GLS UNIVERSITY
Faculty of Computer Applications & Information Technology
Integrated Master of Computer Applications (iMCA)
Semester-V
222304501 SOFTWARE ENGINEERING

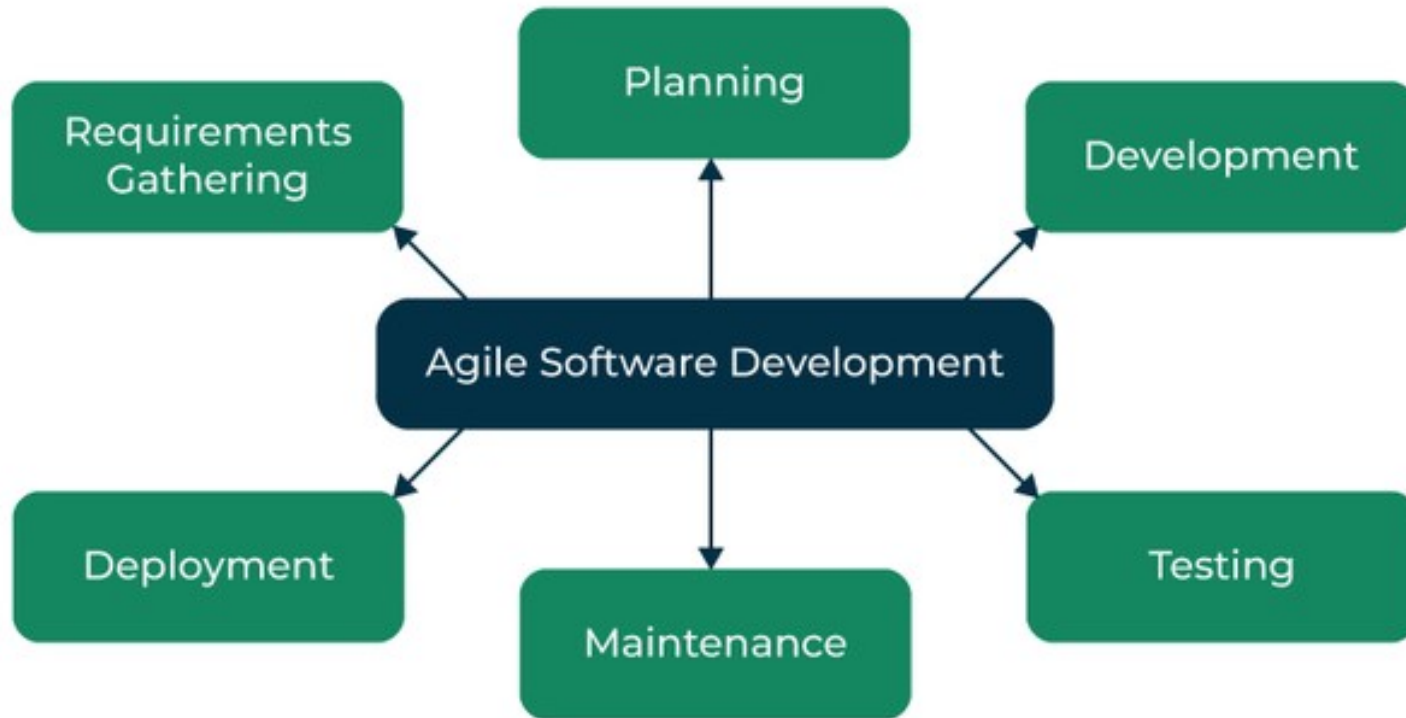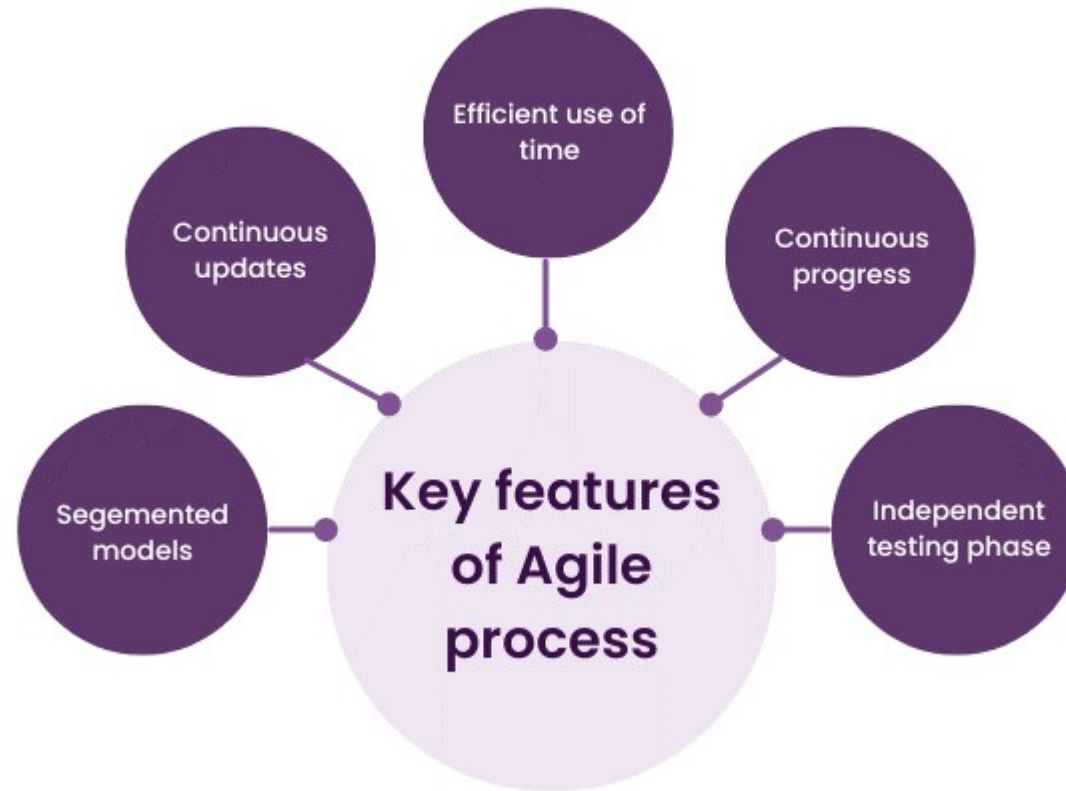## Unit-2 : Software development - An Agile Approach

