



Software Development Methodologies

Lecturer: **Raman Ramsin**

Lecture 11

Agile Methodologies: ASD



Adaptive Software Development (ASD)

- Introduced by James Highsmith in 1997.
- A refined and extended version was introduced in 2000.
- Evolved from a RAD process and based on the teachings of complexity theory.
- Strives to present a change-tolerant, *adaptive* Speculate-Collaborate-Learn alternative to the *classical* Plan-Design-Build and the *iterative* Plan-Build-Revise lifecycles.



ASD: Hypotheses

- All aspects and constituents of the development effort (business environment, people, requirements, resources, methods, etc.) are highly volatile.
- Building complex systems is an evolutionary process extremely difficult to achieve unless special measures are taken to facilitate collaboration among the people who are somehow involved in or affected by the development of the system.
- The uncertain and unpredictable nature of the development leaves developers no alternative but to use short iterations, or *cycles*. In order to bound the development effort and keep it focused, a specific mission, a set of components to develop, and a time box are defined for each cycle.



ASD: Approach

- Iterations should be planned, but plans are only risk-driven *speculations*, requiring revision after each iteration of the cycle.
- The actual design and implementation of the system components becomes a by-product of intense *collaboration*.
- For the process to be adaptive, group reviews are performed at the end of each cycle, to enable the people involved to *learn* from the experience and implement the lessons learned in the process.



[Highsmith 1997]

