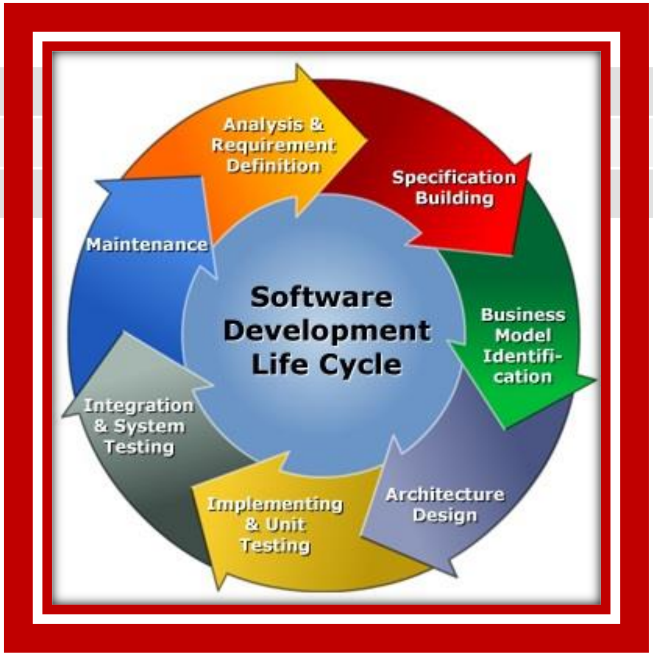
Component 1 | 1.2.3| Software development

Specification & learning objectives

	A Level	Specification point description
	1.2.3a	Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development
	1.2.3b	The relative merits and drawbacks of different methodologies and when they might be used
	1.2.3c	Writing and following algorithms
		Different test strategies, including black and white box testing and alpha and beta testing
		Test programs that solve problems using suitable test data and end user feedback, justify a test strategy for a given situation

Resources

PG Online textbook page ref: 52-63
Hodder textbook page ref: 116-122
CraignDave videos for SLR 6



Analysis & Requirement Definition
Specification Building
Business Model Identifi-cation
Architecture Design
Implementing & Unit Testing
Integration & System Testing
Maintenance
Software Development Life Cycle

Component 1 | 1.2.3| Software development

Key question: How are large scale programming projects undertaken?

A key part of any computer system is the software that runs on it.
For many organisations, much of this software is bespoke (custom made) and can run to thousands or even millions of lines of code.
It is common to have whole teams of people employed to write code. This needs organisation and discipline.
And for this to happen there needs to be a software development methodology in place.
Software development methods have been developed over the years which offer:

- A way of capturing what the system needs to do
- Ways of breaking up the system into manageable chunks
- Ways of keeping track of changes
- Ways of allocating staff
- Ways of finding and correcting errors (debugging)
- Ways of planning and carrying out tests

Key question: How are large scale programming projects undertaken?

SDLC: When developing software you go through stages known as the system development lifecycle, these are the steps involved in developing the software.

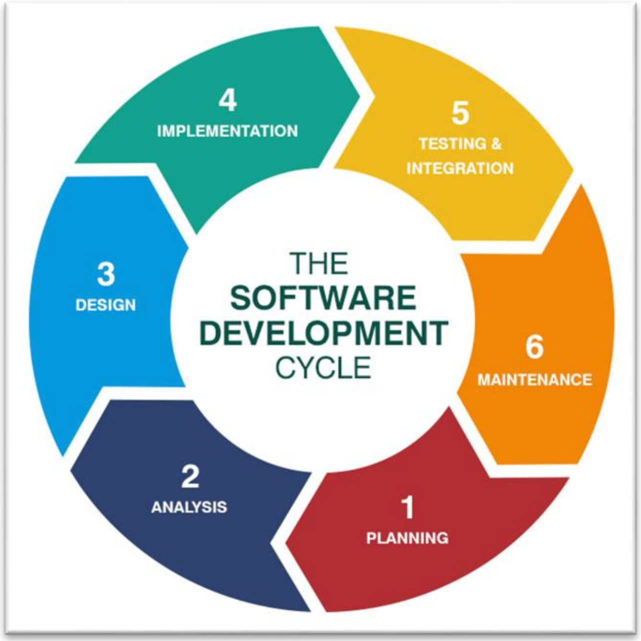
Waterfall Model: In the waterfall model these development processes are shown to be flowing down in the model hence why it is called the waterfall model. Each stage in the waterfall cycle must be done before the next one can be completed and is in linear structure.

Agile methodologies: When the requirements of the software is constantly changing whilst the software is being developed.

