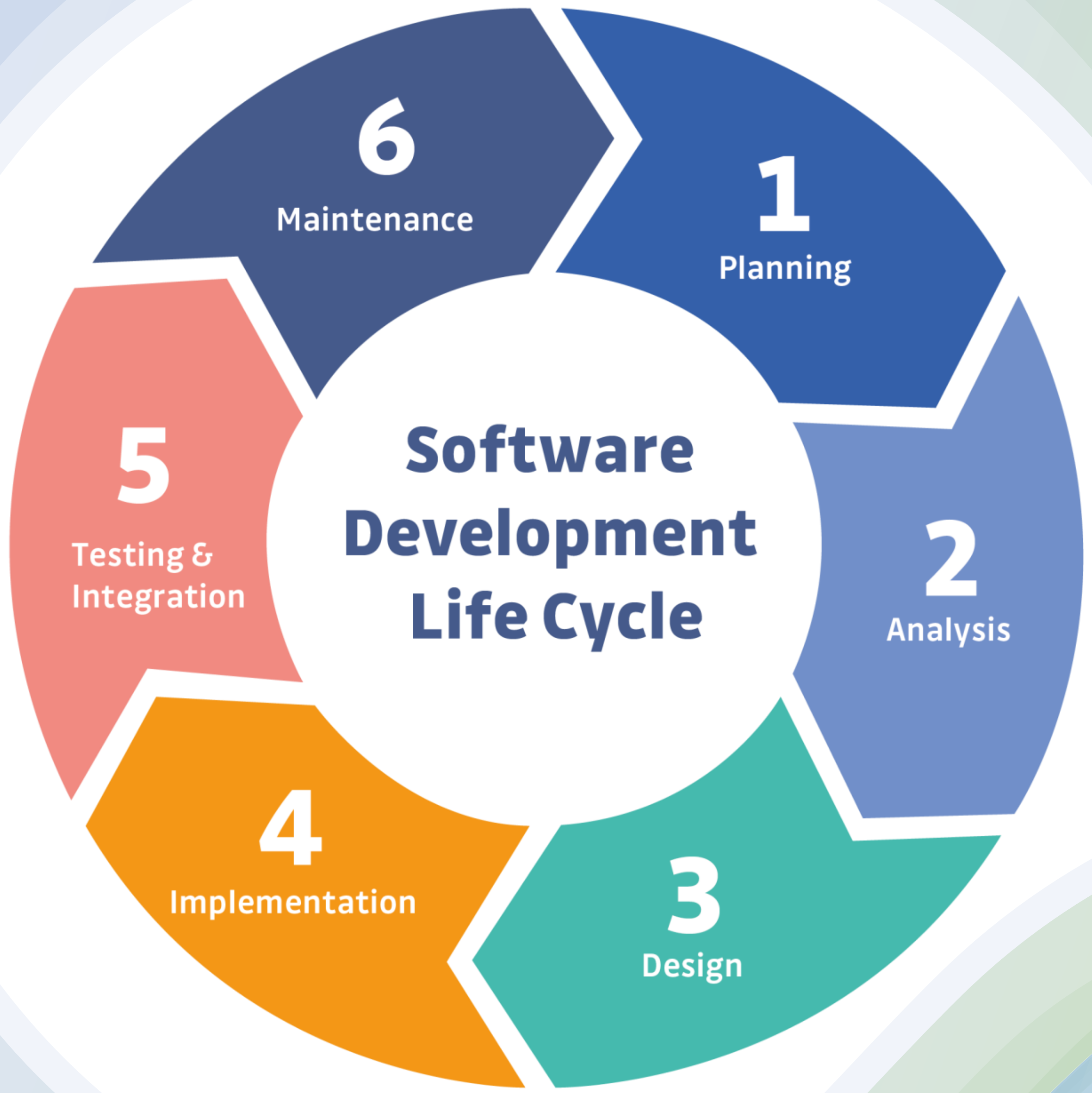
ILP Batch 1 Business Orientation

Reverse Knowledge Transfer

Nathaniel Yeldo
Trainee


Software Development Life Cycle
1
Planning
2
Analysis
3
Design
4
Implementation
5
Testing & Integration
6
Maintenance

Agile

- Agile is a project management and product development approach that prioritizes flexibility, collaboration, and customer satisfaction
- Key Principles:
    - Individuals and interactions over processes and tools
    - Working solutions over comprehensive documentation
    - Customer collaboration over contract negotiation
    - Responding to change over following a plan

