CINEC CAMPUS
Beyond A Graduate

BSc (Hons) in Software Engineering

Course Code: GS3207
Software development methodology Lecture
Ethics & Professionalism

Software development methodology

Objectives

Recall the different Software Development Methodologies

Exploring Ethics in Software Development

Emphasizing Professionalism

Learning Outcomes

- Comprehensive understanding of Software Development methodologies
- Explain professionalism and ethics in software development.

A software development methodology

- A software development methodology is a structured approach or framework.
- It is used by software developers.
- It aims to plan, design, build, test, deploy, and maintain software systems.
- It provides a systematic and organized way to manage the software development process.
- It ensures that projects are completed efficiently and effectively.
- Different methodologies have emerged over time, each with its own set of principles, practices, and processes.

Software Development Methodologies

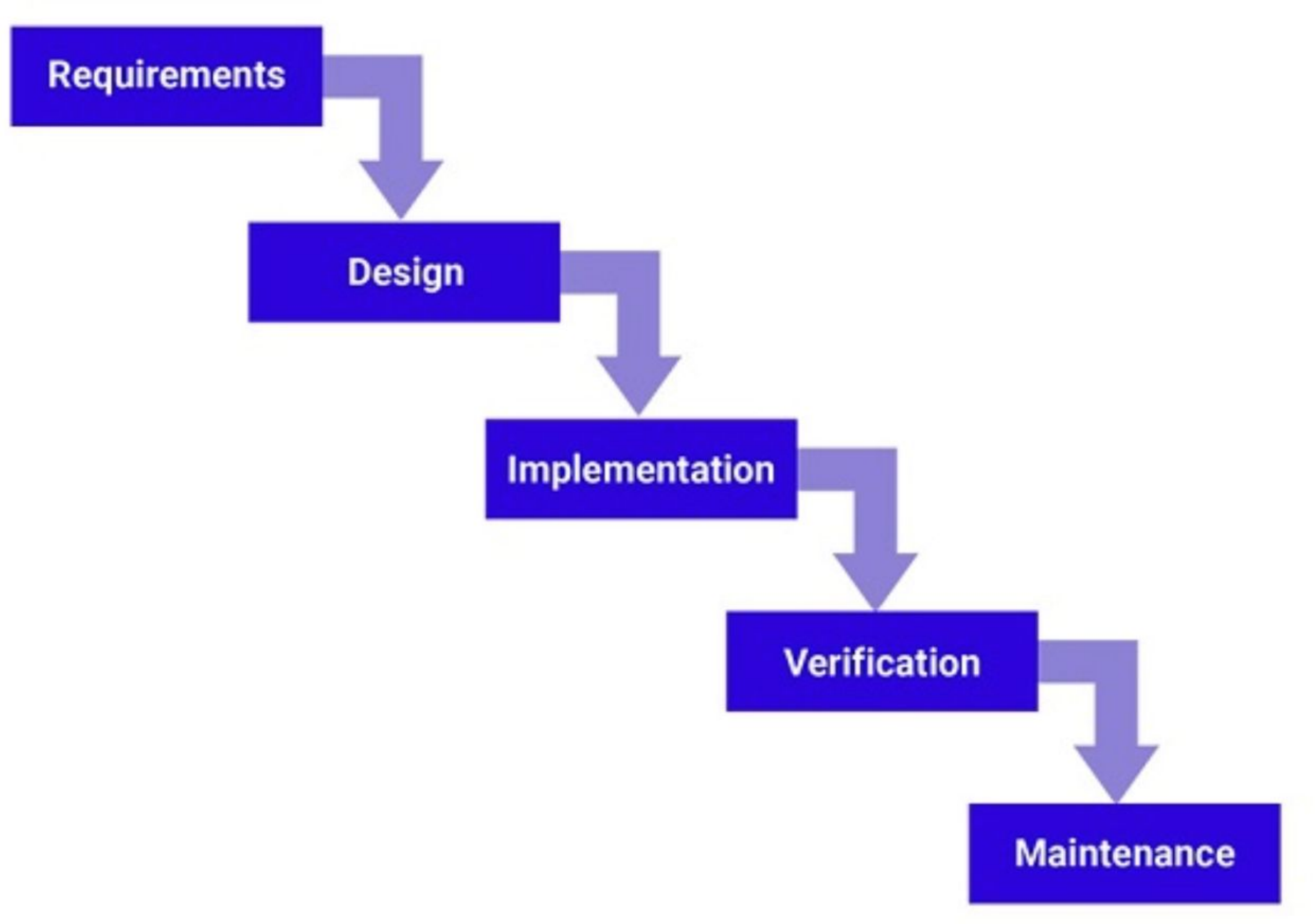
- **Waterfall Model**
- **Agile Development**
- **Rapid Application Development (RAD)**
- **DevOps**
- Iterative and Incremental Development
- Scrum
- Kanban
- Extreme Programming (XP)
- Spiral Model