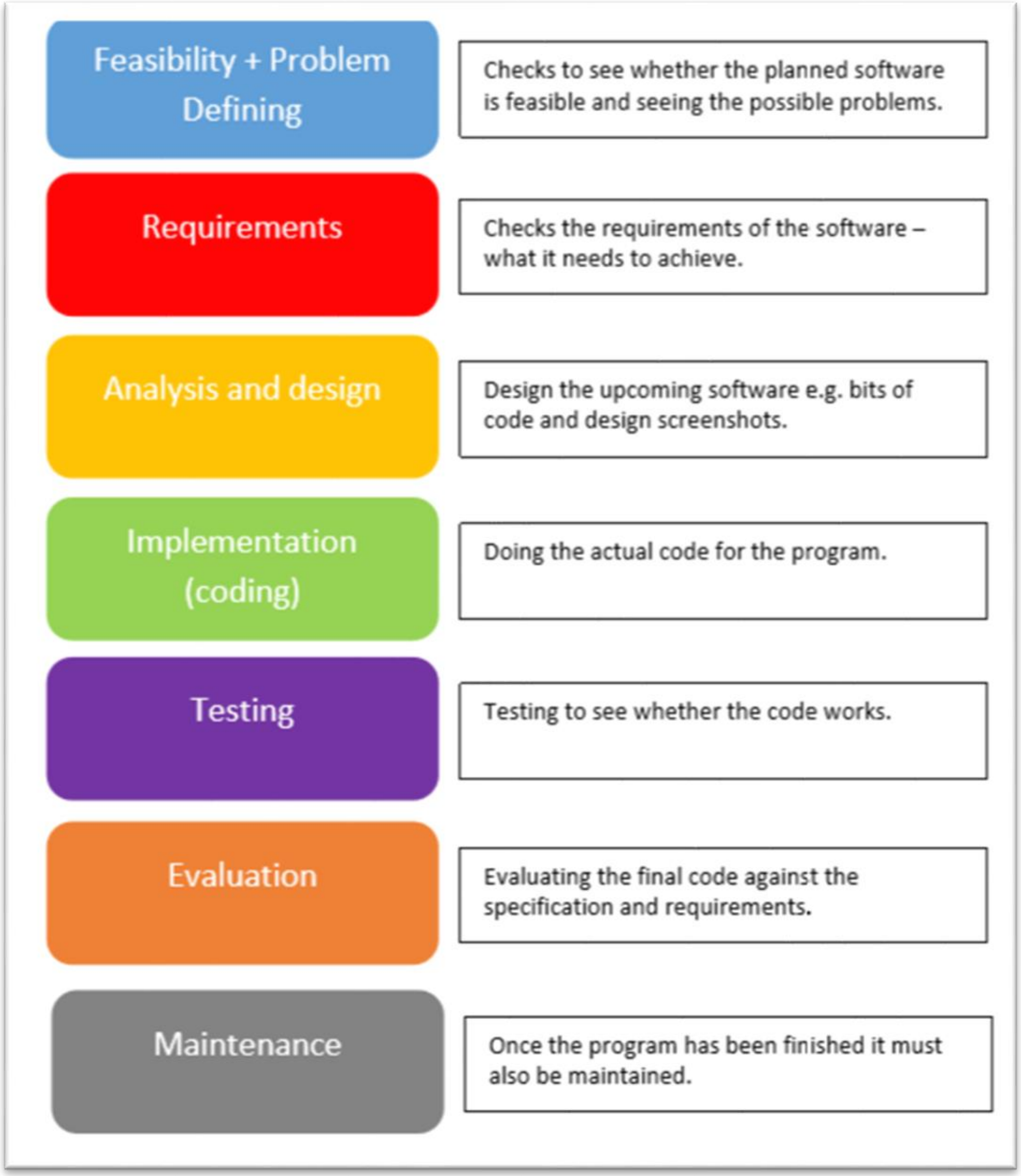
Extreme programming: Two programmers sitting side by side at one computer whilst the code is written, one writes the code whilst the other checks the validity of the code.

Spiral model: The spiral model is based on risk, and is favoured for large, expensive and complicated projects.

RAD: Rapid application development has no detailed planning, uses minimal planning in favour of rapid prototyping.

