Chapter-5 Project Scope Management

📌 Introduction

Project Scope Management is a **crucial knowledge area** in project management that focuses on defining, documenting, and controlling the scope of a project. Scope refers to the **boundaries of the project**, including the work required to deliver the project's objectives. Effective scope management ensures that projects **stay on track, meet stakeholder expectations, and avoid scope creep** (uncontrolled expansion of project scope).

📌 Importance of Scope Management

- Prevents **scope creep** by clearly defining what is included and excluded in the project.
- Ensures **stakeholder alignment** by documenting requirements and expectations.
- Helps in **resource planning** by specifying deliverables and tasks.
- Improves **project control** by monitoring scope changes and their impact.
- Contributes to **on-time and within-budget** project delivery.

📌 Key Processes in Project Scope Management

Project Scope Management consists of **six major processes** under the Planning and Monitoring & Controlling process groups.

📌 1. Plan Scope Management (Planning Phase)

This process involves creating a **Scope Management Plan**, which provides guidelines for defining, validating, and controlling project scope. .

Inputs

✔ Project Charter
✔ Project Management Plan
✔ Enterprise Environmental Factors (EEFs)
✔ Organizational Process Assets (OPAs)

Tools & Techniques

🔹 Expert Judgment
🔹 Data Analysis
🔹 Meetings

Outputs

📄 Scope Management Plan
📄 Requirements Management Plan

📌 2. Collect Requirements (Planning Phase)

This process gathers and documents **stakeholder needs** to define project requirements. It ensures that all **functional and non-functional requirements** are understood before defining the scope.

Inputs

✔ Project Charter
✔ Scope Management Plan
✔ Stakeholder Register

Tools & Techniques

- 🔹 Interviews
- 🔹 Focus Groups
- 🔹 Surveys/Questionnaires
- 🔹 Brainstorming
- 🔹 Document Analysis
- 🔹 Prototypes
- 🔹 Requirements Workshops
- 🔹 Benchmarking
- 🔹 Decision-Making Techniques (Voting, Multi-Criteria Decision Analysis)
- 🔹 Observation

Outputs

📄 Requirements Documentation
📄 Requirements Traceability Matrix

📌 3. Define Scope (Planning Phase)

This process involves developing a **detailed project scope statement** that outlines the project's **boundaries, deliverables, constraints, and exclusions**.

Inputs

✔ Project Charter
✔ Scope Management Plan
✔ Requirements Documentation
✔ Organizational Process Assets

Tools & Techniques

- 🔹 Expert Judgment
- 🔹 Product Analysis
- 🔹 Alternatives Analysis
- 🔹 Facilitated Workshops
- 🔹 Decision-Making Techniques

Outputs

📄 Project Scope Statement
📄 Updated Project Documents

📌 4. Create Work Breakdown Structure (WBS) (Planning Phase)

The **WBS** breaks down the project scope into **smaller, manageable components**, making it easier to estimate costs, schedule tasks, and allocate resources.

Inputs

✔ Project Scope Statement
✔ Scope Management Plan
✔ Requirements Documentation
✔ Enterprise Environmental Factors
✔ Organizational Process Assets

Tools & Techniques

- 🔹 Decomposition
- 🔹 Expert Judgment

Outputs

📄 Work Breakdown Structure (WBS)
📄 WBS Dictionary
📄 Scope Baseline

📌 5. Validate Scope (Monitoring & Controlling Phase)

This process ensures that completed project deliverables meet **customer expectations and acceptance criteria**.

Inputs

✔ Project Management Plan
✔ Requirements Documentation
✔ Requirements Traceability Matrix
✔ Verified Deliverables
✔ Work Performance Data

Tools & Techniques

- 🔹 Inspection
- 🔹 Decision-Making Techniques

Outputs

📄 Accepted Deliverables
📄 Change Requests
📄 Work Performance Information

📌 6. Control Scope (Monitoring & Controlling Phase)

This process ensures that **changes to project scope** are properly managed and documented.