All methodologies have different strengths and weaknesses and exist for different reasons

Waterfall Model

- Traditional software development method
- It is a rigid linear model that consists of sequential phases -requirements, design, implementation, verification, maintenance
- It is focusing on distinct goals.
- Each phase must be 100% complete before the next phase can start.
- There is no process for going back to modify the project or direction.

Waterfall Model

- 

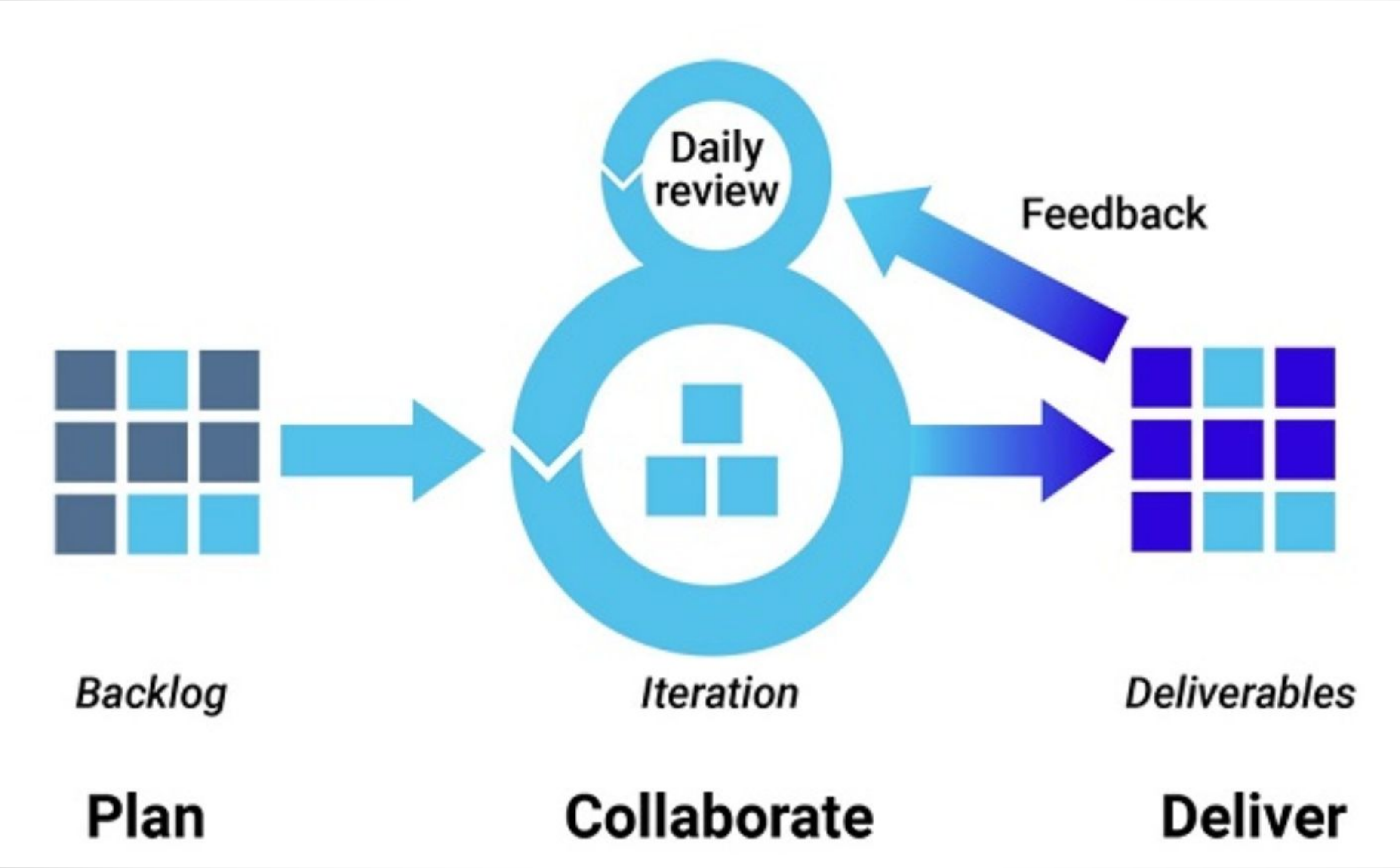
Pros & Cons: Waterfall Model

- Pros:
  - Easy to understand and manage.
  - Projects with clear objectives and stable requirements can best use the waterfall method.
  - Less experienced project managers and project teams, as well as teams whose composition changes frequently, may benefit the most from using the waterfall development methodology.
- Cons:
  - It is often slow and costly due to its rigid structure and tight controls.

Agile development methodology

- Teams use the agile development methodology to minimize risk (such as bugs, cost overruns, and changing requirements) when adding new functionality.
