ECE 811 – SOFTWARE ENGINEERING

WATERFALL MODEL FOR SOFTWARE DEVELOPMENT- TEST YOUR KNOWLEDGE

SECTION A: MULTIPLE CHOICE

1. **The Waterfall Model is characterized by:**
   a) Parallel development phases
   b) Iterative cycles
   c) Sequential, non-overlapping phases
   d) Continuous customer collaboration
   Answer: c) Sequential, non-overlapping phases
   Explanation: Waterfall requires each phase (e.g., Requirements, Design) to be fully completed before the next begins, with no overlap.

2. **In which phase are user needs formally documented?**
   a) Design
   b) Implementation
   c) Requirements Gathering
   d) Testing
   Answer: c) Requirements Gathering
   Explanation: This phase produces the Software Requirements Specification (SRS), capturing all functional/non-functional needs.

3. **Maintenance in Waterfall typically involves:**
   a) Rewriting the entire system
   b) Adding major new features
   c) Bug fixes and minor enhancements
   d) Skipping documentation updates
   Answer: c) Bug fixes and minor enhancements
   Explanation: Maintenance addresses post-deployment issues but avoids major scope changes to preserve structure.

4. **Waterfall is MOST suitable when:**
   a) Requirements are ambiguous
   b) Technology is experimental
   c) Regulatory documentation is essential
   d) User feedback is needed biweekly
   Answer: c) Regulatory documentation is essential
   Explanation: Industries like healthcare or aerospace value Waterfall's rigorous phase documentation for compliance.

5. **A key risk of the Waterfall Model is:**
   a) Early prototype delivery
   b) High customer involvement
   c) Late discovery of design flaws
   d) Frequent requirement changes
   Answer: c) Late discovery of design flaws

Explanation: Testing occurs after implementation, so critical errors may only surface late, causing expensive rework.

6. **The Design phase directly follows which phase?**
   a) Testing
   b) Requirements
   c) Implementation
   d) Deployment
   Answer: b) Requirements
   Explanation: The sequence is Requirements → Design → Implementation → Testing → Deployment → Maintenance.

7. **Testing in Waterfall is typically:**
   a) Conducted alongside coding
   b) Performed by developers during implementation
   c) A separate phase after coding
   d) Skipped for small projects
   Answer: c) A separate phase after coding
   Explanation: System testing only begins once implementation is fully complete.

Section B: True/False (5 Questions)

8. **Waterfall allows revisiting previous phases easily.**
   Answer: False
   Explanation: Returning to earlier phases (e.g., changing requirements post-design) is costly and discouraged.

9. **The V-Model is a variant of Waterfall that emphasizes testing alignment.**
   Answer: True
   Explanation: The V-Model maps testing phases (e.g., unit, system tests) to corresponding design/requirements stages.

10. **Waterfall works well for Agile projects.**
    Answer: False
    Explanation: Agile requires iterative development and flexibility, opposite to Waterfall's rigid structure.

SECTION C: SHORT ANSWER

11. **Name two deliverables of the Requirements phase.**
    Answer: Software Requirements Specification (SRS), Use Cases.
    Explanation: The SRS details functional/non-functional needs; use cases describe user interactions.

12. **Why is Waterfall criticized for customer satisfaction?**
    Answer: Customers see working software only at the end, with no mid-process feedback opportunities.
    Explanation: Late delivery increases the risk of unmet expectations.

SECTION D: SCENARIO-BASED

23. **A project requires FDA approval for medical device software. Why might Waterfall be chosen?**
    Answer: Its phase-gated documentation provides auditable records for regulatory compliance.
    Explanation: Regulators require traceable proof of requirements/design/testing rigor—Waterfall's strength.

SECTION E: CRITICAL THINKING

28. **How does Waterfall's "Big Design Upfront" (BDUF) approach become a liability in innovative projects?**
    Answer: BDUF assumes all requirements are predictable. In innovation, needs evolve with prototyping/user feedback, causing redesign delays.
    Explanation: Example: AI projects often pivot based on early data—Waterfall can't accommodate this flexibly.