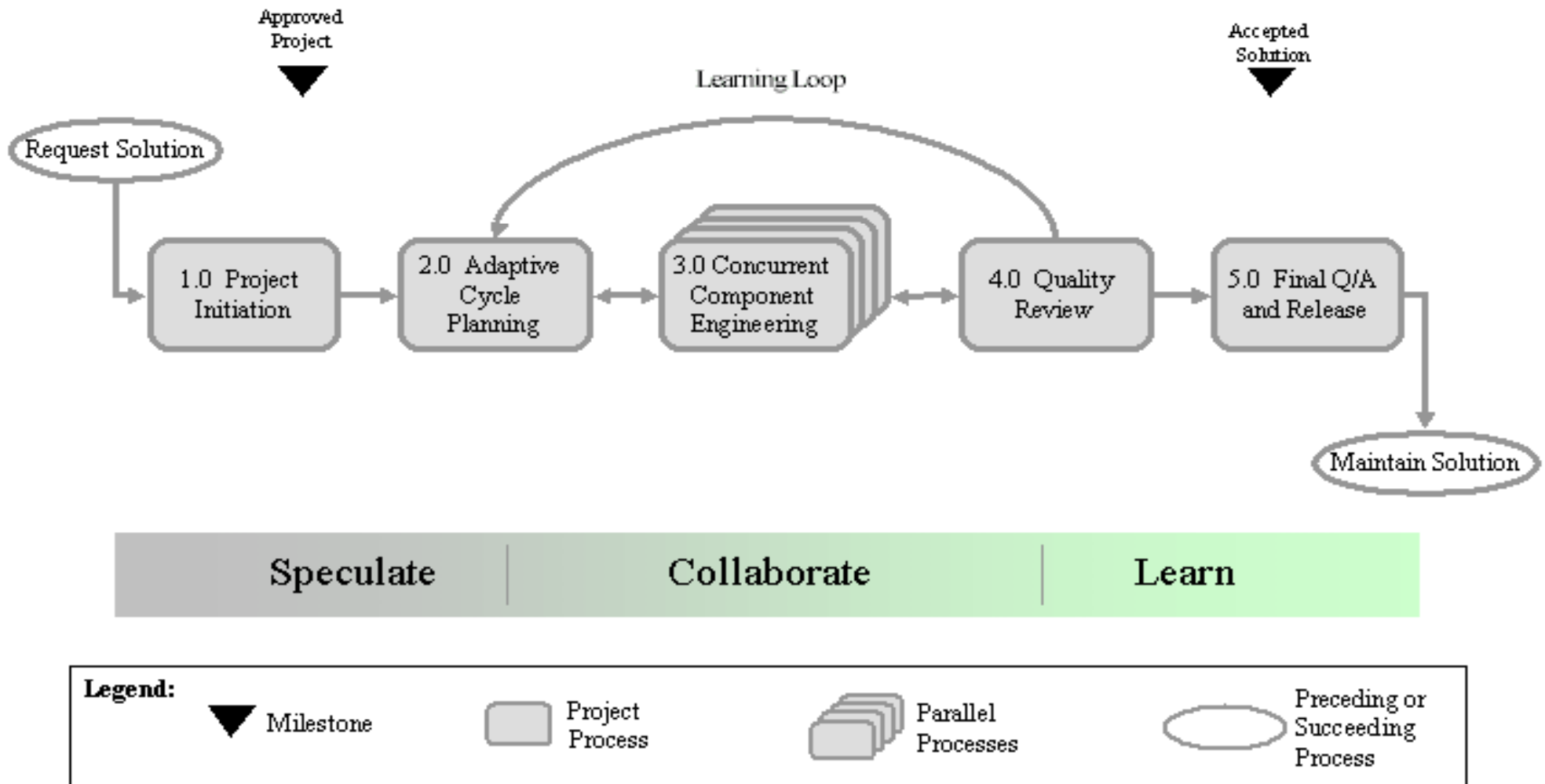


ASD: Process

1. *Project Initiation*: understanding the project's objectives and estimating its size and scope, exploring the constraints and the risks involved, organizing the development teams, identifying high-level requirements, and specifying success criteria.
2. *Iterative Development Phases*:
 1. *Adaptive Cycle Planning*: setting time frames for the project and the development cycles, defining the components that should be developed, assigning the components to cycles, and scheduling the iterations; the plan will be revised at the start of each iteration.
 2. *Concurrent Component Engineering*: concurrent design and implementation of the components assigned to individual cycles.
 3. *Quality Review*: conducting group reviews of the components produced and rectifying the problems confronted.
3. *Final Q/A and Release*: validating the produced system and deploying it into the working environment.



ASD: Process



[Highsmith 2000]

