A quick reference material

Kallil de Araujo

May 25, 2023

Contents

Abstract

When studying for job interviews I frequently had to spend quite some time researching possible concepts that could come up. So, to make my life less complicated, I created this file with the objective of helping me save some time when studying for those situations.

Chapter 1

Methodologies

In the IT context, a methodology refers to a structured approach or set of principles and practices that guide the planning, development, implementation, and management of IT projects or processes. It provides a systematic framework for organizing and executing tasks, as well as a common language and set of tools to facilitate collaboration among team members. A methodology encompasses a range of activities, techniques, and guidelines tailored to address specific challenges and ensure successful outcomes in IT projects.

A methodology typically outlines a step-by-step process for undertaking IT initiatives, including project initiation, requirements gathering, design, development, testing, deployment, and maintenance. It may incorporate various methodologies or frameworks, such as Agile, Waterfall, Scrum, or Lean, depending on the specific needs of the project. The choice of methodology often depends on factors such as project size, complexity, team structure, and organizational culture.

A well-defined methodology brings several benefits to the IT context. It promotes consistency and standardization in project execution, allowing for more predictable outcomes and better resource management. It helps identify and mitigate risks early in the project lifecycle, facilitating proactive decision-making. Additionally, a methodology supports effective communication and collaboration among project stakeholders, ensuring shared understanding and alignment of objectives. By providing a structured approach, a methodology enhances project transparency, facilitates progress tracking, and enables continuous improvement through feedback and lessons learned.

1.1 Agile

Agile is an iterative and collaborative approach that focuses on flexibility and adaptability. It emphasizes frequent communication, incremental development, and delivering working software in short iterations.

1.2 Waterfall

Waterfall is a sequential and linear approach where each phase of a project is completed before moving to the next one. It follows a structured process with defined milestones, making it suitable for projects with well-defined requirements and minimal changes.

1.3 Scrum

Scrum is an Agile framework that utilizes cross-functional teams to deliver software incrementally. It involves iterative sprints, daily stand-up meetings, and continuous collaboration to foster flexibility and rapid development.

1.4 Kanban

Kanban is a visual project management approach that focuses on continuous delivery and workflow optimization. It uses a Kanban board to visualize tasks, limit work in progress, and optimize the flow of work.

1.5 Lean

Lean methodology aims to eliminate waste and maximize value in the software development process. It emphasizes continuous improvement, customer focus, and efficient resource utilization.

1.6 DevOps

DevOps combines software development (Dev) and IT operations (Ops) to enhance collaboration and streamline the software delivery lifecycle. It focuses on automation, continuous integration, continuous delivery, and close collaboration between development and operations teams.

1.7 Rapid Application Development (RAD)

RAD is a methodology that emphasizes rapid prototyping and iterative development. It involves user feedback and frequent iterations to quickly build and refine software applications.

1.8 Extreme Programming (XP)

XP is an Agile methodology that emphasizes close collaboration, continuous feedback, and a high degree of customer involvement. It emphasizes practices like test-driven development, pair programming, and frequent releases.

1.9 Spiral

The Spiral methodology is a risk-driven approach that combines elements of both waterfall and iterative development. It involves iterative cycles where the project progresses through planning, risk analysis, prototyping, and customer evaluation.

1.10 Feature-Driven Development (FDD)

FDD is an iterative and incremental methodology focused on delivering tangible features quickly. It emphasizes domain modeling, iterative development, and feature-level planning and tracking.

Chapter 2

Important ETL Concepts

In the context of IT, ETL stands for Extract, Transform, Load, which is a process used to integrate and consolidate data from various sources into a target system or data warehouse. ETL plays a crucial role in data management and enables organizations to extract valuable insights and make informed business decisions.

