3.3 INCREMENTAL PROCESS MODELS (from pressman 5th ed)

3.3.1 The Incremental Model

3.3.2 The RAD Model

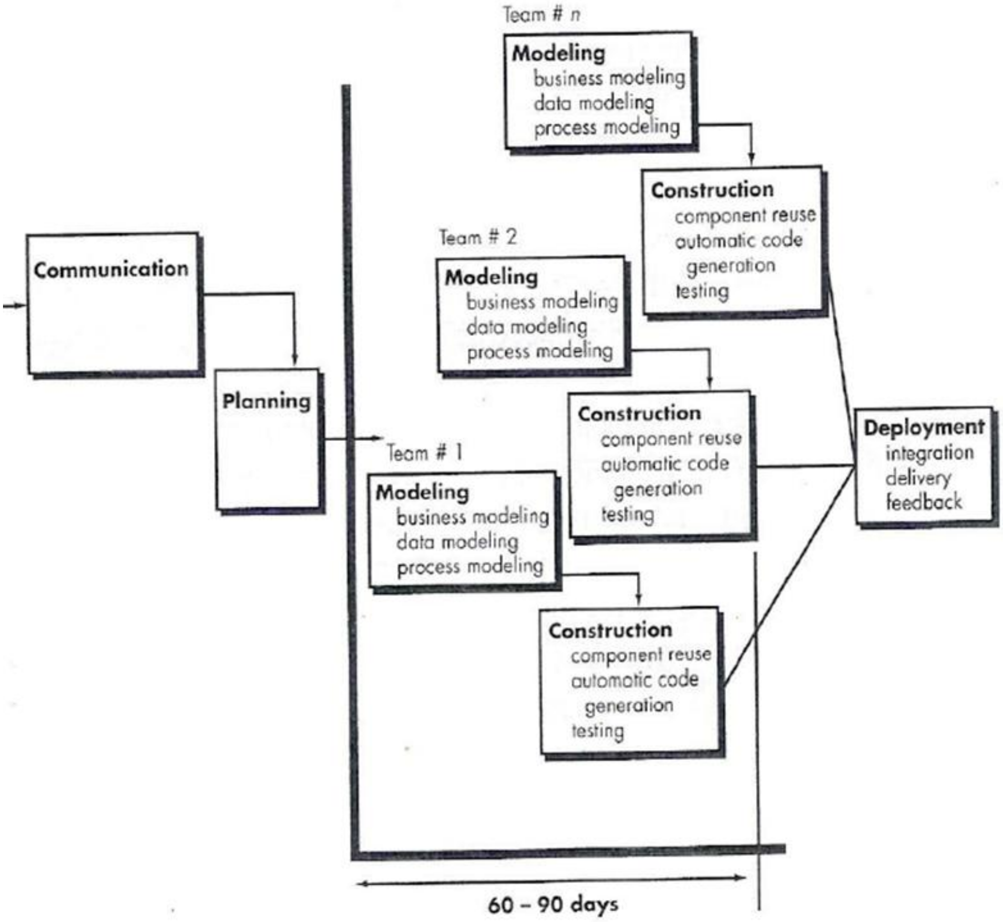
Rapid Application Development (RAD) is an incremental software process model that emphasizes a short development cycle. The RAD model is a **"high-speed" adaptation of the waterfall model,** in which rapid development is achieved by using a **component-based construction approach**. If requirements are well understood and project scope is constrained,[4] the RAD process enables a development team to create a "fully functional system" within a very short time period (e.g., 60 to 90 days) [MAR91].

[3] It is important to note that an incremental philosophy is also used for all "agile" process models

Communication works to understand the business problem and the information characteristics that the software must accommodate. **Planning** is essential because multiple software teams work in parallel on different system functions. **Modeling** encompasses three major phases—business modeling, data modeling and process modeling—and establishes design representations that serve as the basis for RAD's construction activity. **Construction** emphasizes the use of preexisting software components and the application of automatic code generation. Finally, d**eployment** establishes a basis for subsequent iterations, if required [KER94j.

Obviously, the time constraints imposed on a RAD project demand "scalable scope" [KER94j.

If a business application can be modularized in a way that enables each major function to be completed in less than three months (using the approach described above), it is a candidate for RAD. Each major function can be addressed by a separate RAD team and then integrated to form a whole.



. In fact, many software projects have poorly defined requirements at the start. In such cases prototyping or evolutionary approaches (Section 3.4) are much better process options. See [REI95].

ISACA 2006:

RAD supports the analysis, design, development and implementation of individual application systems. However. RAD does not support the planning or analysis required to define the information needs of the enterprise as a whole or of a major business area of the enterprise. RAD provides a means for developing systems faster while reducing cost and increasing quality. This is done by automating large portions of the SDLC, imposing rigid limits on development time frames and reusing existing components. The RAD methodology has four major stages:

1.The **concept definition stage** defines the business functions and data subject areas that the system will support and determines the system scope.

2.The **functional design stage** uses workshops to model the systems data and processes and build a working prototype of critical system components.

3. The **development stage** completes the construction of the physical database and application system, builds the conversion system, and develops user aids and deployment work plans.

4. The **deployment stage** includes final-user testing and training, data conversion and the implementation of the application system.

RAD uses prototyping as its core development tool no matter which underlying technology is used.
The greatest advantage of RAD is the shorter time frame for the development of a system.
A potential risk with prototyped systems is that the finished system will have poor controls