Inputs

✔ Project Management Plan
✔ Requirements Documentation
✔ Requirements Traceability Matrix
✔ Work Performance Data
✔ Organizational Process Assets

Tools & Techniques

🔹 Data Analysis (Variance & Trend Analysis)
🔹 Configuration Management System
🔹 Change Control Process

Outputs

📄 Work Performance Information
📄 Change Requests
📄 Updated Project Documents
📄 Updated Scope Baseline

📌 Conclusion

Project Scope Management ensures that the **project stays within its defined boundaries** by systematically collecting, defining, and validating scope. Proper application of **inputs, tools, and techniques** helps project managers prevent **scope creep**, maintain stakeholder alignment, and ensure smooth project execution. 🚀

📌 Scope Creep and How to Manage It

📌 What is Scope Creep?
Scope creep occurs when **new requirements or features** are added to a project without proper approval, often leading to **budget overruns, missed deadlines, and resource strain**.

📌 How to Prevent Scope Creep?

- Clearly define **project scope and requirements** at the start.
- Use a **formal change control process**.
- Maintain a **Requirements Traceability Matrix (RTM)**.
- Regularly validate scope with **stakeholders**.

Work Breakdown Structure (WBS)

A **Work Breakdown Structure (WBS)** is a hierarchical **decomposition** of a project into smaller, manageable components. It helps in organizing and defining the total scope of the project by breaking it down into **deliverables and work packages**.

---

1. Purpose of WBS

- **Divides the project scope** into smaller, manageable tasks.
- **Improves planning and control** by assigning responsibilities.
- **Enhances cost estimation** and resource allocation.
- **Helps track progress** and identify dependencies.

---

2. Structure of a WBS

A WBS is typically represented in a **tree diagram** or **outline format**, following a **top-down approach**. It consists of **multiple levels**, where:

1. **Level 1:** Project Name (Overall Deliverable)
2. **Level 2:** Major Phases or Deliverables
3. **Level 3:** Sub-deliverables or Key Work Categories
4. **Level 4 and Below:** Work Packages (Smallest Work Units)

3. Types of WBS

1. **Deliverable-Based WBS** – Focuses on project outcomes (e.g., software, reports, or physical products).
2. **Phase-Based WBS** – Breaks work into project phases (e.g., Initiation, Planning, Execution, Closure).

---

4. Benefits of Using WBS

✅ **Clarity & Scope Definition** – Clearly defines what is included in the project.
✅ **Better Task Allocation** – Assigns responsibilities to teams.
✅ **Improved Cost & Time Estimation** – Helps estimate time, cost, and effort accurately.
✅ **Effective Risk Management** – Identifies potential risks in each phase.

Question 1:**Rolling wave vs decompostion method of WBS**

Difference Between Rolling Wave and Decomposition Method in WBS

Both **Rolling Wave Planning** and **Decomposition** are techniques used to break down project work into manageable components in a **Work Breakdown Structure (WBS)**, but they differ in approach and application.

---

1. Rolling Wave Planning

Definition

Rolling Wave Planning is a **progressive elaboration** technique where **detailed planning is done for near-term tasks**, while **future tasks are left at a higher level** and are refined as more information becomes available.

Key Features

- Used when **full details of future work are unclear**.
- Plans evolve **as the project progresses**.
- Commonly used in **Agile, research projects, and evolving scopes**.

Example

A **software development project**:

- In the early stages, the WBS includes **only high-level tasks** for later phases (e.g., testing and deployment).
- As the project progresses, **more details are added**, such as specific test cases and deployment strategies.

---

2. Decomposition Method

Definition

Decomposition is a **top-down** approach where the project scope is broken down into **smaller, manageable work packages** at the beginning of the project.

Key Features

- Requires **detailed planning upfront**.
- Uses a **structured breakdown** of work deliverables.
- Typically applied in **predictive (Waterfall) project management**.

Example

A **construction project**:

- The **entire building process** (foundation, structure, plumbing, electrical, finishing) is **fully broken down into smaller work packages** from the start.
- This allows for clear resource allocation, scheduling, and budgeting.

Comparison Table

Feature	Rolling Wave Planning	Decomposition Method
Approach	Progressive elaboration	Top-down breakdown

Feature	Rolling Wave Planning	Decomposition Method
Detail Level	Near-term is detailed; future phases remain high-level	Entire project is fully broken down upfront
Applicability	Evolving or uncertain projects	Well-defined projects
Project Type	Agile, research, R&D	Construction, manufacturing, IT waterfall projects
Example	Software development (features are defined later)	House construction (foundation, walls, plumbing planned in detail)

Question 2.**What do you mean by gold plating and scope creeping? differentiate between project scope and product scope with an example for each concept**

📌 Gold Plating vs. Scope Creep

1️⃣ Gold Plating

Gold plating occurs when a project team **intentionally adds extra features or enhancements** beyond the agreed-upon project scope without client approval. While it may seem beneficial, it can lead to unnecessary costs, schedule delays, and potential dissatisfaction if the additional features are not needed or create complications.

