	Akhil Bharatiya Maratha Shikshan Parishad's Anantrao Pawar College of Engineering & Research	
	Record No.: ACA/D/021 **Revision: 00** | **DoI: 02/01/2023**	

EVENT REPORT

Name of Event: FARM Software Development Process Training

Date of Event: 16th January 2024

Time of event: 03.00 pm to 04.30 pm

Name of Event Coordinator: Prof. Dewendra Bharambe

Target Audience with count:

Students:

Computer Engineering→ 05
Information Technology→06
Electronics & Telecommunication→06
AI & DS → 05

Faculty: 06

Brief Description of Event:

Capgemini has organized online training on FARM Software Development Process Training with collaboration of IICARE Foundation Dr. Santosh Bhosale sir. Software Development Life cycle, Agile Software, Software Development, Scrum, Kanban, Extreme Programming, Lean Software Development, FARM High level Architecture, Testing & Maintenance participated by Computer Engineering, Information Technology, Electronics & Telecommunication, Artificial Intelligence & Data Science staff and students.

SDLC (Software Development Life Cycle)

The Software Development Life Cycle (SDLC) is a systematic process for planning, creating, testing, deploying, and maintaining software applications or systems. It provides a framework for developers and project managers to follow, ensuring that software development projects are well-organized and produce high-quality software. While there are various SDLC models, the following phases are common to most of them.

Anantrao Pawar College of Engg. & Research Pune ★ Pune-9 ★

	Akhil Bharatiya Maratha Shikshan Parishad's Anantrao Pawar College of Engineering & Research	
	Record No.: ACA/D/021 Revision: 00	DoI: 02/01/2023

EVENT REPORT

Planning:

Define the project scope, goals, and objectives. Assess the feasibility of the project, considering technical, economic, legal, and scheduling factors. Create a project plan, including timelines, resource requirements, and risk assessment.

Analysis:

Gather and analyze requirements from stakeholders, including end-users. Document functional and non-functional requirements to understand what the software should do and how it should perform.

Design:

Develop the system architecture and design based on the requirements. Create detailed technical specifications, including data structures, algorithms, and user interfaces. Define the overall system architecture, subsystems, and their interactions.

Implementation (Coding):

Write the actual code based on the design specifications. Follow coding standards and best practices. Conduct unit testing to ensure individual components work as intended.

Testing:

Perform various testing activities, including unit testing, integration testing, system testing, and acceptance testing. Identify and fix defects or issues found during testing. Ensure that the software meets the specified requirements and quality standards.

Deployment:

Release the software for production use. Conduct user training, if necessary. Monitor and address any issues that arise during the initial deployment phase.

Maintenance:

Address bugs and issues identified post-deployment. Make necessary updates and enhancements to keep the software relevant. Handle ongoing support and maintenance activities. It's important to note that the SDLC is not a linear process, and iterations or feedback loops may occur between

Anantrao Pawar College of Engg. & Research Pune Pune-9

phases. Additionally, different SDLC models, such as Agile, Waterfall, or Spiral, may prioritize and organize these phases in different ways to suit the specific needs of a project or organization.

