

2.2 Rapid Application Development (RAD) Model

- **Rapid application development (RAD)** is an incremental software development process model that emphasizes an extremely short development cycle.
- The RAD model is a “high-speed” adaptation of the linear sequential model in which rapid development is achieved by using component-based construction.
- If requirements are well understood and project scope is constrained, the RAD process enables a development team to create a “fully functional system” within very short time periods (e.g., 60 to 90 days).

Used primarily for information systems applications, the RAD approach encompasses the following phases:

2.1 Communication:

Communication works to understand the business problem and the information characteristics that the software must accommodate.

2.2 Planning:

Planning is essential because multiple software teams work in parallel on different system functions.

2.3 Modeling:

Modeling encompasses three major phases- business modeling, data modeling, and process modeling.

a) Business modeling:

- The information flow among business functions is modeled in a way that answers the following questions:
- What information drives the business process?
- What information is generated?
- Who generates it?
- Where does the information go?
- Who processes it?

b) Data modeling:

- The information flow defined as part of the business modeling phase is refined into a set of data objects that are needed to support the business.
- The characteristics (called *attributes*) of each object are identified and the relationships between these objects defined.

c) Process modeling:

- The data objects defined in the data modeling phase are transformed to achieve the information flow necessary to implement a business function.
- Processing descriptions are created for adding, modifying, deleting, or retrieving a data object.

2.4 Construction:

Construction emphasizes on-

a) Component reuse and Code generation:

- Rather than creating software using conventional third generation programming languages the RAD process works to reuse existing program components (when possible) or create reusable components (when necessary).
- In all cases, automated tools are used to facilitate construction of the software.

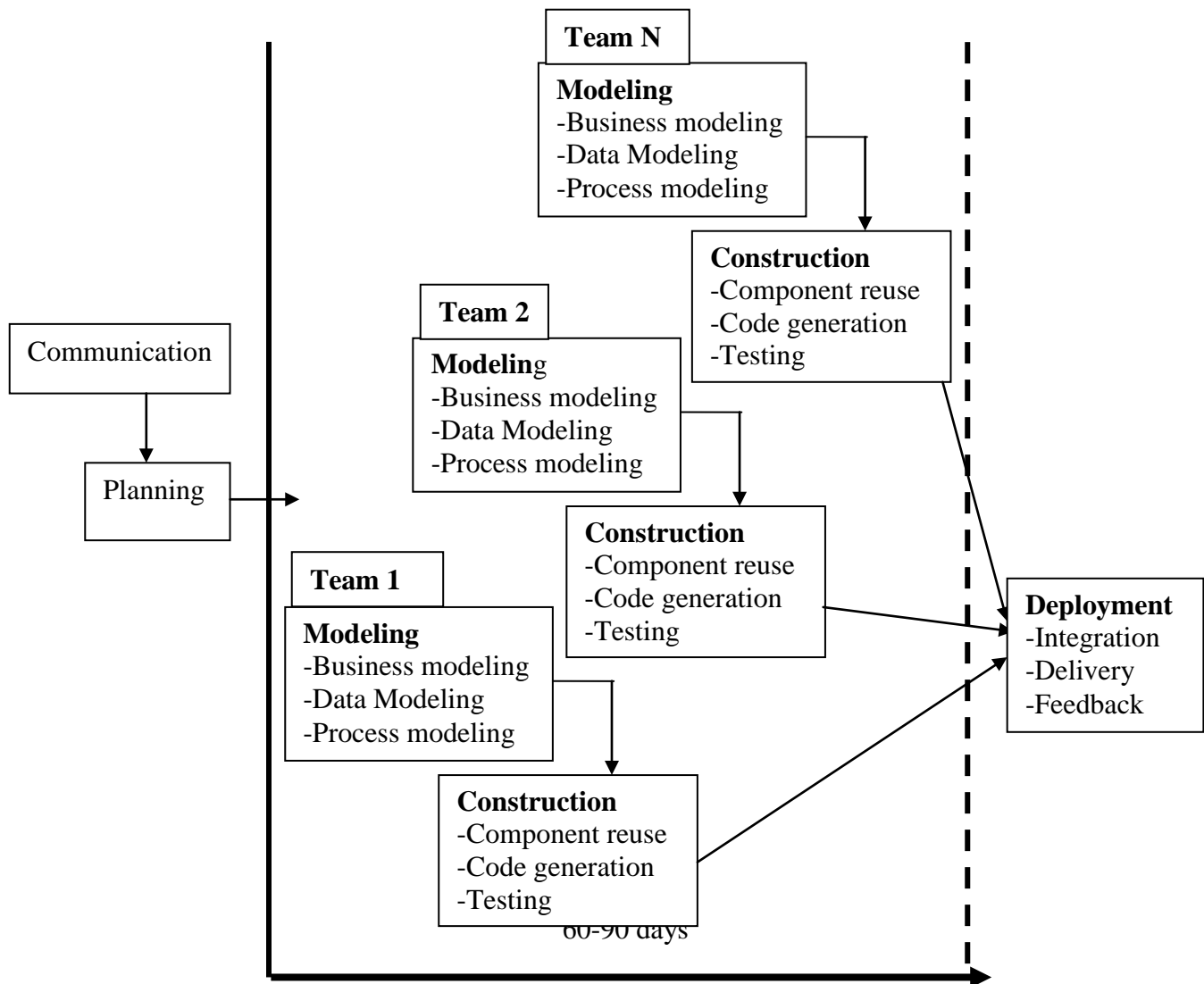


Figure: RAD Model

b) Testing:

- Since the RAD process emphasizes reuse, many of the program components have already been tested.
- This reduces overall testing time. However, new components must be tested and all interfaces must be fully exercised.

Advantages of RAD model:

1. The time required to develop the software is drastically reduced due to a reduced requirement analysis (business requirements documentation and software requirement specification) and planning stage.
2. All the software prototypes produced can be kept in a repository for future use.
3. The reusability of the components also enhances the speediness of the process of software development.
4. It is much easier for a project manager to be accurate in estimating project costs which of course means that project cost controls are easier to implement and manage as well.
5. It is a big cost saver in terms of project budget as well as project time and cost due to reusability of the prototypes.
6. If a component is being picked for the repository, it is already tested and hence need not be tested again. This helps in saving time required for testing.
7. The project management requirements are collected in a dynamic manner. Every time there is a prototype ready, requirements are studied and matched. If there are any additional requirements, these are then included in the next prototype built.

8. There is a strong and continuous participation of the project sponsor who keeps giving feedback in the whole process. Hence the end user satisfaction level is higher when the end result is produced.
9. It promotes better documentation through written test cases.

Drawbacks of RAD model:

Like all process models, the RAD approach has drawbacks:

1. For large but scalable projects, RAD requires sufficient human resources to create the right number of RAD teams.
2. RAD requires developers and customers who are committed to the rapid-fire activities necessary to get a system complete in a much abbreviated time frame. If commitment is lacking from either constituency, RAD projects will fail.
3. Not all types of applications are appropriate for RAD. If a system cannot be properly modularized, building the components necessary for RAD will be problematic.
4. If high performance is an issue and performance is to be achieved through tuning the interfaces to system components, the RAD approach may not work.
5. RAD is not appropriate when technical risks are high. This occurs when a new application makes heavy use of new technology or when the new software requires a high degree of interoperability with existing computer programs.