Exercise 10

1.

Task \ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Requirements & Specs	X	X													
UX/UI Design		X	X	X											
AR Prototype			X	X	X	X									
Tetris Mechanics				X	X	X	X	X							
AR Integration						X	X	X	X	X					
Testing & QA									X	X	X	X			
Launch Prep & Marketing										X	X	X	X		
Beta Release												X	X		
Deployment & Release														X	

2.
Personnel: 5 devs x 15 wks x 8k euros = 600k
1 UX/UI designer: 6wks x 5k = 30k
2 Test engineers x 6 wks x 5k = 48k
AR license = 20 k
Hardware : 15k
Infrastracture : cloud build/test servers.. = 15k
Marketing : 50k
total = 758k
—-------- per-unit license of €30 for approx. ~30 k users
—-------- one-time app fee in a consumer store could be €5–10

3.

To compress the schedule:

Cross-functional “squads”: Pair each AR dev with a Tetris-engine dev to parallelize prototype → integration.

Dedicated QA Automation engineer from week 4, building test harnesses as features land.

UX/UI designer embedded with devs (no handoff delays).

1 Project Manager/Scrum Master to unblock dependencies in real-time.

DevOps engineer early (wk 1–3) to automate CI/CD, device provisioning, and builds.

4.

Hybrid Agile (Scrum + Kanban):

Scrum sprints of 2 weeks for feature planning, demo

Kanban for bugfixes/AR SDK issues to handle unpredictable AR integration challenges.

Continuous Integration/Delivery: Every commit triggers build + unit tests; feature branches merged only on passing CI.

Why?

AR integration is research-heavy and benefits from Kanban's flexibility.

Scrum's sprint demos keep stakeholders engaged.

Hybrid ensures adaptability without losing cadence.

5.
– Ship core Tetris play in 2D, add AR overlay later as "Phase 2."
– Hand off AR rendering or QA automation to a specialized vendor to regain schedule.
– Delay non-critical "social sharing" or "skins store" to post-launch.
– Negotiate with stakeholders for additional funds tied to clear, smaller deliverables.
– Hire contract developers/testers to blitz through current backlog and stabilize.