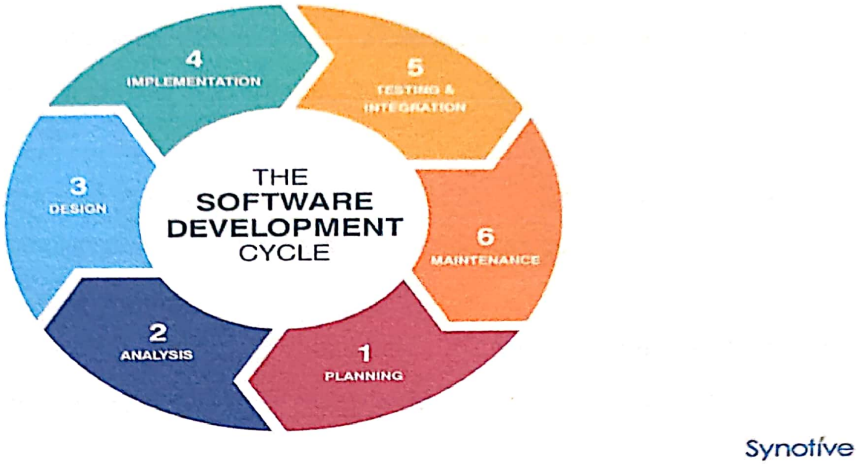

Fig- Software Development Life Cycle

Agile Software Development

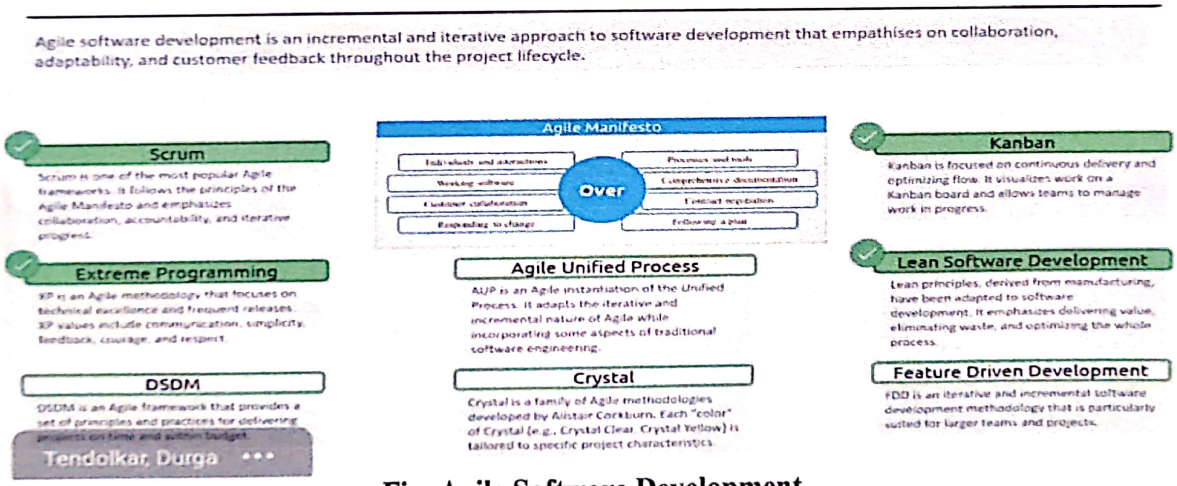

Fig. Agile Software Development

Agile Software Development is an iterative and incremental approach to software development that prioritizes flexibility, collaboration, and customer satisfaction. It emerged as a

response to the limitations of traditional, sequential development methodologies. Some key principles and concepts associated with Agile include:

Iterative Development:

Agile projects are divided into small increments, typically called iterations or sprints. Each iteration results in a potentially shippable product increment.

Collaboration:

Emphasizes regular communication and collaboration among team members, stakeholders, and customers. Cross-functional teams work together throughout the project.

Customer Involvement:

Customers and end-users are involved throughout the development process. Continuous feedback is sought and incorporated to ensure the final product meets user expectations.

Adaptability:

Agile methodologies are designed to be adaptable to changes in requirements, even late in the development process. Changes are seen as a natural part of the development cycle.

Continuous Delivery:

Focuses on delivering a working product incrementally and regularly. Prioritizes delivering high-value features first.

