Scrum,ASD,DSDM

Scrum

scrum

- The scrum team is self-organizing in that there is no overall team leader who decides which person will do which task or how a problem will be solved.
- Scrum teams are supported by two specific roles.
- The first is a ScrumMaster, who can be thought of as a coach for the team, helping team members use the Scrum process to perform at the highest level.
- The product owner (PO) is the other role, and in Scrum software development, represents the business, customers or users, and guides the team toward building the right product.

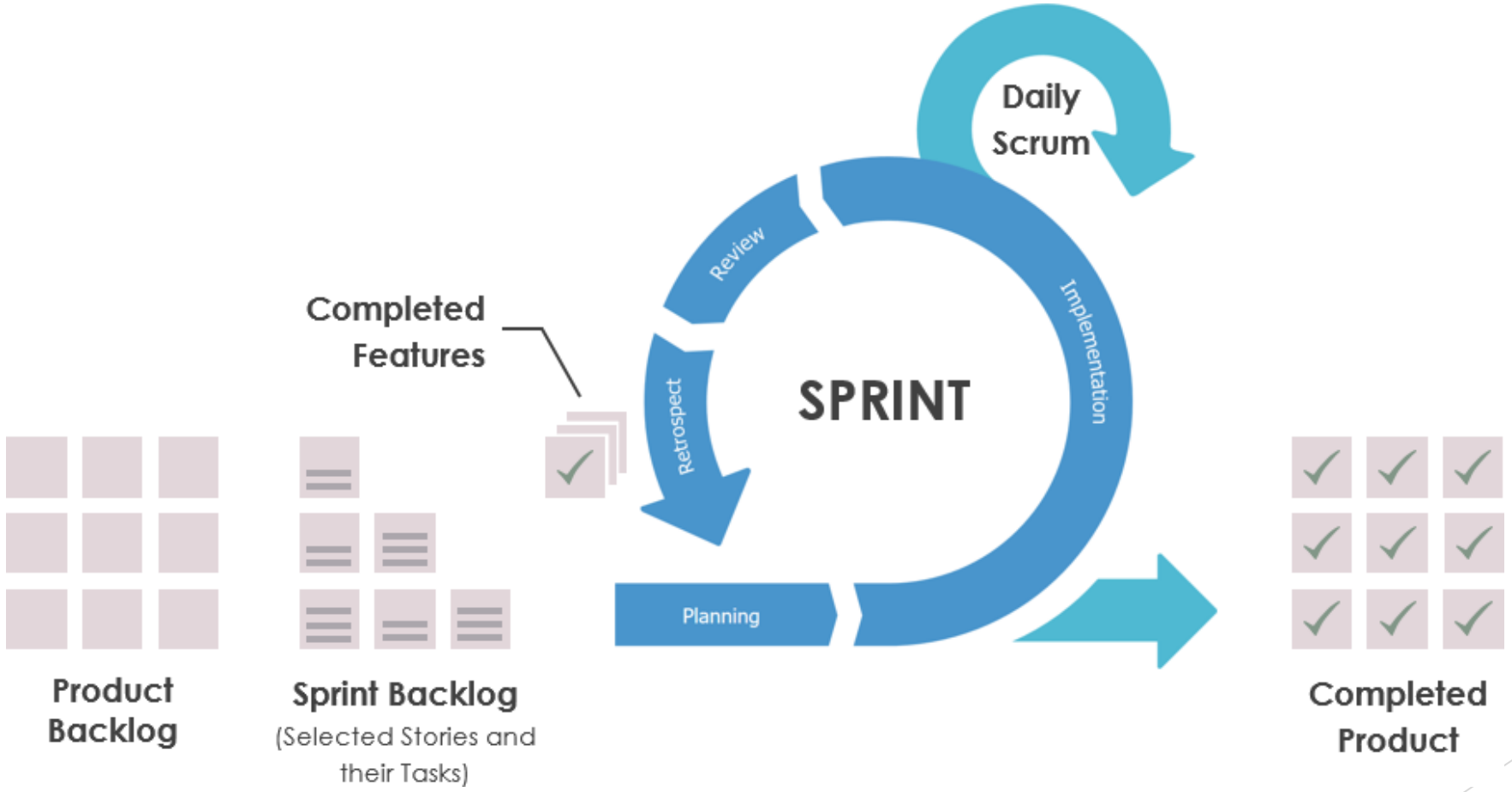
Scrum

- In the scrum model, a project is divided into small parts of work that can be incrementally developed and delivered over time boxes that are called sprints.
- The software therefore gets developed over a series of manageable chunks.(sprints)
- Each sprint typically takes only a couple of weeks to complete.
- At the end of each sprint, stakeholders and team members meet to assess the progress made.
- The stakeholders suggest to the development team any changes needed to features that have already been developed and any overall improvements that they might feel necessary.