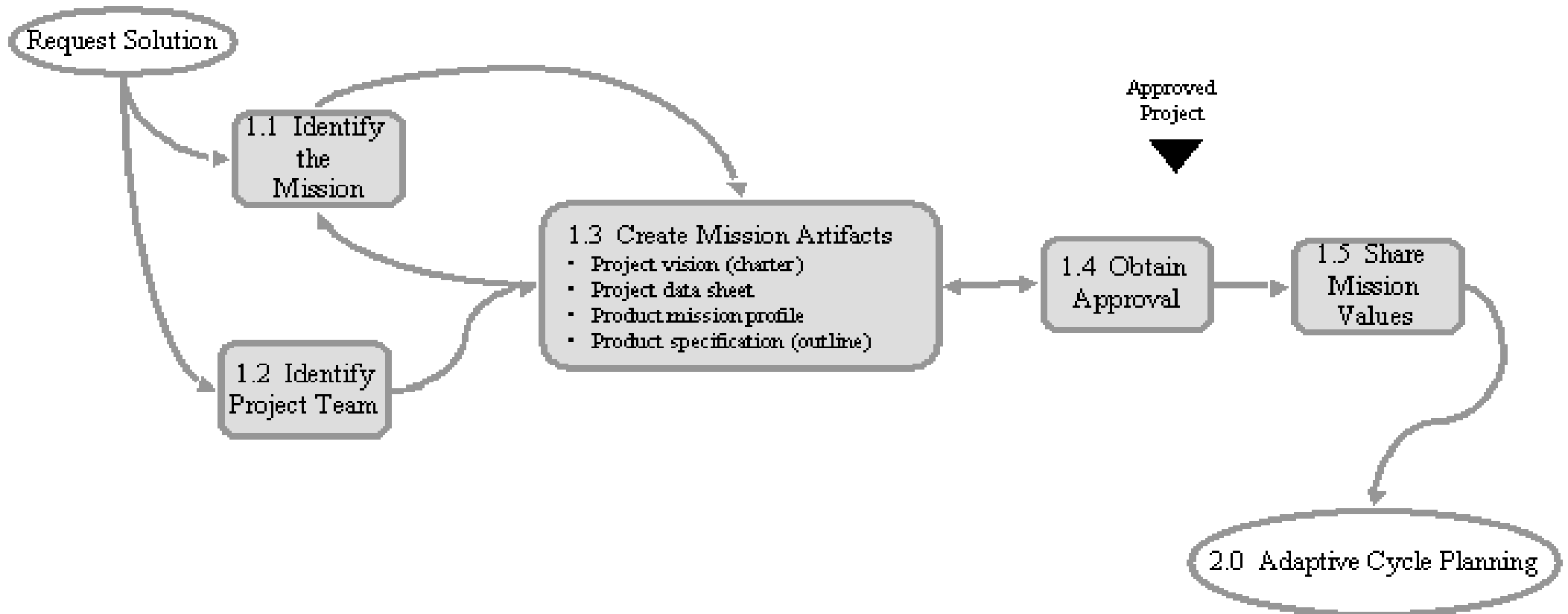


ASD Process: Project Initiation

1. *Specify the Project Mission*, which defines the objectives to be achieved and broad requirements to be satisfied by the project.
2. *Organize project team(s)*.
3. *Create the Mission Artefacts*; the necessary information for producing the artefacts is usually obtained through JAD sessions.
4. *Obtain approval* of the clients/sponsors and the permission to go ahead with the project.
5. *Share mission values* among the project community, through discussing and agreeing on quality objectives and evaluation criteria.



ASD Process: Project Initiation



[Highsmith 2000]



ASD Process: Project Initiation – *Mission Artefacts*

1. *Project Vision (Charter)*, which sets boundaries on the following:
 1. Scope, size, and context of the project.
 2. Resources allocated to the project.
 3. Project staff; defining the skills, knowledge, and authority required.
 4. Communication among the people involved in or affected by the project (*Project Community*).
2. *Product Mission Profile*, which identifies the primary factors governing the product's success.
3. *Product Specification* (outline), which contains the results of systems analysis and modeling, to be enriched in depth and breadth in later phases; typically includes a list of requirements and models of the system showing the overall functionality, major object classes, and their interactions.
4. *Project Data Sheet*, which is a one-page document summarizing the overall knowledge so far accumulated about the project; typically includes: project objectives, clients and sponsors, development teams, main features (functionality), overall scope (as a *Context Diagram*), resources, benefits and implications, milestones, constraints, priorities, and the key risks involved.

