1.2 SOFTWARE & SOFTWARE DEVELOPMENT · 1.2.3

Software development methodologies

Waterfall, agile, extreme programming, spiral and RAD: what each is and when to use it. Spec 1.2.3(a)(b).

01 The methodologies

Waterfall Sequential stages, each finished before the next. Fixed requirements.

Agile Iterative cycles with user feedback. Changing requirements.

XP Agile + pair programming, frequent releases, test-driven, on-site customer.

Spiral Iterative prototypes with risk analysis. Large/high-risk.

RAD Fast prototyping & user feedback. Speed-focused.

02 When to use which

Fixed reqs Waterfall — clear plan, good documentation.

Changing reqs Agile / XP — adapt each cycle.

Large & risky Spiral — manage risk per loop.

Need speed RAD — quick usable prototypes.

Always Justify the choice for the scenario.

03 Waterfall vs agile

Waterfall

One sequence. Well documented, simple to manage. Hard/costly to change; customer sees software late.

Agile

Short repeating cycles, working software each time. Adapts to change; needs close customer involvement, less documentation.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Seven slips on methodology questions.

01

No methodology is "best" — match it to the project. Waterfall suits **fixed** requirements.

02

Describe XP with **specifics**: pair programming, frequent releases, test-driven, on-site customer.

03

Spiral = iterative + **risk analysis**; **waterfall** = a single sequence. Don't confuse them.

04

Agile/XP win when requirements **change** and feedback must be frequent.

05

"Discuss/compare" needs **merits and drawbacks** of each, not just descriptions.

06

Waterfall's weakness: customer sees working software **late**, so errors surface late.

07

Always end an extended answer with a **justified recommendation for the given scenario** — that is where the top band marks are.