Scrum:

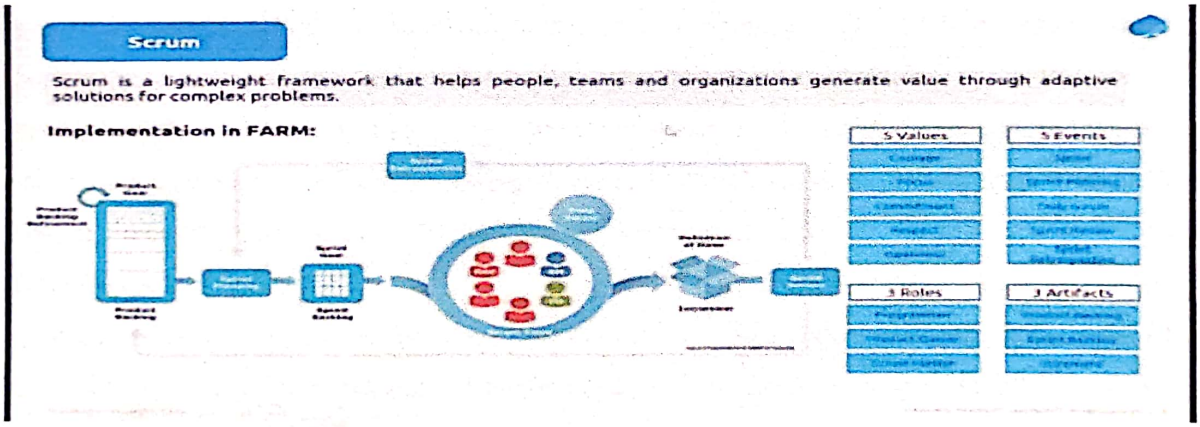

Fig. Scrum

Anantrao Pawar College of Engg. & Research Pune-9 Pune

	Akhil Bharatiya Maratha Shikshan Parishad's Anantrao Pawar College of Engineering & Research	
	Record No.: ACA/D/021 Revision: 00	DoI: 02/01/2023

EVENT REPORT

Scrum is an agile framework for managing and organizing work, particularly in software development. It provides a flexible and iterative approach to project management, emphasizing collaboration, adaptability, and customer satisfaction. Scrum was originally designed for software development teams, but its principles have been successfully applied to various industries and projects.

Key components of Scrum include:

Roles:

Product Owner: Represents the stakeholders and ensures that the team is working on the most valuable features. Scrum Master: Facilitates the Scrum process, removes impediments, and ensures that the team follows the Scrum framework. Development Team: Cross-functional group responsible for delivering a potentially shippable product increment at the end of each sprint.

Artifacts:

Product Backlog: A prioritized list of all features, enhancements, and fixes needed to deliver a product. Sprint Backlog: The subset of the product backlog chosen for a particular sprint, representing the work the team commits to completing during that sprint. Increment: The sum of all completed product backlog items at the end of a sprint.

Events:

Sprint Planning: A meeting at the beginning of each sprint where the team selects items from the product backlog and plans how to complete them. Daily Scrum: A daily stand-up meeting where the team discusses progress, plans for the day, and addresses any obstacles. Sprint Review: A meeting at the end of each sprint to review and demonstrate the completed work to stakeholders. Sprint Retrospective: A reflection meeting at the end of each sprint to discuss what went well, what could be improved, and how to implement those improvements.

Time Boxes:

Sprint: A fixed time period (usually 2-4 weeks) during which a potentially shippable product increment is created. Daily Scrum: A time-boxed meeting typically lasting 15 minutes to ensure quick communication and coordination within the team. Scrum promotes transparency,

Anantrao Pawar College of Engg. & Research Pune Pune-9

inspection, and adaptation, allowing teams to respond quickly to changing requirements and deliver incremental value. It is one of the most widely adopted agile methodologies and has been successful in improving the efficiency and effectiveness of teams in various industries.

Kanban:

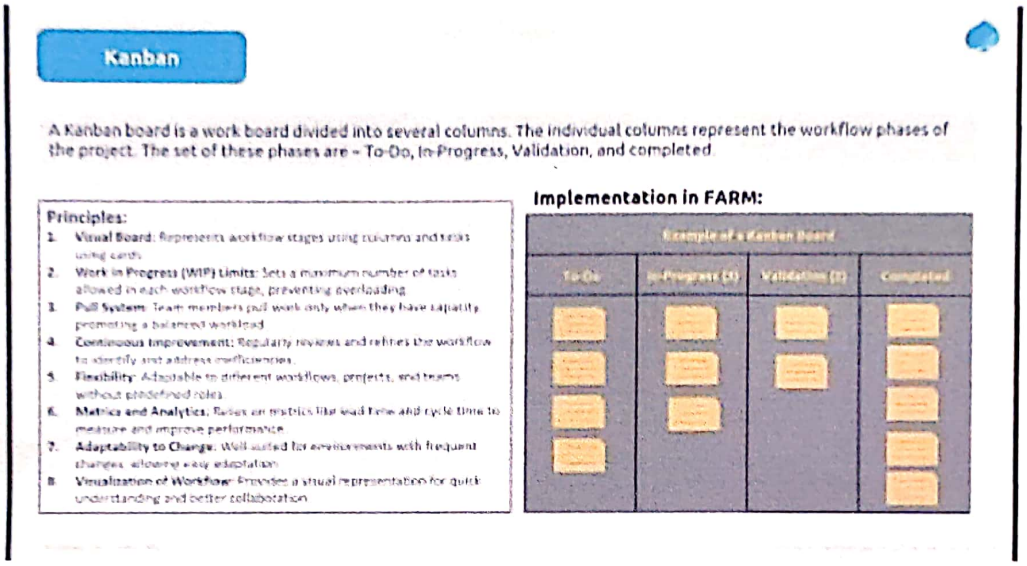

Fig. Kanban

Kanban is a popular method for managing and improving the flow of work in a process, often used in software development and project management, but applicable to various industries. The term "Kanban" originated from Japanese manufacturing systems and means "visual signal" or "card."

Key principles of Kanban include:

Visualization:

The work process is visualized on a Kanban board, typically using cards or sticky notes on a board with columns representing different stages of the process.Work in Progress (WIP) Limits: Each column on the Kanban board has a WIP limit, restricting the number of items allowed in that stage simultaneously. This helps prevent overloading and improves flow.Pull System: Work

Anantrao Pawar College of Engg. & Research ★ Pune-9 ★ Pune

is pulled through the system as capacity becomes available, rather than pushed onto the next stage. This ensures that teams only work on what is necessary, avoiding unnecessary bottlenecks.

Continuous Improvement:

Kanban encourages a culture of continuous improvement. Teams regularly review their processes, identify areas for improvement, and make incremental changes.

Feedback Loops:

Kanban emphasizes quick feedback loops, allowing teams to adapt and respond to changes efficiently.

Kanban is highly flexible and can be adapted to various workflows. It can be used in combination with other methodologies such as Scrum, and it is often implemented in iterative and incremental development environments.

The Kanban method has found applications beyond software development, including areas like product development, customer support, and business processes where visualizing work and optimizing flow are beneficial.

Empowered Teams:

Encourages self-organizing teams that can make decisions independently. Teams have the flexibility to adapt to changing requirements.

Continuous Improvement:

Regularly assesses and improves the development process. Retrospectives are conducted to identify areas for improvement.

Visibility and Transparency:

Work progress is visible to the team and stakeholders. Transparency helps in identifying issues early and making informed decisions. Agile methodologies aim to enhance collaboration, respond quickly to changes, and deliver value to customers in a more efficient and adaptive manner compared to traditional methodologies like Waterfall. Popular Agile practices include Scrum, Kanban, Extreme Programming (XP), and more. Each has its own set of practices and principles that align with the Agile philosophy.

Anantrao Pawar College of Engg. & Research Pune Pune-9

	Akhil Bharatiya Maratha Shikshan Parishad's Anantrao Pawar College of Engineering & Research	
	Record No.: ACA/D/021 Revision: 00	DoI: 02/01/2023
	EVENT REPORT	