Agile Ceremonies

- Sprint Planning
- Daily Standup
- Sprint Review
- Sprint Retrospective

Feature	Scrum	Kanban
Framework	Structured with roles and ceremonies	Flexible without predefined roles
Time-Boxing	Fixed-length iterations (Sprints)	Continuous flow, no fixed time boxes
Roles	Specific roles (Scrum Master, etc.)	No prescribed roles, uses existing team roles
Planning	Fixed planning in Sprints	Continuous planning and adaptability
Changes and Flexibility	Changes discouraged during a Sprint	Emphasizes flexibility and adaptability
Visualization	Sprint Burndown charts and boards	Kanban boards for continuous visualization

Microsoft Teams for Effective Communication

Features:

Chat

Video or Audio Calls

Benefits:

Efficient

Centralized

Enhanced Collaboration

Tips:

Use Channels

@Mentions

Schedule Meetings

Best Practices:

Set Status

Encourage Open Communication

Ensure Security

Project Management with JIRA

JIRA is an Agile project management tool

Promotes collaboration and transparency

Key Features:

- **Task Tracking:** Customizable boards for progress
- **Workflows:** Adaptable processes
- **Dashboards:** Visual project data for decisions

Collaborative Environment:

- **Real-Time Updates:** Instant team visibility
- **Comments and Attachments:** Enhanced communication
- **Integration:** Seamless workflow with other tools

Backlog Management in JIRA

Understanding Backlog:

Epics: High-level work containers

Features: Groups related user stories

User Stories: Granular tasks for end-user functionality

Managing Tasks:

Task Creation: Break down user stories

Assignment: Assign tasks for ownership

Sprint Backlog: Plan and track tasks per sprint

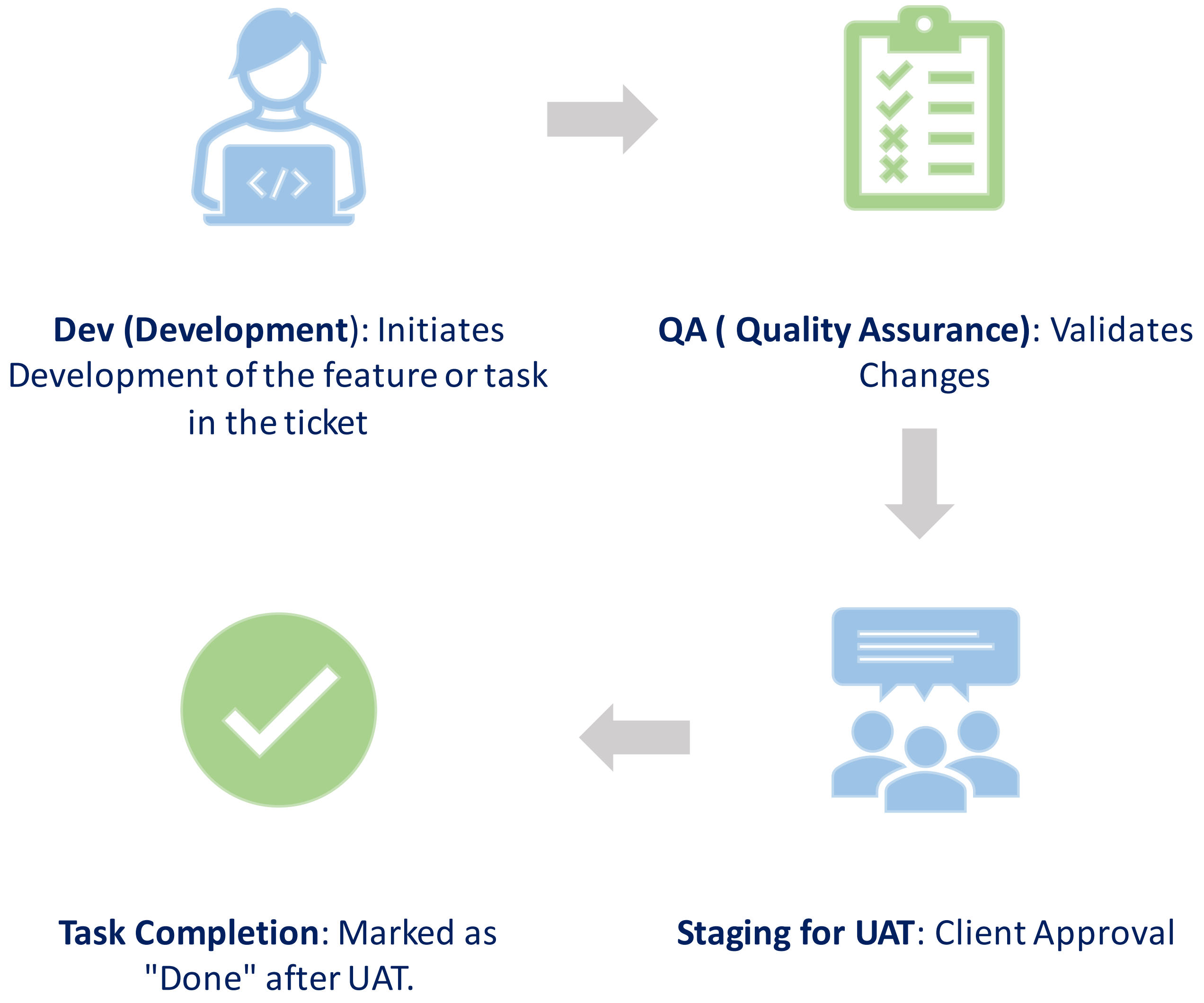
Visualizing Backlog:

