Lean Six Sigma in Construction Projects with Python

Lean Philosophy

The Toyota Production System pioneered the Lean philosophy in the automobile industry and has since been adopted and adapted across various sectors, including construction. At its core, Lean is about maximizing value and minimizing waste.

Lean provides a framework for optimizing the entire process in project delivery, not just isolated activities. As defined by the Lean Construction Institute, Lean Project Delivery is a holistic approach that seeks to manage and improve project design and delivery, leading to improved efficiency and higher-quality outcomes.

The Lean Six Sigma comes into play as it presents a powerful approach to project delivery. The primary purpose of Lean is to eliminate waste and ensure smooth workflow, while Six Sigma focuses on reducing variation and defects in the processes. When used together, these two methodologies provide a comprehensive toolset to increase efficiency, reduce waste, and ensure high-quality outcomes.

Let's start by implementing Lean principles to minimize waste in project delivery. The seven types of waste (or Muda in Lean terminology) in project delivery could be overproduction, inventory, transportation, over-processing, waiting, unnecessary motion, and defects. Identifying and eliminating these waste types will help improve project delivery efficiency.

Let's use Python to plot a bar chart to visualize the waste types in project delivery. We first need to import the necessary libraries and then create our data.

In [1]:
```
import matplotlib.pyplot as plt

waste_types = ['Overproduction', 'Inventory', 'Transportation', 'Over-processing', 'Waiting', 'Unnecessary Motion', 'Defects']
waste_values = [20, 15, 30, 10, 50, 25, 40]  # hypothetical values representing the amount of waste

plt.figure(figsize=(10, 6))
plt.barh(waste_types, waste_values, color='skyblue')
plt.xlabel('Amount of Waste')
plt.title('Types of Waste in Project Delivery')
plt.show()
```

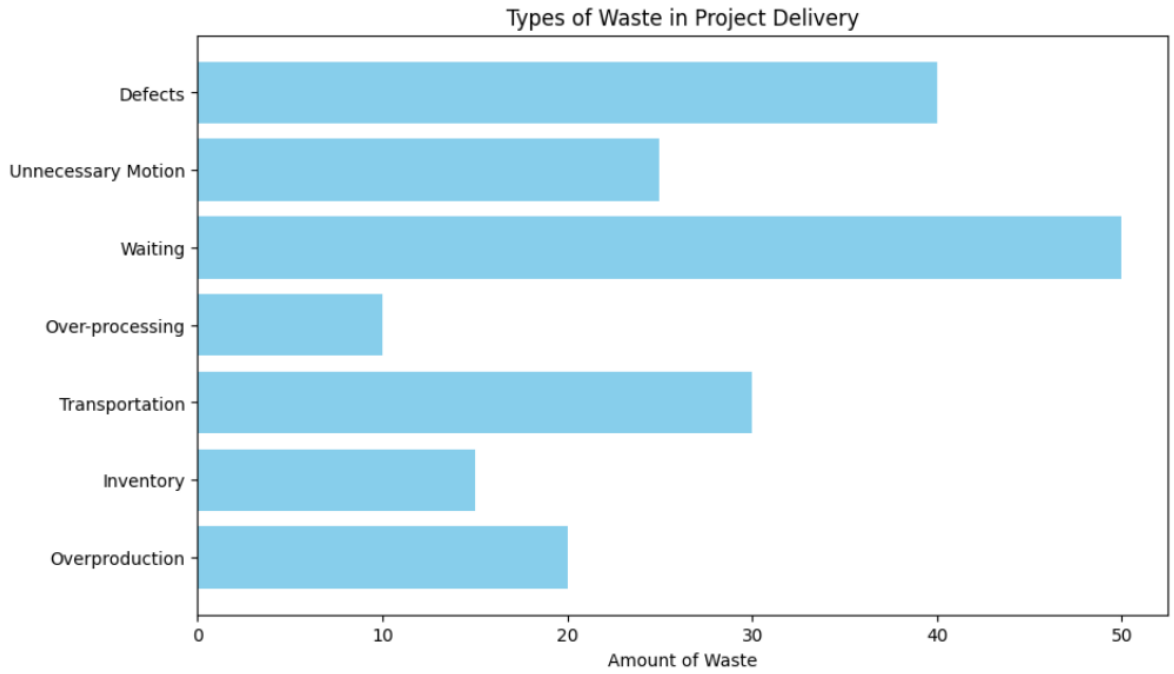

Figure 1: Sources of Waste with hypothetical values

In Figure 1, we've created a horizontal bar chart that showcases different types of waste and their hypothetical quantities in a project delivery process. Identifying these waste types will aid in determining where Lean efforts should be targeted.

