Lecture Notes: Engineering Project Management

1. What is an Engineering Project?

1.1 Definition

- **Engineering Project**: In computing and electronics, an engineering project is a structured effort to develop or enhance systems, hardware, or software solutions. This involves applying principles from computer science and electrical engineering to solve specific problems or create new technologies.

1.2 Characteristics

- **Temporary Nature**: Projects have a defined start and end, focusing on achieving specific objectives such as developing a new software application or designing an electronic circuit.
- **Uniqueness**: Each project aims to produce a distinct product or result, such as a custom-built IoT device or a specialized software tool.
- **Constraints**: Projects must adhere to time, cost, and quality constraints while delivering innovative solutions.
- **Cross-Functional Teams**: Often involves collaboration between software developers, hardware engineers, data scientists, and other specialists.

1.3 Examples

1.3.1 Computing

- **Software Development Project**: Developing a new mobile application for task management that integrates with various online services.
- **Database Management System (DBMS) Upgrade**: Enhancing an existing DBMS to improve performance, security, and scalability.
- **Cloud Migration**: Moving an organization's data and applications from on-premises servers to a cloud platform to achieve greater flexibility and cost savings.

1.3.2 Electronics

- **Consumer Electronics Design**: Designing a new smartwatch with advanced health monitoring features.
- **Embedded Systems Development**: Creating a custom microcontroller-based system for an automated home security solution.
- **PCB Design and Fabrication**: Designing and manufacturing a printed circuit board (PCB) for a new electronic device.

2. What is Engineering Project Management?

2.1 Definition

- **Engineering Project Management**: In the context of computing and electronics, it involves applying project management techniques to plan, execute, and oversee projects related to software development, hardware design, and system integration. It aims to deliver high-quality products or solutions within defined constraints.

2.2 Objectives

- **Deliverables**: Ensure the successful delivery of software applications, hardware systems, or integrated solutions meeting the project specifications.
- **Scope Management**: Define and manage what is included in the project, such as features of a software application or components of an electronic device.
- **Time Management**: Develop and adhere to a project schedule, using tools to track progress and manage deadlines.
- **Cost Management**: Budgeting for development costs, including personnel, hardware, software, and other resources.
- **Quality Management**: Implement quality assurance practices to ensure that the final product meets user requirements and industry standards.

2.3 Key Processes

2.3.1 Initiation

- **Project Charter**: A document authorizing the project, outlining objectives such as developing a new app or designing a new electronic component.
- **Stakeholder Analysis**: Identifying stakeholders such as end-users, project sponsors, and regulatory bodies, and understanding their needs and expectations.

2.3.2 Planning

- **Scope Definition**:
  - **Software Project**: Define features, functionalities, and performance criteria for a new software application.
  - **Hardware Project**: Specify components, design requirements, and performance specifications for an electronic device.
- **Schedule Management**:
  - **Gantt Charts**: Schedule tasks for software development phases or hardware design and testing.
  - **Critical Path Method (CPM)**: Identify key tasks for software release or electronic product development to minimize delays.
- **Cost Management**:

    - **Cost Estimation**: Estimate costs for software development tools, licenses, and hardware components.
    - **Budgeting**: Allocate funds for development, testing, and deployment phases.
- **Risk Management**:
    - **Risk Identification**: Identify risks such as software bugs, hardware malfunctions, or integration issues.
    - **Risk Analysis**: Assess potential impacts on project timelines and costs.
    - **Risk Response Planning**: Develop strategies for addressing risks, such as implementing rigorous testing protocols.
- **Quality Management**:
    - **Quality Assurance (QA)**: Implement testing processes, code reviews, and validation for software projects.
    - **Quality Control (QC)**: Perform hardware testing and inspection to ensure that components meet design specifications.

2.3.3 Execution

- **Team Management**: Manage software developers, hardware engineers, and testers, ensuring clear communication and task delegation.
- **Communication Management**: Share updates on project progress, issues, and changes through regular meetings and reports.

2.3.4 Monitoring and Controlling

- **Performance Reporting**: Use tools to track software development progress or hardware design milestones.
- **Change Management**: Handle changes to project scope, such as adding new features to an app or modifying hardware specifications.

2.3.5 Closing

- **Final Deliverables**: Complete and deliver the final software or hardware product to stakeholders.
- **Post-Project Review**: Evaluate the project’s success, including performance, budget adherence, and stakeholder satisfaction.
- **Formal Acceptance**: Obtain sign-off from stakeholders on project deliverables and performance.

2.4 Tools and Techniques

2.4.1 Project Management Software

- **JIRA**: Used for tracking software development tasks and issues.
- **GitHub/GitLab**: Platforms for version control and collaboration on software projects.
- **Altium Designer**: Software for PCB design and electronics layout.

2.4.2 Risk Management Tools

- **Risk Matrix**: Evaluate and prioritize risks associated with software and hardware projects.
- **Monte Carlo Simulation**: Analyze uncertainties in project timelines and costs for complex software and hardware projects.

2.4.3 Cost Estimation Techniques

- **Analogous Estimating**: Use historical data from similar software or hardware projects to estimate costs.
- **Parametric Estimating**: Apply statistical methods to estimate costs based on parameters like lines of code or PCB board area.
- **Bottom-Up Estimating**: Break down project tasks into smaller components and estimate costs for each, then aggregate.

