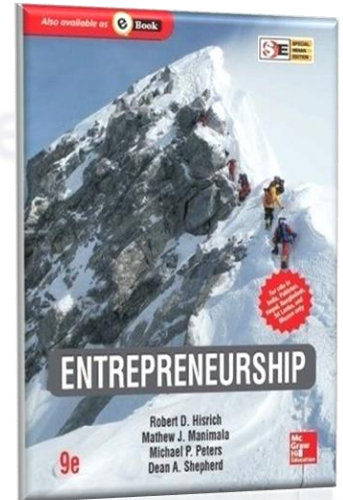


# Entrepreneurship & Leadership (HSS-421)

(2+0)



Department of Computer  
Software Engineering  
Karachi Campus

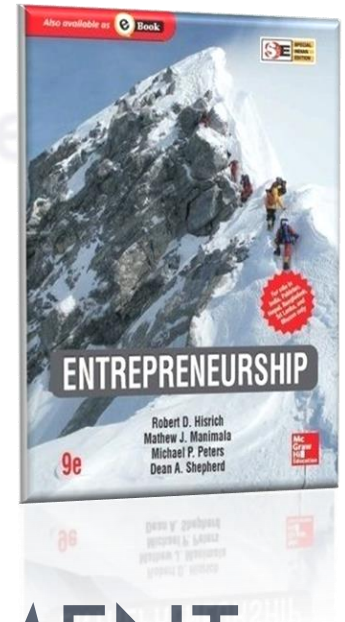


**ENGR. SYED RIZWAN ALI**  
Assistant Professor,  
Head of Business Incubation Center,  
Bahria University (Karachi Campus)





Department of Computer  
Software Engineering  
Karachi Campus



## PRODUCT DEVELOPMENT LIFECYCLE

Lecture No 09

By Engr. Syed Rizwan Ali

In this Lecture will cover ..

- Understand Different Software Development Methodologies: Explain the principles, processes, and key differences of various software development methodologies such as Agile, Scrum, Waterfall, DevOps, and Lean.
- Application of Methodologies: Ability to select and apply appropriate software development methodologies based on project requirements, team dynamics, and business goals.
- Conceptualization of MVP: To identify core functionalities that address the primary needs of the target audience.

In this Lecture will cover ..

- MVP Design and Development Skills: Focusing on essential features that align with the product vision and user needs.
- Understanding Iterative Development: the concept of iterative development, including planning, development, testing, and reviewing in cycles.
- Feedback Loop Integration: Understanding its importance in continuous improvement and adaptation of the product.
- Advantages and Limitations: Analyze the advantages and limitations of each methodology in different scenarios.



Government	Percentage
Current government	85%
Previous government	15%



# Product Development Lifecycle

1. **Overview:** This stage involves selecting an appropriate development methodology that aligns with the project's needs, team dynamics, and business goals.
  - Common methodologies include Agile, Scrum, Waterfall, Lean, and DevOps. Each has its unique approach to managing the development process, with Agile and Scrum being particularly popular for their flexibility and emphasis on continuous improvement.

# Product Development Lifecycle

## 1. Overview:

- In software product development, the definition refers to outlining the software's purpose, functionality, and requirements, while the importance lies in ensuring that the development process aligns with business goals and user needs, ultimately determining the project's success and viability.

# Product Development Lifecycle

## 2. Key stages: Conceptualization, Development, Testing, Deployment, and Maintenance:

- In software product development, the definition refers to outlining the software's purpose, functionality, and requirements, while the importance lies in ensuring that the development process aligns with business goals and user needs, ultimately determining the project's success and viability.
- **Conceptualization:** This initial stage involves brainstorming and defining the idea behind the software. It's where needs are identified, objectives



# Product Development Lifecycle

## 2. Key stages: Conceptualization, Development, Testing, Deployment, and Maintenance:

- are set, and preliminary research is conducted to validate the feasibility of the project.
- **Development:** In this phase, the actual software creation takes place. Based on the requirements gathered during conceptualization, developers write code and build the software's architecture and components.
- **Testing:** After development, the software undergoes rigorous testing to identify and fix bugs, ensuring the

# Product Development Lifecycle

## 2. Key stages: Conceptualization, Development, Testing, Deployment, and Maintenance:

- product meets quality standards and functions as intended.
- **Deployment:** Once tested and finalized, the software is deployed to a production environment where it becomes available for use by its intended users.
- **Maintenance:** Post-deployment, the software requires regular updates and maintenance to address any emerging issues, add new features, and keep it compatible with evolving technology and user needs.

# Product Development Lifecycle

## 2. Key stages: Conceptualization, Development, Testing, Deployment, and Maintenance:

- product meets quality standards and functions as intended.
- **Deployment:** Once tested and finalized, the software is deployed to a production environment where it becomes available for use by its intended users.
- **Maintenance:** Post-deployment, the software requires regular updates and maintenance to address any emerging issues, add new features, and keep it compatible with evolving technology

# Real-world Example

- The role of lifecycle management in entrepreneurial success is pivotal, as it ensures that software products are developed efficiently, meet market demands, and evolve with changing customer needs.
- Effective lifecycle management involves overseeing every stage of development, from initial conception to maintenance, aligning technical processes with business objectives.

