

Introduction to IT Systems for Beginners

Maciej Gowin @ [CoderBrother](#)

All rights reserved

Gathering Requirements

Gathering requirements is the process of eliciting, analyzing, and documenting the needs and expectations of stakeholders for a project or system. It lays the foundation for successful project development.

Importance

- Ensures that the final product meets user needs and expectations.
- Reduces the risk of project scope creep and miscommunication.
- Helps in creating a clear project plan and timeline.

Types of Requirements

- **Functional Requirements:** Specify what the system should do (e.g., features, tasks).
- **Non-Functional Requirements:** Describe how the system performs (e.g., usability, reliability, performance).
- **Technical Requirements:** Define technical specifications, tools, and technologies needed.

Methods for Gathering Requirements

- **Interviews:** One-on-one discussions with stakeholders to understand their needs.
- **Surveys and Questionnaires:** Collecting data from a larger group to identify common needs.
- **Workshops:** Collaborative sessions to brainstorm and discuss requirements with various stakeholders.
- **Observation:** Analyzing users in their environment to understand workflows and pain points.
- **Prototyping:** Creating mockups or models of the system for stakeholders to visualize and provide feedback.

Best Practices

- Engage all relevant stakeholders early and throughout the process.
- Document requirements clearly and maintain traceability.
- Validate requirements with stakeholders to ensure accuracy.

Roadmap and Milestones in Project Development

A **roadmap** is a **strategic plan** that outlines the vision, direction, and progress of a project, detailing how to achieve the goals defined in the requirements-gathering phase.

Milestones are **significant points** or events in the project timeline that indicate progress and completion of key deliverables.

Roadmap Importance

- Provides a clear vision and direction for the project team.
- Helps align stakeholders on project objectives and timelines.
- Facilitates tracking of progress and identification of any delays or issues.

Components of a Roadmap

- **Project Goals:** High-level objectives that the project aims to achieve.
- **Timeline:** An overview of the project schedule, indicating phases and key dates.
- **Key Activities:** Major tasks or phases that need to be completed.
- **Dependencies:** Relationships between tasks, showing how one task may rely on the completion of another.

Roadmap and Milestones Best Practices

- **Regular Updates:** Continuously update the roadmap to reflect changes and progress.
- **Stakeholder Communication:** Regularly communicate milestones and overall progress to stakeholders to maintain alignment.
- **Celebrate Milestones:** Acknowledge the completion of milestones to motivate the team and recognize their hard work.

Prototyping

Prototyping is the process of **creating an early model or sample** of a software application to visualize and test concepts, functionalities, and user interactions before full-scale development.

Purpose

- Helps stakeholders understand the proposed system and its features.
- Facilitates early user feedback to refine requirements and design.
- Reduces risks by identifying issues and misunderstandings early in the development process.

Prototyping Benefit

- Enhances user engagement by involving stakeholders early in the design process.
- Encourages iterative development and continuous improvement.
- Saves time and resources by identifying problems before full-scale development.

Types of Prototypes

- **Low-Fidelity Prototypes**

- Basic sketches or wireframes that represent layout and structure.
- Useful for brainstorming and gathering initial feedback.

- **High-Fidelity Prototypes**

- Detailed and interactive models that closely resemble the final product.
- Allows users to experience functionality and design more realistically.

- **Functional Prototypes**

- Partially developed systems that include core functionalities for testing.
- Useful for validating technical feasibility and design choices.

Prototyping Methods

- **Paper Prototyping:** Creating hand-drawn sketches of the user interface.
- **Digital Wireframes:** Using design tools to create digital representations of screens.
- **Interactive Prototyping:** Utilizing tools (e.g., Figma, Axure) to build clickable prototypes.
- **Agile Prototyping:** Developing prototypes iteratively as part of Agile methodologies.

Minimum Viable Product (MVP)

A **basic version of a product** with enough features to satisfy early adopters and gather user feedback for future development.

Purpose

- Validate product ideas and market demand.
- Test hypotheses and gather user feedback for improvement.

Characteristics

- Focuses on core functionalities.
- Developed for real users.
- Iterative enhancements based on feedback.

Proof of Concept (PoC)

A demonstration or prototype to verify that a certain concept or idea is feasible and can be developed into a viable product.

Purpose

- Assess technical feasibility and practical potential.
- Minimize risks by exploring the idea before full development.

Characteristics

- Often non-functional or limited in scope.
- Used primarily for internal evaluation.
- May not be user-facing or fully developed.

When to Use MVP and PoC

MVP

- When you want to enter the market quickly and test the viability of a product idea.
- Useful for startups looking to attract early users and investors.