Agile Methodology

- Agile Software Development is a software development methodology that values flexibility, collaboration, and customer satisfaction.
- It is based on a set of principles for software development that prioritize individuals and interactions, working software, customer collaboration, and responding to change.
- Agile Software Development is an iterative and incremental approach to software development that emphasizes the importance of delivering a working product quickly and frequently.
- It involves close collaboration between the development team and the customer to ensure that the product meets their needs and expectations.

Why choose agile?

- Teams choose agile so they can respond to changes in the marketplace or feedback from customers quickly without derailing a year's worth of plans. "Just enough" planning and shipping in small, frequent increments lets your team gather feedback on each change and integrate it into future plans at minimal cost.

- Collaborating with customers and teammates is more important than predefined arrangements. And delivering a working solution to the customer's problem is more important than hyper-detailed documentation.

Planning

Requirements Gathering

Development

Agile Software Development

Deployment

Maintenance

Testing

Key features of Agile process

- Efficient use of time
- Continuous updates
- Continuous progress
- Segemented models
- Independent testing phase

Key Features of Agile Process

1) Within software development, the Agile process is segmented into discrete models that designers work on.

2) The client is provided with early and regular chances to see the product, as well as the opportunity to make decisions and adjustments about the project.

3) Compared to the Waterfall methodology, the Agile model is considered unstructured.