With the waste identified, let's implement Six Sigma to minimize variability and improve quality. We can utilize Six Sigma's DMAIC (Define, Measure, Analyze, Improve, Control) methodology in our project delivery process.

To visualize the DMAIC process, we can create a Python pie chart:

In [2]:
```
process_stages = ['Define', 'Measure', 'Analyze', 'Improve', 'Control']
stage_values = [20, 20, 20, 20, 20]  # Equal distribution across all stages

plt.figure(figsize=(7, 7))
plt.pie(stage_values, labels=process_stages, autopct='%1.1f%%', startangle=140)
plt.title('DMAIC Process Distribution in Project Delivery')
plt.show()
```

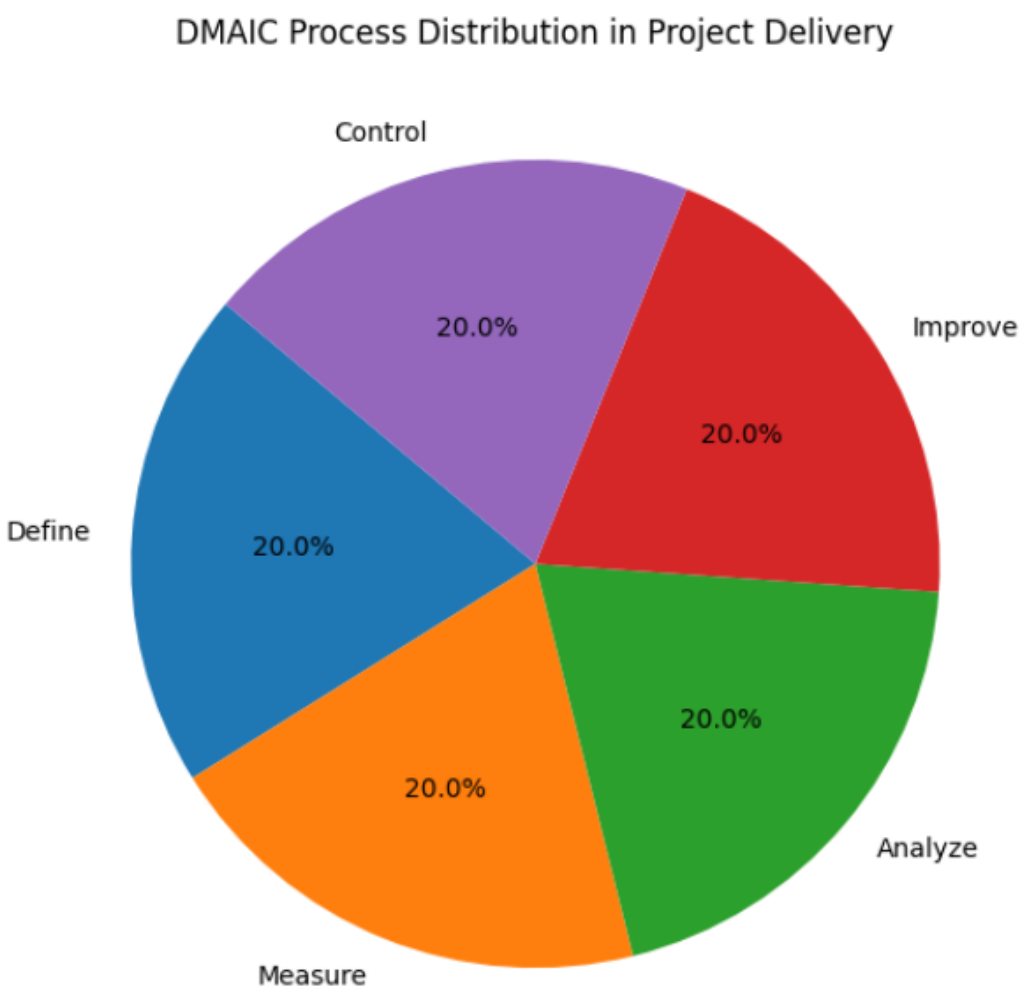

Figure 2: Illustration DMAIC process

In Figure 2, we show an equal distribution of effort across all stages of the DMAIC process. Figure 2 illustrates the importance of each stage in the Six Sigma process.