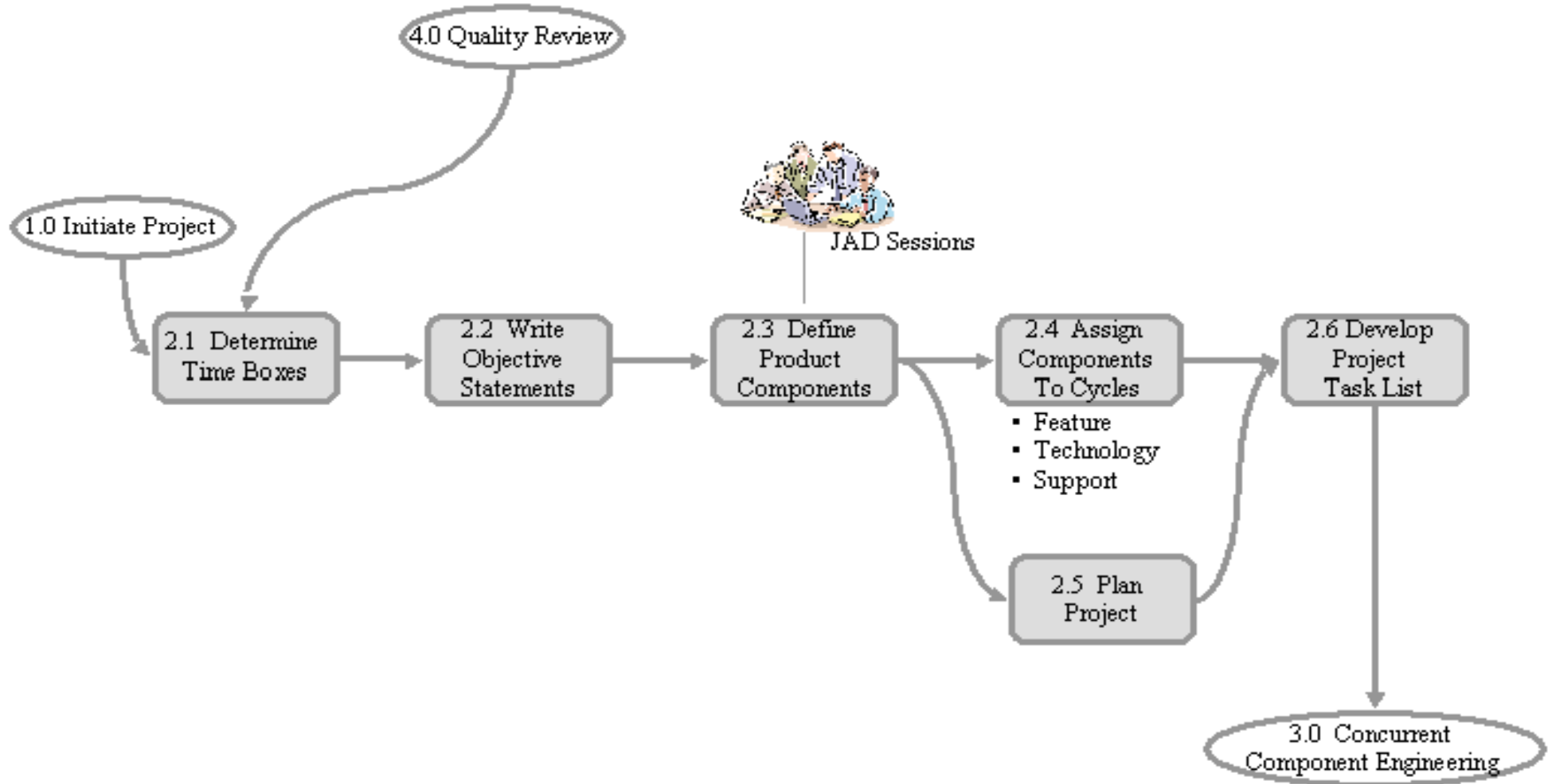


ASD Process: Adaptive Cycle Planning

1. *Determine time boxes* for the entire project and each of the development cycles; cycle time boxes are typically between two to eight weeks in duration.
2. *Write objective statements* for the development cycles.
3. *Define product components* through JAD sessions; components are of three types: *feature components* which are domain-specific and enact the business logic of the system; *technology components* and *support components*, which are domain-independent and provide the technical infrastructure.
4. *Assign components to cycles* according to the risks involved in their development, with consideration given to their interdependencies.
5. *Plan the project*; developing buffered schedules for development cycles and setting up a suitable medium (methods, tools and procedures) for enabling and enhancing collaboration among members of the project community.
6. *Develop a Project Task List*, consisting of the tasks that should be performed during the remaining phases of the project.



ASD Process: Adaptive Cycle Planning



[Highsmith 2000]