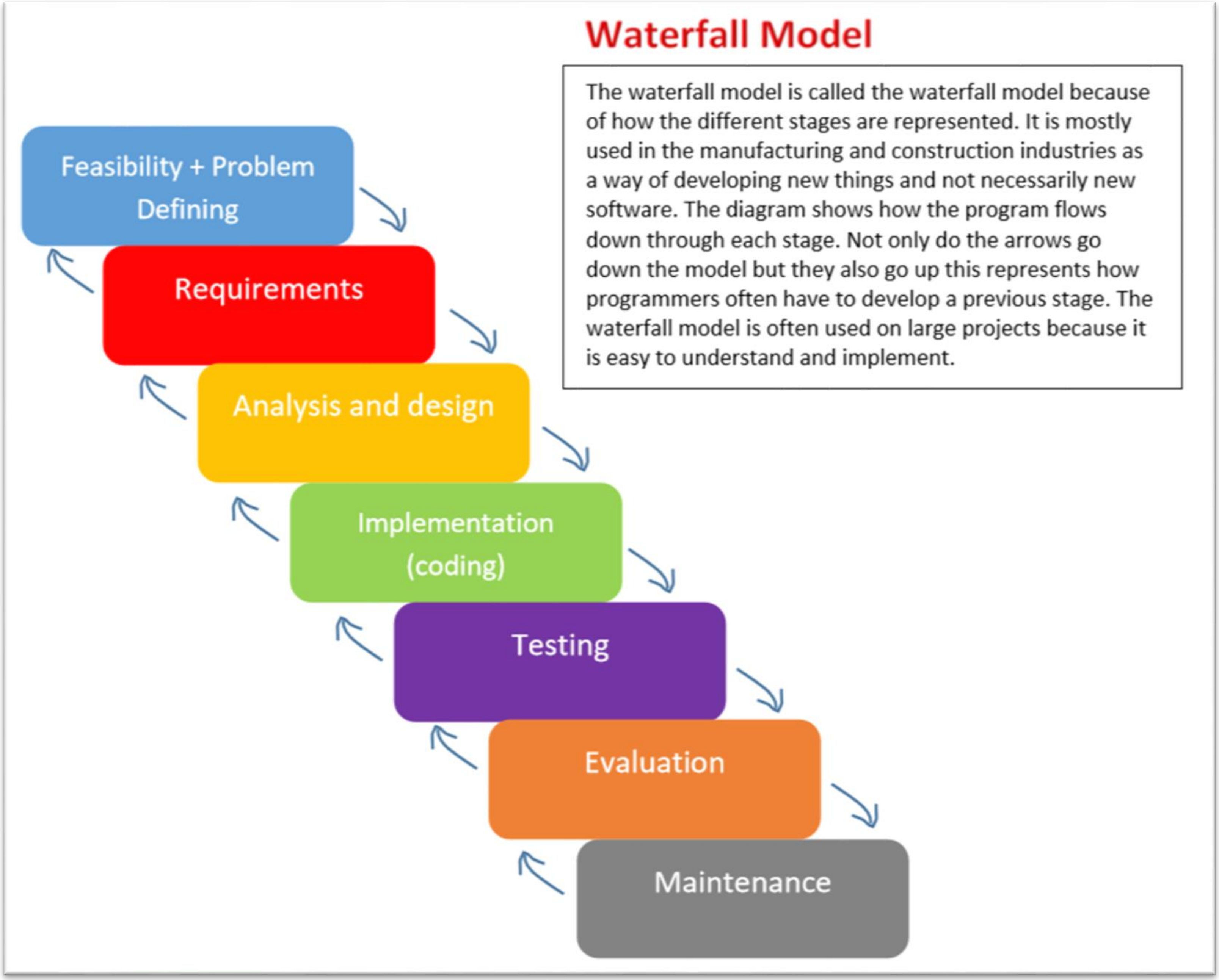


Component 1 | 1.2.3| Software development



Feasibility + Problem Defining	Checks to see whether the planned software is feasible and seeing the possible problems.
Requirements	Checks the requirements of the software – what it needs to achieve.
Analysis and design	Design the upcoming software e.g. bits of code and design screenshots.
Implementation (coding)	Doing the actual code for the program.
Testing	Testing to see whether the code works.
Evaluation	Evaluating the final code against the specification and requirements.
Maintenance	Once the program has been finished it must also be maintained.