3. What are Team Project Requirements?

3.1 Team Formation

- **Roles and Responsibilities**:
  - **Software Engineer**: Develops and tests software components.
  - **Hardware Engineer**: Designs and tests electronic circuits and components.
  - **Systems Architect**: Integrates software and hardware components into a cohesive system.
  - **Quality Assurance Tester**: Ensures that software and hardware meet quality standards.
- **Skill Sets**:
  - **Technical Skills**: Proficiency in programming languages (e.g., Python, C++), circuit design, and hardware debugging.
  - **Managerial Skills**: Project management, resource allocation, and risk management.
  - **Interpersonal Skills**: Communication, teamwork, and problem-solving abilities.

3.2 Communication

- **Communication Plan**:
  - **Purpose**: Ensures that project updates, issues, and changes are communicated effectively.
  - **Channels**: Includes email, chat platforms (e.g., Slack), and project management tools.
  - **Frequency**: Regular updates and meetings to keep stakeholders informed.
- **Meetings**:
  - **Types**: Daily stand-ups for software teams, weekly design reviews for hardware projects, and monthly progress reports for stakeholders.
  - **Agenda**: Focuses on project status, upcoming milestones, and issue resolution.

    - **Minutes**: Document meeting discussions and action items for follow-up.

3.3 Collaboration Tools

- **Project Management Tools**:
    - **Trello/Asana**: Track tasks, assign responsibilities, and monitor progress.
    - **Microsoft Project**: Create detailed project schedules and manage timelines.
- **Document Sharing**:
    - **Cloud Platforms**: Google Drive, SharePoint for storing and sharing project documents.
    - **Version Control**: GitHub for managing code versions and changes.

3.4 Requirements Gathering

- **Stakeholder Input**:
    - **Interviews**: Gather requirements from end-users, clients, or project sponsors.
    - **Surveys**: Collect feedback on desired features or design requirements.
- **Scope Definition**:
    - **Software Project**: Define application features, performance metrics, and integration requirements.
    - **Hardware Project**: Specify component specifications, design parameters, and performance criteria.

3.5 Performance Monitoring

- **Key Performance Indicators (KPIs)**:
    - **Software Project**: Metrics such as code quality, defect rates, and user satisfaction.
    - **Hardware Project**: Metrics such as component reliability, manufacturing yield, and adherence to design specifications.
- **Feedback Mechanisms**:
    - **Performance Reviews**: Assess individual and team performance, and address any issues.
    - **Issue Resolution**: Processes for resolving technical issues or project delays.

4. Engineering Career Paths and Marketplaces

4.1 Engineering Career Paths

4.1.1 Traditional Engineering Roles

- **Software Developer**:
    - **Responsibilities**: Design, code, test, and debug software applications.
    - **Skills**: Proficiency in programming languages, software design principles, and debugging techniques.

- **Hardware Engineer**:
    - **Responsibilities**: Design and test electronic circuits and systems.
    - **Skills**: Circuit design, PCB layout, and electronic testing.
- **Systems Engineer**:
    - **Responsibilities**: Integrate software and hardware components into a cohesive system.
    - **Skills**: Systems integration, requirement analysis, and system testing.

4.1.2 Specialized Engineering Roles

- **Embedded Systems Engineer**:
    - **Responsibilities**: Develop and optimize embedded software for microcontrollers and other embedded systems.
    - **Skills**: Embedded programming, real-time operating systems (RTOS), and hardware-software interfacing.
- **Network Engineer**:
    - **Responsibilities**: Design, implement, and manage network infrastructure and security.
    - **Skills**: Networking protocols, network security, and troubleshooting.

4.1.3 Management and Consulting

- **Engineering Manager**:
    - **Responsibilities**: Oversee engineering teams, manage projects, and ensure alignment with organizational goals.
    - **Skills**: Leadership, project management, and strategic planning.
- **Consulting Engineer**:
    - **Responsibilities**: Provide expert advice and solutions to clients on technical challenges.
    - **Skills**: Technical expertise, problem-solving, and client management.
- **Entrepreneur/Start-up Founder**:
    - **Responsibilities**: Launch and manage new technology ventures, from concept to market.
    - **Skills**: Innovation, business management, and market analysis.

4.2 Marketplaces for Engineering Professionals

4.2.1 Traditional Employment

- **Technology Companies**: Firms like Google, Microsoft, Intel, and Apple.
- **Research Institutions**: Universities, research labs, and technology research centers.
- **Government Agencies**: Departments and agencies working on technology and infrastructure projects.

4.2.2 Emerging Marketplaces

- **Tech Start-ups**: Innovative companies focusing on emerging technologies like AI, IoT, and blockchain.
- **Freelance and Contract Work**: Platforms like Upwork or Freelancer for short-term projects or consulting roles.
- **Remote Work**: Opportunities for software development, hardware design, and system integration that can be performed remotely.

4.2.3 Professional Networks and Associations

- **Professional Organizations**: IEEE (Institute of Electrical and Electronics Engineers), ACM (Association for Computing Machinery).
- **Online Platforms**: LinkedIn for networking, GitHub for showcasing projects, and specialized forums for technology discussions.

4.3 Skills and Qualifications

- **Technical Skills**: Advanced knowledge in specific computing or electronics domains, such as software development or circuit design.
- **Soft Skills**: Communication, teamwork, and leadership abilities essential for effective project management and collaboration.
- **Certifications**: Relevant certifications such as Certified ScrumMaster (CSM) for agile projects, or CompTIA Network+ for network engineering.

Lecture Summary

This is an overview of engineering project management specifically tailored to the fields of computing and electronics, including examples, processes, and career paths.

Group work

Consider any previous assignment/work of your choice and qualify it as a project.