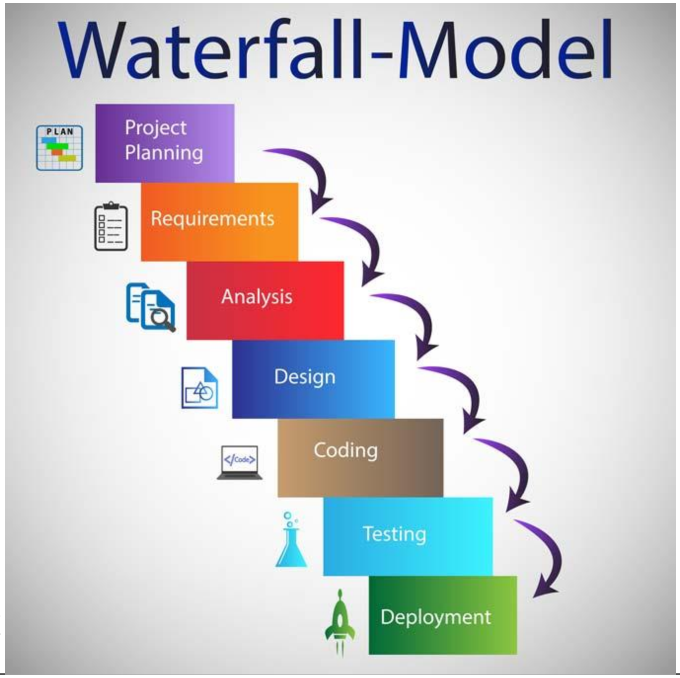
ERROR TYPES	
Syntax Error	An error in the code – incorrectly typed, missing punctuation etc
Logical Error	An error which, although allows the code to run, produces incorrect outcomes
EOF Error	The *End of File* has been reached, whilst the computer is waiting for a snippet to be completed.
Type Error	Attempting to use data incorrectly – adding 1 to a string etc
Name Error	Using a variable before its declaration
Indentation Error	Loops or functions are incorrectly indented

Models of Software Development


THE SOFTWARE DEVELOPMENT CYCLE
1 PLANNING
2 ANALYSIS
3 DESIGN
4 IMPLEMENTATION
5 TESTING & INTEGRATION
6 MAINTENANCE


Spiral Development
Cumulative cost
Progress
Objective Identification
Alternate Evaluation
Risk Management
Prototypes
Review
Design
Code
Integration
Test
Implementation
Next Phase Planning
Release
Product Development


AGILE DEVELOPMENT
Develop functionality 1
Test
Integrate
Develop functionality 2
Test
Integrate
Develop function-ality n
Test
Integrate
Demo release
Client feedback
Make changes
System testing
ALL FUNCTIONALITIES COMPLETE?
NO
YES
Test
Release to market
NEXT ITERATION


Waterfall-Model
PLAN
Project Planning
Requirements
Analysis
Design
</code>
Coding
Testing
Deployment