Extreme Programming

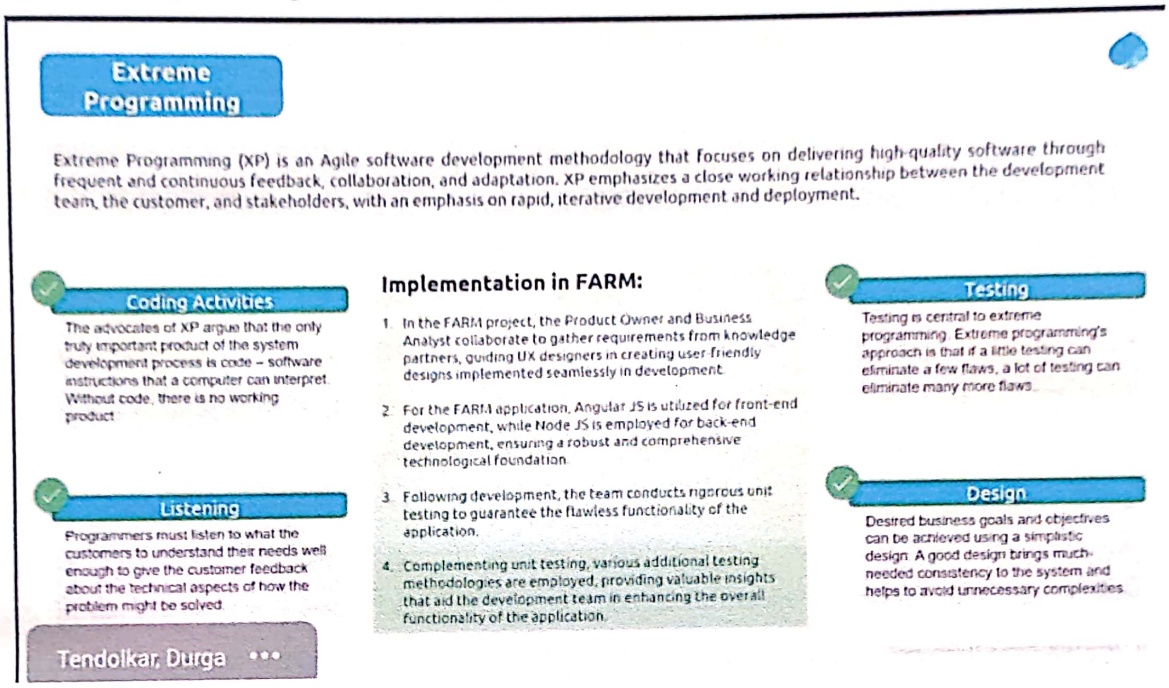

Fig. Extreme Programming

Extreme Programming (XP) is a software development methodology that emphasizes flexibility, collaboration, and customer satisfaction. It was introduced by Kent Beck in the late 1990s and is one of the agile methodologies designed to improve software development processes.

Key principles and practices of Extreme Programming include:

Customer Involvement:

XP encourages close and continuous collaboration between developers and customers throughout the project. This ensures that the software meets the customer's needs and can adapt to changing requirements.

Iterative and Incremental Development:

XP promotes short development cycles, typically one to three weeks, during which a small part of the system is built and delivered. This iterative approach allows for regular feedback and adjustments.

Anantrao Pawar College of Engg. & Research Pune Pune-9

	Akhil Bharatiya Maratha Shikshan Parishad's Anantrao Pawar College of Engineering & Research	
	Record No.: ACA/D/021 Revision: 00	DoI: 02/01/2023
	EVENT REPORT	

Frequent Releases:

The software is released in small, frequent increments, providing tangible value to the customer at the end of each iteration. This helps in early detection of issues and allows for quick adjustments.

Pair Programming:

In XP, developers work in pairs, sharing a single computer. This practice promotes knowledge sharing, reduces errors, and enhances the overall quality of the code.

Test-Driven Development (TDD):

TDD is a technique where tests are written before the code is implemented. This ensures that the code meets the specified requirements and helps in maintaining a reliable and robust codebase.

Continuous Integration:

Code changes are integrated frequently, and automated tests are run to detect and fix issues early in the development process. This practice ensures that the software is always in a working state.

Refactoring: Developers are encouraged to continuously improve the codebase by refactoring – restructuring the code without changing its external behavior. This helps in maintaining code quality and adaptability.