sprint

- Scrum projects are broken down into small and consistent time intervals referred to as sprints.
- Sprint is one timeboxed iteration of a continuous development cycle.
- Within a Sprint, planned amount of work has to be completed by the team and made ready for review.
- They can be as short as a few days and generally are no longer than 3 – 4 weeks.

sprint


Daily
Scrum
Review
Implementation
Completed
Features
Retrospect
SPRINT
Planning
Product
Backlog
Sprint Backlog
(Selected Stories and
their Tasks)
Completed
Product

- During an agile Scrum sprint, the Scrum team takes a small set of features from idea to coded and tested functionality. At the end, these features are done, meaning coded, tested and integrated into the evolving product or system.
- On each day of the sprint, all team members should attend a daily Scrum meeting, including the ScrumMaster and the product owner.
- This meeting is timeboxed to no more than 15 minutes. During that time, team members share what they worked on the prior day, will work on that day, and identify any impediments to progress.
- The Scrum model sees daily scrums as a way to synchronize the work of team members as they discuss the work of the sprint.

Main Roles

- ScrumMaster: The ScrumMaster is the team's coach, and helps Scrum practitioners achieve their highest level of performance.
- In the Scrum process, a ScrumMaster differs from a traditional project manager in many ways, including that this role does not provide day-to-day direction to the team and does not assign tasks to individuals.
- The product owner is responsible for prioritizing the backlog during Scrum development, to ensure it’s up to par as more is learned about the system being built, its users, the team and so on.
- The third and final role in Scrum project management is the Scrum team itself.
- Although individuals may join the team with various job titles, in Scrum, those titles are insignificant. Scrum methodology states that each person contributes in whatever way they can to complete the work of each sprint.

Advantageous

1. Flexibility and Adaptablity
2. Minimum time to market
3. Lower costs.
4. Customer satisfaction.
5. Provides continuous feedvback
6. Generate high quality software.

- Disavantageous

1. Training and skill required
2. Organizational transformation
3. Scalability

Adaptive Software Development (ASD)

- Adaptive Software Development (ASD) is a direct outgrowth of an earlier agile framework Rapid Application Development (RAD).

- It aims to enable teams to quickly and effectively adapt to changing requirements or market needs by evolving their products with lightweight planning and continuous learning.

- The ASD approach encourages teams to develop according to a three-phase process: speculate, collaborate, learn.

- Project managers John Highsmith and Sam Bayer are credited with inventing the Adaptive Software Development methodology in the early 1990s
- They developed ASD as a more iterative and shorter-interval version of the Rapid Application Development (RAD) agile framework.

- ASD's strengths include:

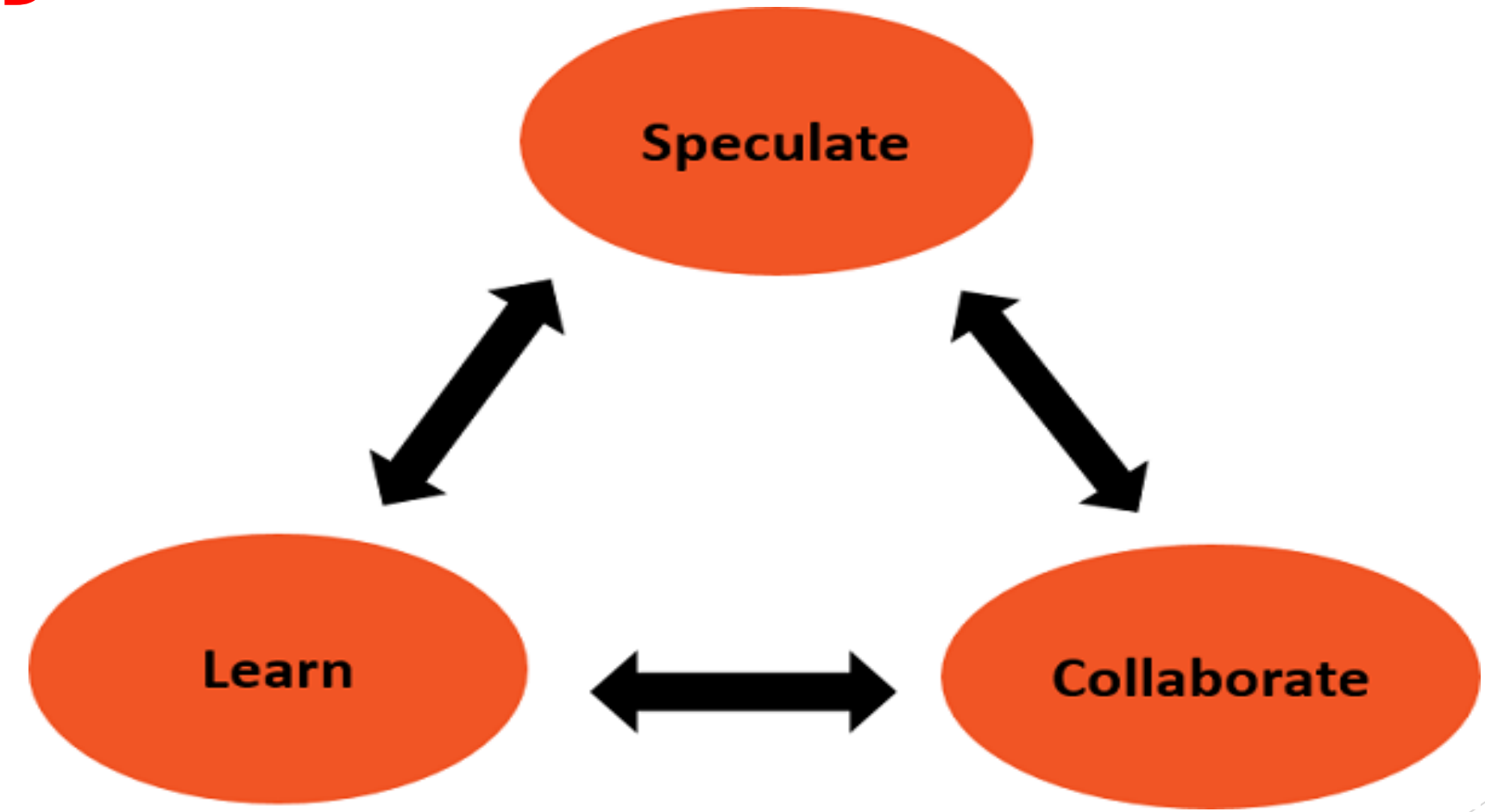
1. Focused on the end users, which can lead to better and more intuitive products
2. Allows for on-time and even early delivery
3. Encourages more transparency between developers and clients

- Adaptive Software Development is cyclical like the Evolutionary model, with the phase names reflecting the unpredictability in the complex systems.

