

TITLE: PROJECT MANAGEMENT AND SYSTEM DEVELOPMENT

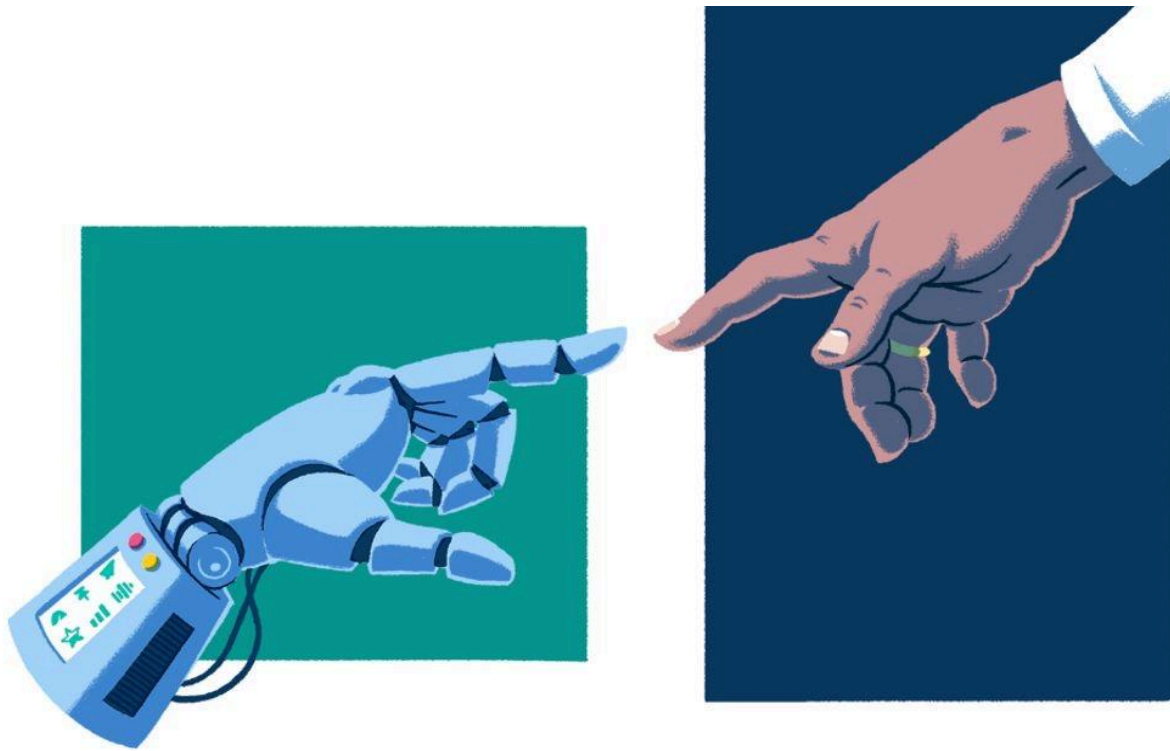


Table of contents

- Speaker's experience
- What is project management and system development (PM & SD)
- Waterfall vs agile
- How has PM & SD been used in our program
- Core skills to master
- Reflection from each member

Group members

1. Santhya Ravichandran (A25CS0345)
 2. Annchalee Anon (A25CS0194)
 3. Muhammad Yasir Nawaz (A25CS0015)
 4. Siti Zulaikha binti Abdol Razid (A23KM0250)
 5. Nilima Rahman (A24CS4107)
-

Speaker's experience

The speaker, Ts. Hj. Abdul Alim, is a UTM Graduate who has served as a Head of Technology and Innovation. He has been involved in various company ventures and has contributed to the development of multiple projects. The speaker possesses in-depth experience in project management and system development.

What is project management and system development

System development is the entire process of developing a software application which consists of phases such as defining, designing, testing, and implementation. System development is usually accompanied by a System Development Life Cycle (SDLC) to aid in the efficient management of the development process. Project Management (PM) ensures projects stay on track, on time and within budget. Project Management also allows members of different departments to collaborate effectively such as designers, testers, and developers.

There are two main Project Management strategies, Waterfall and Agile. Waterfall strategy is the more traditional approach and involves a linear step by step approach to each phase of development.

Agile strategy is the modern approach and consists of an iterative approach where work is broken into smaller "sprints".

Differences between Waterfall and Agile

Aspect	Waterfall	Agile
Development Approach	Linear and sequential process	Iterative process using sprints
Best Use Case	Fixed and clearly defined requirements	Frequently changing requirements
Team Size & Coordination	Large teams, less coordination	Small or mid-sized team coordination
Customer Involvement	Involved only at milestone	continuous involvement throughout
Flexibility & Feedback	Low flexibility, late feedback	High flexibility, continuous feedback
Feature Prioritization & Risk	No prioritization, higher risk of failure	Prioritized features, reduced risk
Frameworks	No specific framework	Scrum, Kanban

How has Project Management and System Development been used in our program?

