Methodologies:

1. GDD as documentation
2. Agile-like
    a. Less documentation, more user involvement
    b. Scrum
        i. For game mechanics and features (programming part)
        ii. Periodically, have prototypes, unpredictable (but goal must be clear)
        iii. May have informal Sprint Reviews (stakeholders rarely come)
        iv. If Scrum is used in design too, artists and devs may become independent
        v. Definition of Done not used (because the goal is “fun” – that is not correct definition)
        vi. Retrospectives may be used or not
        vii. In small studios team lead may be SM
        viii. Often there is producer (or game designer) that fulfills some of PO’s and SM’s duties
        ix. Team lead also gets some of SM’s responsibilities
        x. Teams not understanding Scrum enough – so there is “cargo cult”
        xi. ABC sprints
        xii. Scrum of scrums
    c. Kanban
        i. For testing
        ii. For asset creation (creative part)
        iii. Arts have many stages
        iv. Predictable, many little tasks
        v. From “Is Agile not agile enough?” [2021] not used in a good way because of lack of knowledge
    d. Scrumban
    e. ScrumBut
    f. Water-Scrum-fall
    g. RAD
    h. XP
    i. Ad-hoc (developer-driven ?)
    j. FDD (feature driven)
3. Traditional
    a. Waterfall
    b. Iterative
    c. Spiral
    d. Evolutionary
    e. Incremental / Staged delivery
4. Hybrid
    a. CBGD (component based)
    b. Modular
    c. Empirical ? (see Software Engineering Research, Management and Applications, 2006. Fourth International Conference on, pp. 371-377)
    d. Generic (gameplay loop formal language)
    e. Reuse
    f. IEEE SS&E
    g. ETVX
    h. SDLC

i. MDD (model driven)
j. RUP
k. V model
l. Lean
m. Game waterfall

Stages:

1. Simple
    a. Concept
        i. Fuzzy project vision
        ii. Even if Scrum is used, it's in ad-hoc style
            1. Short sprints, less defined goals, rapidly changing backlog
        iii. Kanban may be used
            1. General milestones as goals for prototyping, poor risk and time estimation
        iv. Maybe better use ad-hoc, RAD, incremental?
    b. Pre-production
        i. Validating and iterating on prototypes
    c. Production
        i. Clear project vision
        ii. Programming using Scrum
            1. With proper testing
        iii. Arts using Kanban
    d. Post-production
        i. Testing, marketing, maintaining, updating
        ii. Good with Scrum (almost fixed project vision)

Problems:

1. Maintaining clear project vision
2. Scope management problems (feature cutting, feature creep)
3. Multidisciplinary team dynamics
4. Schedule problems
5. Poor work culture

Agile adoption challenges:

1. Stakeholders must respect the process
2. Development team must have more autonomy
3. Employees need experience with agile
4. In post-production (or in live games – game-as-a-service) arts and programming can be meshed together, with Scrumban
5. If studios use Agile under pressure or because of its popularity, they not understand its logic and use it in a wrong way, so they get communication problems
6. Agile alone is not enough for multidisciplinary team
7. There should be one person in PO role (like game designer, with single project vision) or frequent meeting for understanding of common project vision
8.

Examples:

1. +, Scrum with collocation practices of XP (less communication problems)
    a. May be good to use seating arrangement with artists and programmers together
2. -, ScrumBut (no understanding of principles)
3. +, Scrumban (arts and programming mixed)
4. +-, There may be some people with common game vision – in role of PO (2 designers)