✅ **Example of Gold Plating:**
A software development team **adds an extra feature** to an app (such as a chatbot) that was not requested by the client. This takes additional time and resources but was never part of the original project scope.

2️⃣ Scope Creep

Scope creep refers to **uncontrolled changes or continuous expansion of the project scope** without proper authorization. It often results from unclear requirements, stakeholder influence, or poor change management, leading to budget overruns, delays, and project failure.

❌ **Example of Scope Creep:**
A client initially requests a **basic e-commerce website**, but later keeps adding new requirements, such as **multi-language support, custom dashboards, and AI-powered recommendations**, without adjusting the timeline or budget.

📌 Project Scope vs. Product Scope

Aspect	Project Scope 🚀	Product Scope 🎯
Definition	Defines **all the work** required to complete the project successfully.	Defines the **features and functionalities** of the final product/service.

Aspect	Project Scope 🚀	Product Scope 🎯
Focus	The **tasks, resources, budget, and deadlines** involved in project execution.	The **capabilities, characteristics, and performance** of the final deliverable.
Changes	Controlled through **change management processes**.	Changes might be necessary based on **market demand or user feedback**.
Example	A construction company building a **shopping mall** within a specified budget and timeline.	The final **shopping mall itself**, with features like a food court, parking, and security systems.

📌 Conclusion

- **Gold plating** is the **intentional addition of extra features** beyond requirements.
- **Scope creep** is **uncontrolled expansion** of the project scope.
- **Project scope** focuses on **work execution**, while **product scope** defines the **final product's functionalities**.

Question 3:**Illustrate a WBS for a project of website design of pokhara university**

📌 Work Breakdown Structure (WBS) for Website Design of Pokhara University

A **Work Breakdown Structure (WBS)** divides the project into smaller, manageable components, ensuring efficient planning, execution, and monitoring. Below is a hierarchical **WBS for the website design project of Pokhara University**:

```
1. Website Design for Pokhara University
   ├── 1.1 Project Initiation
   │   ├── 1.1.1 Requirement Gathering
   │   ├── 1.1.2 Stakeholder Identification
   │   ├── 1.1.3 Feasibility Analysis
   │   ├── 1.1.4 Project Charter Approval
   │
   ├── 1.2 Planning Phase
   │   ├── 1.2.1 Define Website Objectives
   │   ├── 1.2.2 Create Work Breakdown Structure (WBS)
   │   ├── 1.2.3 Develop Project Schedule
   │   ├── 1.2.4 Resource Allocation
   │   ├── 1.2.5 Risk Assessment
   │
   ├── 1.3 Website Design
   │   ├── 1.3.1 Wireframing and UI/UX Design
   │   ├── 1.3.2 Approval of Website Mockups
   │   ├── 1.3.3 Front-End Design (HTML, CSS, JavaScript)
   │   ├── 1.3.4 Responsive and Mobile Optimization
   │
   ├── 1.4 Website Development
   │   ├── 1.4.1 Backend Development (Database, Server-Side)
   │   ├── 1.4.2 Content Management System (CMS) Integration
   │   ├── 1.4.3 API Development and Integration
   │   ├── 1.4.4 Security and Authentication Implementation
   │
   ├── 1.5 Content Development
   │   ├── 1.5.1 Academic Programs and Course Details
   │   ├── 1.5.2 Faculty and Staff Information
   │   ├── 1.5.3 News, Events, and Announcements
   │   ├── 1.5.4 Student Portal and Online Services
   │
   ├── 1.6 Testing Phase
   │   ├── 1.6.1 Functional Testing
   │   ├── 1.6.2 Performance Testing
   │   ├── 1.6.3 Security Testing
   │   ├── 1.6.4 Bug Fixing and Debugging
   │
   ├── 1.7 Deployment Phase
   │   ├── 1.7.1 Hosting and Domain Setup
   │   ├── 1.7.2 Website Launch and Go-Live
   │   ├── 1.7.3 Final Approval and Handover
   │
   ├── 1.8 Maintenance and Support
   │   ├── 1.8.1 Regular Updates and Security Patches
   │   ├── 1.8.2 User Training and Documentation
   │   ├── 1.8.3 Feedback Collection and Enhancement
   │   ├── 1.8.4 Backup and Disaster Recovery
```

