Scrum

- Use when: Project requirements are likely to evolve, but you need structure, clear roles, and predictable delivery cycles. Best for software development, SaaS, and startups with small to medium teams.
- Avoid if: The project scope changes daily, or you have very large teams or require heavy documentation.

Kanban

- Use when: You need maximum flexibility, continuous delivery, and visual workflow management. Ideal for support, maintenance, or operational teams where priorities shift rapidly.
- Avoid if: You need strict deadlines, defined roles, or work in a highly regulated environment that requires detailed documentation.

Extreme Programming (XP)

- Use when: Code quality is paramount, requirements change frequently, and you have a small, highly technical team. XP is excellent for projects needing frequent releases and close customer collaboration.
- Avoid if: The team is large, non-technical, or the customer cannot be closely involved.

Adaptive Software Development (ASD)

- Use when: The project is innovative, requirements are unclear or likely to change, and rapid adaptation is needed. ASD suits small, experimental teams[7].
- Avoid if: The project is large, highly structured, or requires extensive documentation.

DSDM

- Use when: You need strong business alignment, active user involvement, and can invest in upfront planning and documentation. DSDM is suitable for large, complex projects in regulated industries.
- Avoid if: The team is small, the project is fast-paced, or you want minimal documentation.

Advantages and Disadvantages

Scrum

Advantages:

- High communication and team synergy
- Predictable delivery with sprints
- Continuous customer feedback

Disadvantages:

- Can be rigid in dynamic environments
- Requires experienced teams and committed roles
- Poor documentation and less control for product owners[711]

Kanban

Advantages:

- Highly flexible and easy to implement
- Visual workflow management
- Continuous delivery and optimized workload[41511]

Disadvantages:

- No deadlines or timeframes can slow progress
- Lack of defined roles may cause ambiguity
- Scaling can be challenging[1511]

Extreme Programming (XP)

Advantages:

- High code quality and frequent feedback
- Deep customer involvement
- Emphasizes improvement and adaptability[472]

Disadvantages:

- Requires constant customer availability
- Not suitable for large or non-technical teams
- Overhead for non-technical environments[72]

Adaptive Software Development (ASD)

Advantages:

- Highly adaptive to change
- Promotes learning and innovation
- Lightweight process[78]

Disadvantages:

- Less structure may not suit all teams
- Limited suitability for large or complex projects[78]

DSDM

Advantages:

- Strong business alignment and rapid deliverables
- Active user involvement
- Comprehensive framework for large teams[193]

Disadvantages:

- Heavy documentation and planning
- Can be rigid and resource-intensive
- Less suitable for small teams or fast-moving projects[193]

Summary Table

Model	Best For	Key Advantages	Main Disadvantages
Scrum	Small/medium teams, evolving products	Structure, feedback, predictability	Rigid, needs experience, minimal docs
Kanban	Support/ops, changing priorities	Flexibility, visual, continuous	No deadlines, role ambiguity
XP	Small, technical teams, frequent change	Code quality, feedback	Needs customer, not for large teams
ASD	Small, innovative projects	Adaptability, learning	Less structure, not for large/comple
DSDM	Large, business-critical projects	Business alignment, user involvement	Heavy docs, resource-intensive

Conclusion

Each Agile model offers unique strengths and is best suited to specific project types, team structures, and organizational needs. Selecting the right framework depends on your priorities: structure and predictability (Scrum), flexibility and flow (Kanban), code quality and feedback (XP), adaptability (ASD), or business alignment and rigor (DSDM)[412].