In our program, which focuses on graphics and multimedia software, Project Management and System Development are applied throughout the software creation process to ensure that projects are completed systematically, on time, and in line with user requirements.

During the planning phase, project objectives and timelines are defined, and tasks are distributed among team members. Before developing a multimedia application, the project scope, target audience, software tools, and deadlines are clearly identified.

In the system development phase, appropriate development models are used, such as Waterfall or Agile. The Waterfall model is commonly applied to projects with clear and fixed requirements, such as designing posters, animations, or interactive presentations. Meanwhile, the Agile approach is used for projects that require frequent feedback and continuous improvement, such as games or interactive multimedia applications.

Additionally, the testing and evaluation stage is crucial to ensure usability, visual consistency, performance, and compatibility across different platforms.

Core Skills to Master & Success Formula

Success in this field isn't just about coding anymore. The talk highlighted three key skills: requirements gathering, system design, and AI-assisted development. Requirements gathering means understanding user needs and turning them into clear goals. System design focuses on building efficient, scalable solutions. And AI-assisted development helps us work smarter with tools like code assistants, while still relying on human judgement. Together, these skills prepare us to be adaptable, future ready professionals.

Crucially, the talk presented a key formula for success in this field; 40% coding and AI skills + 60% system and project management skills = 100 % future ready professionals. Coding and AI skills remain essential, they form the technical backbone that helps professionals create smart, efficient solutions. But success today goes beyond just knowing how to code. The larger 60% lies in system and project management skills: planning, communicating, collaborating, managing risks, and making sound decisions. In short, being future ready means finding the right balance. IT professionals who can blend technical know-how with leadership and management skills will be the ones who stand out, adapt faster, and thrive as technology keeps evolving.

Reflections from each member

Santhya	From this talk, I realized that my future success depends less on writing code and more on disciplined planning to prevent development chaos. I discovered that while AI is a powerful accelerator, I must use it as an assistant rather than a replacement; relying on it blindly without understanding the underlying architecture will only stunt my growth. I now understand that to remain valuable, I must master the "why" behind system design, not just the syntax. Therefore, I commit to continuous self-learning beyond the classroom syllabus to ensure I graduate as a competent, industry-ready professional.
Annchalee	The speaker effectively gives us exposure to the skill sets needed to thrive in this industry through his speech. Since computer science is a field that

	<p>keeps growing every day, I learned that it is crucial for us to be equipped with life-long learning skills so that we can continuously adapt to new technologies, tools and demands. Besides, it is also important that we know how to utilise AI in our work while not fully relying on it. As computer science students, we should know the core concepts and project management skills to be successful.</p>
Yasir	<p>The speaker's talk shifted my view of development from just writing code to understanding the entire Software Development Life Cycle (SDLC). I realized that rigorous planning and testing are what make professional software truly scalable and secure. Additionally, learning about Project Management strategies like Agile showed me how essential communication and structured frameworks are for team success. I now see that being a professional developer requires excelling within a collaborative ecosystem, rather than just solving isolated technical problems.</p>
Siti Zulaikha	<p>From the industrial talk, I learned that success in computer science is not based on programming skills alone. The speaker emphasised that professionals in this field must also master several skills. This includes planning, system design and teamwork. Over the next four years, I aim to improve my ability to understand user needs and translate them into clear system requirements. I also plan to strengthen my system design skills by thinking more holistically about system architecture and data flow rather than focusing only on coding. The talk also strongly emphasised the importance of using AI-assisted development tools responsibly to improve productivity without replacing human decision-making.</p>
Nilima	<p>From the industrial talk, I learned that success in computer science requires more than just strong programming skills. The speaker helped me understand that software development involves planning, system design, teamwork, and continuous learning alongside coding. I realized the importance of clearly understanding user requirements and project goals before starting development, as well as thinking about the overall system architecture instead of focusing only on individual features. The talk also highlighted that while AI-assisted tools can improve productivity and efficiency, they must be used responsibly, with human logic and decision-making remaining central. Overall, the session encouraged me to develop a balanced skill set that combines technical knowledge, system-level thinking, and adaptability to succeed in the industry.</p>

Reference

Palmquist, M. S., Lapham, M. A., Miller, S., Chick, T., & Ozkaya, I. (2013). *Parallel worlds: Agile and Waterfall differences and similarities* (Technical Note CMU/SEI-2013-TN-021, ADA610501). Software Engineering Institute, Carnegie Mellon University.

Link: <https://apps.dtic.mil/sti/pdfs/ADA610501.pdf>