ISO/IEC 12207

IEEE/EIA 12207

SOFTWARE LIFE CYCLE PROCESSES

SUPPORTING LIFE CYCLE PROCESSES
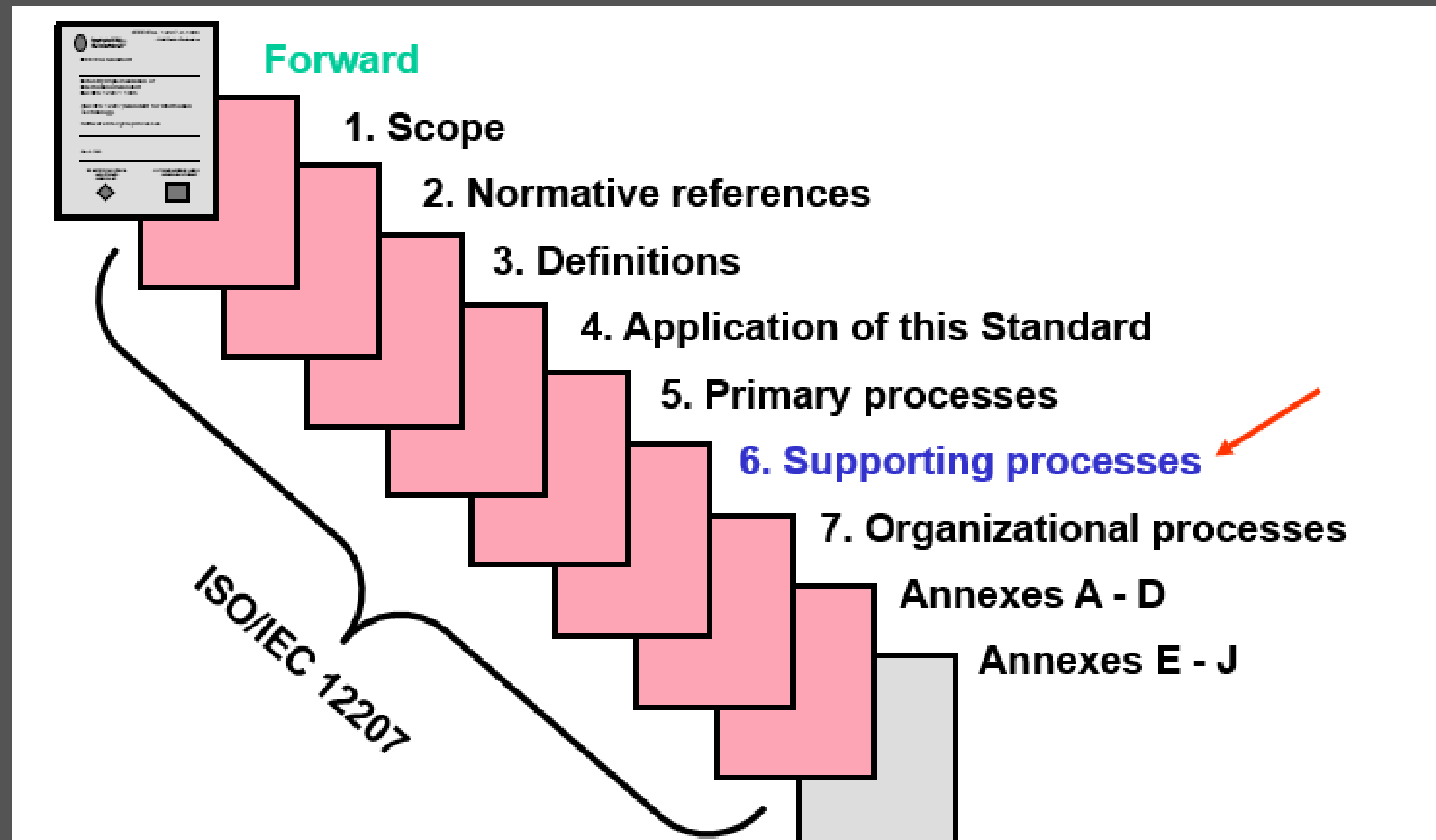
by

Nishant   A053

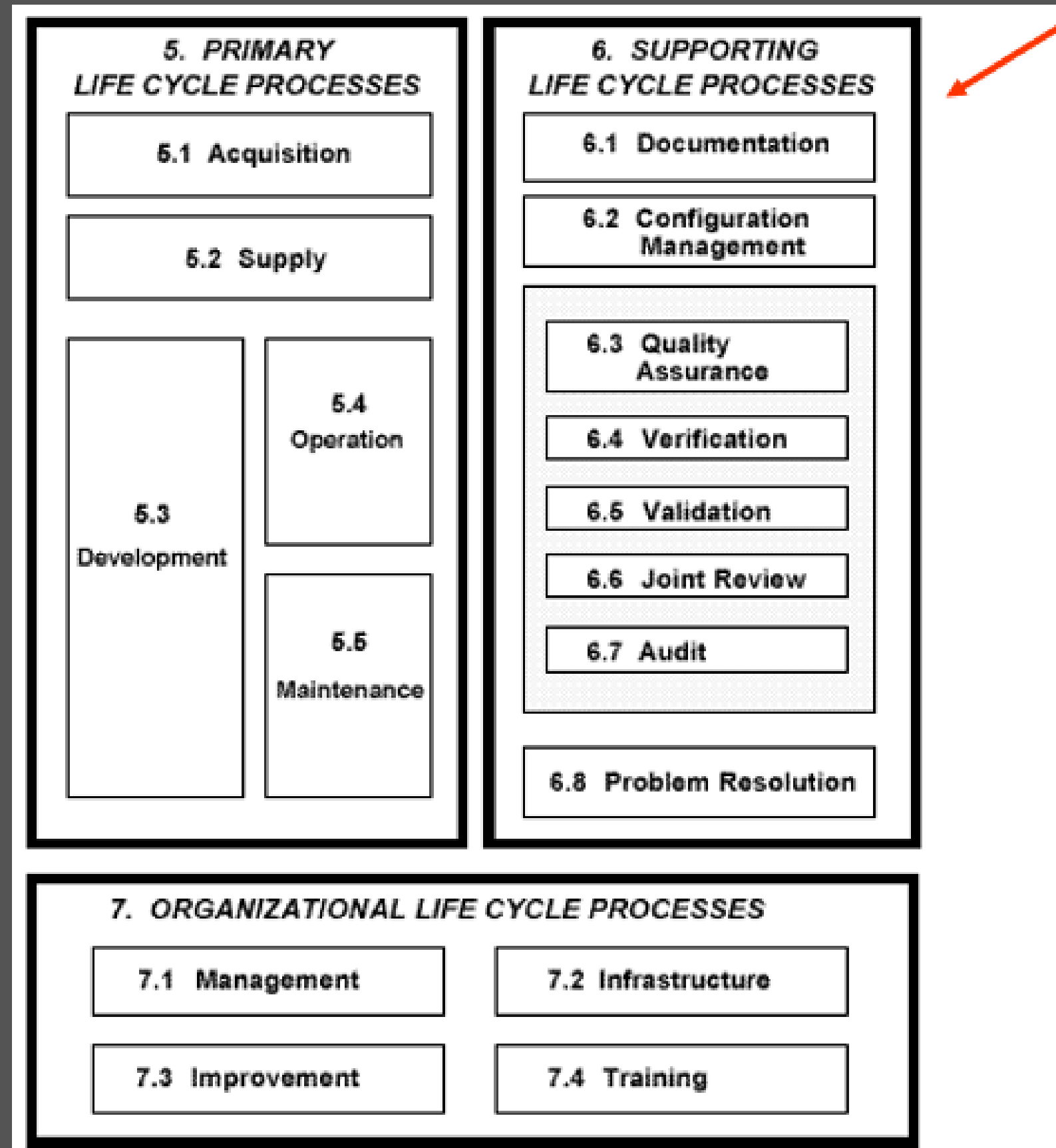Anudeep   A012

OVERVIEW OF 12207

IEEE 12207 is a standard developed by the Institute of Electrical and Electronics Engineers (IEEE) that specifies the processes involved in the software life cycle. The full title of the standard is "IEEE Std 12207-2008, Standard for Systems and Software Engineering - Software Life Cycle Processes." It provides a framework for the development, acquisition, and maintenance of software systems.

OUTLINE OF IEEE/EIA 12207 "SOFTWARE LIFE CYCLE PROCESSES"

12207 LIFE-CYCLE PROCESSES

5. PRIMARY LIFE CYCLE PROCESSES

- 5.1 Acquisition
- 5.2 Supply
- 5.3 Development
- 5.4 Operation
- 5.5 Maintenance

6. SUPPORTING LIFE CYCLE PROCESSES

- 6.1 Documentation
- 6.2 Configuration Management
- 6.3 Quality Assurance
- 6.4 Verification
- 6.5 Validation
- 6.6 Joint Review
- 6.7 Audit
- 6.8 Problem Resolution

7. ORGANIZATIONAL LIFE CYCLE PROCESSES

- 7.1 Management
- 7.2 Infrastructure
- 7.3 Improvement
- 7.4 Training

CONTENT

QUALITY ASSURANCE PROCESS

Quality Assurance Process is one of the primary processes that ensures that the software development and maintenance processes meet their specified objectives and that the resulting products are of high quality

- PROCESS IMPLEMENTATION

- PRODUCT ASSURANCE

- PROCESS ASSURANCE

- ASSURANCE OF QUALITY SYSTEM

VERIFICATION PROCESS

The verification process, as outlined in IEEE 12207, is a part of the software development life cycle and focuses on ensuring that work products (such as software code, documentation, and design specifications) meet specified requirements and standards.