Component 1 | 1.2.3| Software development

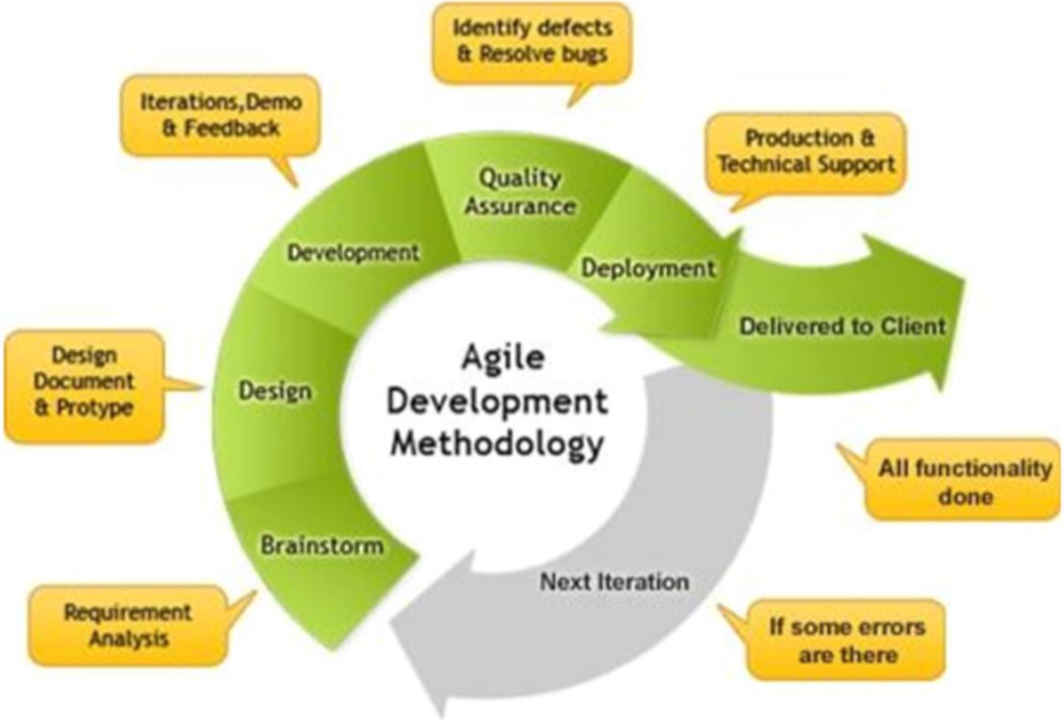


Waterfall Model

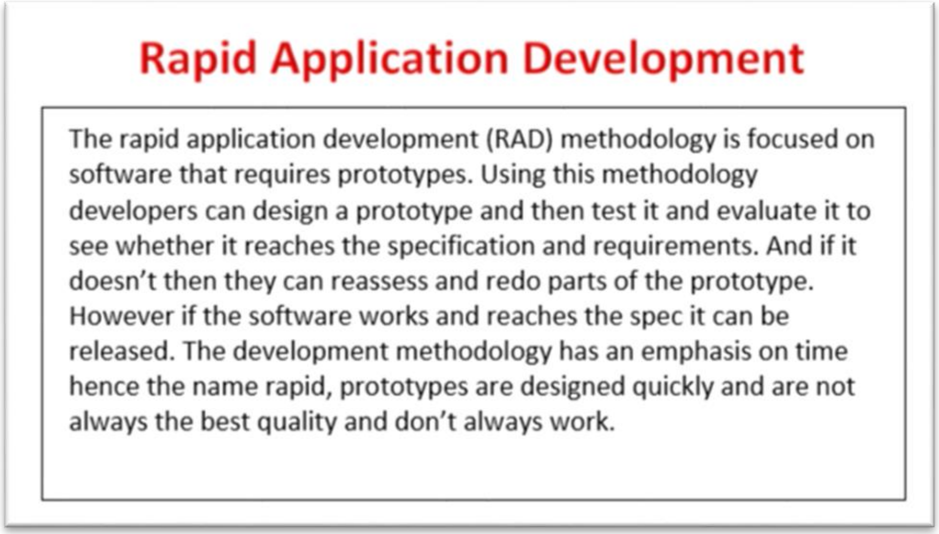
The waterfall model is called the waterfall model because of how the different stages are represented. It is mostly used in the manufacturing and construction industries as a way of developing new things and not necessarily new software. The diagram shows how the program flows down through each stage. Not only do the arrows go down the model but they also go up this represents how programmers often have to develop a previous stage. The waterfall model is often used on large projects because it is easy to understand and implement.

Agile Model

Concentrates on the fact that the requirements will change frequently whilst the program is made. The program is made in an iterative way. This means that with each iteration the requirements change or there are more requirements. This also means the program can constantly be shown to the user who can change what they want. However with this method the price might increase constantly as the requirements increase.

