Assignment #8

1. Computing KPIs for the projects

a. On budget

To compute the budget KPI, we have calculated the deviation between the Actual money that has been spent minus the Estimated required budget in columns **Budget = column L** and **Actual Cost = column M**.
For analyzing if the target has been overpassed, we calculated the ratio percentages between the "Deviation" previously calculated divided per the "Budget estimated" as in the following formula: $\frac{Budget\ Deviation}{Budget}$

As a final step, we compute the average of the estimations that reach the objectives of **Budget Deviation (%)** less than 5% and 10%. Because the resulting percentage of projects that lay between our target is 95,98%, by looking at the Tables sheet, we conclude that the **Budget achievement rate** is 130%.

b. On time

To compute the On time planned KPI, we have calculated the *average* of Development and Production **ratios**, which are firstly calculated following the steps:

- For the **Development** ratio, we compute the difference between the **Development Actual Date** and the estimated **Development Planned date**.
  Afterwards, the result is compared in a ratio by dividing the calculated days resulting in column **Plan Deviation (P)** against the difference between the **Start of development** → *Actual date* and the **Development** → *Planned date*.

- $Plan\ Deviation\ (P) = Development\ Actual\ date\ -\ Development\ Planned\ date$
- $Deviated\ starting\ point = Start\ of\ development\ (actual\ date)\ -\ Development\ Planned\ date$
- $Development\ Ratio = \frac{Plan\ Deviation\ (P)}{Deviated\ starting\ point}$

As a final step, we compute the average of the estimations that reach the objectives of **Plan Deviation (%)** less than 10% and 20%. Because the resulting percentage of projects that lay between our target is 89,85%, by looking at the Tables sheet, we conclude that the **Plan achievement rate (D)** is 110%.

- For the **Production** ratio, we compute the difference between the **Production Actual Date** and the estimated **Production Planned date**. Afterwards, the result is compared in a ratio by dividing the calculated days resulting in column **Plan Deviation (P)** against the difference between the columns **Project Requirements Specification** → *Planned Date* and the **Production** → *Planned Date*.

  - $Plan\ Deviation\ (P)\ =\ Production\ Actual\ date\ -\ Production\ Planned\ date$
  - $Deviation\ to\ Requirements\ =\ Production\ Planned\ date\ -\ Project\ Requirements\ Specification\ Planned\ date$
  - $Production\ Ratio\ =\ \frac{Plan\ Deviation\ (P)}{Deviation\ to\ Requirements}$

  As a final step, we compute the average of the estimations that reach the objectives of **Plan Deviation (%)** less than 10% and 20%. Because the resulting percentage of projects that lay between our target is 89,45%, by looking at the Tables sheet, we conclude that the **Plan achievement rate (P)** is 110%.

- Finally, we repeat the formulas logic to in reference to the Plan achievement rate for the column Plan Deviation (average D, P), which also results in 110%, reasonably because the previous ratios lead both to 110%

c. On error

To compute the On error planned quality KPI, we have calculated the difference between the number of errors and the amount of cases that have been tested. The average of ratios gives us a KPI achievement of 19,37%. Analyzing whether this is a good ratio or not, we have established an objective target of 20% of the average of the projects, which is a good result because the projects are lightly behind the target ratio.

$$Quality\ =\frac{Errors}{Test\ Cases}$$

d. Conclusions

On Budget

- According to the provided KPI achievement table, this performance corresponds to a **Budget Achievement Rate of 130%**, reflecting excellent financial management.

On Time

- The **On Time KPI** considers both the **Development** and **Production Ratios**:
  1. **Development Ratio**: Measures deviations in development completion dates relative to the planned dates.
  2. **Production Ratio**: Evaluates production completion against planned schedules.
- Both ratios achieved a performance of **89.85%** and **89.45% (>=85%)**, respectively, for projects meeting the **< 20% deviation target**, resulting in an overall **Plan**

Achievement Rate of 110%. This performance is satisfactory but indicates potential areas for improvement in schedule adherence.

On Error

- The calculated error rate is **19.37%**, which is slightly below the target of **20%**. This indicates good software quality, with range for minor improvements in reducing errors.

---

2. Relevant KPI metrics

a. Delivery Rate

How to measure it?

Delivery Rate's metric is used to optimize the Lead Time of work items by limiting the Work in Progress, as stated in the Little's Law.

$$\overline{Delivery\ Rate} = \frac{\overline{WiP}}{\overline{Lead\ Time}}$$

This flow KPI can be adjusted to achieve the target by the following strategies:

Reduce WIP: Limiting the number of work items in progress can reduce delays and improve flow.

Optimize Lead Time: Streamlining processes, removing inefficiencies, and improving collaboration can reduce lead time.

Adjust Workflow: Continuous improvement and regularly reviewing bottlenecks in the Kanban system can help increase the Delivery Rate.

Why is relevant?

Using Delivery Rate as a KPI with Little's Law helps the team:

1. Measure the efficiency of their workflow by understanding how WIP and lead time affect delivery speed.
2. Track progress by focusing on how quickly items are being completed.

