1. **Requirement Gathering** – This is the detective work of a project, where business analysts and stakeholders collaborate to unearth the core needs of a system. It’s not just about collecting data but understanding the "why" behind every requirement. The more thorough the gathering, the smoother the execution.
2. **Requirement Analysis** – Think of this as refining raw ideas into polished diamonds. It involves breaking down gathered needs, eliminating contradictions, and ensuring alignment with business objectives. The goal is to create a foundation where every requirement makes sense technically and strategically.
3. **BRD (Business Requirements Document)** – This document is the blueprint of business goals, capturing what the company wants to achieve through a project. It doesn’t focus on how the system will work but what value it will bring. A well-crafted BRD ensures that business and IT teams are on the same page from the start.
4. **FRD (Functional Requirements Document)** – If BRD is the "what," FRD is the "how." It defines system behaviours, interactions, and features in detail. Developers and testers rely on this document to build and validate functionalities, ensuring every button clicked, data processed, and workflow executed aligns with business needs.
5. **Non-FRD (Non-Functional Requirements Document)** – While FRD defines what a system does, non-FRD defines how well it does it. This includes speed, security, reliability, and user experience. A system may work perfectly but still fail if it doesn’t load quickly or ensure data privacy—this document safeguards against such pitfalls.
6. **Stakeholder** – A stakeholder is anyone whose expectations or interests influence the project’s success. They can be champions or blockers, so understanding their perspectives is crucial. Engaging the right stakeholders early ensures fewer surprises and a smoother journey from concept to completion.
7. **Product Owner** – The product owner is the visionary who transforms abstract ideas into tangible priorities. They don’t just define features; they decide what brings the most value to the users. A great product owner balances business goals, user expectations, and technical feasibility to create a product that truly matters.
8. **Use Case** – A use case is like a screenplay for system interactions, defining how users and technology collaborate to achieve a goal. It maps out the possibilities, including expected actions and potential roadblocks, ensuring every scenario is accounted for in system design.
9. **User Stories** – User stories are the heartbeat of agile development, capturing requirements from an end-user’s perspective. They are not just tasks; they tell a story of how a system should function to improve lives. A well-written user story keeps development focused on real-world needs rather than just technical implementations.
10. **Use Case Documents** – These serve as a playbook for system behaviour, detailing actors, triggers, flows, and exceptions. They help developers anticipate real-world usage and ensure no critical function is overlooked. A comprehensive use case document acts as a bridge between business logic and technical execution.
11. **Business Value** – Business value isn’t just about revenue—it’s about impact. Whether it's reducing operational costs, improving user satisfaction, or gaining a competitive edge, every project should be measured by the difference it makes. High-value tasks should be prioritized to maximize benefits in the shortest time.
12. **Complexity Points** – Not all tasks are created equal—some are straightforward, while others require deep problem-solving. Complexity points measure the level of effort, technical challenges, and uncertainties involved. They help teams balance workloads, ensuring efficient use of time and resources.
13. **Tasks –** Tasks are the building blocks of a project, each representing a small yet essential step toward completion. They turn broad objectives into actionable work items, allowing teams to track progress in a tangible way. Well-defined tasks lead to a smooth execution with minimal confusion.
14. **Activity Diagram –** This is the choreography of workflows, illustrating how processes flow from start to finish. It visually captures decisions, actions, and dependencies, making complex business logic easier to understand. A well-structured activity diagram helps uncover inefficiencies before they become bottlenecks.
15. **Product Backlogs –** A product backlog is like a wish list of everything a project aspires to achieve. Constantly evolving, it holds features, improvements, and fixes, all ranked by priority. A well-maintained backlog ensures that teams always work on what matters most.
16. **Sprint Backlogs –** This is a focused to-do list derived from the product backlog, containing everything to be completed within a sprint. It provides a clear scope and goal for the team, ensuring every sprint delivers meaningful progress. A good sprint backlog keeps teams efficient and goal-oriented.
17. **Sprint Planning –** Sprint planning is where big ambitions are broken down into achievable goals. Teams analyse priorities, estimate efforts, and set realistic sprint targets. A well-planned sprint fosters efficiency, ensuring work is structured and aligned with business needs.
18. **User Acceptance Test (UAT) –** UAT is where theory meets reality, as real users validate if the system functions as expected. It’s the last line of defence before deployment, ensuring that the product isn’t just technically sound but also practical and user-friendly.
19. **Sprint Meetings –** These include daily stand-ups, reviews, and retrospectives—each playing a role in keeping teams aligned and productive. Sprint meetings foster communication, remove roadblocks, and ensure continuous improvement throughout the development process.
20. **Sprint Retrospective –** Reflection is key to progress. A sprint retrospective allows teams to analyse what worked, what didn’t, and how to improve. The best teams embrace feedback, turning every sprint into a learning experience that drives future success.