- In all agile methods, teams develop the software in iterations that contain mini-increments of the new functionality.
- There are many different forms of the agile development method, including scrum, crystal, extreme programming (XP), and feature-driven development (FDD).

Agile development methodology



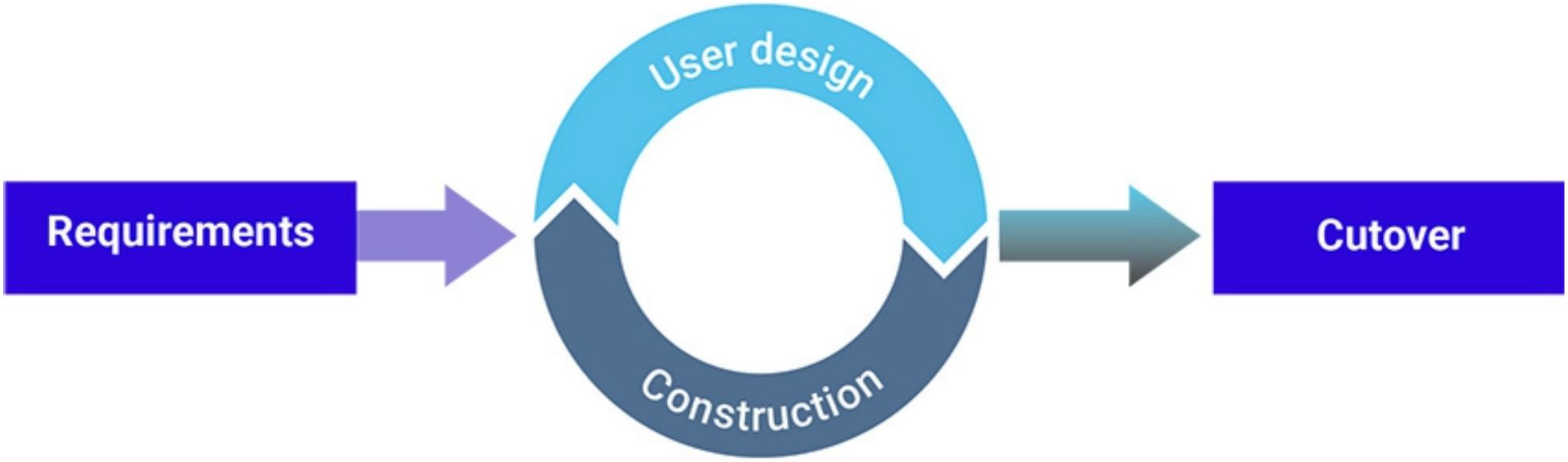
Pros & Cons: Agile Development Methodology

- Pros:
  - The primary benefit of agile software development is that it allows software to be released in iterations.
  - Iterative releases improve efficiency by allowing teams to find and fix defects and align expectation early on.
  - They also allow users to realize software benefits earlier, with frequent incremental improvements.
- Cons:
  - Agile development methods rely on real-time communication, so new users often lack the documentation they need to get up to speed.
  - They require a huge time commitment from users and are labor intensive because developers must fully complete each feature within each iteration for user approval.