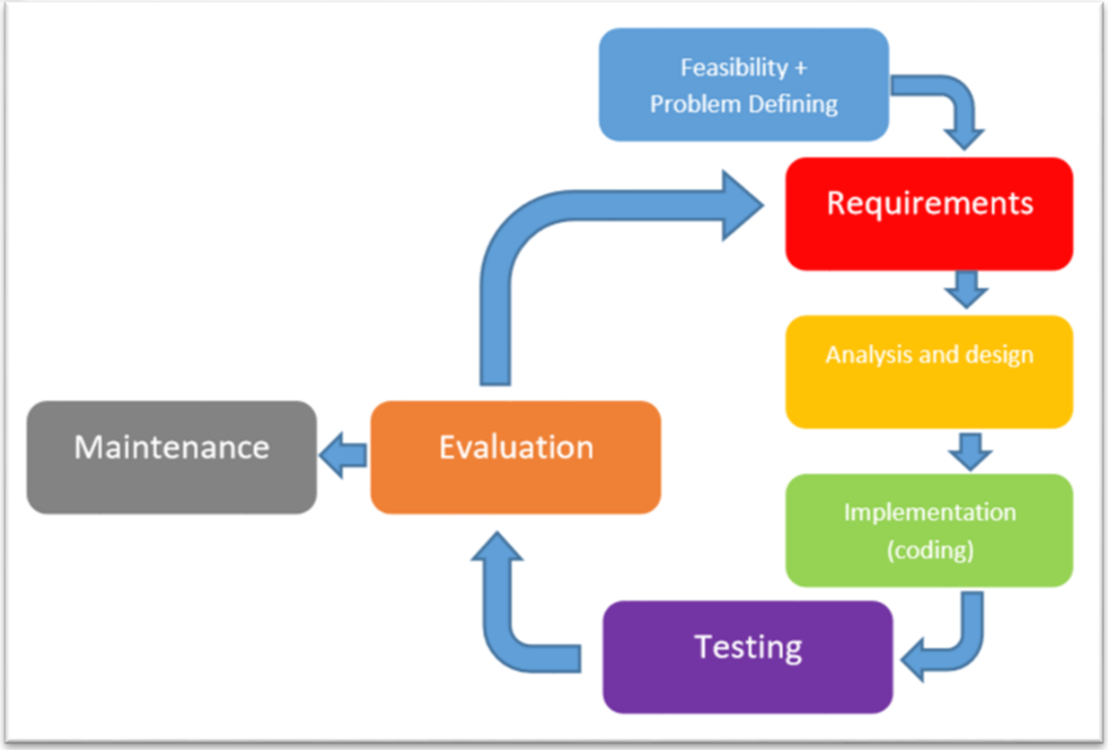


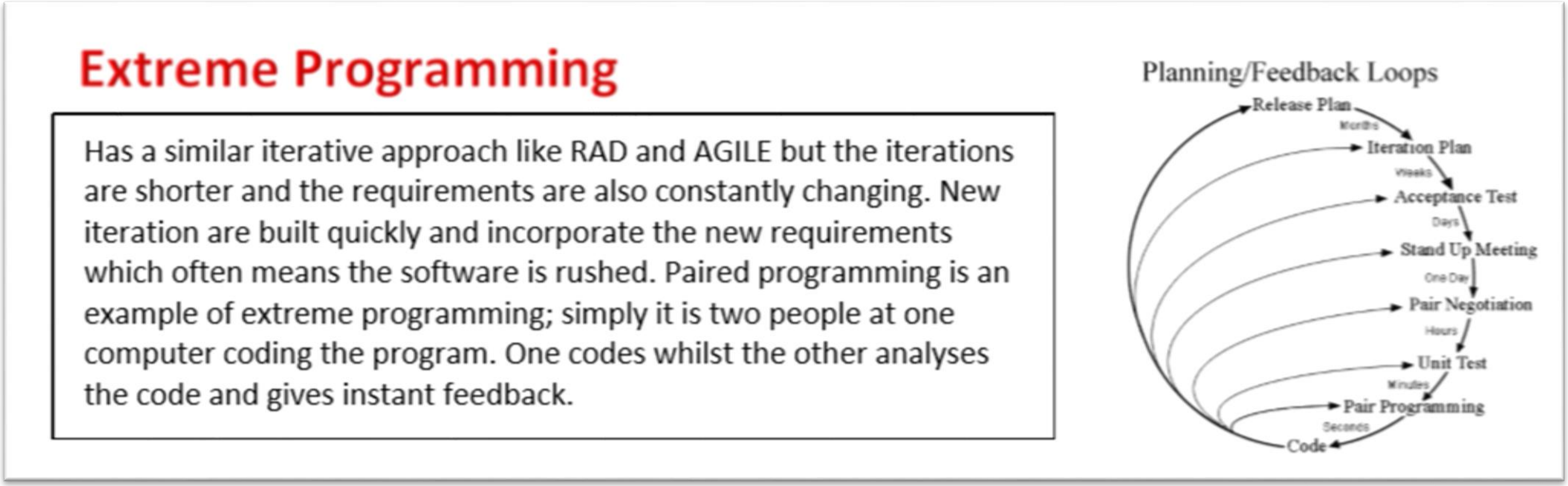
Component 1 | 1.2.3| Software development



Rapid Application Development

The rapid application development (RAD) methodology is focused on software that requires prototypes. Using this methodology developers can design a prototype and then test it and evaluate it to see whether it reaches the specification and requirements. And if it doesn't then they can reassess and redo parts of the prototype. However if the software works and reaches the spec it can be released. The development methodology has an emphasis on time hence the name rapid, prototypes are designed quickly and are not always the best quality and don't always work.