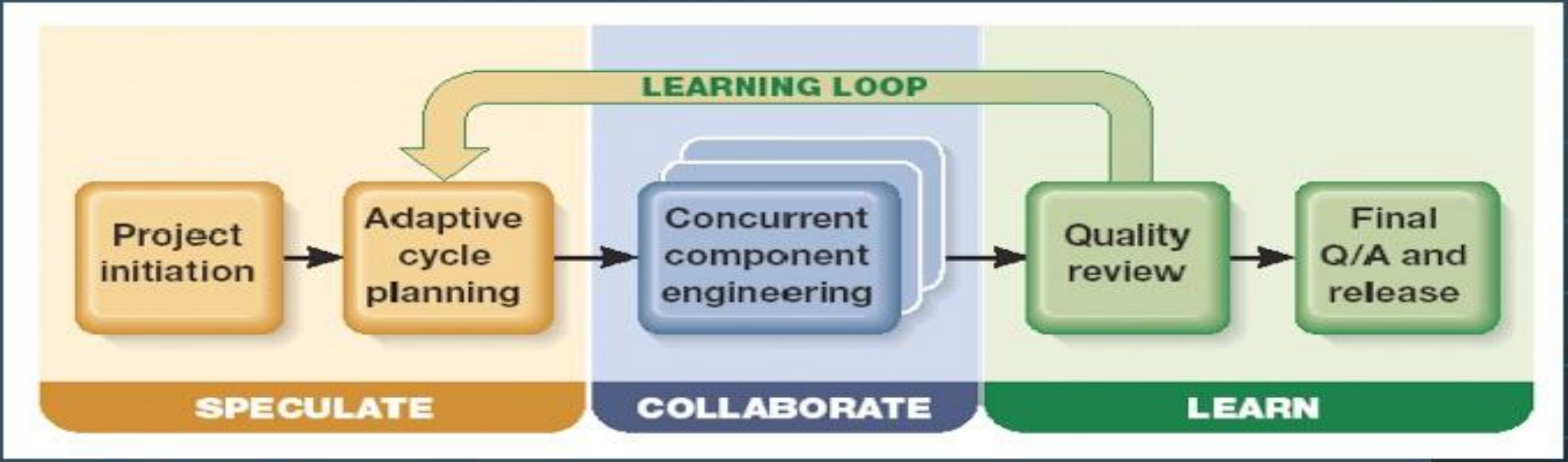
- The phases in the Adaptive development life cycle are –

1. Speculate
2. Collaborate
3. Learn

ASD



Adaptive Software Development



- Short iterations
- Deliverable-centric instead of task-centric

- ASD--Speculate

- The term plan is too deterministic and indicates a reasonably high degree of certainty about the desired result.
- In ASD, the term plan is replaced by the term speculate.
- While speculating, the team does not concentrate only on planning, but it acknowledges the reality of uncertainty in complex problems.
- Speculate encourages exploration and experimentation.
- Iterations with short cycles are encouraged.

ASD--Collaborate

- Complex applications require that a large volume of information be collected, analyzed, and applied to the problem.
- Turbulent environments have high rates of information flow. Hence, complex applications require that a large volume of information be collected, analyzed, and applied to the problem.
- This results in diverse Knowledge requirements that can only be handled by team collaboration.
- Collaborate would require the ability to work jointly to produce results, share knowledge or make decisions.

- ASD--Learn

- The Learn part of the Lifecycle is vital for the success of the project.
- Team has to enhance their knowledge constantly, using practices such as –

1. Technical Reviews
2. Project Retrospectives
3. Customer Focus Groups

- Reviews should be done after each iteration.
- Both, the developers and customers examine their assumptions and use the results of each development cycle to learn the direction of the next.

- The team learns –

  - About product changes
  - More fundamental changes in underlying assumptions about how the products are being developed
  - The iterations need to be short, so that the team can learn from small rather than large mistakes.