4) It is possible to carry out the implementation of smaller projects rapidly. Providing an accurate time estimate for the development of huge projects is challenging.

5) It is possible to correct errors even during the project.

6) The development method is iterative, and the project is carried out in small iterations that last between two to four weeks. There is virtually little planning involved.

Key Features of Agile Process

1)The development of software supersedes the creation of documentation.

2)Each iteration has its independent testing phase. It makes it possible to do regression testing every time new logic or functions are made available for use.

3)When an iteration of Agile testing ends, all shippable features of the product are handed over to the client. The newly released features are immediately available for use. When you have a strong relationship with your clientele, it is beneficial.

4)Both testers and developers collaborate on the project.

5)User acceptability testing is carried out after each sprint.

6)Maintaining close contact with the developers and for everyone to examine the needs and plan jointly is necessary.

Agile Tools

- Agile Development Tools play a crucial role in enhancing collaboration, communication, and overall efficiency within Agile teams.

- These tools are designed to streamline various aspects of the development process, facilitating better project management and communication among team members.

Overview of Popular Agile Tools:

- **Jira:** Widely used for issue tracking, project management, and agile development. It allows teams to plan, track, and manage their work efficiently.

- **Trello:** Known for its simplicity, Trello uses boards, lists, and cards to organize tasks. It's user-friendly and particularly effective for smaller teams.

- **VersionOne:** A comprehensive agile project management tool that supports various methodologies, including Scrum, Kanban, and SAFe.

▥ Board ⌄ | **Team Tasks** ☆ | △ Inktistic ENT | 👥 Workspace visible | Invite | Join board

To Do ⋯

Collect all assets

Schedule client meeting

Approve legal agreement

Create Legal Agreement

+ Add another card

Doing ⋯

Schedule recurring team meeting

Meet with sales team

+ Add another card

Done ⋯

Create deck for presentation ✎

+ Add a card

Importance of Tools in Agile Development:

- **Enhanced Collaboration:** Agile tools provide a centralized platform for team collaboration, fostering better communication and information sharing.

- **Real-time Tracking:** These tools enable real-time tracking of project progress, helping teams to stay on top of tasks and adapt to changes promptly.

- **Transparency:** Agile tools promote transparency by providing visibility into each team member's tasks, timelines, and potential roadblocks.

- **Data-Driven Decision Making:** Through analytics and reporting features, Agile tools empower teams and stakeholders to make informed decisions based on project data.

What Is Scrum?

- Scrum is a project management framework that facilitates team collaboration on complex product and software development projects.

- The good news is that scrum is easy to understand.

- The bad news? It's hard to master.

What Is the Scrum Methodology?

- The scrum methodology emphasizes teamwork in project management. It stresses accountability and iterative progress toward a well-defined goal.

- Scrum is part of agile software development and teams practice agile.

- **The name comes from the sport of rugby, where scrum is a formation where everyone plays a specific role, but everyone is working towards a quick adoption of strategies.**

- The necessary collaboration required for a successful scrum is facilitated by ProjectManager, online and project management software that connects scrum teams wherever they work.

15

Scrum (software development)

Scrum is the type of Agile framework. It is a framework within which people can address complex adaptive problem while productivity and creativity of delivering product is at highest possible values. Scrum uses Iterative process. Silent features of Scrum are:

- Scrum is light-weighted framework
- Scrum emphasizes self-organization
- Scrum is simple to understand
- Scrum framework help the team to work together

Scrum vs. Agile

- Scrum is a part of the agile process, but certainly not the only part. Agile is a large tent, but scrum is an important pillar.

- Think of scrum as a framework by which you can implement agile development.

- Agile doesn't have a set of steps to follow, so scrum provides a means to apply agile to your project.

- There are many frameworks that you can use in agile development, such as extreme programming or feature-driven development, but scrum's simplicity and autonomy are selling points.

- The scrum methodology can also be used as an entry point to other agile practices. It's also not solely a framework for software but can benefit many other kinds of projects.

