OCR A Level Computer Science

Your notes

2.3 Software Development

Contents

- Waterfall Lifecycle
- Agile Programming
- Spiral Model
- Rapid Application Development (RAD)
- Comparing Software Development Models

Your notes

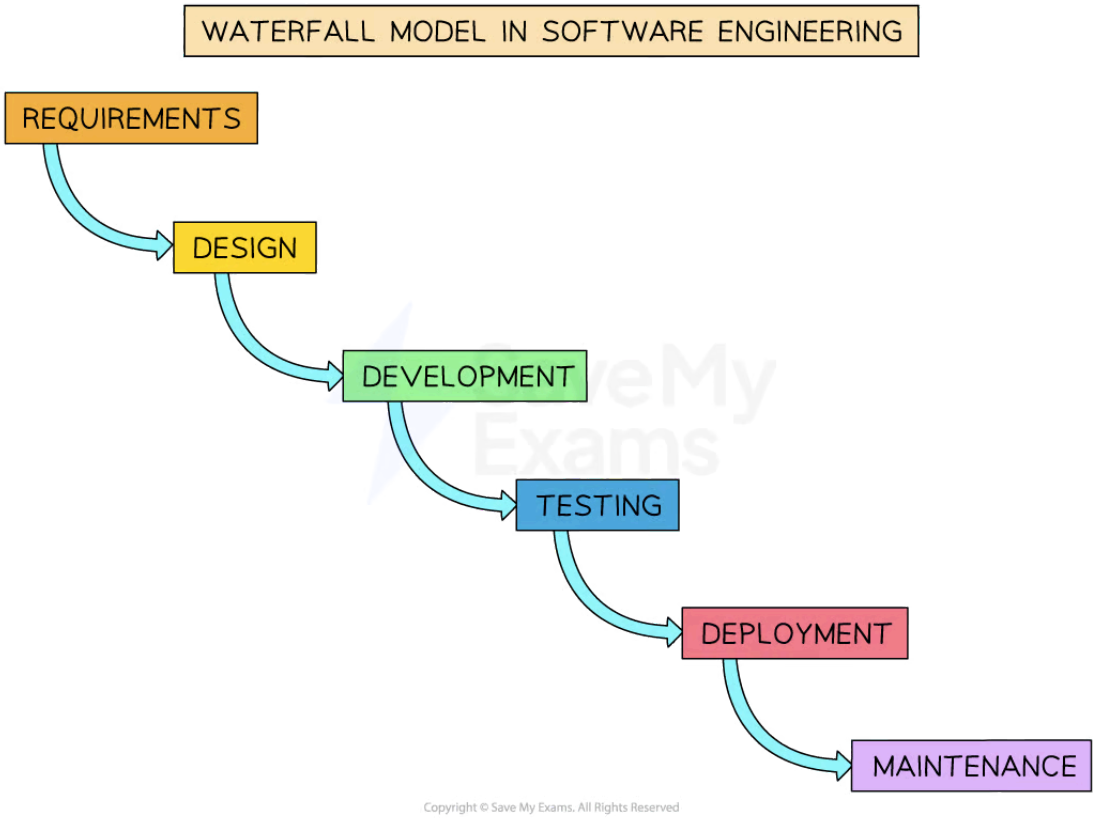
Waterfall Lifecycle

The Waterfall Model is a sequential software development process divided into distinct phases. Each phase must be completed before the next one begins.

Steps in the model:

1. **Requirement Gathering and Analysis:** All possible system requirements to be developed are captured and documented clearly
2. **System Design:** The requirements are translated into a design. Architects and designers define the overall architecture and identify the main components
3. **Implementation:** The actual code is written in this phase based on the design documents, turning the system design into a functional program
4. **Integration and Testing:** All the components and modules are integrated and tested to ensure that the entire system works as expected
5. **Deployment:** The product is released to the market or handed over to the client. It may involve installation, customization, and training
6. **Maintenance:** Post-release, the system needs regular maintenance to fix bugs, improve performance, or add new features

Your notes



The Waterfall Model in Software Engineering

Benefits:

- **Simple and linear:** Easy to understand and follow, with each stage progressing linearly to the next
- **Clear stages and milestones:** Each phase has specific deliverables and milestones, making progress easy to measure
- **Suitable for well-defined projects:** Works best when the requirements are clear and unlikely to change during development

Drawbacks:

- **Inflexible:** Changes are difficult to implement once the project has started, as the model doesn't easily allow for revisiting previous stages
- **Expensive to fix late problems:** If a problem appears later in the development cycle, it can be costly and time-consuming to fix

- **Long development cycle:** The sequential nature may lead to a longer development time, especially if stages are delayed

Suitability:

- **The Waterfall Model is most suitable for projects where requirements are well understood and unlikely to change.** It works well when high quality and compliance are essential, and there is a clear understanding of the project's goals and constraints


Your notes

Your notes

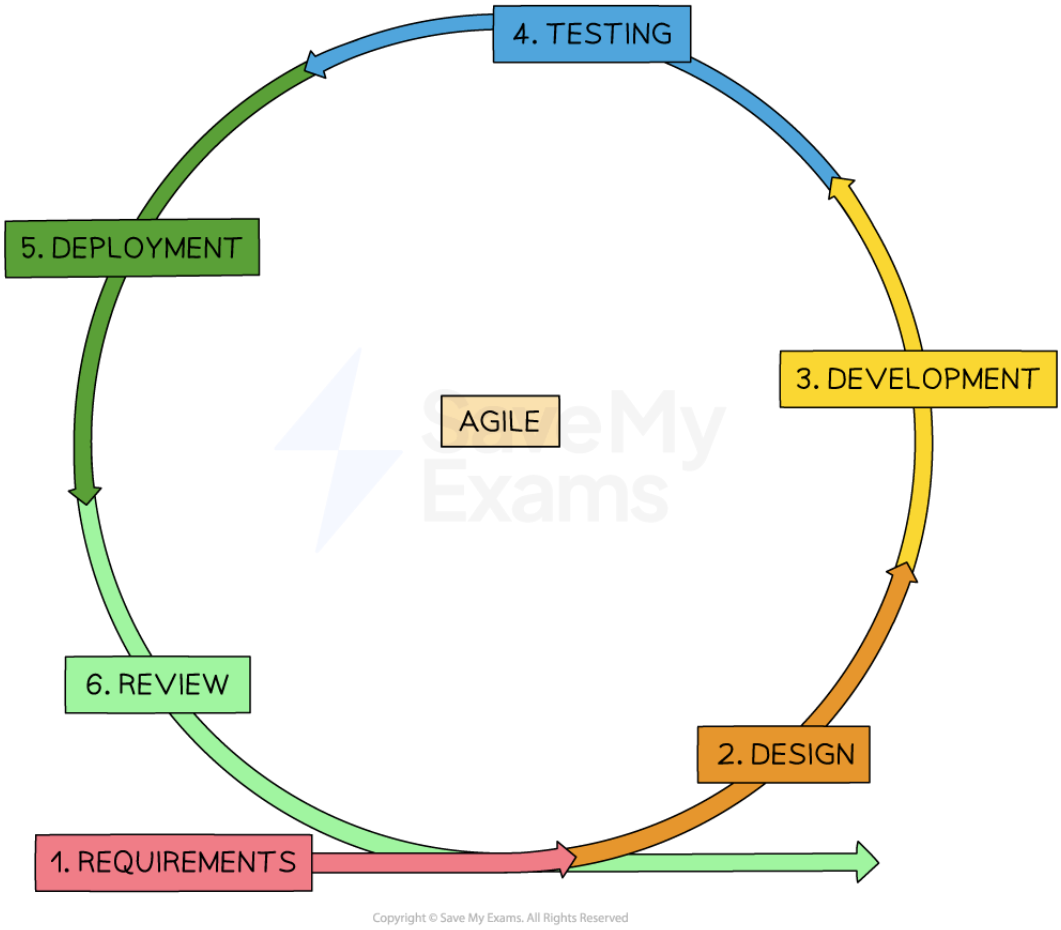
Agile (Extreme Programming)

Extreme Programming (XP) is a type of Agile software development methodology that promotes adaptability and high customer involvement.