- PROCESS IMPLEMENTATION

- CONTRACT VERIFICATION

- PROCESS VERIFICATION

- REQS. VERIFICATION

- DESIGN VERIFICATION

- CODE VERIFICATION

- INTEGRATION VERIFICATION

- DOC. VERIFICATION

VALIDATION PROCESS

In IEEE 12207, the validation process is a critical aspect of software development that focuses on ensuring that the final software product meets the specified requirements and functions correctly in its intended operational environment. The validation process is essential for confirming that the developed software satisfies the needs of the end-users and complies with the defined criteria. Here are key components of the validation process in IEEE 12207:

- Requirements Validation

- System Design Validation

- Software Validation

- Integration and Interface Validation

- Operational Validation

- Compliance Validation

JOINT REVIEW PROCESS

In IEEE 12207, the Joint Review Process is a type of review activity that involves collaboration between different stakeholders, typically from both the development and customer sides, to assess and discuss work products. Joint reviews are intended to facilitate communication, understanding, and agreement on various aspects of the software development process. This process is part of the broader category of reviews and inspections conducted during the software life cycle.

- Collaborative Evaluation

- Communication

- Clarification of Requirement

- Identification of Issues

- Decision-Making

- Formal Documentation

AUDIT PROCESS

In IEEE 12207, the Audit Process refers to a set of activities conducted to independently examine and evaluate processes and work products within the software life cycle. Audits are systematic examinations performed to ensure compliance with defined processes, standards, and requirements. The purpose of the Audit Process is to verify that the software development activities are carried out according to the established plans and procedures and to identify areas for improvement.

- Objective Evaluation

- Independence

- Compliance Checking

- Identification of Nonconformities

- Root Cause Analysis

- Recommendations for Improvement

- Follow-up Activities

PROBLEM RESOLUTION PROCESS

In IEEE 12207, the Problem Resolution Process is a set of activities aimed at identifying, analyzing, and resolving problems that arise during the software development life cycle. This process is essential for managing and mitigating issues that may impact the quality, schedule, or performance of the software being developed. The Problem Resolution Process involves systematic steps to address and correct problems in an effective and timely manner.