Roles in Scrum:

- **Scrum Master:** The scrum master, as his name suggests, is a scrum methodology expert. He guarantees that everybody in the scrum team understands how the framework works and helps them adapt to the agile environment. He leads scrum meetings.

- **Scrum Product Owner:** The scrum product owner manages the product log and oversees sprint planning and actively participates in scrum meetings. In a sense, they act as a project manager because they lead backlog grooming and prioritize user stories to help the teamwork better.

- **Scrum Development Team:** The scrum development team is simply made up of all the team members who develop software or products. They must work closely with the product owner and adhere to the scrum master's suggestions.

What Are the Scrum Events in the Scrum Process?

- **Sprint Planning:** Using the product backlog, teams start with the highest priority items and determine how to achieve this objective. A good tip when sprint planning is to do the due diligence and only start with items that are ready. Also, remember that planning is a short process, so don't get bogged down in the details. Simply get to work on meeting the objectives and keep the plan collaborative. The team should also ask the product owner and stakeholder questions.

- **Daily Scrum Meeting:** These are 15-minute meetings where everybody in the scrum team talks about the tasks they'll be working on during the day and share any roadblocks or difficulties they're facing. There's no need to make this daily scrum meeting longer, as there are other meetings such as sprint reviews and sprint retrospectives to explore more complex topics.

What Are the Scrum Events in the Scrum Process?

- **Sprint Review:** You want to look back on the sprint and see what worked and what didn't. You can then take the information and apply it to future sprints to replicate the positives and reduce the negatives. Begin the sprint review process by thanking participants, offering short introductions and setting ground rules for the discussion.

- **Sprint Retrospective:** The sprint retrospective meeting gives the scrum team a space to reflect on the last sprint and determine what went well and wrong. Stakeholder and customer feedback are also gathered in order to prioritize user stories and improve product performance.

Extreme programming (XP)

- Extreme programming (XP) is one of the most important software development frameworks of Agile models. It is used to improve software quality and responsiveness to customer requirements.

- The extreme programming model recommends taking the best practices that have worked well in the past in program development projects to extreme levels.

Planning/feedback loops

Release plan
Months
Iteration plan
Weeks
Acceptance test
Days
Stand-up meeting
One day
Pair negotiation
Hours
Unit test
Minutes
Pair programming
Seconds
Code
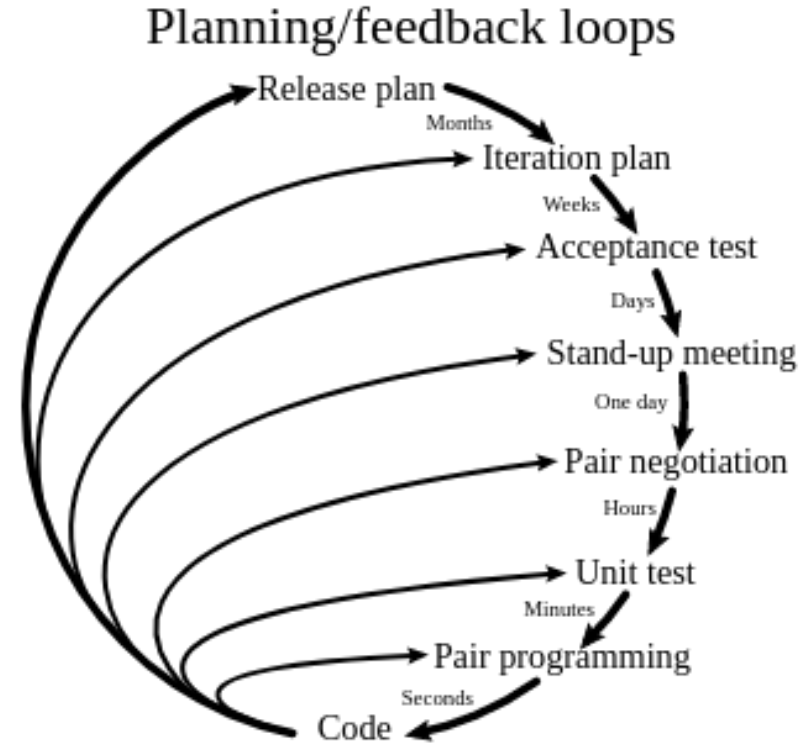
A Case Study on Real-Time Software Development Using Agile Principles

This case study focuses on a fictitious company, TechSprint Innovations, and their real-time software development project aimed at creating a cutting-edge mobile application for event management.

