CHAPTER 1: INTRODUCTION TO RAPID APPLICATION DEVELOPMENT (RAD)

1. INTRODUCTIONS

- RAD refers to a development life cycle designed
- Compare to traditional life cycle it is Faster development with higher quality systems
- It is designed to take advantage of powerful development software like:
  - CASE tools (computer-aided software engineering)
  - Prototyping tools
  - Code generators
The key objectives of RAD are:
- High Speed
- High Quality
- Low Cost

RAD is a people-centered and incremental development approach.

- active user involvement
- all stakeholders
- testing is included – system is tested & reviewed by both developers & user

In short RAD thus enables quality products to be developed faster, saving valuable resources.
2. WHY DO YOU NEED TO BE RAD?

- RAD takes advantage of tools and techniques to restructure the process of building information systems.

- This new process, expand to the entire IS organization, results in a reflective transformation of IS development.

- RAD replaces hand-design and coding processes.

- RAD is a more capable process, faster and less error level than hand coding.

- Traditional development lifecycles are:
  - too slow
  - inflexible to meet the business demands of today’s economy
A new methodology must be implemented one that allows organizations to build software applications

- Faster
- Better
- Cheaper

RAD, in addition to providing a more:

- quality product in less time
- ensures greater customer satisfaction
- end-users join together in the development of the application
- development teams concentrate on delivering prototypes to designated user experts.

Each prototype is tested by those users and returned to the development team for reworking

- giving the users the opportunity to modify the requirements and review the resulting software implementation
3. THE HISTORY OF RAD

- Traditional lifecycles develop in the 1970s, and still widely used today.

- The requirements and design are then frozen and the system is coded, tested, and implemented.

- With such conventional methods, there is a long delay before the customer gets to see any results and the development process can take so long that the customer’s business could fundamentally change before the system is even ready for use.
RAD compresses the step-by-step development of conventional methods into an iterative process.

The RAD approach thus includes:
- developing and refining the data models
- process models
- prototype in parallel using an iterative process

User requirements are:
- developed
- a solution is designed
- the solution is prototyped
- the prototype is reviewed
- user input is provided
- the process begins again
4. ESSENTIAL ASPECTS OF RAD

Rapid Application Development has four essential aspects:
Methodology
People
Management
Tools
4.1 Methodology

Fundamentals of the RAD methodology thus include:

1. Combining the best available techniques and specifying the sequence of tasks that will make those techniques most effective.

2. Using evolutionary prototypes that are eventually transformed into the final product.

3. Using workshops, instead of interviews, to gather requirements and review design.
4. Selecting a set of CASE tools to support modeling, prototyping and code reusability, as well as automating many of the combinations of techniques.

5. Implementing time boxed development that allows development teams to quickly build the core of the system and implement refinements in subsequent releases.

6. Providing guidelines for success and describing drawback to avoid.
Active user involvement throughout the RAD lifecycle ensures that business requirements and user expectations are clearly understood.

RAD takes advantage of powerful application development tools to develop high quality applications rapidly.

Prototyping is used to help users visualize and request changes to the system as it is being built, allowing applications to evolve iteratively.

RAD techniques are also very successful when faced with unstable business requirements or when developing non-traditional systems.

The structure of the RAD lifecycle is thus designed to ensure that developers build the systems that the users really need. This lifecycle, through the following four stages,
1. Requirements Planning
This stage defines the business functions and data subject areas that the system will support and determines the system’s scope.

2. User Design
This stage uses workshops to model the system’s data and processes and to build a working prototype of critical system components.

3. Construction
This stage completes the construction of the physical application system, builds the conversion system, and develops user aids and implementation work plans.

4. Implementation
This stage includes final user testing and training, data conversion, and the implementation of the application system.
4.2 People

- The success of Rapid Application Development is dependent upon the involvement of people with the right skills and talents.

- These people must thus be carefully selected, highly trained, and highly motivated.

- They must be able to use the tools and work together in teams.

- At the Requirements Planning and User Design stages, key end users must be available to participate in workshops.
At the end of the development cycle, the Cutover Team, which handles training and cutover, must also be ready to move quickly.

The key players in a Rapid Application Development project include:

**Sponsor**
A high-level user executive who funds the system and is dedicated to both the value of the new system and to achieving results quickly.

**User Coordinator**
A user appointed by the Sponsor to oversee the project from the user perspective.

**Requirements Planning Team**
A team of high-level users who participate in the Joint Requirements Planning workshop.
**User Design Team**
A team of users who participate in the design workshop.

**User Review Board**
A team of users who review the system after construction and decide whether modifications are necessary before cutover.

**Training Manager**
The person responsible for training users to work with the new system.

**Project Manager**
The person who oversees the development effort.

**Construction (SWAT) Team (Skilled Workers with Advanced Tools)**
Team is a small team of two to six developers who are highly trained to work together at high speed.
Workshop Leader
The specialist who organizes and conducts the workshops for Joint Requirements Planning and Joint Application Design.

4.3 Management

- To successfully introduce rapid development, management must pay careful attention to human motivation.

- Managers should target those professionals whom they deem as ‘Early Adapters.’ ‘Early Adapters’ are those people who see the value of a new methodology and lead the way in making it practical to use.

- These employees are excited about the new methodology and they want to make it work well in their environment.
4.4 Tools

- The RAD methodology uses both computerized tools and human techniques to achieve the goals of high-speed and high quality.

- Examples of tools that can be used in RAD projects are CASE tools. These tools play a key role in eliminating some problems that exist in other models of software development.

- For example, CASE tools can be used to develop models (using, e.g., UML diagrams) and directly generate code based on those models instead of hard coding.
5. Conclusion

- RAD has successfully achieved the objective of reducing costs on project whilst not compromising on quality.

- Encouraging the involvement of customers in the entire process of its development lifecycle.

- RAD has also demonstrated strength in being able to speed up the development process by appropriately combine its methodology, people, management and high tech computer aided tools.

- RAD has also proven to have challenges.