Now, let's consider applying Lean principles to enable the flow of value. It is often done by mapping the value stream - the organization's activities to deliver on a customer request.

We can use the **networkx** library in Python to create a directed graph representing a value stream map. Here's an example of how we might do this.

:

In [3]:
```python
import networkx as nx

G = nx.DiGraph()  # Create a new directed graph G

edges = [
    ("Customer Request", "Define Scope"),
    ("Define Scope", "Design"),
    ("Design", "Development"),
    ("Development", "Testing"),
    ("Testing", "Deployment"),
    ("Deployment", "Customer Delivery"),
]

G.add_edges_from(edges)  # Add edges to the graph

pos = nx.spring_layout(G)  # Position nodes using Fruchterman-Reingold force-directed algorithm
nx.draw(G, pos, with_labels=True, node_color='skyblue', node_size=1500, arrowstyle='->', arrowsize=20, font_size=8)
plt.show()
```

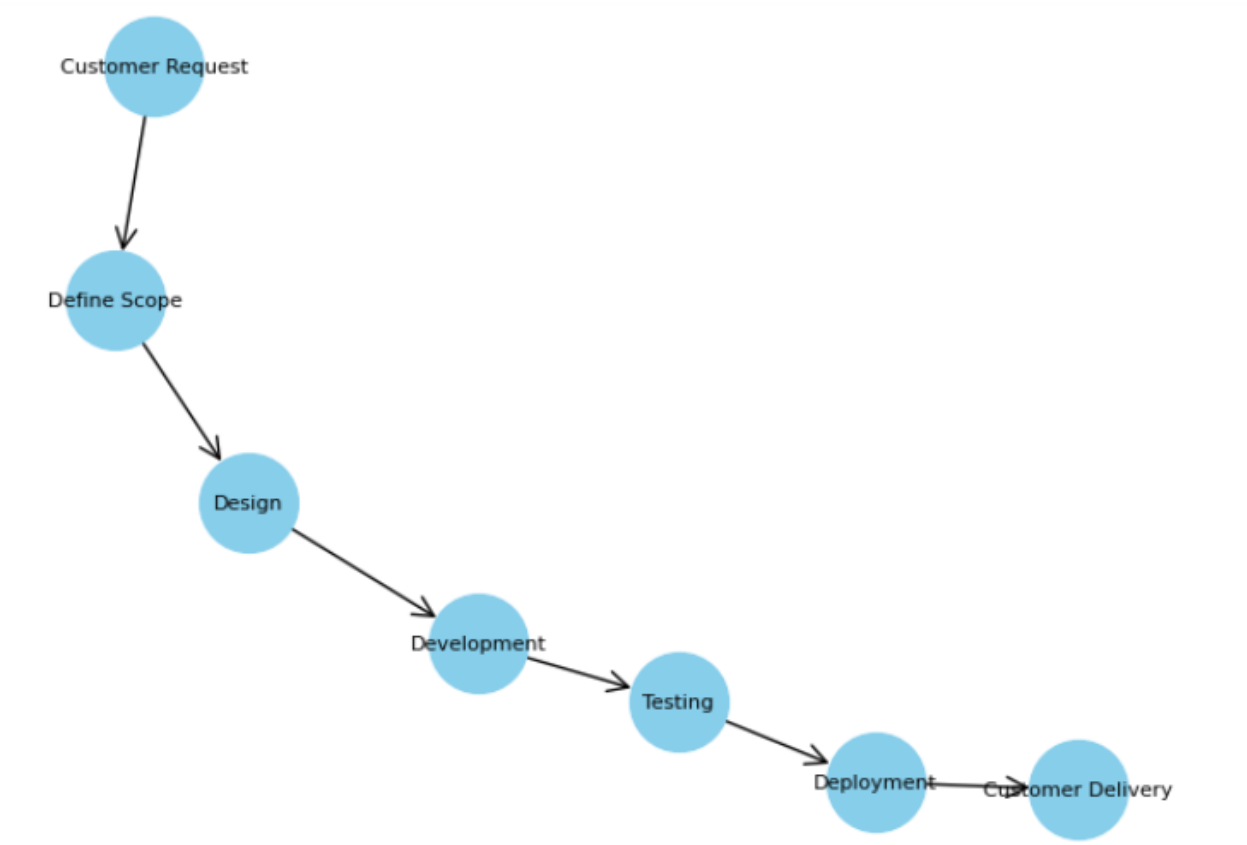

Figure 3: Value Stream map

Figure 3 is a directed graph representing the value stream. Note that Lean is a journey, and it is not a destination. Lean is a continuous improvement process that demands a culture of respect, transparency, and relentless focus on value to the customer. The focus is on eliminating waste and continuously improving the process. Lean is all about 'respect for people' and 'continuous improvement.' It emphasizes delivering value to the customer, eliminating waste, and continuously improving processes.

Let's now discuss the Theory of Lean. The Theory of Lean is based on five principles: Value, Value Stream, Flow, Pull, and Perfection. Here's a brief on each:

1. **Value:** It is about understanding what the customer values in a project, which means you should determine the customer's needs, and the project should meet these needs.
2. **Value Stream:** Identifying the steps in the process that add value and those that do not. The goal is to eliminate the steps that do not add value.
3. **Flow:** After the value and value stream has been identified, the next step is to make the value-added steps flow without interruption.
4. **Pull:** This means that products are only supplied when there is a demand. This reduces overproduction and inventory.
5. **Perfection:** Lean is about continuous improvement and seeking perfection by eliminating waste.

Let's take an example of a hypothetical construction project to demonstrate these principles in action. This will involve tracking the construction activities and their durations, identifying the activities that add value and those that don't (waste), and then using Python to demonstrate this information visually.

In [4]:
```
import matplotlib.pyplot as plt

# Define activities and their durations
activities = ["Design", "Permits & Approvals", "Site Work", "Foundations", "Superstructure", "Exterior Cladding", "Interiors", "I
durations = [60, 30, 40, 30, 80, 50, 60, 10]

# Plot bar chart
plt.bar(activities, durations, color='blue')

# Label the axes
plt.xlabel('Activities')
plt.ylabel('Duration (Days)')

# Rotate X-axis labels for better readability
plt.xticks(rotation='vertical')

# Show the plot
plt.show()
```