3. Optimize delivery rate by balancing WIP and lead time.

By tracking the Delivery Rate, teams can identify potential bottlenecks, understand their capacity, and adjust processes to improve flow and speed, leading to faster delivery and higher productivity. A **higher Delivery Rate** suggests that the team is completing more tasks within a given time period, which is generally a positive sign of productivity and efficiency. A **lower Delivery Rate** might indicate bottlenecks, too much WIP, or delays in lead time, requiring investigation and optimization (e.g., reducing WIP, improving team processes, or addressing delays in workflow stages).

Comparing the **actual Delivery Rate** to **targeted rates** helps in measuring whether the system is functioning at its best level.

b. Effort and Cost per Function

How to measure it?

Effort and Cost per Function Point measures the efficiency of project delivery by calculating the effort and cost required to implement a unit of functionality (Function Point).

This KPI is calculated as:

- **Effort per Function Point (ph/PF)**: Total person-hours (ph) divided by the total Function Points (PF).
- **Cost per Function Point (EUR/PF)**: Total cost in euros (EUR) divided by the total Function Points (PF).

The calculation involves:

1. Using **Estimate Sheets** that document:
    - Effort in person-hours (ph).
    - Cost in euros (EUR).
    - Project size in Function Points (PF).
2. Taking a sample of projects and computing the **simple average** for both ratios (ph/PF, and EUR/PF).

Function Points (FP) are a measure used to estimate the size,
complexity, and functionality of a software application.
To calculate Function Points, the Function Point Analysis (FPA)
method is typically used, which involves the following steps:
- Identify Function Types
- Determine Complexity Weight for Each Component

- Calculate the Unadjusted Function Points (UFP)
- Determine the Value Adjustment Factor (VAF)- Calculate the Adjusted Function Points (AFP) = UFP x VAF

Why is it relevant?

Effort and Cost per Function Point is a key productivity KPI, enabling teams and organizations to:

1. **Measure and Improve Efficiency**:
    - Tracks how efficiently resources (effort and budget) are being used to deliver functionality.
2. **Enable Comparisons**:
    - Provides a consistent unit (Function Point) for comparing productivity across projects and teams.
3. **Monitor Progress**:
    - Shows improvements over time by comparing actual results to annual targets.

4. **Support Budgeting and Estimation**:
    - Helps refine estimates for future projects based on historical data.

A lower Effort and Cost per Function Point indicates higher productivity and cost-effectiveness, while higher values suggest areas where processes or resource allocation could be optimized. Regular tracking of this KPI drives continuous improvement and ensures alignment with annual improvement goals.

c. Incidents metrics

How to measure it?

Incidents reported are counted and **classified** depending on the gravity and the impact on the user. It can be classified into occurrences, service failures and incidents reported by users, and is **tracked** with monthly dashboards that visually show the status of the failures and are used to set the target **objectives** on improving your results compared to the previous period of time.

The service provided for solving the incident can be measured as a KPI by computing the average of **response times** for each failure reported, meaning the time spent in solving it between the moment the incident is reported and the time it is closed. The objectives are set following the Service Level Agreement **standards**: customer and company agree on for example how many days will the company take to solve the incident, and the goal KPI is to achieve them for example at least in 85% of the cases.

Why is relevant?

For a company it is relevant to track the **quality** of the products and services that it provides, and there are two approaches that *evaluate* so: during testing and with incidents.

The standard error metrics are computed in the testing phase before the product can be delivered, tracking the ratio of failed tested cases among the tests. But after the product is delivered the errors can still occur and they **impact the** *prestige* of the company more, because the product was supposed to be tested and delivered with the best *quality*. These types of errors are handled in the form of incidents and *did not* lay on the *planned* life cycle of developing a *project*, but need to be solved in the form of responses.

Therefore if we had to choose relevant KPIs to *represent* our company, we would show the incidents KPIs because measures the failed cases that the customer experiences after it is delivered and deployed and if this ratio is good, means that the user is going to be **satisfied**: better than having to invest into customer support, is to deliver with so much *quality that does not need to have support*.

d. Conclusions

- The **Delivery Rate KPI** provides critical insight into the efficiency and effectiveness of your Kanban system. By tracking this metric, you can continuously monitor and optimize your workflow, leading to faster delivery and improved team performance. The goal is to maintain a consistent and high delivery rate while managing WIP and lead time effectively.
- **Effort and Cost per Function Point** focuses on productivity and cost-efficiency, highlighting how effectively resources are utilized to deliver functionalities. By tracking this KPI, organizations can optimize resource allocation, control budgets, and improve estimation accuracy for future projects. Achieving lower effort and cost ratios per Function Point indicates streamlined workflows and higher overall efficiency.
- Failures directly impact customer satisfaction and reflect the quality of what has been delivered. Focusing on **incident KPIs** allows companies to minimize the need for customer support, ensuring a higher level of product quality and a better user experience.