Brief Overview:

- **Industry:** Technology and Event Management

- **Project Scope:** Development of a mobile app for event planning, ticketing, and attendee engagement.

- **Background:** TechSprint Innovations recognized the need for a dynamic and user-friendly app to streamline event management processes.

Challenges Faced and Addressed:

Changing Requirements:
    Challenge: Midway through development, stakeholders decided to add a feature for live streaming of events, altering the initial scope
    Agile Solution: Agile's iterative cycles allowed the team to incorporate this change seamlessly, ensuring the final product met evolving needs.

Communication Gaps:
Challenge: Communication breakdowns between the development team and marketing, leading to misaligned promotional activities.
Agile Solution: Daily stand-up meetings and Agile collaboration tools facilitated clearer communication, aligning marketing efforts with development milestones.

Challenges Faced and Addressed:

Uncertain Timelines:

Challenge: Initial difficulty in estimating the time required for implementing complex features like real-time analytics.

Agile Solution: Through iterative development and regular feedback, the team refined estimates, adapting to evolving complexities and ensuring timely delivery.

Results and Benefits:

Faster Time-to-Market: Result: The Agile approach enabled the team to release a minimum viable product (MVP) for basic event management quickly, garnering user feedback early in the development process.

Improved Customer Satisfaction: Result: Continuous customer involvement and feedback loops allowed for the incorporation of user preferences, leading to a final product that exceeded initial expectations.

Enhanced Team Collaboration: Result: Daily stand-ups, pair programming sessions, and collaborative tools fostered a cohesive team environment. Team members reported increased satisfaction and productivity.

Examples of projects that are well-suited for each methodology:

- **Waterfall Methodology:**
  Example Project: Building a Bridge
  Reasoning: Construction projects, such as building a bridge, often have well-defined requirements, a sequential flow of tasks, and minimal changes once construction begins. The Waterfall methodology is suitable when the project can be clearly defined at the outset, and changes during development are expected to be minimal.

- **Agile Development Methodology:**
  Example Project: Mobile App Development
  Reasoning: Agile is well-suited for projects with evolving requirements and a need for frequent adaptations. In mobile app development, customer feedback and changing market trends can impact the project. Agile methodologies, with their iterative and incremental nature, allow for flexibility and continuous refinement of the product.

Examples of projects that are well-suited for each methodology:

Scrum Methodology:
Example Project: E-commerce Website Development
Reasoning: Scrum is particularly effective for projects with a complex scope and a need for regular releases. In e-commerce website development, features such as product catalog, shopping cart, and payment processing can be developed and released incrementally. Scrum's sprint-based approach aligns well with this iterative development cycle
The V-Model:
Embedded Systems Development:
Reasoning: Embedded systems often have clearly defined requirements and a fixed architecture. The V-Model's step-by-step approach aligns well with the structured and deterministic nature of embedded systems development.

Examples of projects that are well-suited for each methodology:

Automated Testing Framework Development:
Reasoning: When developing automated testing frameworks, the requirements for test cases and functionalities are typically well-documented upfront. The V-Model ensures that testing is closely aligned with each development phase, helping to identify defects early in the process.

Hardware Development with Fixed Specifications:
Reasoning: For projects where the hardware specifications are well-established and unlikely to change, the V-Model provides a systematic approach to design, development, and testing, minimizing the risk of late-stage changes impacting the project timeline.