RAPID APPLICATION DEVELOPMENT AND SERVICE-ORIENTED ARCHITECTURE

BY NIRJALA K.C.

Rapid Application Development

What is Rapid application development?

RAD is an adaptive software development approach where a software prototype is rapidly updated based on user feedback and iteratively delivered until it meets all client requirements.
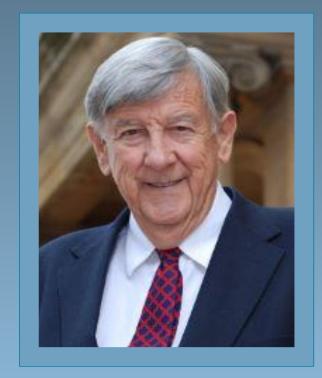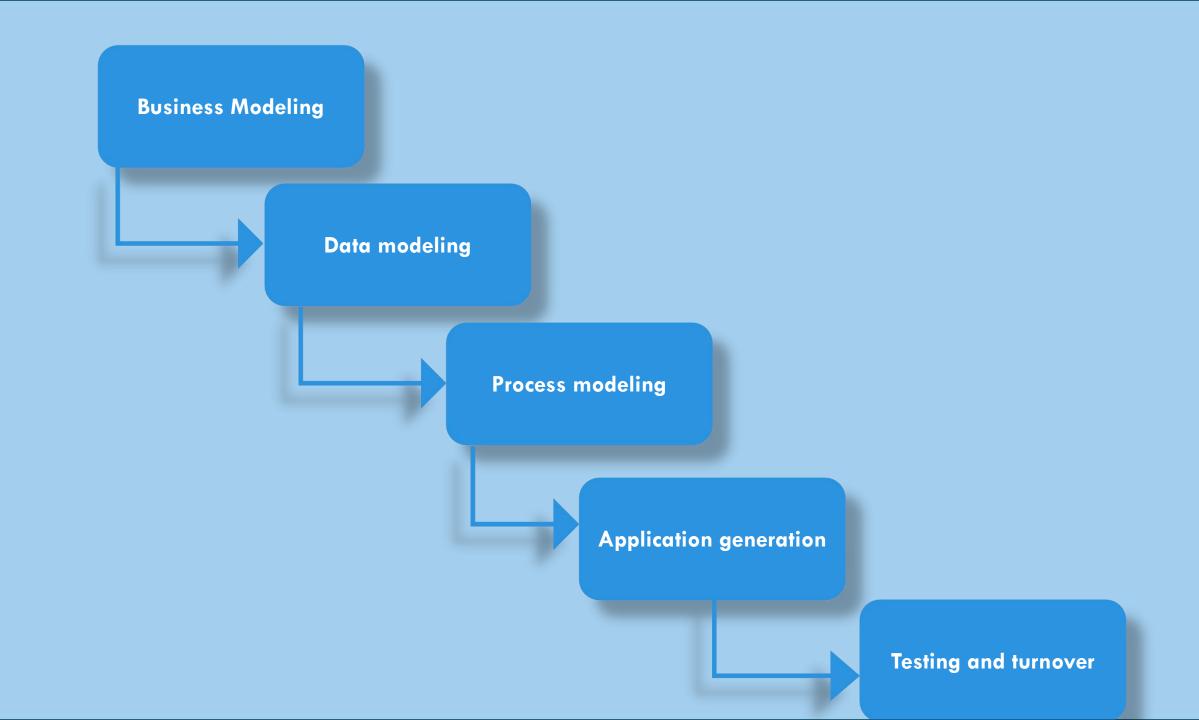
RAD was first introduced by author James Martin in 1991. Martin recognized and then took advantage of the endless malleability of software in designing development models. Rapid Application Development (RAD) is a methodology focusing on delivering rapidly through continuous feedback and frequent iterations.

RAD (Rapid Application development) Model

❑ RAD is a software development process based on prototyping without any specific planning.

❑The RAD model was first proposed by IBM in the 1980's

❑The RAD model is a type of incremental process model in which there is an extremely short development cycle.