Backlog View: Prioritize and organize

Drag-and-Drop: Easily reorder items

Filtering and Searching: Efficient item management

Team Workflow



Dev (Development): Initiates Development of the feature or task in the ticket

QA (Quality Assurance): Validates Changes

Staging for UAT: Client Approval

Task Completion: Marked as "Done" after UAT.

Importance of Clean and Readable Code

IMPORTANCE

- Efficiency
- Collaboration
- Bug Prevention
- Scalability
- Professionalism
- Overall Impact
- Code Reusability

BEST PRACTICES

- Descriptive Naming
- Consistent Formatting
- Comments for Clarity
- Modularization
- Avoid Magic Numbers
- Regular Code Reviews

Version Control System - GitHub

- Utilize GitHub for Version Control
- Branching
- Handling Conflicts
- Commit Best Practices
- Gitignore File
- Remote Repositories
- Code Review

IDE - Visual Studio & Debugging

- Powerful IDE for coding, testing, and debugging
- Supports multiple programming languages
- Debugging tools seamlessly embedded in Visual Studio
- Utilization of breakpoints in the IDE for debugging and **Step Into, Over, and Out**

Low-Level Design (LLD)

- **Definition:** Detailed design phase bridging high-level design and coding
- **Components:** Data Structures, Algorithms, Interfaces, UML Diagrams
- **Coding Guidelines:** Follow standards, encourage reuse and modularity
- **Testing:** Identify test cases and plan error handling
- **Collaboration:** Team coordination, thorough documentation
- **Tools:** Use Case Tools for UML, Version Control