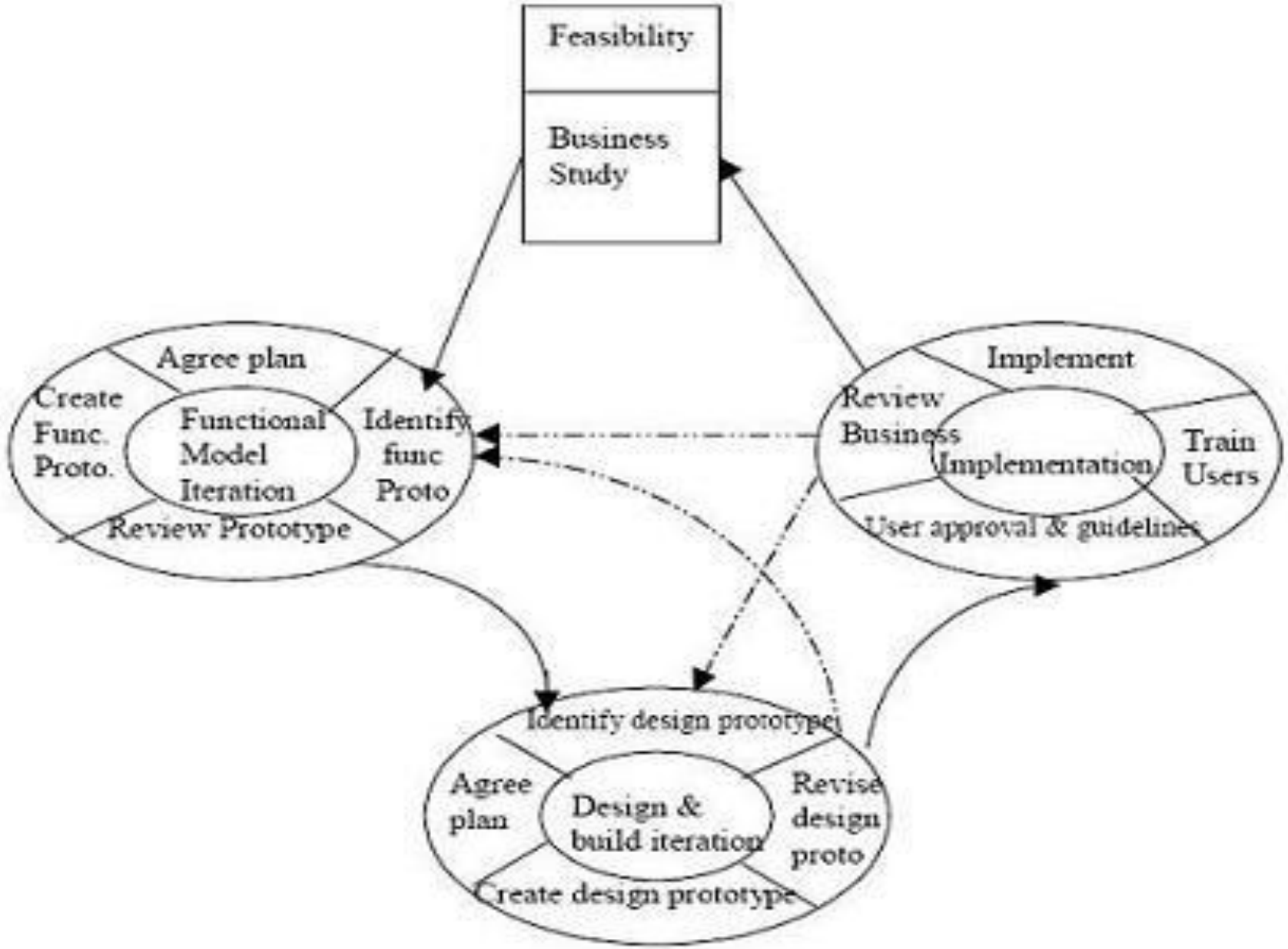
Advantageous

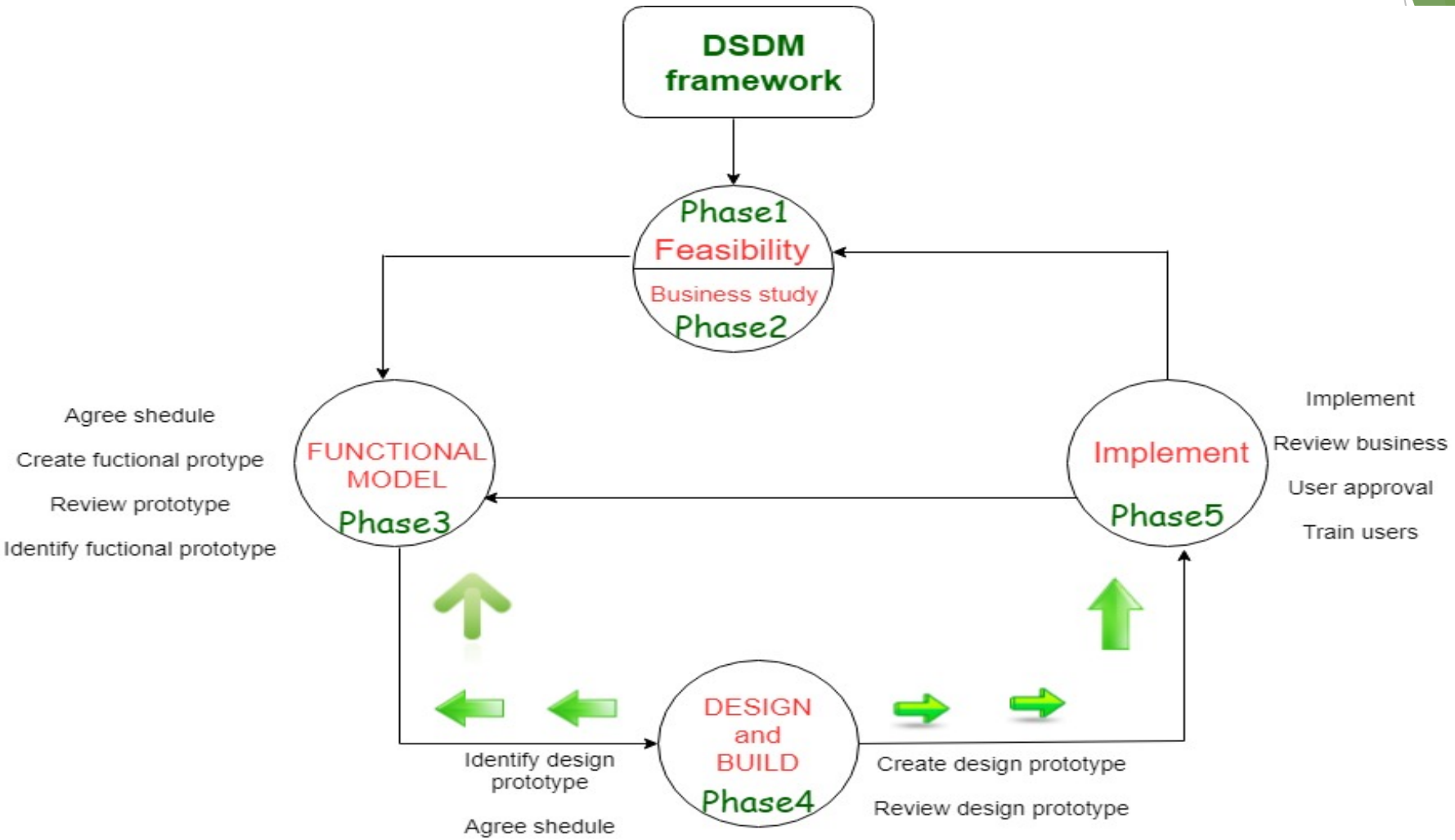
1. Extensive stakeholders engagement
2. Highly transparent
3. Early delivery
4. Focuses on the user
5. Generate high quality software.

Dynamic Systems Development Method (DSDM)

- DSDM is an agile software development methodology.
- It is an iterative, incremental approach that is largely based on the Rapid Application Development (RAD) methodology.
- DSDM differs from more common agile approaches as it encompasses the entire project lifecycle and not just software development.
- DSDM incorporates project management disciplines and provides mechanisms to ensure that the project benefits are clear, the proposed solution is feasible and there are solid foundations in place before detailed work is started.

- First released in 1994
- In 2014, DSDM released the latest version of the method in the 'DSDM Agile Project Framework'
- The method provides a four-phase framework consisting of:
  - Feasibility and business study
  - Functional model / prototype iteration
  - Design and build iteration
  - Implementation


Feasibility
Business Study
Agree plan
Create Func. Proto.
Functional Model Iteration
Identify func Proto
Review Prototype
Implement
Review Business
Implementation
Train Users
User approval & guidelines
Identify design prototype
Agree plan
Design & build iteration
Revise design proto
Create design prototype



Dynamic Systems Development Method life cycle

Principles of a DSDM

1. Focus on the business need
2. Deliver on time
3. Collaborate
4. Never compromise quality
5. Build incrementally from firm foundations
6. Develop iteratively
7. Communicate continuously and clearly
8. Demonstrate control

Core techniques

- Timeboxing: is the approach for completing the project incrementally by breaking it down into splitting the project in portions, each with a fixed budget and a delivery date.
- MoSCoW: is a technique for prioritizing work items or requirements. It is an acronym that stands for:
  - MUST have
  - SHOULD have
  - COULD have
  - WON'T have
- Prototyping:. It enables the early discovery of shortcomings in the system and allows future users to ‘test-drive’ the system.
- This way good user involvement is realized, one of the key success factors of DSDM, or any System Development project for that matter.

Bibliography

- ASD: https://www.tutorialspoint.com/adaptive_software_development/adaptive_software_development_lifecycle.htm
- DSDM

https://www.solutionsiq.com/agile-glossary/dynamic-systems-development-method-dsdm/

https://en.wikipedia.org/wiki/Dynamic_systems_development_method