❑When the requirement are fully understand and the component –based construction approach is adopted then the RAD model is used.

❑The RAD model enables a development team to create a fully functional system with a concise time period.

❑It is a concept that product can be developed faster and of higher quality through.

❑Gathering requirements using workshops or focus groups

❑Prototyping and early , reiterative user testing or design.

❑The reuse of software components.

❑Less formality in reviews and other team communication.

Phase 1: Business modelling

At this stage, information flow between different business functions is defined by answering the following question

Who generates the data?

Where does the information go?

Who processes it?

Phase 2: Data modeling

The data collected from business modeling is refined into a set of data objects (entities) that are needed to support the business.

The attributes of each group are identified, and the relationship between them is defined.

Phase 3: Process modeling

The information object defined in the data modeling phase are transformed to achieve the data flow necessary to implemented a business functions processing descriptions are created for adding modifying deleting retrieving a data object.

Phase 4: Application generation

   Automated tools construction of the software ,Even they use the 4th GL techniques.

Phase 5: Testing and turnover

   Many of the programming Components have already been tested Since, RDA emphasis reuse. This reduces the overall testing time. But the new part must be tested, and all interface must be exercised.

Advantages

➢ **Speed and Efficiency:** RAD significantly reduces development timelines, making it suitable for time-sensitive projects.

➢ **User Satisfaction**: Active user involvement ensures that the final product meets user needs and expectations.

➢ **Reduced Development Costs:** Faster development cycles can lead to cost savings.

➢ **Flexibility**: RAD adapts well to projects with evolving or unclear requirements.

➢ **Early Prototyping**: Early prototypes allow stakeholders to visualize the end product and provide valuable feedback.

Drawbacks

– Complexity: The rapid pace of RAD can make it challenging to manage complex projects effectively.

– Limited Documentation: Minimal planning and documentation may lead to a lack of comprehensive project documentation.

– Scope Creep: Frequent changes and adaptations can result in scope creep if not managed rigorously.

– Resource Demands: RAD requires skilled and dedicated team members who can work closely with end-users.

– Not Suitable for All Projects: RAD may not be the best fit for projects with well-defined and stable requirements.
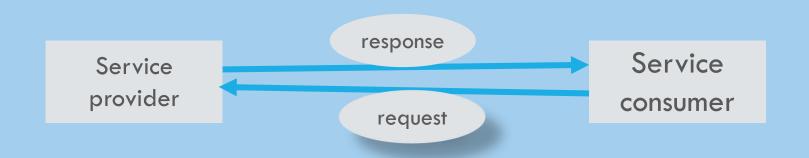
What is service provider?

Service-oriented architecture (SOA) is a method of software development that uses software components called services to create business applications. Each service provides a business capability, and services can also communicate with each other across platforms and languages.

There are two major roles within Service-oriented Architecture:

Service provider: The service provider is the maintainer of the service and the organization that makes available one or more services for others to use. To advertise services, the provider can publish them in a registry, together with a service contract that specifies the nature of the service, how to use it, the requirements for the service, and the fees charged.

Service consumer: The service consumer can locate the service metadata in the registry and develop the required client components to bind and use the service.

response

Service provider → Service consumer

request

Advantages of SOA:

Service reusability: In SOA, applications are made from existing services. Thus, services can be reused to make many applications.

Easy maintenance: As services are independent of each other they can be updated and modified easily without affecting other services.

Platform independent: SOA allows making a complex application by combining services picked from different sources, independent of the platform.

Availability: SOA facilities are easily available to anyone on request.

Reliability: SOA applications are more reliable because it is easy to debug small services rather than huge codes

Scalability: Services can run on different servers within an environment, this increases scalability

Disadvantages of SOA:

High overhead: A validation of input parameters of services is done whenever services interact this decreases performance as it increases load and response time.

High investment: A huge initial investment is required for SOA.

Complex service management: When services interact they exchange messages to tasks. the number of messages may go in millions. It becomes a cumbersome task to handle a large number of messages.