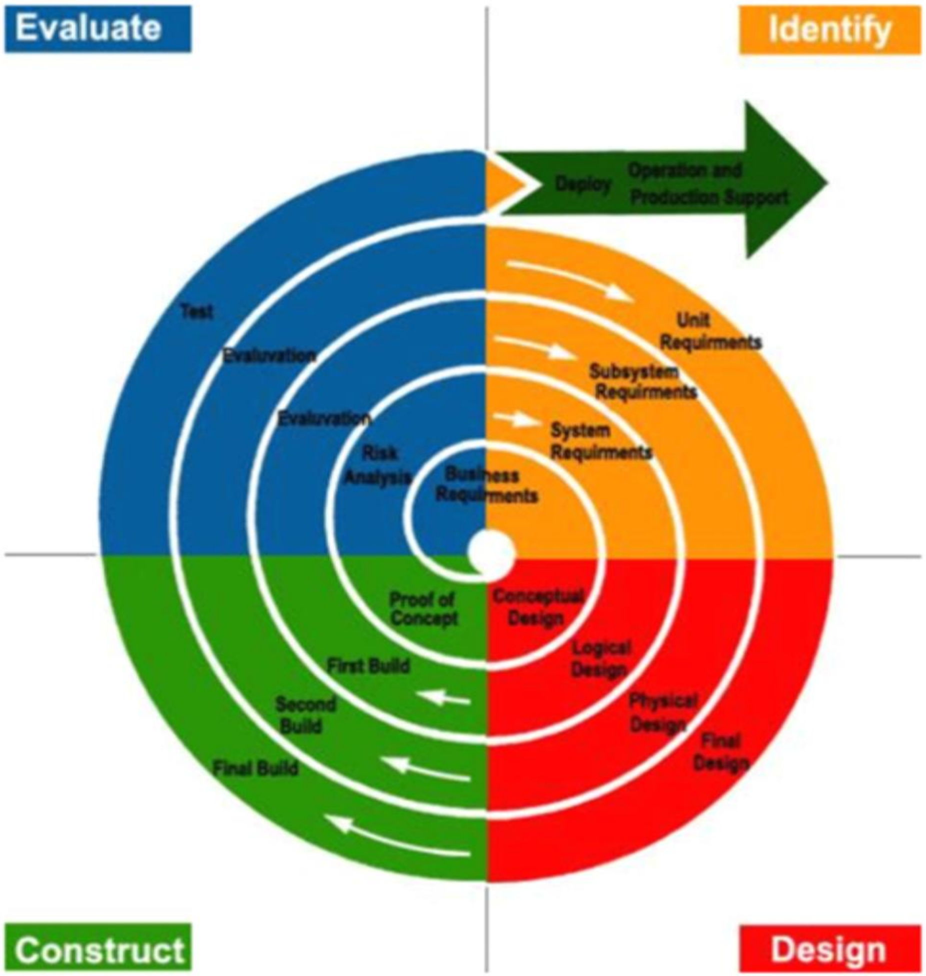


Extreme Programming

Has a similar iterative approach like RAD and AGILE but the iterations are shorter and the requirements are also constantly changing. New iteration are built quickly and incorporate the new requirements which often means the software is rushed. Paired programming is an example of extreme programming; simply it is two people at one computer coding the program. One codes whilst the other analyses the code and gives instant feedback.

Spiral Model

The spiral development methodology has an emphasis on risk, and therefore acknowledges the risk that impacts lots of large scale development projects. The model has 4 stages evaluate, identify, construct and design each taking up a section of the circle. The first section looks at requirements such as unit, subsystem, system and business requirements and also looks at risk and issues that might arise. The next section implements the design of the program and the next stages is to construct the program including the code and graphics and is therefore the first prototype. The next section allows the evaluation of the program to test it, evaluate it against spec and then evaluate risk.