Simple Design:

XP advocates for keeping the design of the system as simple as possible to minimize complexity and make the code more maintainable. This does not mean sacrificing functionality but rather focusing on simplicity and clarity.

Collective Code Ownership:

All team members are responsible for the codebase, and there is no single gatekeeper. This promotes a sense of shared responsibility and encourages collaboration.

Sustainable Pace:

XP recognizes the importance of maintaining a sustainable work pace to avoid burnout and ensure long-term productivity. Teams are encouraged to work at a consistent and manageable pace.

Anantrao Pawar College of Engg. & Research Pune-9 Pune

Extreme Programming is particularly suitable for projects where requirements are expected to change frequently and where a high level of customer involvement is essential. It provides a set of practices that aim to deliver high-quality software in a flexible and adaptive manner.

Lean Software Development

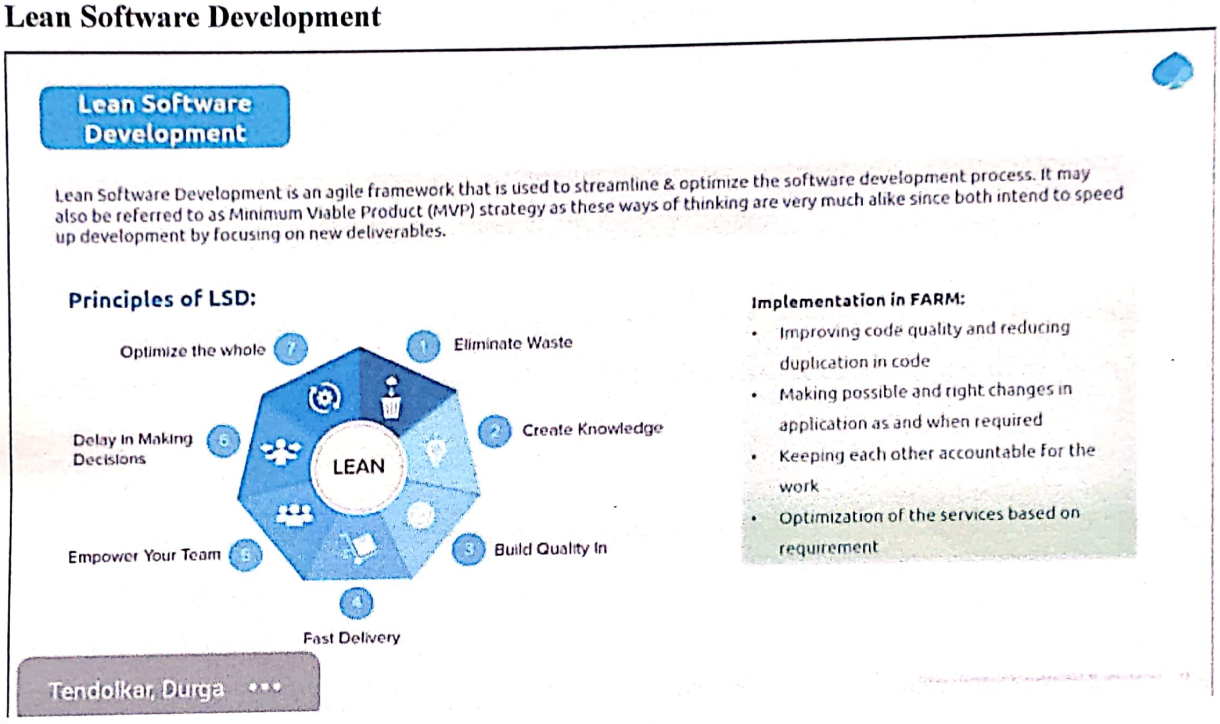

Fig. Lean Software Development

Lean Software Development is an approach to software development that originated from manufacturing principles, particularly the Toyota Production System. It emphasizes delivering value to the customer while minimizing waste in the development process. The principles of Lean Software Development focus on efficiency, continuous improvement, and customer satisfaction. Here are some key principles and concepts associated with Lean Software Development:

EVENT REPORT

Customer Value:

The primary focus is on delivering value to the customer. This involves understanding customer needs and developing software that meets those needs.

