Software Development KPIs (Translated from Dutch)

Here is the translation of the provided software development Key Performance Indicators (KPIs), categorized by area and level.

Analysis

- **S1.1** Gathering and validating functional requirements for a software system with a single stakeholder according to a standard method.
- **S1.2** Defining acceptance criteria for the aforementioned functional requirements.
- **S2.1** Performing a requirements analysis for a software system with multiple stakeholders, considering quality attributes including security.
- **S2.2** Performing an analysis to formulate and validate the functionality, security, design, interfaces, etc., of an existing system or component.
- **S2.3** Creating an acceptance test based on quality attributes.
- **S3.1** Performing a requirements analysis for a software system with multiple stakeholders in the context of existing systems.
- **S3.2** Defining acceptance criteria based on quality attributes and a performed risk analysis, including attention to security aspects.

Advice

- **S1.1** Making recommendations about specific requirements of a software system based on research into existing, comparable systems.
- **S2.1** Advising on the procurement and selection of software components when developing a software system, where cost aspects may play a role.
- **S2.2** Advising on a part of an architecture or a limited software system.
- **S2.3** Advising on the use of prototypes for validating requirements.
- **S3.1** Advising regarding the choice of software architecture or existing software frameworks, where cost aspects and quality characteristics such as availability, performance, security, and scalability play a role.
- **S3.2** Advising on the approach for processing and consulting large amounts of data with attention to privacy.
- **S3.3** Advising on the setup of a software development process, including the testing process.

Design

- **S1.1** Creating a design for a software system, including a database, using modeling techniques according to a standard method.
- **S2.1** Creating a design for a software system, taking into account the use of existing components and libraries.
- **S2.2** Applying design quality criteria, considering security aspects and different

types of devices.
- **S2.3** Creating a design for a system that can process and consult large amounts of data.
- **S2.4** Determining the quality of the design, for example, through review or prototyping, considering the formulated quality attributes.
- **S2.5** Creating test designs according to a given test strategy.
- **S3.1** Creating a software architecture for a software system, composed of existing and new systems, considering multiple stakeholders and quality characteristics, including security and scalability.
- **S3.2** Creating a test strategy for system testing.

Realisation (Implementation/Building)

- **S1.1** Building, testing, and deploying a simple software system. Setting up, populating, and querying a database is part of the software system.
- **S2.1** Building and deploying a software system consisting of multiple subsystems, using existing components.
- **S2.2** Integrating software components into an existing system, monitoring integrity, security, and system performance, among other things.
- **S2.3** Performing, monitoring, and reporting on unit, integration, regression, and system tests, with attention to security aspects.
- **S3.1** Building and deploying a scalable software system that connects with existing systems, possibly in the cloud, according to the designed architecture using existing frameworks.
- **S3.2** Applying test automation when performing tests.

Manage & Control

- **S1.1** Setting up and using a management system to support software development in a team context.
- **S2.1** Managing and using a development pipeline to support software development in teams, enabling possibilities such as continuous integration.
- **S2.2** Applying methods and techniques to manage a software development process and ensure its quality.
- **S3.1** Performing configuration, change, and release management in coordination with infrastructure management.
- **S3.2** Setting up a development pipeline with automated build and test infrastructure.