Module 1

RAD, DevOps, CMM Model

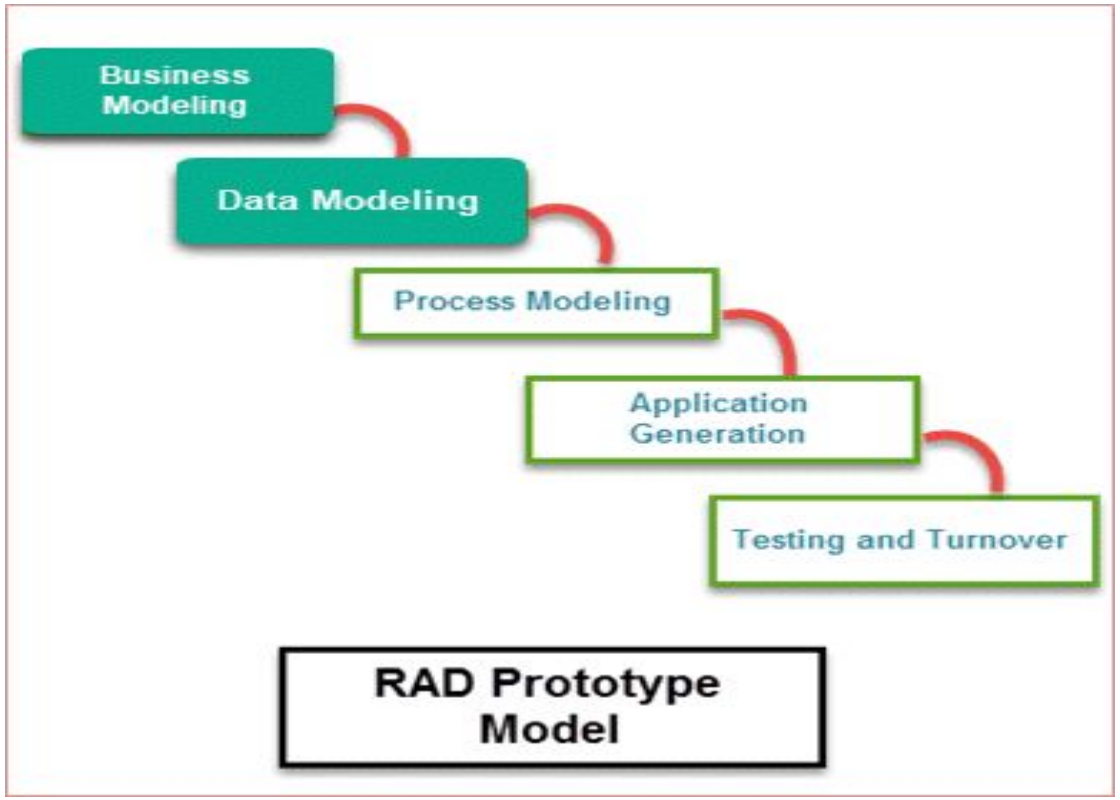
What is RAD Model?

- RAD Model or Rapid Application Development model is a software development process.
- It is based on prototyping without any specific planning.
- In RAD model, there is less attention paid to the planning and more priority is given to the development tasks.
- It targets at developing software in a short span of time.

SDLC RAD modeling has following phases:

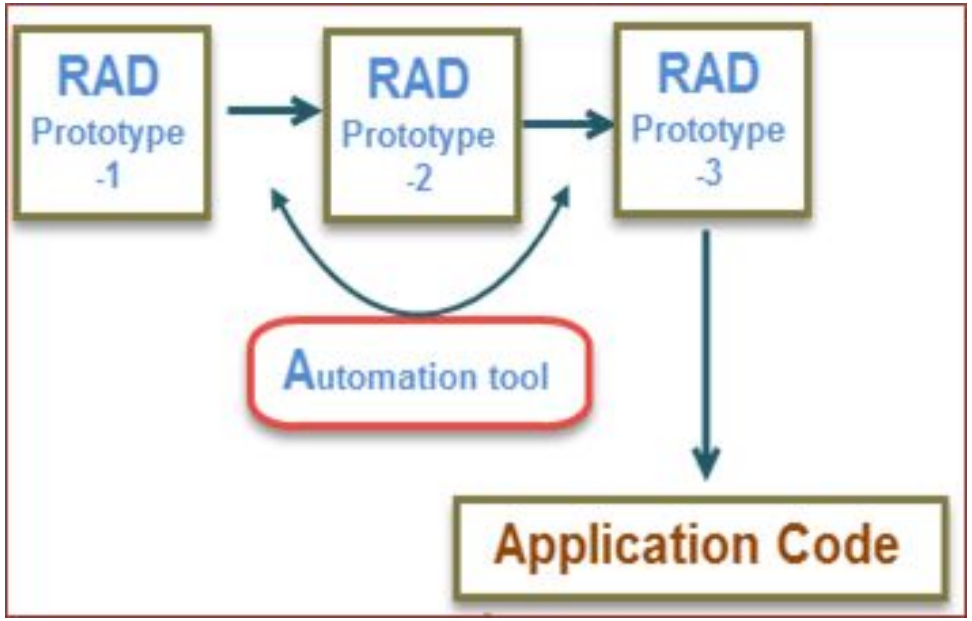
- Business Modeling
- Data Modeling
- Process Modeling
- Application Generation
- Testing and Turnover

RAD Model Diagram:



- It focuses on input-output source and destination of the information.
- It emphasizes on delivering projects in small pieces; the larger projects are divided into a series of smaller projects.
- The main features of RAD modeling are that it focuses on the reuse of templates, tools, processes, and code.

RAD Model in Software Engineering:



Different Phases of RAD Model:

- There are following five major phases of Rapid Application Development Model:

RAD Model Phases	Activities performed in RAD Modeling
Business Modeling	•On basis of the flow of information and distribution between various business channels, the product is designed
Data Modeling	•The information collected from business modeling is refined into a set of data objects that are significant for the business

Process Modeling	•The data object that is declared in the data modeling phase is transformed to achieve the information flow necessary to implement a business function
Application Generation	•Automated tools are used for the construction of the software, to convert process and data models into prototypes
Testing and Turnover	•As prototypes are individually tested during every iteration, the overall testing time is reduced in RAD.

When to use RAD Methodology?

- When a system needs to be produced in a short span of time (2-3 months)
- When the requirements are known
- When the user will be involved all through the life cycle
- When technical risk is less
- When there is a necessity to create a system that can be modularized in 2-3 months of time
- When a budget is high enough to afford designers for modeling along with the cost of automated tools for code generation

Advantages of RAD Model	Disadvantages of RAD Model
•Flexible and adaptable to changes	•It can’t be used for smaller projects
•It is useful when you have to reduce the overall project risk	•Not all application is compatible with RAD
•It is adaptable and flexible to changes	•When technical risk is high, it is not suitable
•It is easier to transfer deliverables as scripts, high-level abstractions and intermediate codes are used	•If developers are not committed to delivering software on time, RAD projects can fail

•Due to code generators and code reuse, there is a reduction of manual coding	•Reduced features due to time boxing, where features are pushed to a later version to finish a release in short period
•Due to prototyping in nature, there is a possibility of lesser defects	•Reduced scalability occurs because a RAD developed application begins as a prototype and evolves into a finished application
•Each phase in RAD delivers highest priority functionality to client	•Progress and problems accustomed are hard to track as such there is no documentation to demonstrate what has been done
•With less people, productivity can be increased in short time	•Requires highly skilled designers or developers

What Is DevOps?

- DevOps is a set of practices, tools, and a cultural philosophy.
- It automate and integrate the processes between software development and IT teams.
- It emphasizes team empowerment, cross-team communication and collaboration, and technology automation.