Steps in the model:

1. **Planning:** Customers and developers define the scope and priorities, including writing user stories
2. **Design:** Developers create a simple design for the system that can evolve over time
3. **Coding:** Coding is done in small increments with frequent integration and pair programming to ensure quality
4. **Testing:** Continuous testing throughout the coding process, with automated tests developed before or alongside the code
5. **Integration:** Regular and frequent integration of code, with continuous integration tools ensuring that the system works as a whole
6. **Feedback and Iteration:** Constant feedback from stakeholders is used to guide future development cycles, leading to iterative and incremental development

Your notes



The Extreme Programming (Agile) Model

Benefits:

- **Highly adaptable:** Can quickly respond to changes in requirements, even late in development
- **Frequent communication:** Encourages constant communication between team members and stakeholders
- **Quality focus:** Emphasizes technical excellence and good design, with continuous testing throughout the development cycle
- **Customer collaboration:** Encourages working closely with customers to ensure the developed product meets their needs

Drawbacks:

- **Requires experienced team members:** Can be challenging to implement without knowledgeable and skilled developers
- **Intensive collaboration can lead to burnout:** The constant communication and collaboration can be tiring for team members
- **May lack documentation:** The focus on adaptability and immediate coding may result in insufficient documentation
- **Scope creep:** The flexible nature may lead to uncontrolled changes in requirements


Your notes

Suitability:

- **Extreme Programming is most suitable for small to medium-sized projects** where requirements can change and customer involvement is high

Your notes

Spiral Model

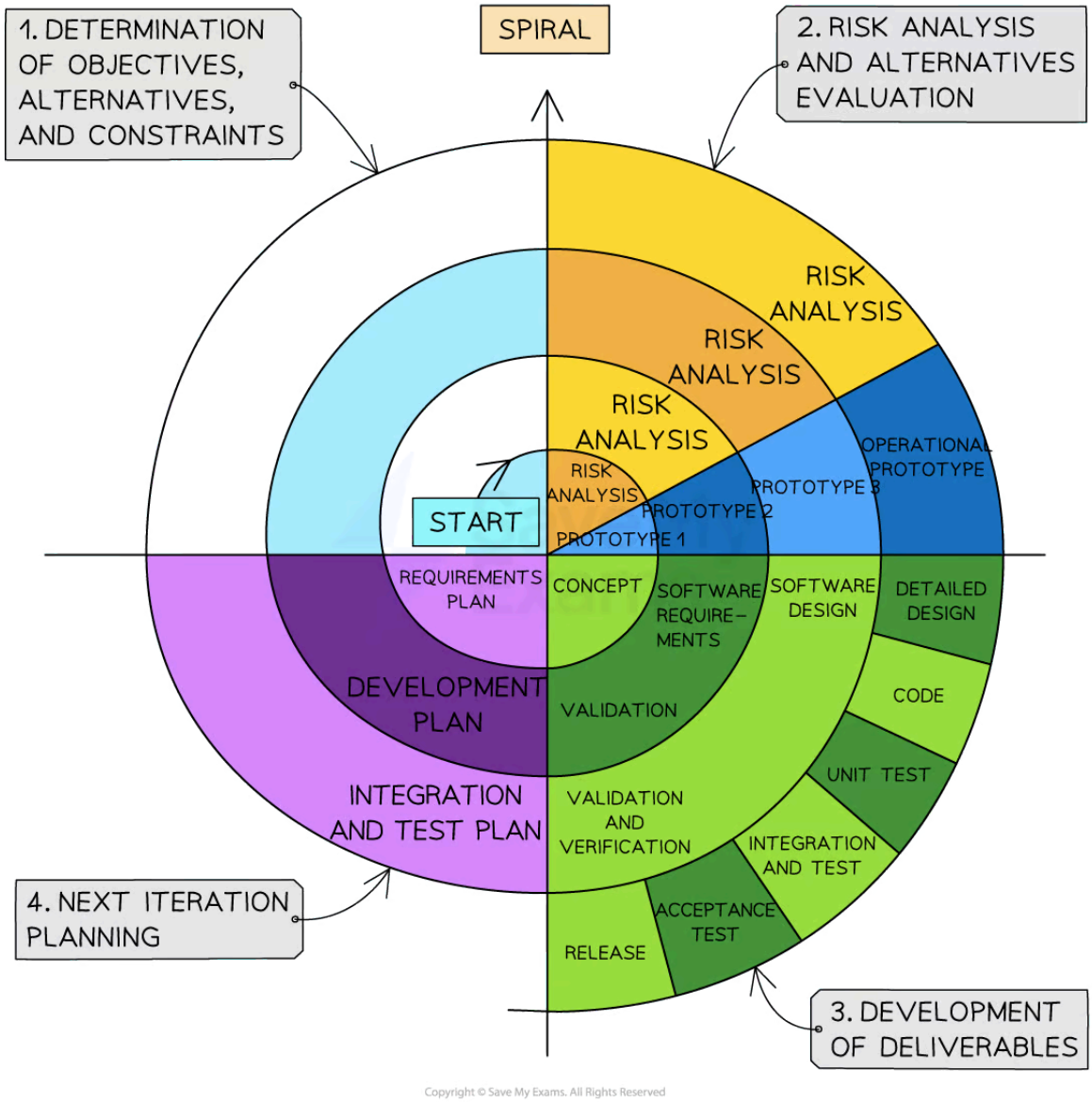
The Spiral Model is a software development methodology that combines aspects of both iterative (agile) and sequential (waterfall) processes.