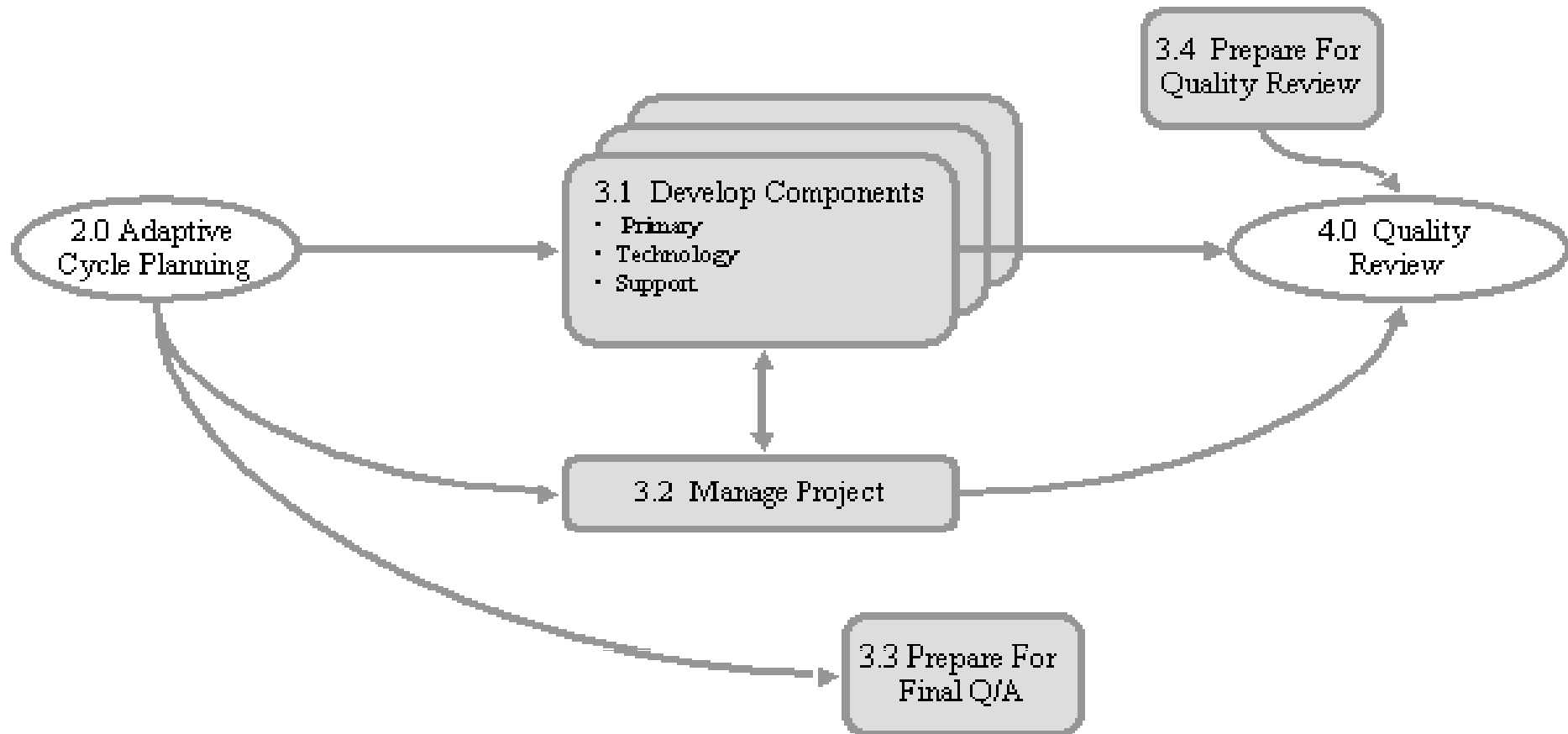


ASD Process: Concurrent Component Engineering

1. *Develop the components* assigned to the cycle. Working components are typically developed concurrently by development teams working in parallel and are delivered as *builds* on a daily or weekly basis. The produced builds are immediately fed into an integration process. Testing and refactoring are ongoing processes during this activity.
2. *Manage the project* through continuous monitoring and control. Maintaining the inter- and intra-team collaboration and keeping the cycle on the right track are the main concerns.
3. *Prepare for final Q/A* by developing system-level test plans and test cases.
4. *Prepare for quality review* by planning the review meetings to take place in the Quality Review phase.



ASD Process: Concurrent Component Engineering



[Highsmith 2000]