How it emerged:

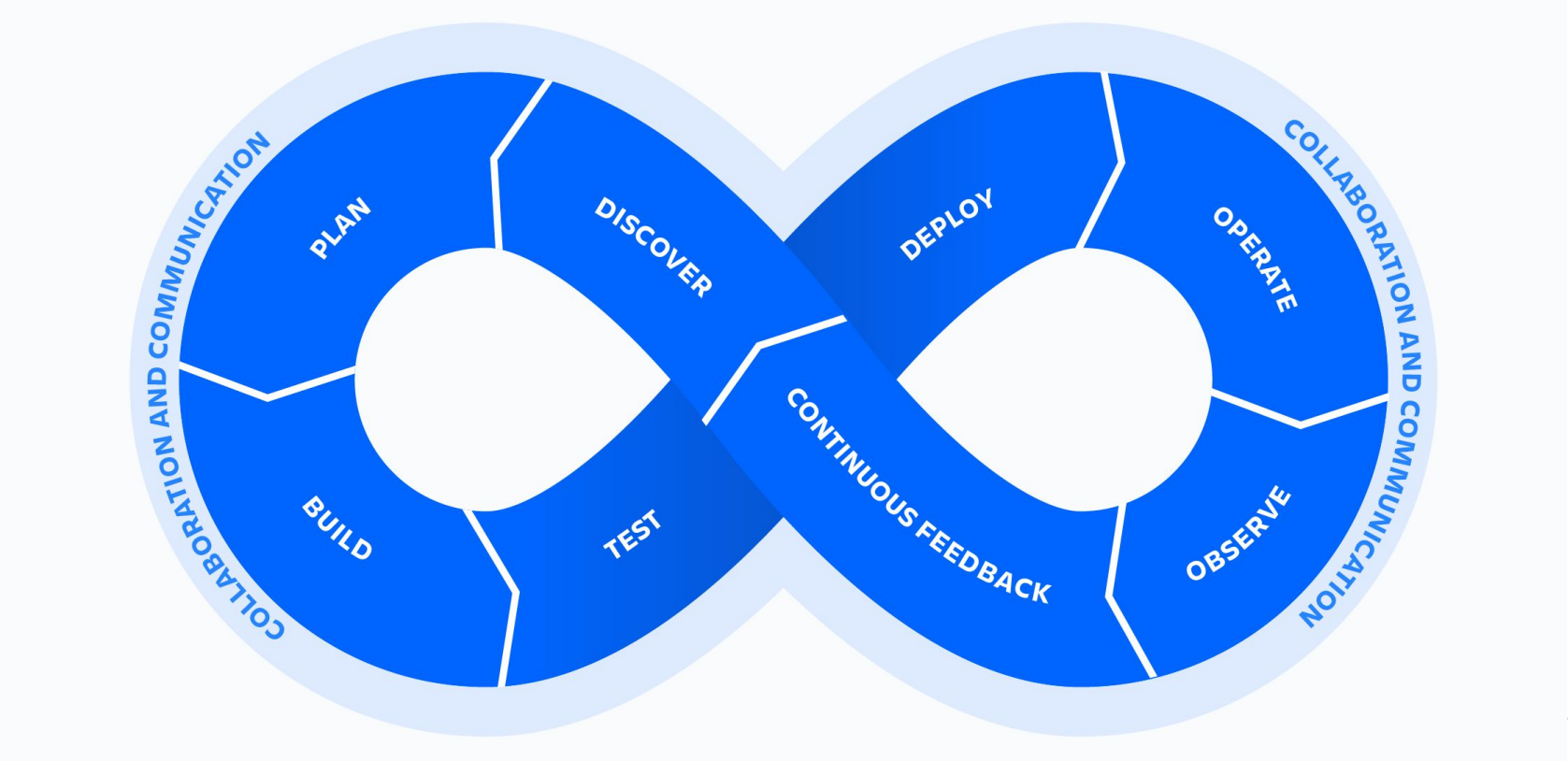
- The DevOps movement began around 2007 when the software development and IT operations communities raised concerns about the traditional software development model.
- where developers who wrote code worked apart from operations who deployed and supported the code.
- The term DevOps, a combination of the words development and operations, reflects the process of integrating these disciplines into one, continuous process.

Devops isn't any single person's job. It's everyone's job.