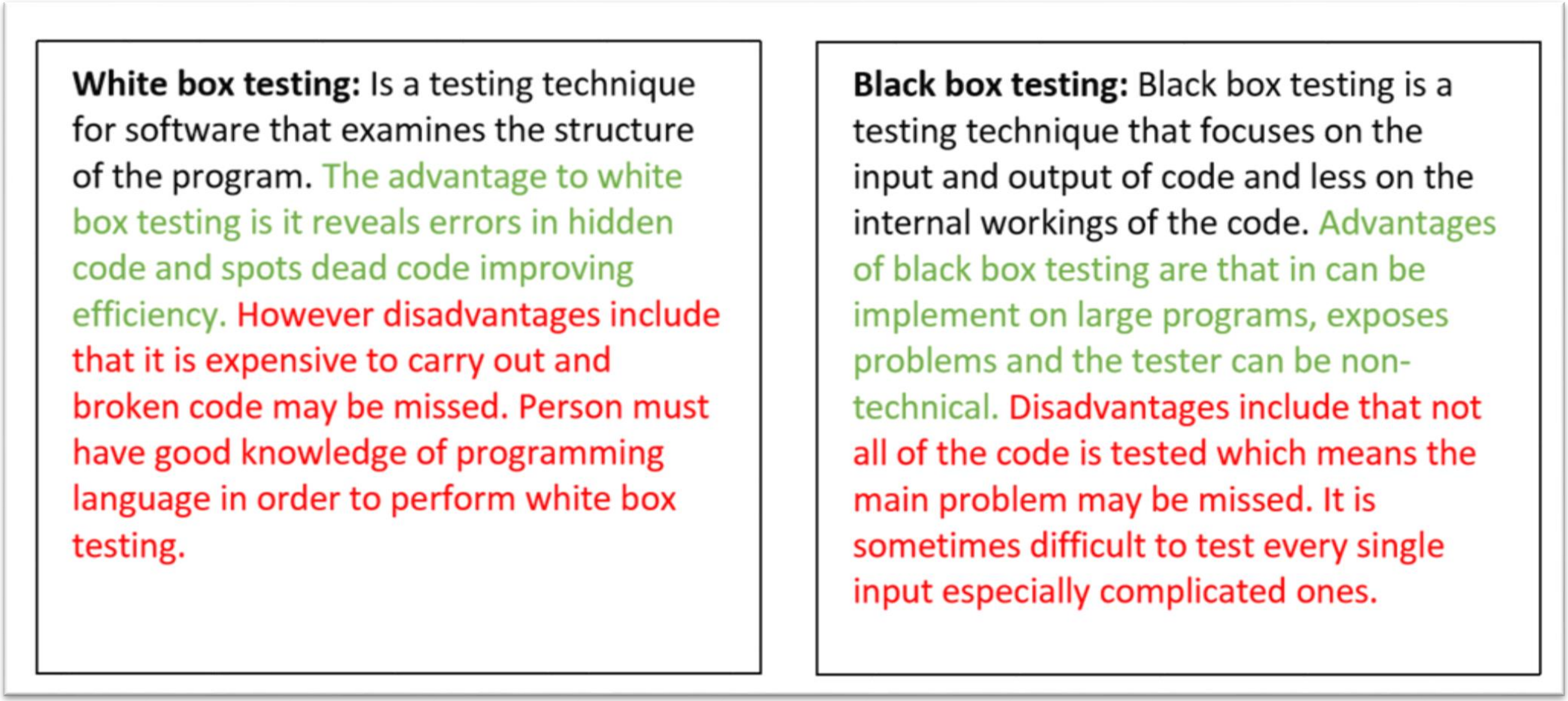


Component 1 | 1.2.3| Software development

Waterfall Methodology	Rapid Application Development (RAD)	Spiral Methodology	Agile Methodology	Extreme Programming
Simple to understand the approach and therefore easy to manage and see if it is running to schedule.	Requirement don't have to be clear because the client gets to see the software frequently.	Has an emphasis on risk. Good for large projects due to the risk management.	Concentrates on software requirement constantly changing, code is made in a iterative way.	Has an emphasis on code meaning that is good quality and is efficient.
Lot of risk associated because user doesn't get to see it until the end and requirements may have been misunderstood.	The code lacks efficiency and is not good for large software projects.	However risk management is expensive and a good risk analyser is a highly specialised skill. Communication must be maintained with client.	High cost and communication must be maintained with client.	The client needs a full time representative working with the development team. The development must be geographically close to each other.

Component 1 | 1.2.3| Software development

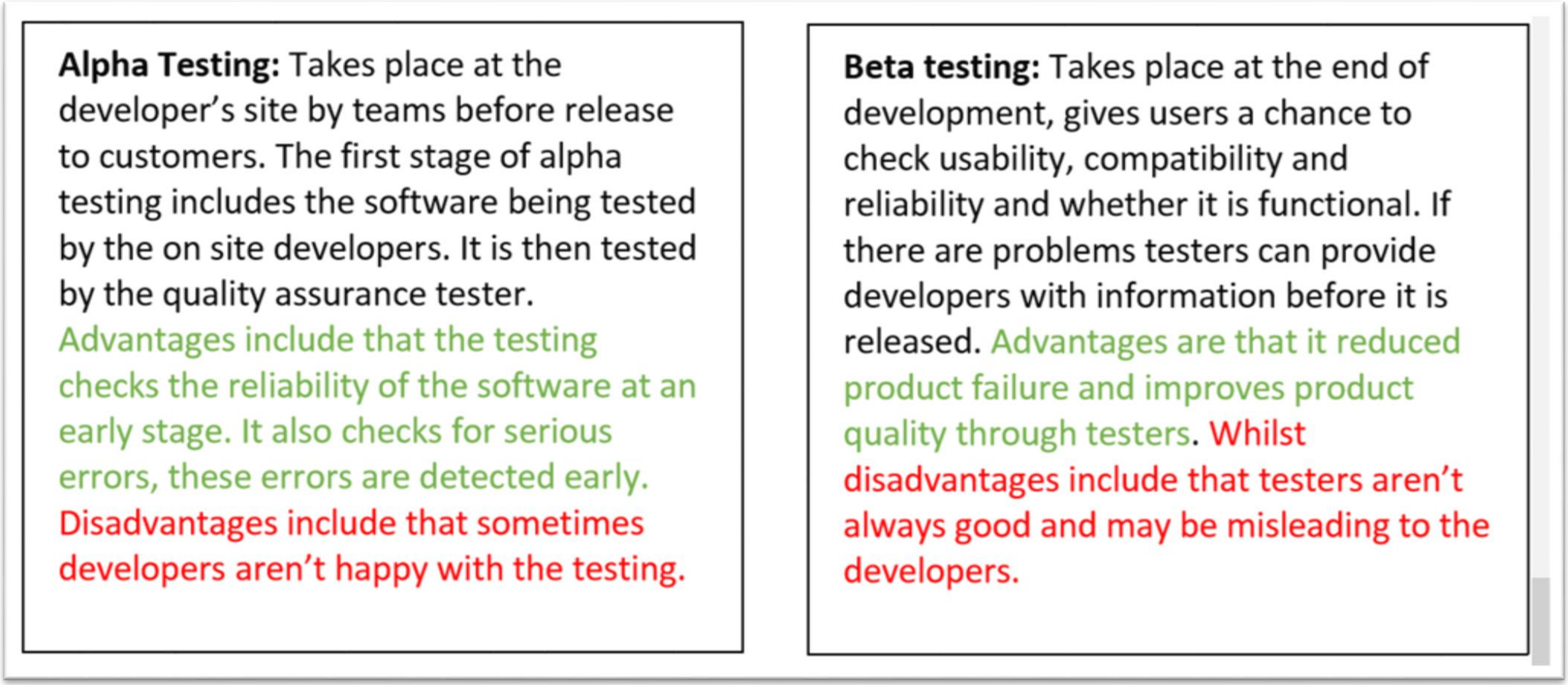
Key question: What are the features of the different ways a program can be tested?