This **WBS** helps to **organize** the project into structured deliverables, ensuring **clarity, accountability, and efficiency** throughout the **website design process**. Let me know if you need a **graphical** version of the WBS! 🚀

Question 4:**Why scope validation and scope control are important for a project?**

Importance of Scope Validation and Scope Control in a Project

1️⃣ Scope Validation: Ensuring Deliverables Meet Requirements

Scope validation is the process of formally accepting the completed project deliverables. It involves **reviewing, inspecting, and approving** the project work to ensure it meets stakeholder expectations.

✅ Why is Scope Validation Important?

- **Prevents Misalignment with Stakeholder Expectations**: Ensures that the final product matches what was initially agreed upon.
- **Reduces Rework and Cost Overruns**: Detecting scope deviations early prevents expensive rework.
- **Formal Acceptance**: Provides documented approval from stakeholders, reducing disputes later.
- **Improves Customer Satisfaction**: Ensures that the product or service meets quality and functional expectations.
- **Ensures Deliverables Are Complete**: Avoids incomplete handovers and gaps in the final product.

📌 **Example**: If a website development project reaches the final stage, scope validation ensures that all agreed-upon features (e.g., student login, course catalog, online admission portal) are fully functional before the university formally accepts the website.

2️⃣ Scope Control: Managing Changes Effectively

Scope control involves monitoring and managing project scope changes to prevent unauthorized expansions (scope creep). It ensures that any changes go through a proper **change control process**.

✅ Why is Scope Control Important?

- **Prevents Scope Creep**: Uncontrolled changes can lead to project delays and cost overruns.
- **Maintains Project Schedule and Budget**: Ensures that any scope changes are reviewed and their impact assessed.
- **Improves Project Efficiency**: Keeps the team focused on the approved deliverables.
- **Reduces Risk of Project Failure**: Unplanned expansions can drain resources and derail the project.
- **Enhances Change Management**: Ensures a structured approval process for modifications.

📌 **Example**: If the university requests an additional **AI chatbot feature** after the website development has started, scope control ensures the request is formally assessed for feasibility, budget impact, and timeline adjustments before approval.

🔹 Conclusion

Both **Scope Validation** and **Scope Control** are critical for successful project execution:

- **Scope Validation** ensures that the deliverables meet the agreed-upon requirements and are officially accepted.
- **Scope Control** ensures that project changes are managed properly, preventing unauthorized scope expansion.

💡 **Without scope validation, deliverables might not meet stakeholder expectations, leading to rework. Without scope control, uncontrolled changes can derail the project, causing delays and budget overruns.** 🚀

Question 5 -chapter 4:**Develop the project charter for development of pokhara university's mobile application by considering the required parameter of your choice**

PROJECT CHARTER

Pokhara University Mobile Application Development

---

1. PROJECT INFORMATION

Project Name: Pokhara University Mobile Application Development
Project Sponsor: Pokhara University Administration
Project Manager: [Project Manager Name]
Project Start Date: [MM/DD/YYYY]
Project End Date: [MM/DD/YYYY]
Project Budget: $XX,XXX

---

2. PROJECT PURPOSE & JUSTIFICATION

Pokhara University aims to improve accessibility and operational efficiency by developing a mobile application for students, faculty, and administrators. This application will serve as a **centralized digital platform** for academic activities, communication, and university services, reducing manual processes and increasing engagement. The project is expected to enhance the overall student experience and administrative workflow.

---

3. PROJECT OBJECTIVES

- Develop an **Android and iOS** mobile application with a user-friendly interface.
- Provide **real-time access** to academic records, notices, and course materials.
- Enable **secure authentication** for students, faculty, and staff.
- Integrate **course registration, exam results, and library services**.
- Improve communication through **push notifications and messaging features**.
- Ensure **data security and compliance** with university policies.