Robert Krohn

HEAD OF ENGINEERING, DEVOPS AT ATLASSIAN

The DevOps lifecycle



- The DevOps lifecycle consists of eight phases representing the processes, capabilities, and tools needed for development (on the left side of the loop) and operations (on the right side of the loop).
- Throughout each phase, teams collaborate and communicate to maintain alignment, velocity, and quality.

Discover

- Building software is a team sport.
- In preparation for the upcoming sprint, teams must workshop to explore, organize, and prioritize ideas.
- Ideas must align to strategic goals and deliver customer impact.
- Agile can help guide DevOps teams.

Plan

- DevOps teams should adopt agile practices to improve speed and quality.
- Agile is an iterative approach to project management and software development that helps teams break work into smaller pieces to deliver incremental value.

Build

- Git is a free and open source version control system.
- It offers excellent support for branching, merging, and rewriting repository history, which has led to many innovative and powerful workflows and tools for the development build process.

Test

- Continuous integration (CI) allows multiple developers to contribute to a single shared repository.
- When code changes are merged, automated tests are run to ensure correctness before integration.
- Merging and testing code often help development teams gain reassurance in the quality and predictability of code once deployed.

Deploy

- Continuous deployment (CD) allows teams to release features frequently into production in an automated fashion.
- Teams also have the option to deploy with feature flags, delivering new code to users steadily and methodically rather than all at once. This approach improves velocity, productivity, and sustainability of software development teams.

Operate

- Manage the end-to-end delivery of IT services to customers.
- This includes the practices involved in design, implementation, configuration, deployment, and maintenance of all IT infrastructure that supports an organization’s services.