The first step in the ETL process is extraction, where data is collected from different sources such as databases, files, APIs, or external systems. This involves retrieving relevant data based on defined criteria and requirements. Once the data is extracted, it undergoes transformation, which involves cleaning, validating, and structuring the data to ensure consistency, accuracy, and compatibility with the target system. Transformations may include data cleansing, data enrichment, data aggregation, and applying business rules or calculations.

Finally, the transformed data is loaded into the target system, typically a data warehouse or a data mart. This step involves mapping the transformed data to the appropriate tables or entities in the target system, ensuring data integrity and maintaining the overall data structure. The loaded data can then be used for reporting, analysis, and decision-making purposes.

2.1 Data Extraction

Understanding various methods and techniques for extracting data from different sources such as databases, files, APIs, or web scraping.

2.2 Data Transformation

Familiarity with data transformation techniques, including data cleansing, data validation, data enrichment, and data aggregation.

2.3 Data Quality

Knowledge of data quality assessment and improvement techniques to ensure accurate and reliable data in the ETL process.

2.4 Data Integration

Understanding how to integrate data from multiple sources into a unified and coherent format, resolving data inconsistencies, and handling data mapping and data merging.

2.5 Data Modeling

Proficiency in data modeling concepts and techniques, including dimensional modeling, relational modeling, and schema design to support efficient data storage and retrieval.

2.6 ETL Architecture

Understanding the overall architecture and components of ETL systems, including staging areas, data warehouses, data marts, and data pipelines.

2.7 ETL Tools and Technologies

Familiarity with popular ETL tools such as Informatica PowerCenter, Talend, SSIS, and understanding their functionalities, capabilities, and best practices.

2.8 Data Governance

Knowledge of data governance principles, policies, and practices, including data lineage, data security, data privacy, and compliance regulations.

2.9 Performance Optimization

Skills in optimizing ETL processes for performance and scalability, including data partitioning, indexing, parallel processing, and query optimization.

2.10 Error Handling and Logging

Understanding techniques for handling errors and exceptions during the ETL process, implementing error logging, and implementing appropriate error handling strategies.

2.11 Change Data Capture (CDC)

Knowledge of CDC techniques to capture and process incremental data changes efficiently, ensuring synchronization between source and target systems.

2.12 Metadata Management

Understanding the importance of metadata in ETL processes, including metadata extraction, storage, and management for data lineage, data profiling, and data documentation.

2.13 Workflow Orchestration

Familiarity with workflow management tools such as Apache Airflow, Oozie, or Luigi for orchestrating complex ETL workflows and scheduling jobs.

2.14 Data Security

Knowledge of data security principles and practices, including encryption, access controls, and data anonymization techniques to protect sensitive data during ETL operations.

2.15 Data Warehousing

Understanding the fundamentals of data warehousing, including star schemas, snowflake schemas, slowly changing dimensions (SCDs), and dimensional hierarchies.

2.15.1 Slowly Changing Dimensions (SCDs)

Slow Changing Dimensions refer to the handling and management of slowly changing data in a data warehousing environment. In ETL processes, SCDs are used to track and manage changes to dimensional attributes over time.

SCDs are essential because they enable the capture and representation of historical data in a data warehouse. This is particularly useful when analyzing trends, performing historical comparisons, or conducting time-series analysis.

There are different types of Slow Changing Dimensions, including:

- Type 1 SCD: Overwrite the existing dimension attribute with the new value, effectively losing the historical information. This approach is suitable when historical data is not needed or when the dimension attribute is not expected to change over time.

- Type 2 SCD: Maintain a separate row for each change in the dimension attribute, creating a new record with an updated attribute value and an assigned surrogate key. This approach allows for historical tracking but can lead to data redundancy.

- Type 3 SCD: Add new columns to the dimension table to track specific changes, typically storing the current value and previous value of the attribute. This approach offers limited historical tracking but can help maintain simplicity in the data model.

- Type 4 SCD: Create a separate history table to store all changes to the dimension attribute, while the main dimension table only contains the current value. This approach provides comprehensive historical tracking while minimizing redundancy in the main dimension table.