- Problem Identification

- Logging and Documentation

- Analysis and Classification

- Assigning Responsibility

- Resolution Planning

- Implementation of Solutions

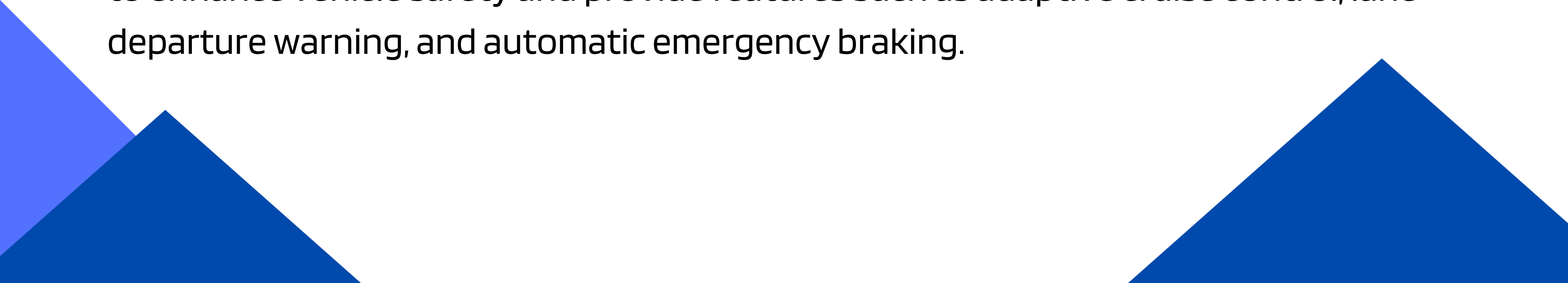- Verification and Validation

- Closure and Documentation

CASE STUDY

Automotive Software Development

Project: Development of Advanced Driver Assistance System (ADAS) Software
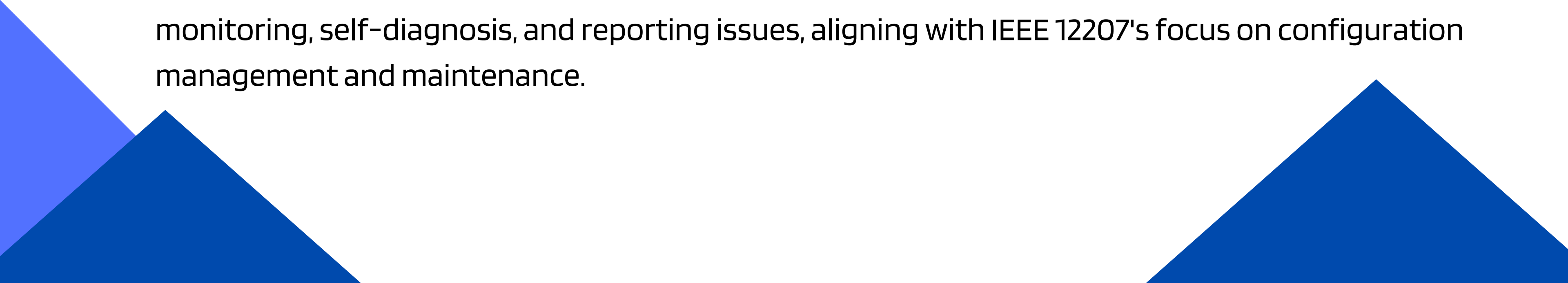
Description:

In this case study, we'll explore the application of IEEE 12207 standards in the development of software for an **Advanced Driver Assistance System (ADAS)** in the automotive industry. ADAS involves the integration of various software components to enhance vehicle safety and provide features such as adaptive cruise control, lane departure warning, and automatic emergency braking.

Contd.....

Key Components,features/Requirements for the Automotive Software:

1. **Sensor Integration**: The software interacts with sensors like cameras, radar, and lidar to capture real-time data about the vehicle's surroundings.
2. **Decision-Making Algorithms**: Complex algorithms are developed to process sensor data and make decisions, such as detecting obstacles, recognizing road signs, and determining optimal driving paths.
3. **HMI (Human-Machine Interface):** The software includes interfaces for drivers to receive alerts, warnings, and engage with ADAS features through the vehicle's dashboard display.
4. **Integration with Vehicle Control Systems:** ADAS software must seamlessly integrate with the vehicle's control systems, such as the engine control unit (ECU) and braking system.
5. **Continuous Monitoring and Diagnostics**: The system includes mechanisms for continuous monitoring, self-diagnosis, and reporting issues, aligning with IEEE 12207's focus on configuration management and maintenance.
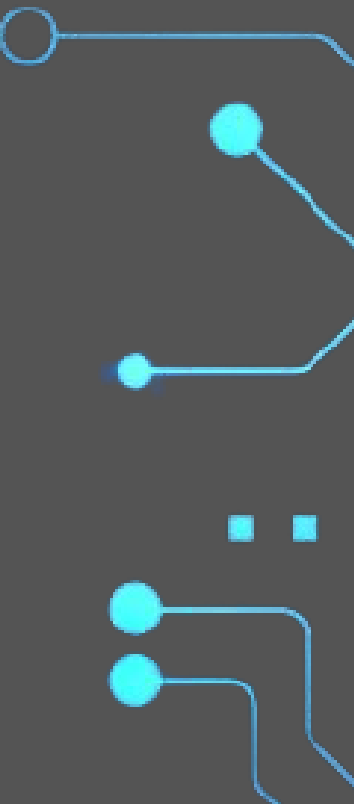
Contd.....

Application of IEEE 12207:

- Planning Phase

- Development and Integration Phase

- **Validation and Verification Phase**

- **Joint Review Process**

- **Audit Process**

- **Problem Resolution Process**

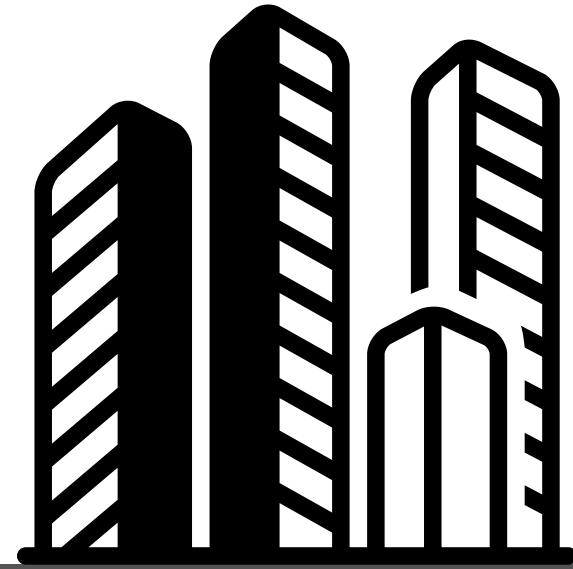- Operation and Maintenance Phase
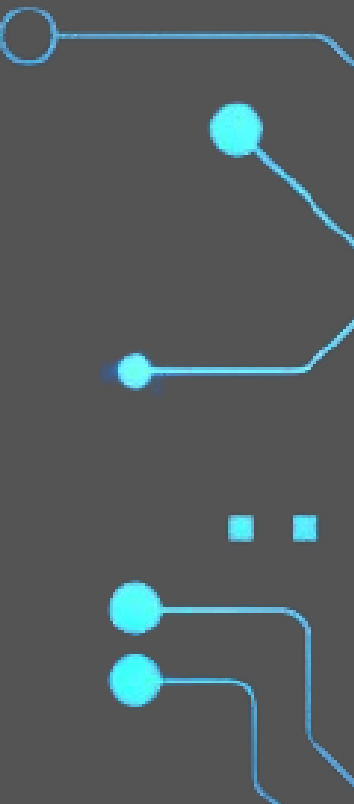
- Documentation

ADVANTAGES OF 12207
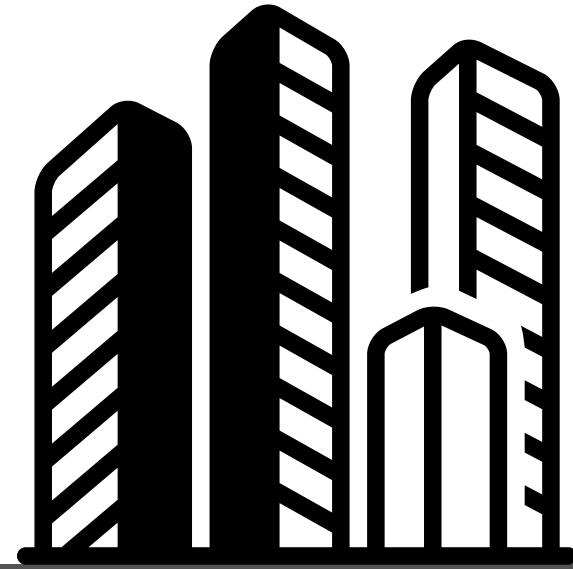
- Standardization

- Interoperability

- Risk Reduction

- Quality Improvement

- Compliance with Regulations

DISADVANTAGES OF 12207

- Complexity

- Rigidity

- Resource Intensive

- Not One-Size-Fits-All

- Training Requirements

12207 VS 9001

Aspect	IEEE 12207	ISO 9001
Scope	Software life cycle processes	Quality management system for any industry
Focus	Software development and maintenance	Overall quality management across various processes
Applicability	Specific to the software industry	Applicable to any industry or organization
Processes	Defines processes related to software development, testing, configuration management, and maintenance	Encompasses a broader range of processes, including management, resource, and support processes
Documentation	Emphasizes detailed documentation for traceability, quality assurance, and process compliance	Requires documentation but focuses on documentation essential for quality management and improvement
Certification	Not explicitly a certification standard but provides guidance for software processes	Can lead to ISO 9001 certification for organizations
Customer Focus	Addresses customer requirements within the software context	Emphasizes meeting customer needs across all organizational processes
Risk Management	Considers risk management within the software life cycle	Encourages risk-based thinking for overall process improvement
Adaptability	Perceived as more tailored to software development environments	Offers flexibility for implementation in various industries
International Recognition	Recognized in the software engineering field internationally	Widely recognized and adopted across diverse industries
Continuous Improvement	Emphasizes continuous improvement in software processes	Focuses on continual improvement in the overall quality management system

CONCLUSION

- In conclusion, IEEE 12207 stands as a cornerstone in the realm of software engineering, delineating a comprehensive framework that orchestrates the entire software life cycle. From conception to retirement, its structured processes ensure efficiency, reliability, and adherence to quality standards.

- Furthermore, the international recognition of IEEE 12207 underscores its relevance and significance in the global software engineering landscape. Its emphasis on continuous improvement aligns seamlessly with the dynamic nature of the industry, fostering innovation and resilience.

Thank you