A) software process is a structured set of activities, tasks and steps followed to devlop, deliver and maintain software in a systamatic and efficient manner.
The differene is that a process defines what needs to be done and in what order, while a methodology defines how it will be done using specific practices, tools, and techniques.
If a software development house does not follow any systematic process, it may face problems such as inconsistent quality, poor communication, difficutly in tracking progress, delays, budget overruns, lack of documentation, difficulty in maintaince, and higher risk of project failure.

The major phases in waterfall model are requirement analysis, system design, implementation (coding), integration and testing, deployment, and maintenance. Among these, the maintenance phas usally consumes the maximum effort, as the software often requires updates, bug fixes, and enhancements after delivery.

a) Water fall model
Reason: Requirements are clear, stable and easy to define up

b) Spiral model
Reason: High risk and lack of experience require iterative risk analysis and prototyping.

c) V-model.

d) Incremental model
Reason: Can deliver in stages, adding features gradually while users start using parts of the system early.

e) Spiral model
Reason: Extremely large, complex and high risk system needs continous risk management and iterative development.

f) Prototying model
Reason: User interface and usability need to be refined through prototypes based on user feedback.

g) Waterfall model
Reason: Requirements and funtionality are well-defined from the start. changes are minimal.

h) Prototyping model
Reason: GUI design depends heavily on user feedback and requires multiple re-finements.

4. A prototype is an early, simplified version of a software product created to demonstrate its features, design, or functionality before full development. It is especially beneficial when requirements are unclear, the user interface is important, the technology is new, or user feedback is essential. While building a prototype adds initial efforts, it does not always increase the overall cost, as it can prevent expensive changes later by clarifying requirements early. The main advantage of constructing a prototype first include refining requirements, obtaining early user feedback, identifying flaws early, reducing risks, and improving communication between developers and clients.

5. The spiral life cycle is considered a meta-model because it incorporates features of multiple traditional life cycle models and organises them in a risk driven, iterative framework. It does not prescribe a fixed sequence of activities; instead, each iteration, or spiral, can adopt the most suitable model for that phase of development. This flexibility allow the spiral model to act as a framework within which other models are applied based on projects needs and risk analysis.

6. The spiral model is suitable for large, complex, and high risk projects using new technology, and projects with unclear or changing requirements. The no. of loops is not fixed, it depends on project size, complexity, and risks. Each loop represents one development phase, and more loops are needed for complex projects to refine requirements, reduce risks, and build system gradually.

7. Iterative Waterfall Model- Advantages: Simple, easy to manage, clear stages and deliverables, best for stable requirements.
Ex: Payroll system, compiler.
Spiral model- Advantages: Handles high risk, supports changing requirements, combines multiple models, focuses on risk analysis.
Ex: Satellite communication software, ERP, AI navigation.

When to use :
Waterfall : Small, well-defined, low-risk projects.
Spiral : Large, complex, high-risk, evolving-requirement projects.

8. The evolution life cycle model is more suitable for products that can be developed and delivered in incremental versions, where user feedback shapes future releases.
Ex: Word processors, mobile apps and e-commerce websites - features are delivered in stages, refined over time.
The spiral model is more suitable for large, complex, and high risk systems requiring continous risk assessment and prototyping.
Ex: Air traffic control systems, satellite communication software, and large defense systems - where failure costs are high and requirements evolve.

9. Irrespective of the life cycle model used, final documentation are prepared as if the product were developed using the classical waterfall model because documentation needs to be clear, structured, and easy to ~~use~~ follow. The waterfall sequence - requirements, design, implementation, testing and maintaince provides a logical order that helps in standardization, Compliance, maintenance, Review and verification.
This makes the documentation usable even if the actual development followed Agile, Spiral, or any other model.

1a For a project where the costumer is likely to change requirements frequently, the spiral model (or Agile model) is most suitable, because

i) Supports frequent changes by allowing requirement refinement in each iteration.

ii) Risk analysis at every loop reduces chances of failure,

iii) Continous costomer involvement ensures the project evolves according to updated needs.

iv) Prototyping helps validate changes before full implementation.