Unit Testing

Small, isolated tests done by Developers

Helps in early bug detection and makes the code stable

Can be automated by integrating with CI/CD

Aim for high test coverage and hence recognize areas lacking tests

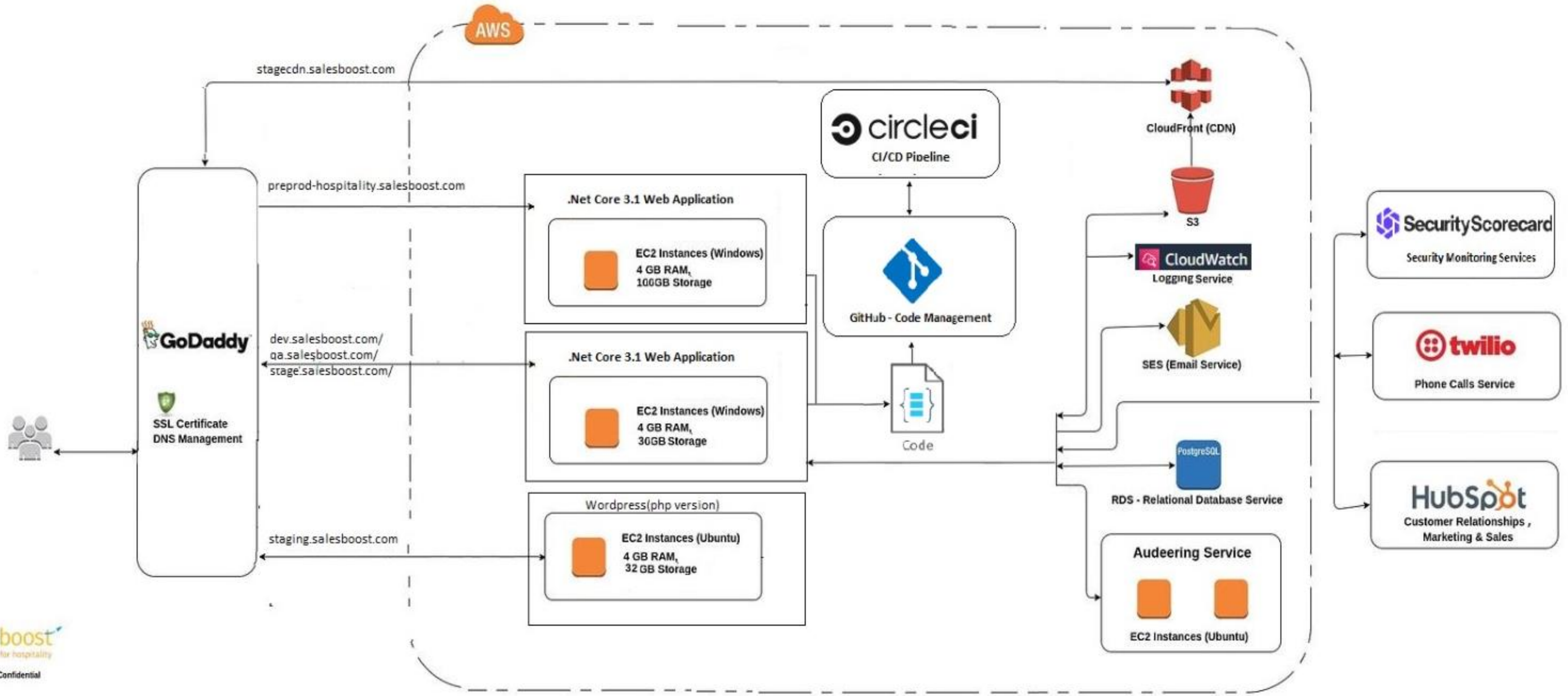
AAA (Arrange, Act, Assert) pattern is followed for writing test cases

Tests should be independent and named with clarity.

CI/CD Overview

- **Continuous Integration (CI)**: Merge code changes regularly and validate changes automatically
- **Continuous Deployment (CD)**: Automate deployment and ensure consistent, error-free releases
- **Tools**: CircleCI (Cloud-based CI/CD Tool)
- **Key Benefits:**
    - Automate repetitive tasks
    - Quick feedback on code changes
    - Rigorous testing before deployment

Project Architecture



esboost for hospitality
oost Confidential

SonarQube

- Code Quality Management Tool.
- Used for:
  - Static Analysis.
  - Duplication Detection.
  - Code Coverage Analysis.
  - Security Vulnerability Scanning.
- Seamless CI/CD Integration.
- Early Issue Detection.
- Continuous Improvement.

Quality Assurance & Testing

- Systematic process for ensuring high-quality products.
- Stages
    - Planning
    - Execution
    - Monitoring
    - Feedback and Improvement

Production Release

- Production DB and Application Build.
- Deploy to Production, Update Settings.
- Execute migration process.
- Verify AWS CloudWatch logs.
- Perform Smoke Test.
- Verify successful deployment.
- Backup Application, Git Tagging.

Project Documentation – Confluence Page

- Roadmap Plan
- Product Requirements
- Non-functional Requirements
- Retrospectives
- Configuration Files
- Meeting Notes
- References
- Technical Documents (LLD, Architecture design, etc.)

Business Orientation - Takeaways

Project Management:
- Daily standups, JIRA, and Git.
- Understanding of SDLC and agile methodologies.

Technical Skills:
- Efficient use of Microsoft Teams and Azure DevOps.
- Practical learning of Kanban, Scrum, and collaboration tools.

Agile Methodologies and Collaboration:
- Participation in Agile ceremonies.
- Exploration of collaboration tools like Microsoft Teams.

Technical Tools and Security:
- Introduction to SonaQube for security.
- Real-world experience in addressing production issues.

Documentation and Communication:
- Utilization of Confluence for project documentation.
- Emphasis on effective issue reporting and secure coding.

Career Growth:
- Mentorship advice and understanding company culture.
- Focus on personal ownership and positive work attitudes.

Thank You!