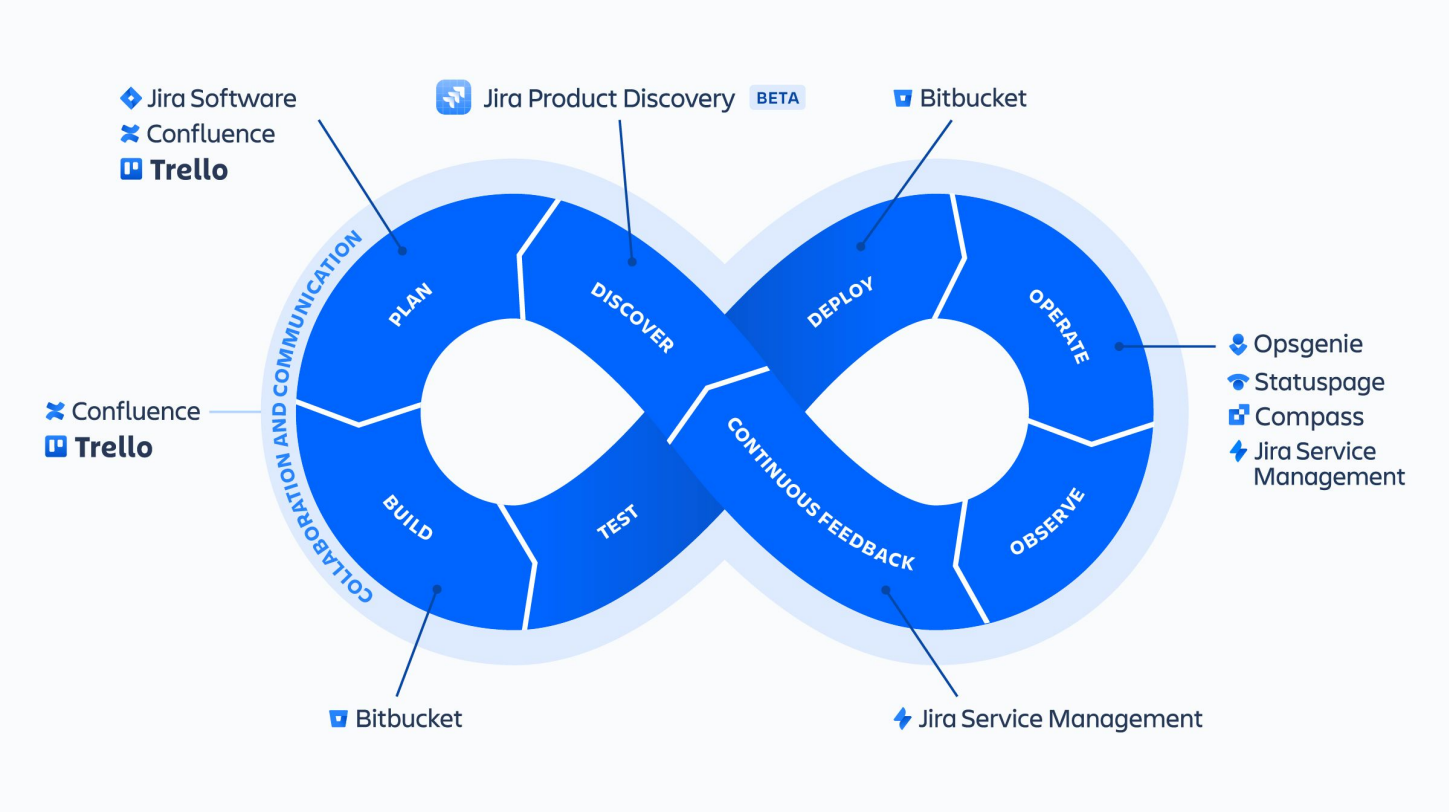
Observe

- Quickly identify and resolve issues that impact product uptime, speed, and functionality.
- Automatically notify your team of changes, high-risk actions, or failures, so you can keep services on.

Continuous feedback

- DevOps teams should evaluate each release and generate reports to improve future releases.
- By gathering continuous feedback, teams can improve their processes and incorporate customer feedback to improve the next release.

DevOps tools



What are the benefits of DevOps?

Speed

Teams that practice DevOps release deliverables more frequently, with higher quality and stability. In fact, the DORA 2019 State of DevOps report found that elite teams deploy 208 times more frequently and 106 times faster than low-performing teams. Continuous delivery allows teams to build, test, and deliver software with automated tools.

Improved collaboration

The foundation of DevOps is a culture of collaboration between developers and operations teams, who share responsibilities and combine work. This makes teams more efficient and saves time related to work handoffs and creating code that is designed for the environment where it runs.

Rapid deployment

By increasing the frequency and velocity of releases, DevOps teams improve products rapidly. A competitive advantage can be gained by quickly releasing new features and repairing bugs.

Quality and reliability

Practices like continuous integration and continuous delivery ensure changes are functional and safe, which improves the quality of a software product. Monitoring helps teams keep informed of performance in real-time.