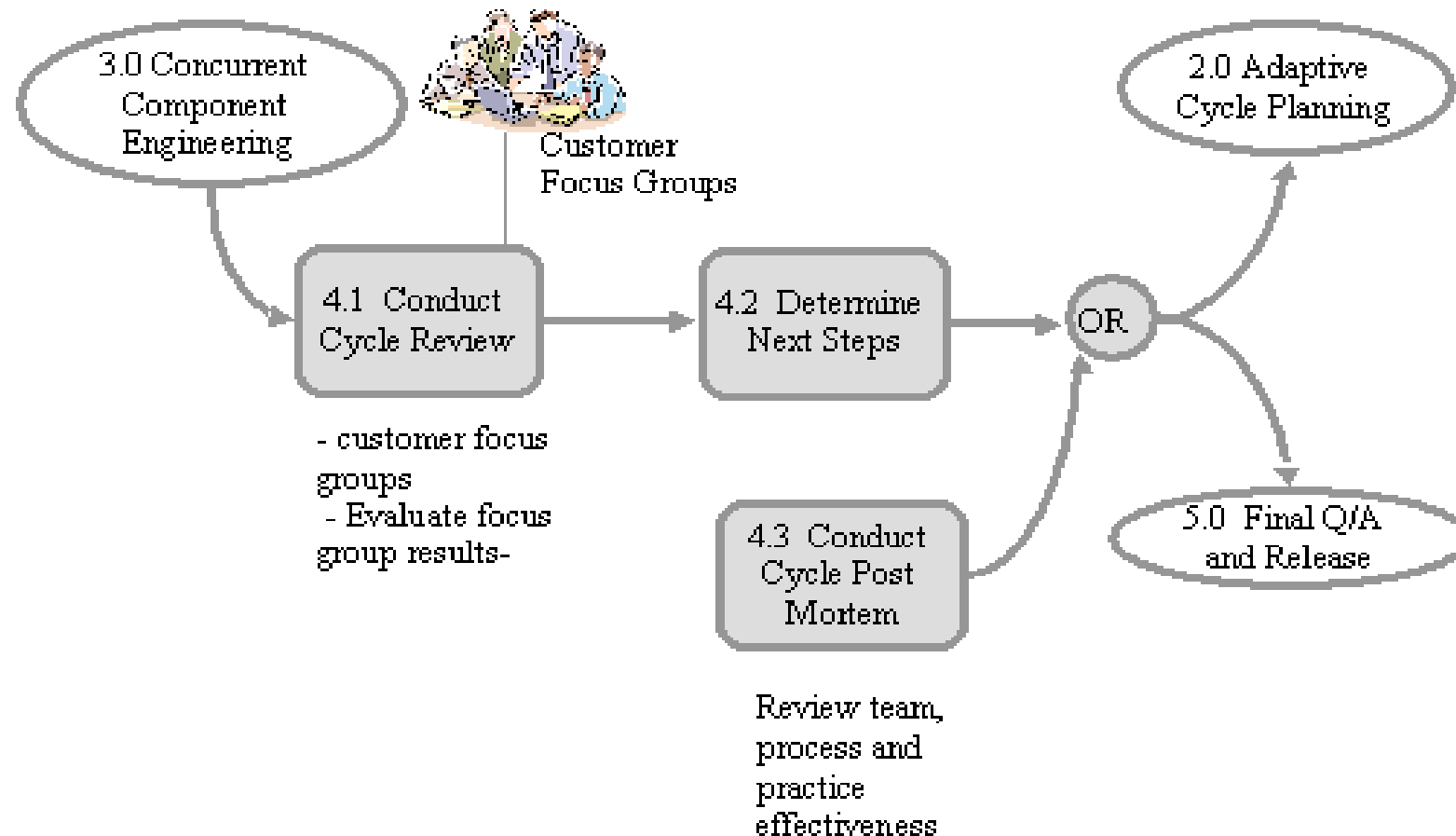


ASD Process: Quality Review

1. *Conduct cycle review* by holding facilitated customer focus group sessions. The result of the cycle is presented to the customers. The feedback and change requests are carefully documented in order to be considered in later iterations.
2. *Determine next step*; decision is made on whether another iteration cycle should be initiated, or the system should be prepared for release.
3. *Conduct cycle post-mortem*, which typically involves reviewing the performance of the teams and the effectiveness of the methods used. The problems are then rectified so as not to adversely affect the next iterations.



ASD Process: Quality Review



[Highsmith 2000]

