ASSIGNMENT – 3

1. BUILD

A definition is a statement that explains the meaning of a word, phrase, concept, or idea. It serves to clarify understanding by providing a concise explanation of what something is or what it entails

2. PROJECT

A project is a temporary endeavor undertaken to create a unique product, service, or result. It typically has a defined beginning and end, with specific objectives, constraints, and resources allocated to achieve its goals.

3. VERSION

Versions refer to different iterations or variations of a product, software, or any creative work. Each version typically represents an updated or modified version of the original, often incorporating improvements, bug fixes, or new features.

4. DEPLOYEMENT

Deployment refers to the process of making a software application or system available and operational for use by its intended users.

Deployment often involves tasks such as installing, configuring, and launching the software to ensure it functions as intended in its target environment.

5. BETA:

Beta refers to a phase in software development where a pre-release version of a product is made available to a limited group of users for testing and feedback purposes. This version is not yet considered final and may contain bugs or incomplete features.

Beta testing allows developers to gather real-world usage data and identify issues before the official release, helping to improve the product's quality and user experience.

6. WALKTHROUGH :

A walkthrough is a step-by-step guided explanation or demonstration of a process, task, or procedure. It's often used in various contexts, such as software development or learning environments, to provide clear instructions or guidance on how to complete a particular activity.

7. BRS :

BRS stands for "Business Requirements Specification." It is a document that outlines the high-level objectives, goals, and needs of a business for a particular project or system. The BRS typically describes what the business expects the project or system to accomplish, without detailing how it will be implemented.

It serves as a foundation for the development team to understand the business requirements and design a solution that meets those needs.

8. CRS :

cRS typically stands for "Customer Relationship Management" in business contexts. However, in other contexts, it can refer to "Common Reporting Standard" in the financial domain. Here's a simple definition for each:

Customer Relationship Management (CRM): CRS is a strategy, technology, or system used by businesses to manage interactions and relationships with current and potential customers. It involves collecting and analyzing data to improve customer satisfaction, retention, and ultimately drive sales and profitability.

Common Reporting Standard (CRS): CRS is an international standard for the automatic exchange of financial account information between tax authorities. It aims to combat tax evasion by ensuring that financial institutions report information about their customers' accounts to the relevant tax authorities.

9. RAD :

RAD stands for "Rapid Application Development." It's a software development methodology that prioritizes rapid prototyping and iterative development over planning and documentation. RAD aims to accelerate the development process by emphasizing user feedback and collaboration between developers and stakeholders.

It typically involves creating working prototypes quickly, gathering feedback, and making continuous improvements to deliver a functional product in shorter timeframes compared to traditional development methods.

10 . **PROTOTYPE**

A prototype is a preliminary version or sample of a product, system, or concept that is created to test and demonstrate its feasibility, functionality, and design. It's often developed early in the design process to allow stakeholders to visualize and interact with the proposed solution, gather feedback, and identify potential improvements or issues before full-scale production or implementation.

11.REFACTOR :

To "refactor" in software development means to restructure or rewrite existing code without changing its external behavior. It involves making improvements to the code's internal structure, organization, or readability to enhance maintainability, scalability, or performance.

Refactoring aims to clean up the codebase, remove redundancies, improve clarity, and make it easier to understand and modify in the future without altering its observable functionality.