# Real-world Example

- **Airbnb's Journey Through Lifecycle Management:**
- **Conceptualization:** Airbnb began with a simple concept - allowing people to rent out spare rooms to travelers. The founders started with a basic website to test this idea during a design conference in San Francisco.
- **Development:** After validating the idea, they developed a more robust platform to facilitate room rentals. This included features for listings, bookings, and payments.

# Real-world Example

- **Airbnb's Journey Through Lifecycle Management:**
- **Testing:** As Airbnb grew, testing became crucial to ensure the platform could handle increased traffic and various user interactions without issues.
- **Deployment:** Airbnb regularly updated its platform, deploying new features like enhanced search functionality, user verification systems, and mobile app development.
- **Maintenance:** They continuously refine the user experience, address technical issues, and add features, like experiences and adventures, to keep the platform relevant and user-friendly.



# Real-world Example

- **Airbnb's Journey Through Lifecycle Management:**
- **Impact on Entrepreneurial Success:**
- **Scalability and Adaptability:** Lifecycle management allowed Airbnb to scale its operations globally while adapting to different market needs and regulatory environments.
- **Customer-Centric Approach:** By focusing on user feedback during the testing and maintenance phases, Airbnb has continually improved its service, leading to high customer satisfaction and repeat business.

# Real-world Example

- **Airbnb's Journey Through Lifecycle Management:**
- **Impact on Entrepreneurial Success:**
- **Innovation:** Airbnb's approach to managing its product lifecycle has fostered continuous innovation, helping it to stay ahead of competitors and expand its market presence.
- Airbnb's success story illustrates how effective lifecycle management can transform a simple idea into a global enterprise by ensuring that the software product continually evolves and stays aligned with business goals and customer expectations.

# Methodologies: Tailoring to Project Needs

## 3. Selecting the Right Software Development Methodology:

- **Assess Project Scope:** Size, complexity, and criticality.
- **Understand Team Dynamics:** Skills, experience, and communication.
- **Align with Business Goals:** Time-to-market, budget, and ROI expectations.
- **Case in Point:** Agile for fast-paced startups, Waterfall for well-defined projects.

# Methodologies: Tailoring to Project Needs

- 4. **Applying the Chosen Methodology Effectively:**
  - **Effective Implementation of Software Development Methodologies:**
  - **Customization is Key:** Tailor methodologies to the project's unique challenges and objectives rather than following them rigidly.
  - **Empower the Team:** Encourage team ownership over processes and decisions to promote engagement and accountability.
  - **Iterative Improvement:** Regularly review and adapt the methodology to refine efficiency and effectiveness throughout the project lifecycle.

# Methodologies: Tailoring to Project Needs

- 4. **Applying the Chosen Methodology Effectively:**
  - **Effective Implementation of Software**
  - **Adapt and Oversee:** Customize the methodology to fit project specifics.
  - **Facilitate Collaboration:** Tools and practices to support team engagement.
  - **Continuous Evaluation:** Monitor progress, gather feedback, and refine processes.
  - **Real-World Success:** How a hybrid Agile-Waterfall approach benefited a tech giant.

# MVP Conceptualization: Focusing on Core Functionalities

- **Identify Core Needs:** Engage with potential users to determine the most critical problem your product will solve.
- **Prioritize Features:** Select key features that directly address these needs, ensuring they form the product's foundation.
- **Balance Value & Simplicity:** Aim for the simplest version of your product that still delivers value, making it both viable and desirable to early adopters.



# Mastering MVP Design and Development

- **Streamline Feature Set:** Concentrate on essential features that deliver the product's promise and fulfill user expectations.
- **Align with Vision:** Ensure every feature aligns with the overarching product vision to maintain coherence and direction.
- **User-Centric Approach:** Incorporate user feedback early and often to refine the MVP in alignment with real user needs.

# Embracing Iterative Development Cycles

- **Cyclical Process:** Break down the development into manageable iterations, each encompassing planning, development, testing, and review.
- **Continuous Improvement:** Use feedback from each cycle to refine the product, ensuring it evolves to meet user needs effectively.
- **Adaptability and Learning:** Foster a flexible development environment that learns from each iteration, enhancing product quality and team expertise.

# Feedback Loop Integration

- **Integrating Feedback Loops for Continuous Improvement:**
- **Critical for Evolution:** Emphasize how iterative feedback is essential for the product to adapt and grow with user needs and market changes.
- **Mechanisms for Gathering:** Outline methods such as user testing, surveys, and analytics to collect actionable feedback.
- **Responsive Adjustments:** Stress the importance of quickly implementing changes based on feedback to continually enhance the product.

# Weighing Methodologies in Varied Scenarios

- **Advantages and Limitations of Development Methodologies**
- **Agile Advantages:** Flexibility, rapid iteration, and customer feedback. Ideal for dynamic projects with changing requirements.
- **Waterfall Limitations:** Inflexible structure, difficult to adapt to change. Can be challenging for projects that aren't well-defined from the start.
- **Context Is Crucial:** Select methodologies based on project size, complexity, customer involvement, and team expertise.

**Thanks**  
**Q&A**