Rapid Application Development

- Rapid application development (RAD) is a condensed development process that produces a high-quality system with low investment costs.
- This RAD process allows our developers to quickly adjust to shifting requirements in a fast-paced and constantly changing market.
- The ability to quickly adjust is what allows such a low investment cost.
- This method contains four phases: requirements planning, user design, construction, and cutover.
- The user design and construction phases repeat until the user confirms that the product meets all requirements.

Rapid Application Development



Pros & Cons: RAD Methodology

Pros:

- Rapid application development is most effective for projects with a well-defined business objective and a clearly defined user group, but which are not computationally complex.
- RAD is especially useful for small to medium projects that are time sensitive.

Cons:

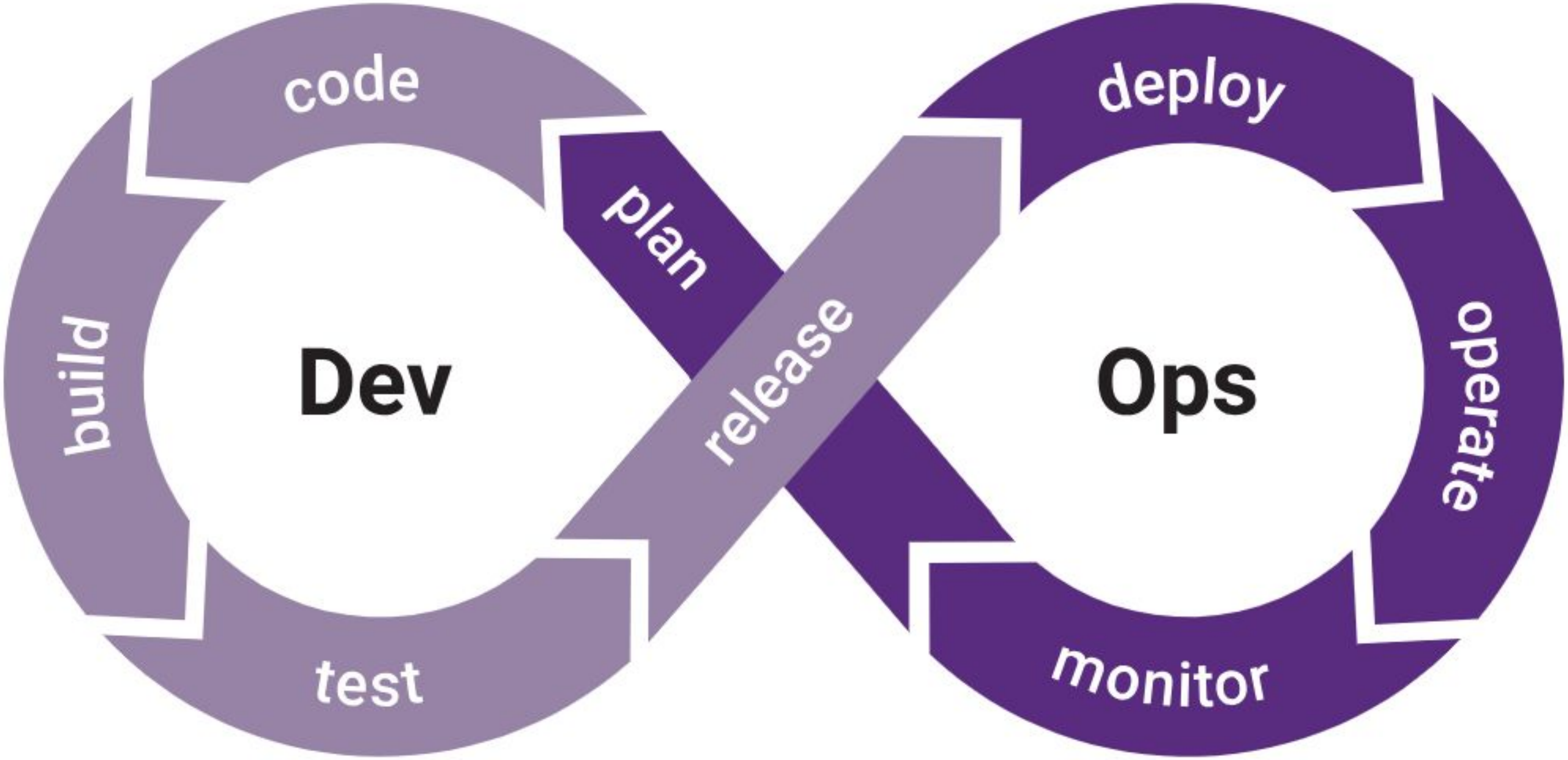
- Rapid application development requires a stable team composition with highly skilled developers and users who are deeply knowledgeable about the application area.
- Deep knowledge is essential in a condensed development timeline that requires approval after each construction phase.
- Organizations that don’t meet these requirements are unlikely to benefit from RAD.