Elimination of Waste:

Lean principles aim to eliminate waste in the development process. This includes unnecessary features, redundant code, and any activities that do not directly contribute to delivering value.

Continuous Improvement (Kaizen):

The development process is seen as an ongoing effort for improvement. Teams regularly reflect on their processes and seek ways to make them more efficient.

Pull System:

Work is pulled through the development process based on customer demand. This contrasts with a push system, where work is pushed through the process whether or not there is customer demand.

Just-In-Time (JIT):

Work is done just in time to meet customer demand, reducing the need for excess resources and minimizing delays.

Empowered Teams:

Cross-functional, self-organizing teams are a key component of Lean. Team members are empowered to make decisions and take ownership of the development process.

Optimize the Whole:

Instead of optimizing individual components, Lean focuses on optimizing the entire value stream. This involves looking at the end-to-end process to identify and eliminate bottlenecks and inefficiencies.

Build Integrity In:

Quality is built into the development process from the beginning rather than being addressed as a separate step. This reduces defects and ensures that the software meets customer requirements.

Visual Management:

Anantrao Pawar College of Engg. & Research Pune ★ Pune-9 ★

	Akhil Bharatiya Maratha Shikshan Parishad's Anantrao Pawar College of Engineering & Research	
	Record No.: ACA/D/021 Revision: 00	DoI: 02/01/2023

EVENT REPORT

Capgemini's FARM Software Development Process Training benefits to students:

Skill Development:
The training can provide students with practical skills in software development, focusing on financial and agricultural recommendation models. This can include programming languages, data analysis, machine learning, and other relevant technologies.

Industry-Relevant Knowledge:
Capgemini, being a prominent global consulting and technology services company, is likely to provide insights into industry best practices and real-world applications. This can help Students Bridge the gap between theoretical knowledge and industry requirements.

Hands-on Experience:
The training is likely to include hands-on projects and exercises, allowing students to apply the concepts they learn in a practical setting. This experience is invaluable for building confidence and competence in software development.

Networking Opportunities:
Training sessions conducted by Capgemini may include interactions with industry professionals, creating networking opportunities for students. Building connections with professionals in the field can be beneficial for future career prospects.

Career Advancement:
Acquiring skills in financial and agricultural recommendation models can enhance students' employability. This training may serve as a stepping stone for students looking to enter the fields of finance, agriculture technology, or data science.

Problem-Solving Skills:
Working on recommendation models for financial and agricultural scenarios involves addressing real-world problems. This can sharpen students' problem-solving skills and their ability to apply technical knowledge to practical challenges.

Project Collaboration:
The training may involve collaborative projects, simulating real-world development scenarios where students learn to work in teams. Collaboration is a key skill in the professional world, and this experience can be beneficial for their future careers.

Career Guidance:
Capgemini, as a global IT and consulting company, may provide insights into career paths within the industry. This can help students make informed decisions about their future careers and understand the various roles available in financial and agricultural software development.

Anantrao Pawar College of Engg. & Research ★ Pune-9 ★ Pune

	Akhil Bharatiya Maratha Shikshan Parishad's Anantrao Pawar College of Engineering & Research	
	Record No.: ACA/D/021 Revision: 00	DoI: 02/01/2023
	EVENT REPORT	

Capgemini's FARM Software Development Process Training benefits to Faculties:

Specialized Knowledge Transfer:
Faculty member's gain specialized knowledge in the development of Financial Agricultural Recommendation Models, which can enhance their understanding of the intersection between finance and agriculture.

Cutting-Edge Technology Exposure:
Capgemini, being a leading global consulting and technology services firm, is likely to provide insights into the latest technologies and methodologies used in software development for financial and agricultural applications.