---

4. PROJECT SCOPE

In-Scope:

✔ Mobile application for Android and iOS platforms.
✔ Student, faculty, and administrative portals.
✔ Integration with university ERP and database systems.
✔ Push notifications for academic updates.
✔ User authentication and role-based access.
✔ Testing, deployment, and user training.
✔ Maintenance and post-launch support.

Out of Scope:

✘ Hardware procurement or upgrades.
✘ Third-party software licensing beyond initial ERP integration.
✘ Content creation for learning materials.

---

5. STAKEHOLDERS

Stakeholder	Role & Responsibility

Stakeholder	Role & Responsibility
Pokhara University Administration	Project sponsor, budget approval, and governance.
IT Department	Infrastructure support, security, and compliance.
Faculty Members	Provide feedback, testing, and content input.
Students	End users, adoption, and feedback.
Software Development Team	Design, development, testing, and deployment.
ERP Team	Integration with existing university systems.

6. DELIVERABLES & MILESTONES

Milestone	Deliverable	Completion Date
Project Initiation	Project charter approval	MM/DD/YYYY
Requirement Analysis	Business and technical requirements document	MM/DD/YYYY
UI/UX Design	Wireframes, prototypes, and design approval	MM/DD/YYYY
Development Phase 1	Core functionalities (login, dashboard, notifications)	MM/DD/YYYY
Development Phase 2	Advanced features (course reg., results, library)	MM/DD/YYYY
Testing & Quality Assurance	Bug fixes and performance testing	MM/DD/YYYY
Beta Launch	User acceptance testing (UAT)	MM/DD/YYYY
Full Deployment	Public release on Play Store & App Store	MM/DD/YYYY
Post-Launch Support	Ongoing maintenance and feature updates	Ongoing

7. BUDGET ESTIMATE

Component	Estimated Cost (USD)
App Design & Development	$XX,XXX
ERP Integration	$X,XXX
Cloud Hosting & Security	$X,XXX
Testing & Quality Assurance	$X,XXX
Training & Documentation	$X,XXX
Maintenance & Support	$X,XXX per year
Total Estimated Cost	**$XX,XXX**

8. RISK MANAGEMENT

Risk	Probability	Impact	Mitigation Strategy
Scope Creep	High	High	Define clear requirements and enforce change management.
Security Breach	Medium	High	Implement encryption, two-factor authentication, and security audits.
Technical Challenges	Medium	Medium	Allocate skilled developers and conduct thorough testing.
User Adoption Issues	Low	High	Conduct training, gather feedback, and iterate based on user needs.
Integration Failures	Medium	High	Perform compatibility checks with university ERP before deployment.

9. PROJECT ASSUMPTIONS & CONSTRAINTS

Assumptions:

- University ERP system is accessible for integration.
- Adequate funding and human resources are available.
- Stakeholders actively participate in feedback and testing.
- Users (students, faculty) will adopt and engage with the application.

Constraints:

- Fixed budget and delivery timeline.
- Compliance with university policies and data privacy regulations.
- Platform compatibility (Android & iOS development considerations).

10. SUCCESS CRITERIA

✔ Successful development and launch of the mobile application.
✔ At least **80% adoption rate** among students and faculty within six months.
✔ Improved administrative efficiency through process automation.
✔ User satisfaction rating of **4.5+ stars** in app stores.
✔ No major security or data breaches.

11. APPROVAL SIGNATURES

Name	Role	Signature	Date
[Sponsor Name]	Project Sponsor	___________	MM/DD/YYYY
[Project Manager]	Project Manager	___________	MM/DD/YYYY
[IT Department Head]	IT Head	___________	MM/DD/YYYY

Name	Role	Signature	Date
[ERP Integration Lead]	ERP Team	___________	MM/DD/YYYY

This document serves as the official Project Charter for the Pokhara University Mobile Application Development. Any changes to scope, timeline, or budget must be reviewed and approved by relevant stakeholders.

Question 6 chapter-5:**Describe about PM plan.What are the contents that a PM plan must address? Dicuss in brief**

Project Management Plan (PM Plan)

A **Project Management Plan (PM Plan)** is a formal, approved document that outlines how a project will be executed, monitored, controlled, and closed. It serves as a **roadmap** for project execution, providing guidance on processes, roles, and responsibilities to ensure successful project completion. The PM Plan integrates all subsidiary plans and project baselines, ensuring alignment with project objectives.