DevOps deployment methodology

DevOps is not just a development methodology but also a set of practices that supports an organizational culture.

DevOps deployment centers on organizational change that enhances collaboration between the departments responsible for different segments of the development life cycle, such as development, quality assurance, and operations.

DevOps deployment methodology



Pros & Cons :DevOps deployment methodology

Pros:

- Significantly reducing time to market
- Lowering the failure rate of new releases
- Shortening the lead time between fixes
- Minimizing disruption
- Maximizing reliability
- Improving customer satisfaction
- Improve product quality
- Employee productivity and efficiency.

Pros & Cons :DevOps deployment methodology

Cons:

- Some customers don't want continuous updates to their systems.
- Some industries have regulations that require extensive testing before a project can move to the operations phase.
- If different departments use different environments, undetected issues can slip into production.
- Some quality attributes require human interaction, which slows down the delivery pipeline.

Exploring Ethics in Software Development

- Ethics and professionalism are crucial aspects in any software development methodology.

Next we will explore ethics and professionalism apply to software development:

Transparency and Honesty:

Requirement Gathering:

- Ethical software development begins with honest and transparent communication with stakeholders during the requirements gathering phase.
- Developers and project managers must ensure that client expectations are realistic and that requirements are clearly defined.

Data Security and Privacy

Design and Development:

- Professionals in a Waterfall model must ensure that data security and privacy considerations are integrated into the design and development phases.
- Ethical responsibility involves protecting user data and ensuring that sensitive information is handled securely.

Quality Assurance

Testing:

- Professionalism requires rigorous testing and quality assurance throughout the development process.
- Ethical developers should not compromise on the quality of the software, ensuring that it meets the specified requirements and functions correctly.

Client Interactions:

Project Management:

- Project managers and developers must maintain professionalism when dealing with clients.
- This includes delivering regular project status updates, managing expectations, and addressing client concerns promptly and honestly.

Documentation and Accountability

Project Documentation:

- Ethical software development involves thorough documentation of project plans, design specifications, and test results.
- This documentation ensures transparency and accountability

Change Management:

Change Requests:

- When clients request changes to the project scope or requirements, professionals in a model should handle these requests ethically by assessing the impact on the project timeline, budget, and scope.
- Ethical decisions should be made regarding whether to accept or reject changes.

Adherence to Deadlines:

Project Timelines:

- Meeting project deadlines is essential .
- Ethical professionals must work diligently to meet these deadlines and inform stakeholders promptly if delays are anticipated.

Professional Development:

Continual Learning:

- Ethical software developers in a environment should commit to ongoing professional development.
- Staying current with industry best practices and technologies is essential to delivering high-quality software.

Conflict Resolution:

Dispute Resolution:

In cases of disputes or conflicts within the development team or with clients, professionals should engage in ethical conflict resolution practices that prioritize fairness and cooperation.

Accountability and Responsibility:

Defects and Failures:

- When defects or failures occur in the software, professionals should take responsibility, conduct root cause analysis, and implement corrective actions promptly and transparently.

Legal Compliance:

Intellectual Property:

- Ethical professionals should respect intellectual property rights, such as copyrights and patents, and ensure that the software they develop does not infringe on these rights.

Sustainability

Environmental Responsibility:

Considerations for sustainability and minimizing the environmental impact of software should also be part of ethical software development practices.

Brief of the Next Lecture

Capability Maturity Model Integration for software

safety-critical system

Thank You.