Steps in the model:

1. **Planning:** Define the objectives, alternatives, and constraints for the current phase of the project
2. **Risk analysis:** Identify and assess potential risks, and plan mitigation strategies
3. **Engineering:** Develop the next version of the product, including design, coding, testing, and integration
4. **Evaluation and feedback:** Review the progress with stakeholders, and plan the next iteration

The process repeats, spiralling through these stages, with each spiral loop representing a development phase until the final product is ready.

Your notes



The Spiral Model of Software Development

Benefits:

- **Flexibility:** Allows for changes and adaptations at various stages of development
- **Risk management:** Emphasizes risk assessment and mitigation, helping to identify and address issues early

- **Strong customer involvement:** Encourages feedback and input from clients throughout the development process
- **Incremental releases:** Provides early partial working solutions, enabling early usage and feedback


Your notes

Drawbacks:

- **Complexity:** Can be more complex and harder to manage compared to other methodologies
- **Time-consuming:** The emphasis on planning, risk management, and iterations may lead to a longer development process
- **Expensive:** Often requires more resources, particularly in risk assessment and iterative design
- **Not suitable for small projects:** The extensive planning and risk management might be overkill for simple or small-scale projects

Suitability:

- **The Spiral Model is most suitable for large, complex projects** where requirements may change, and risk management is essential

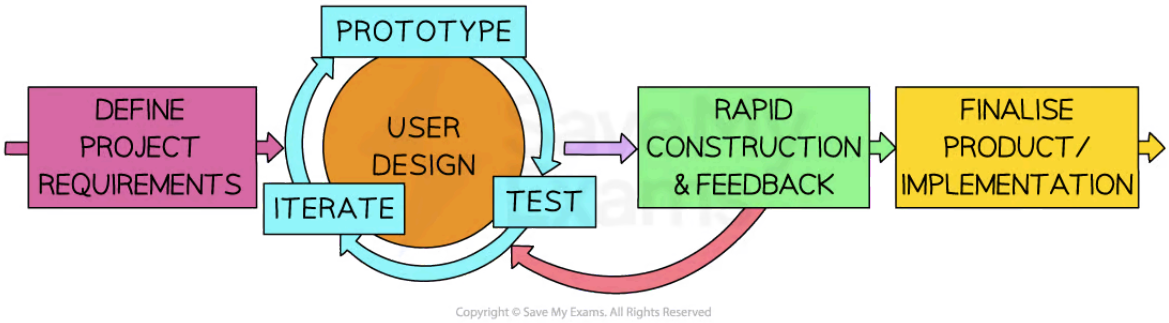
Your notes

Rapid Application Development (RAD)

- Rapid Application Development (RAD) is a software development methodology that emphasises fast and iterative development

Steps in the model:

1. **Requirement planning:** Gather general system requirements, define constraints and assumptions
2. **User design and prototyping:** Collaborate with users to develop prototypes, ensuring alignment with user needs
3. **Construction or iterative development:** Build the system incrementally, with continuous user feedback and adaptation
4. **Cutover or deployment:** Transition the product into the live environment, including user training, support, and documentation
5. **Maintenance and updates:** Continue to adapt and improve the system based on user feedback and needs



Rapid Application Development (RAD) Model of Software Development

Benefits:

- **Speed:** Enables rapid development and delivery of a high-quality system at a relatively low investment cost
- **User involvement:** Clients are involved throughout the development process, ensuring that the system aligns with user needs and expectations
- **Flexibility:** Allows for changes and adaptations to be made quickly as requirements evolve

- **Incremental development:** Promotes development in small increments, with constant feedback and adaptation

Drawbacks:

- **Dependent on strong team collaboration:** Requires skilled and collaborative team members, which can be a challenge to assemble
- **Potential lack of quality:** The focus on speed might lead to skipping rigorous testing or documentation, impacting the quality
- **Not suitable for small projects:** The intense collaboration and iterative approach might be overkill for simple or small-scale projects
- **Can lead to scope creep:** The flexible nature may lead to uncontrolled changes in requirements

Suitability:

- Rapid Application Development is most suitable for projects where **rapid delivery is required** and where requirements can be developed and refined **on the go**


Your notes

Your notes

Comparing Software Development Models

- You need to be able to compare software development models, and be able to recommend a model for a given scenario
- The below table summarises some of the key processes, benefits, drawbacks, and suitability of each

Method	Tasks and Processes	Benefits	Drawbacks	Suitability
Waterfall Lifecycle	Linear stages: Requirements, Design, Coding, Testing, Deployment, Maintenance	Simple, clear stages, well-defined projects	Inflexible, expensive late changes, long development cycle	Well-defined projects with stable requirements
Agile (XP)	Iterative: Planning, Designing, Coding, Testing, Reviewing	Flexible, constant customer involvement, rapid delivery	Scope creep, collaboration dependency, not for large projects	Fast-paced projects with changing requirements
Spiral Model	Cyclic: Planning, Risk Analysis, Engineering, Evaluation	Risk management focus, tailored customer feedback	Complex, expensive, expert risk assessment required	Complex projects needing extensive risk management
Rapid Application Development (RAD)	Phases: Requirements Planning, User Design, Construction, Cutover	Faster development, user feedback, adaptable	Collaboration dependence, less scalable, quality issues	Projects requiring quick delivery with active user participation

WORKED EXAMPLE

The software team that produces De-Duplicator is adding a new feature that can detect duplicated images the previous version could not. The software team must decide which methodology they will use for the project. Some members of the group suggest extreme

programming, whilst others would prefer to use the waterfall lifecycle. Discuss the two methodologies and justify which you would recommend.

[12]

Your notes

How to answer this question:

- Introduce both Extreme Programming (XP) and the Waterfall Lifecycle, highlighting key features
- Compare and contrast the two methodologies, explaining how they are similar and different
- Identify specifics from the scenario - the software is receiving a new feature, and there is a team of people on the project
- Recommend one methodology based on the given context, providing justification

Answer:

Answer that gets full marks:

The software team for De-Duplicator is facing a choice between two methods to develop a new version of the software: Extreme Programming (XP) and the Waterfall Lifecycle.

XP is like building something with the ability to keep changing and improving it as you go. It's flexible and allows for continuous feedback and changes. This method works well when you need to keep adjusting things during the project. In the case of the De-Duplicator project, where the team is adding a new feature, this flexibility could be a real advantage so the team can discover the best way to implement it.

On the other hand, the Waterfall model is more like following a strict step-by-step guide. It's organised but not very flexible if you need to make changes later on. While this structure can be good for some projects, it could make it tough for the De-Duplicator team if they uncover problems in the software while adding the new feature.

Given what the De-Duplicator project needs, XP seems to be the better choice. Its flexibility and ability to adapt to changes align well with a project that might require ongoing improvements or customer feedback loops. The Waterfall model's lack of flexibility could make things difficult if requirements change. So, Extreme Programming would likely be the recommended choice for this project, letting the team keep things flexible and continuously aligned with what users need.