Industry-Relevant Curriculum:
Training programs conducted by industry experts like Capgemini are designed to align with current industry trends and standards, ensuring that faculties receive up-to-date and relevant information.

Practical Implementation Insights:
Capgemini may share practical insights into the implementation of financial and agricultural recommendation models, offering faculties a real-world perspective on the challenges and solutions in this domain.

Networking Opportunities:
Faculty members may have the chance to network with professionals from Capgemini and other participants, fostering collaborations and potential research partnerships.

Enhanced Teaching Capabilities:
Faculty members can bring back their learnings to the classroom, enriching their teaching methodologies with practical examples and case studies from the financial agricultural domain.

Research Collaboration Opportunities:
Training programs may open doors for collaborative research opportunities between faculties and Capgemini, allowing for joint projects that can contribute to advancements in the field.

Skill Development:
Faculties can enhance their skills in software development, data analytics, and machine learning, which are likely to be integral components of Financial Agricultural Recommendation Models.

Anantrao Pawar College of Engg. & Research Pune Pune-9

Visual tools, such as Kanban boards, are often used to make the development process and work status easily visible. This helps teams to manage their work and identify areas for improvement.

Respect for People:

Lean emphasizes the importance of respecting and empowering individuals within the development team. This includes recognizing their expertise and providing an environment that supports continuous learning and improvement.

Lean Software Development provides a framework for creating more efficient and effective software development processes, ultimately leading to higher customer satisfaction and better outcomes. It has gained popularity in agile development environments and is often used in conjunction with other methodologies such as Scrum.

FARM High Level Architecture

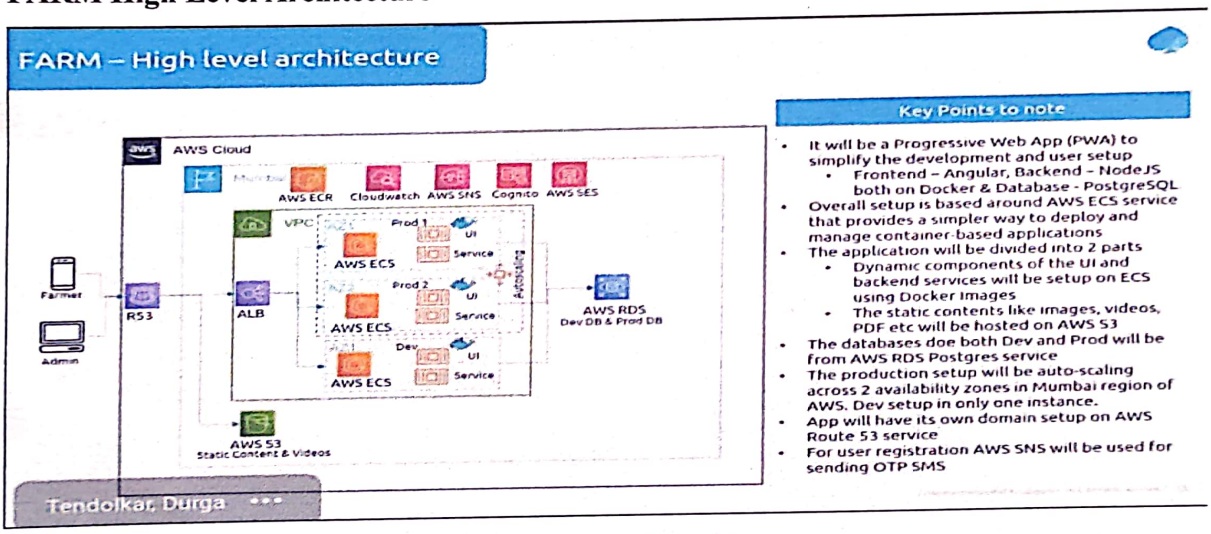

Fig. Farm High Level Architecture

Date:- 03/02/2024

03/02/2024
Event Coordinator

for 03/02/2024.
Head of Department

Principal

Anantrao Pawar College of Engg. & Research
Pune
Pune-9