Key Contents of a Project Management Plan

A **comprehensive PM Plan** should address the following key areas:

1. Inputs

1. **Project Charter** – The foundational document that outlines the project's objectives, stakeholders, and high-level scope.
2. **Outputs from Other Planning Processes** – Various subsidiary plans such as Scope Management, Schedule Management, and Risk Management Plans contribute to the overall PM Plan.
3. **Enterprise Environmental Factors (EEF)** – Organizational culture, industry regulations, government policies, and market conditions.
4. **Organizational Process Assets (OPA)** – Templates, past project documentation, standards, and best practices followed within the organization.

---

2. Tools and Techniques

1. **Expert Judgment** – Experienced project managers, stakeholders, and subject matter experts provide insights.
2. **Data Gathering** – Includes brainstorming, interviews, and research to collect project information.
3. **Interpersonal & Team Skills** – Facilitation, conflict resolution, and leadership skills for effective planning.
4. **Meetings** – Regular planning sessions and stakeholder discussions ensure alignment on project requirements.
5. **Decision-Making Techniques** – Methods such as multi-criteria decision analysis, voting, and prioritization to finalize planning components.

---

3. Outputs

1. **Project Management Plan Document** – A formal, approved document consolidating all subsidiary plans and baselines.
2. **Project Baselines**
   - **Scope Baseline** – WBS, scope statement, and project boundaries.
   - **Schedule Baseline** – Project timeline and milestones.
   - **Cost Baseline** – Estimated budget, cost breakdown, and funding sources.
3. **Subsidiary Plans** (Scope, Schedule, Cost, Quality, Risk, Communication, Resource, Procurement, and Stakeholder Management Plans).
4. **Change Management Procedures** – Defines how changes in scope, time, and cost will be managed.

Conclusion

A well-defined **PM Plan** ensures a structured approach to managing projects, reducing risks, and improving efficiency. It serves as a **blueprint** for guiding the project from initiation to completion, ensuring stakeholder alignment and project success. 🚀

Question 6:**Explain project and product scope. What are the key points to consider while collecting requirements for a project?**

Project Scope vs. Product Scope

1. Project Scope

- Defines the **work required** to deliver a product, service, or result with specified features and functions.
- Focuses on **processes, resources, deliverables, and constraints** necessary to complete the project.
- Example: Developing a **university mobile application** involves defining tasks such as UI/UX design, coding, testing, deployment, and support.

2. Product Scope

- Defines the **features and functionalities** of the product being delivered.
- Focuses on **product specifications, performance, and quality requirements**.
- Example: The university mobile app's **product scope** includes features like **student login, exam results, notifications, and library access**.

Key Difference:

- **Project Scope = "How" the work is done (Processes & Activities).**
- **Product Scope = "What" is being delivered (Features & Functionalities).**

Key Points to Consider While Collecting Requirements for a Project

Collecting requirements is a crucial step in project planning to ensure stakeholder needs are well understood. Key considerations include:

1. **Engaging Stakeholders** – Identify key stakeholders (clients, users, management) to gather expectations.
2. **Clear Objectives** – Define what the project must achieve to align with business goals.

3. **Functional & Non-Functional Requirements** – Capture both **features (functional)** and **performance, security, and usability aspects (non-functional)**.
4. **Use Multiple Techniques** – Use **interviews, surveys, workshops, brainstorming, and prototyping** to collect comprehensive requirements.
5. **Prioritization of Requirements** – Categorize requirements as **must-have, should-have, or nice-to-have** to manage scope effectively.
6. **Avoid Ambiguity** – Ensure requirements are **clear, measurable, and testable** to avoid misunderstandings later.
7. **Compliance & Constraints** – Consider **legal, regulatory, budget, and timeline constraints** while gathering requirements.
8. **Traceability & Documentation** – Maintain a **Requirements Traceability Matrix (RTM)** to track and verify requirements throughout the project lifecycle.
9. **Scope Validation** – Continuously review and validate requirements with stakeholders to prevent scope creep.

Conclusion

A well-defined **project scope** ensures that the necessary work is executed, while a clear **product scope** guarantees that the final deliverable meets user needs. Effective **requirement collection** reduces risks, prevents misunderstandings, and ensures project success. 🚀