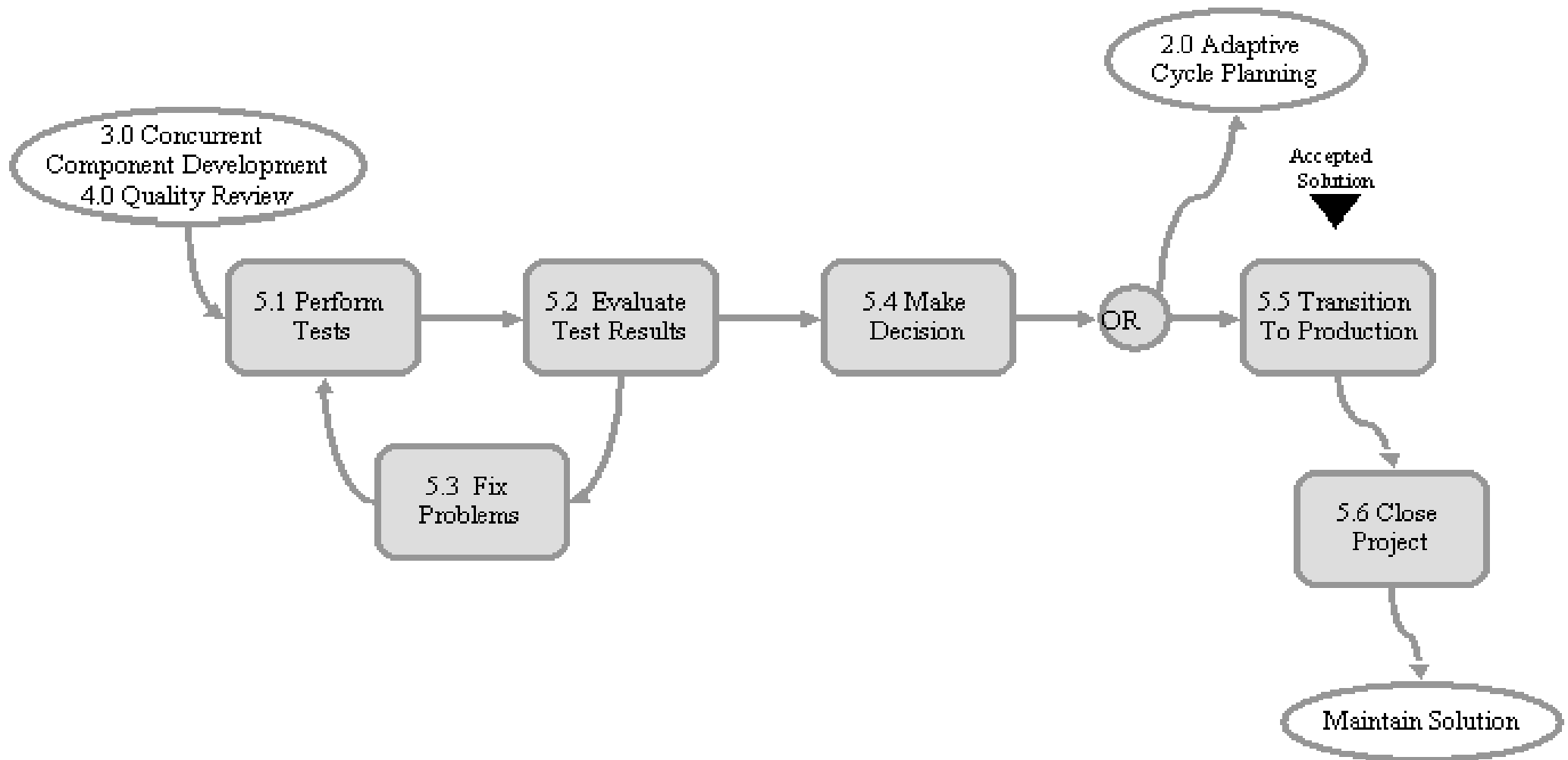


ASD Process: Final Q/A and Release

1. *Perform tests*, with the main purpose of system-level validation.
2. *Evaluate the test results*.
3. *Fix the problems*.
4. *Make a decision* based on the test results, whether to release the system or to start a new development cycle.
5. *Transition to production*; typically involving deployment activities including system conversion, training, and preparation of documents.
6. *Close the project*, which, in addition to the usual wrapping-up and termination procedures, also includes a project post-mortem summarizing the lessons learned from the execution of the project.



ASD Process: Final Q/A and Release



[Highsmith 2000]



ASD: Strengths and Weaknesses

■ Strengths

- Iterative-incremental process
- Based on structural, functional and behavioural modeling of the problem domain and the system
- Well-worked-out process
- Special attention to quality assessment and control (Q/A is performed at all levels: per-project and per-iteration)
- Component-based development approach
- Adaptive (tuneable) process; through risk-driven planning, conducting reviews, and revising the plans and the development process according to what has been learnt during the iterations
- Extensive use of JAD sessions for information gathering and decision making



ASD: Strengths and Weaknesses

■ Strengths (Contd.)

- Stress on the importance of a collaborative environment for the development to be successful: a *User Community* is established and a suitable medium of collaboration (methods, tools and procedures) is set up.
- Test-based development
- Refactoring for simplifying the code
- Continuous integration
- Stress on parallel development of components by collaborating teams of developers, thus speeding up the process
- Traceability to requirements through ongoing validation and quality review



ASD: Strengths and Weaknesses

■ Weaknesses

- Not scalable
- Over-dependence on inter-human communication
- Need for intensive project monitoring and control in order to maintain inter-team and intra-team collaboration during component development
- Seamlessness not addressed
- No clear-cut design effort
- Model-phobic
- No specific models prescribed
- Physical configuration modeling is ignored (even though necessary in component-based development).
- Lack of formalism



References

- Highsmith, J., "Messy, exciting, and anxiety-ridden: Adaptive software development". *American Programmer* 10, 4 (April), pp. 23-29, 1997.
- Highsmith, J., *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. Dorset House, 2000.