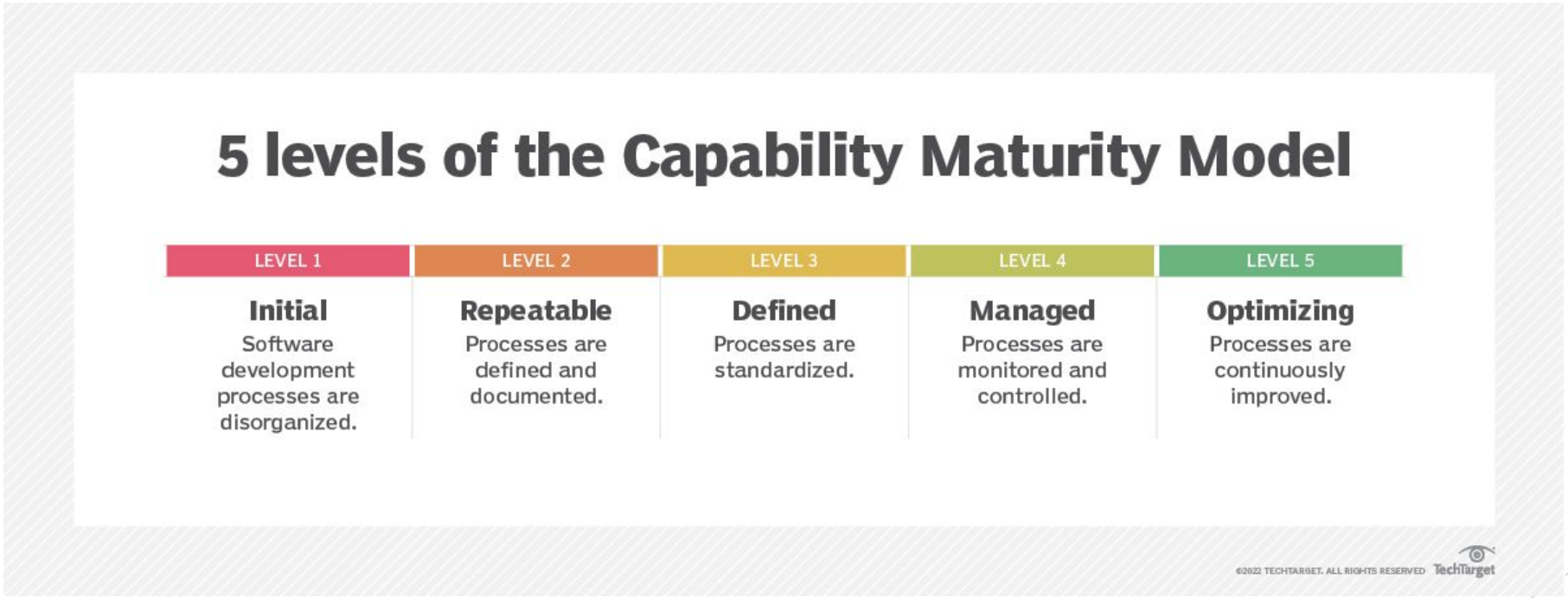
Security

By integrating security into a continuous integration, continuous delivery, and continuous deployment pipeline, DevSecOps is an active, integrated part of the development process. Security is built into the product by integrating active security audits and security testing into agile development and DevOps workflows.

What is Capability Maturity Model (CMM)?

- The Capability Maturity Model (CMM) is a methodology used to develop and refine an organization's software development process.
- The model describes a five-level evolutionary path of increasingly organized and systematically more mature processes.

- CMM was developed and is promoted by the Software Engineering Institute (SEI), a research and development center sponsored by the U.S. Department of Defense (DOD) and now part of Carnegie Mellon University. SEI was founded in 1984 to address software engineering issues and, in a broad sense, to advance software engineering methodologies.


5 levels of the Capability Maturity Model
LEVEL 1
Initial
Software development processes are disorganized.
LEVEL 2
Repeatable
Processes are defined and documented.
LEVEL 3
Defined
Processes are standardized.
LEVEL 4
Managed
Processes are monitored and controlled.
LEVEL 5
Optimizing
Processes are continuously improved.
©2022 TECHTARGET. ALL RIGHTS RESERVED
TechTarget

CMM's five levels of maturity for software processes

- Initial:
- At the initial level, processes are disorganized, ad hoc and even chaotic. Success likely depends on individual efforts and is not considered to be repeatable. This is because processes are not sufficiently defined and documented to enable them to be replicated.
- Repeatable:
- At the repeatable level, requisite processes are established, defined and documented. As a result, basic project management techniques are established, and successes in key process areas are able to be repeated.

- Defined:
- At the defined level, an organization develops its own standard software development process. These defined processes enable greater attention to documentation, standardization and integration.
- Managed:
- At the managed level, an organization monitors and controls its own processes through data collection and analysis.
- Optimizing:
- At the optimizing level, processes are constantly improved through monitoring feedback from processes and introducing innovative processes and functionality.

Activity for Next Lecture:

- Prepare a PPT of 2 Slides at most:
- Roll Nos: 1-10 (Real time application of **Waterfall** model)
- Roll Nos: 11-20 (Real time application of **Scrum** model)
- Roll Nos: 21-30 (Real time application of **Spiral** model)
- Roll Nos: 31-40 (Real time application of **V-Model** model)
- Roll Nos: 41-50 (Real time application of **RAD** model)
- Roll Nos: 51-60 (Real time application of **DevOps** model)
- Roll Nos: 61-71 (Real time application of **CMM** model)