PoC

- When exploring new technologies or innovative ideas to determine if they are achievable.
- Suitable for projects requiring validation before investing significant resources.

Task definition

The process of breaking down high-level requirements into actionable tasks that can be assigned, tracked, and completed within a project.

Importance

- Provides clarity and direction for the development team.
- Ensures that all aspects of the requirements are addressed.
- Facilitates project planning, resource allocation, and progress tracking.

Steps to Convert Requirements into Tasks

1. Analyze Requirements

- Review and understand each requirement in detail to identify specific needs and expectations.

2. Prioritize Requirements

- Assess the importance and urgency of each requirement to determine which tasks should be completed first.

3. Break Down Requirements

- Decompose each requirement into smaller, manageable tasks or sub-tasks.
- Consider dependencies and logical groupings.

Steps to Convert Requirements into Tasks

4. Define Acceptance Criteria

- Establish clear criteria for what constitutes task completion to ensure quality and alignment with requirements.

5. Assign Tasks

- Allocate tasks to team members based on their skills and availability, ensuring a balanced workload.

6. Create a Timeline

- Develop a timeline for task completion, incorporating milestones and deadlines to track progress.

Project and Task Management Tools

Project and task management tools are **software applications that help** teams plan, execute, and monitor projects effectively.

They facilitate collaboration, organization, and communication within teams.

Benefits of Using Management Tools

- **Enhance productivity** by streamlining workflows and improving task visibility.
- Enable better **resource allocation** and deadline management.
- **Foster collaboration** and communication among team members and stakeholders.
- **Centralized information** for easy access and tracking.
- Improved accountability through task assignments and **status updates**.
- **Enhanced collaboration** with features for communication and file sharing.
- **Data-driven insights** for better decision-making and project planning.

Popular Project Management Tools

| Tool | Description | Key Features |
|--------------------------|--|---|
| Jira | A powerful tool for agile project management and issue tracking. | Scrum and Kanban support, Custom workflows, Reporting |
| Trello | A visual project management tool using boards and cards for tasks. | Kanban boards, Checklists, Due dates, Labels |
| Asana | A task management tool that helps teams organize and track work. | Task assignments, Project timelines, Dashboards |
| Microsoft Project | A comprehensive project management software for planning and scheduling. | Gantt charts, Resource management, Task dependencies |

Project Methodologies

Project methodologies are structured frameworks or approaches used to plan, execute, and manage projects. They provide guidelines and best practices for teams to achieve project objectives efficiently.

Importance

- Ensures consistency and standardization in project execution.
- Enhances communication and collaboration among team members.
- Improves the likelihood of project success by providing clear processes.

Common Project Methodologies

| Methodology | Description | Characteristics |
|------------------|---|--|
| Waterfall | A linear and sequential approach where each phase must be completed before moving to the next. | Clear project phases; Fixed requirements; Suitable for smaller, well-defined projects. |
| Agile | An iterative and incremental approach that focuses on collaboration, customer feedback, and flexibility. | Short iterations (sprints); Continuous improvement; Regular stakeholder feedback. |
| Scrum | A framework within Agile that uses fixed-length sprints to deliver incremental value. | Defined roles (Scrum Master, Product Owner, Development Team); Daily stand-ups; Sprint reviews and retrospectives. |
| Kanban | A visual management method for workflow optimization, focusing on continuous delivery without overloading team members. | Visual boards; Work in progress limits; Flexible and adaptive. |
| Lean | A methodology focused on maximizing value while minimizing waste in processes. | Continuous improvement; Efficiency-oriented; Emphasis on customer value. |

Comparing Project Methodologies

| Aspect | Waterfall | Agile | Scrum | Kanban |
|-----------------------------|-----------------------|----------------------------|----------------------|------------------------------|
| Flexibility | Low | High | High | High |
| Customer Involvement | Limited to start/end | Continuous | Continuous | Continuous |
| Delivery | At the end | Incremental | Incremental | Continuous |
| Documentation | Extensive | Minimal | Minimal | Minimal |
| Best Suited For | Clear, fixed projects | Complex, evolving projects | Software development | Ongoing, continuous projects |

Choosing the Right Methodology

- **Project Size and Complexity:** Larger, more complex projects may benefit from Agile or Scrum, while smaller projects may be suited for Waterfall.
- **Team Structure:** The skills and experience of team members can influence the choice of methodology.
- **Stakeholder Engagement:** High stakeholder involvement may warrant an Agile or Scrum approach for regular feedback.
- **Timeline and Flexibility:** If the project requires quick adaptations, Agile or Kanban methodologies may be more appropriate.
- **Nature of the Project:** Understand if the project is research-oriented, product development, or operational improvement to select the most suitable methodology.