White box testing: Is a testing technique for software that examines the structure of the program. The advantage to white box testing is it reveals errors in hidden code and spots dead code improving efficiency. However disadvantages include that it is expensive to carry out and broken code may be missed. Person must have good knowledge of programming language in order to perform white box testing.

Black box testing: Black box testing is a testing technique that focuses on the input and output of code and less on the internal workings of the code. Advantages of black box testing are that in can be implement on large programs, exposes problems and the tester can be non-technical. Disadvantages include that not all of the code is tested which means the main problem may be missed. It is sometimes difficult to test every single input especially complicated ones.

Key question: What are the features of the different ways a program can be tested?



Alpha Testing: Takes place at the developer's site by teams before release to customers. The first stage of alpha testing includes the software being tested by the on site developers. It is then tested by the quality assurance tester. Advantages include that the testing checks the reliability of the software at an early stage. It also checks for serious errors, these errors are detected early. Disadvantages include that sometimes developers aren't happy with the testing.

Beta testing: Takes place at the end of development, gives users a chance to check usability, compatibility and reliability and whether it is functional. If there are problems testers can provide developers with information before it is released. Advantages are that it reduced product failure and improves product quality through testers. Whilst disadvantages include that testers aren't always good and may be misleading to the developers.

Typical exam questions

1. A software company has been commissioned to produce a new sales solution for a supermarket. This is a major piece of work for the company and the solution will span many stores as well as serving the needs of their head offices. It is estimated the problem with take up to 14 months to develop fully and it is essential it is developed on time, to budget and to the user's requirements.

The software teams first major decision is if they should use the extreme programming, spiral, agile or waterfall methodology for development.

State two methodologies you would recommend and provide a justification of your choices. **[8]**

Methodology 1:

Methodology 2:

Component 1 | 1.2.3| Software development

Target: □ Overall grade: □

Minimum expectations & learning outcomes

□	Terms 75-80 from your A Level Key Terminology should be included and formatted.
□	You must include some illustrations of the various software development methodologies.
□	You must include a comparison (e.g. in a table) of the advantages and disadvantages of the software development methodologies.
□	You must include at least one program you have written with associated pseudocode.
□	You must include a section which outlines the main testing strategies and describes when these test strategies might occur during the software development process.
□	Answer the exam questions.

Feedback

Breadth	Depth	Presentation	Understanding
□ All	□ Analysed	□ Excellent	□ Excellent
□ Most	□ Explained	□ Good	□ Good
□ Some	□ Described	□ Fair	□ Fair
□ Few	□ Identified	□ Poor	□ Poor

Comment & action required

Component 1 | 1.2.3| Software development

Reflection & Revision checklist

Confidence	Clarification
☹ 😐 ☺	Candidates need to understand the following models that can be followed to produce a software system; the waterfall lifecycle, agile methodologies (specifically extreme programming); the spiral model and rapid application development).
☹ 😐 ☺	Candidates need to understand the tasks, processes, benefits and drawbacks of each model and the similarities and differences between each. They need to understand where each model is most suitable to use, and be able to justify the use in a situation.
☹ 😐 ☺	Candidates need to be able to write algorithms using pseudocode and/or program code.
☹ 😐 ☺	Candidates need to be able to follow the code as shown in the OCR pseudocode guide, but are not expected to write code in this syntax. Candidate's code is not expected to be syntactically correct, but must use appropriate code structures.