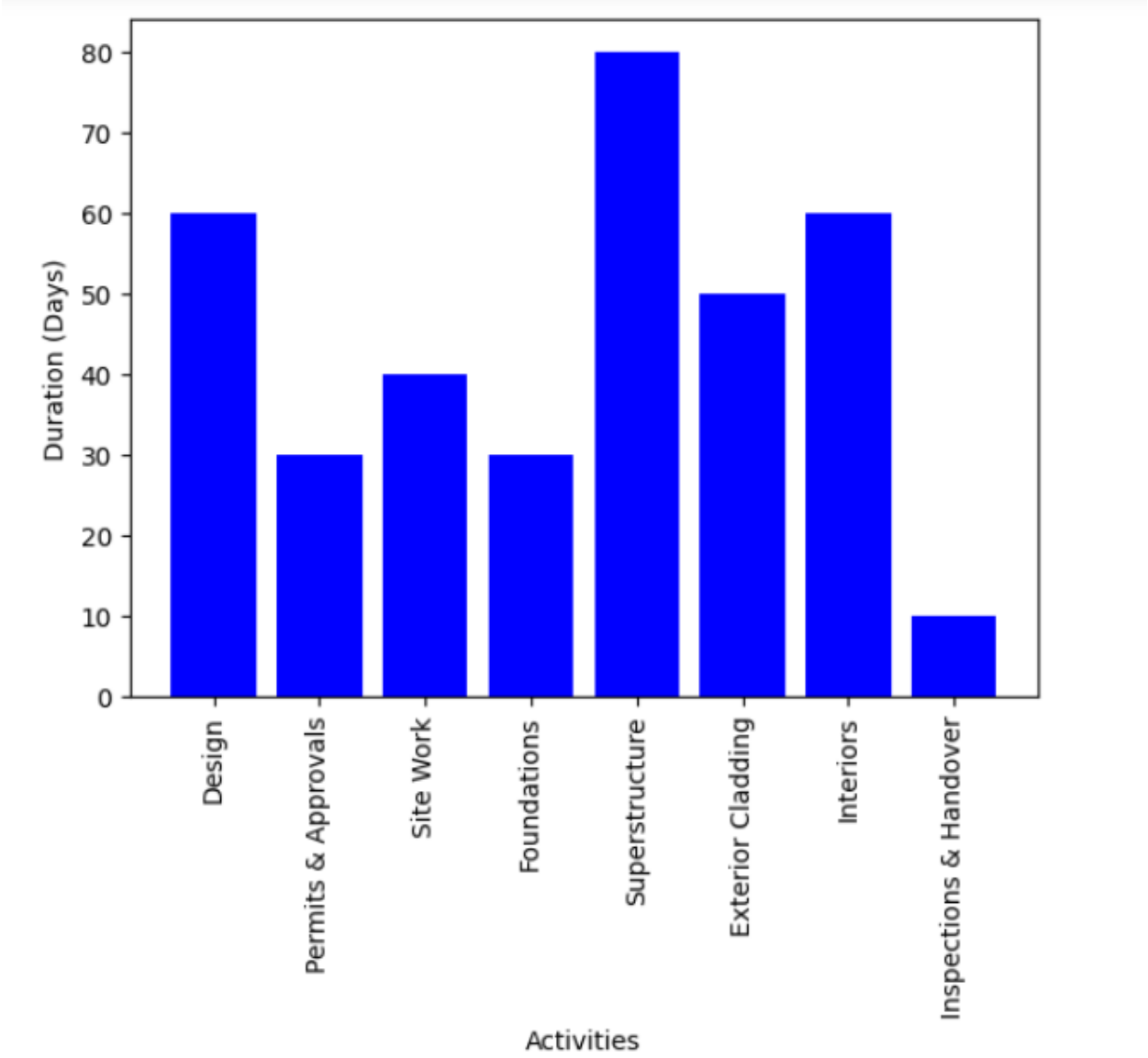

Figure 4: Construction activities and durations.

In Figure 4, we produced a simple bar chart, showing the duration of each activity in a construction project. Here, we haven't differentiated between value-adding activities and non-value-adding ones.

Now, let's use color coding to identify value-adding and non-value-adding activities, typically known as 'waste' in lean methodology. This visualization can help us understand how much of our project timeline is taken up by activities that do not add direct value to the product. For simplicity, let's consider 'Permits & Approvals' and 'Inspections & Handover' as non-value-adding activities.

In [6]:

```
# Define activities and their durations
activities = ["Design", "Permits & Approvals", "Site Work", "Foundations", "Superstructure", "Exterior Cladding", "Interiors", "I
durations = [60, 30, 40, 30, 80, 50, 60, 10]

# Identify non-value adding activities
non_value_activities = ["Permits & Approvals", "Inspections & Handover"]

# Assign colors based on the type of the activity
colors = ['blue' if activity not in non_value_activities else 'red' for activity in activities]

# Plot bar chart
plt.bar(activities, durations, color=colors)

# Continue from the previous code snippet

plt.xlabel('Activities')
plt.ylabel('Duration (Days)')

# Create a legend
import matplotlib.patches as mpatches
blue_patch = mpatches.Patch(color='blue', label='Value Adding Activities')
red_patch = mpatches.Patch(color='red', label='Non-value Adding Activities')
plt.legend(handles=[blue_patch, red_patch])

# Rotate X-axis labels for better readability
plt.xticks(rotation='vertical')

# Show the plot
plt.show()
```

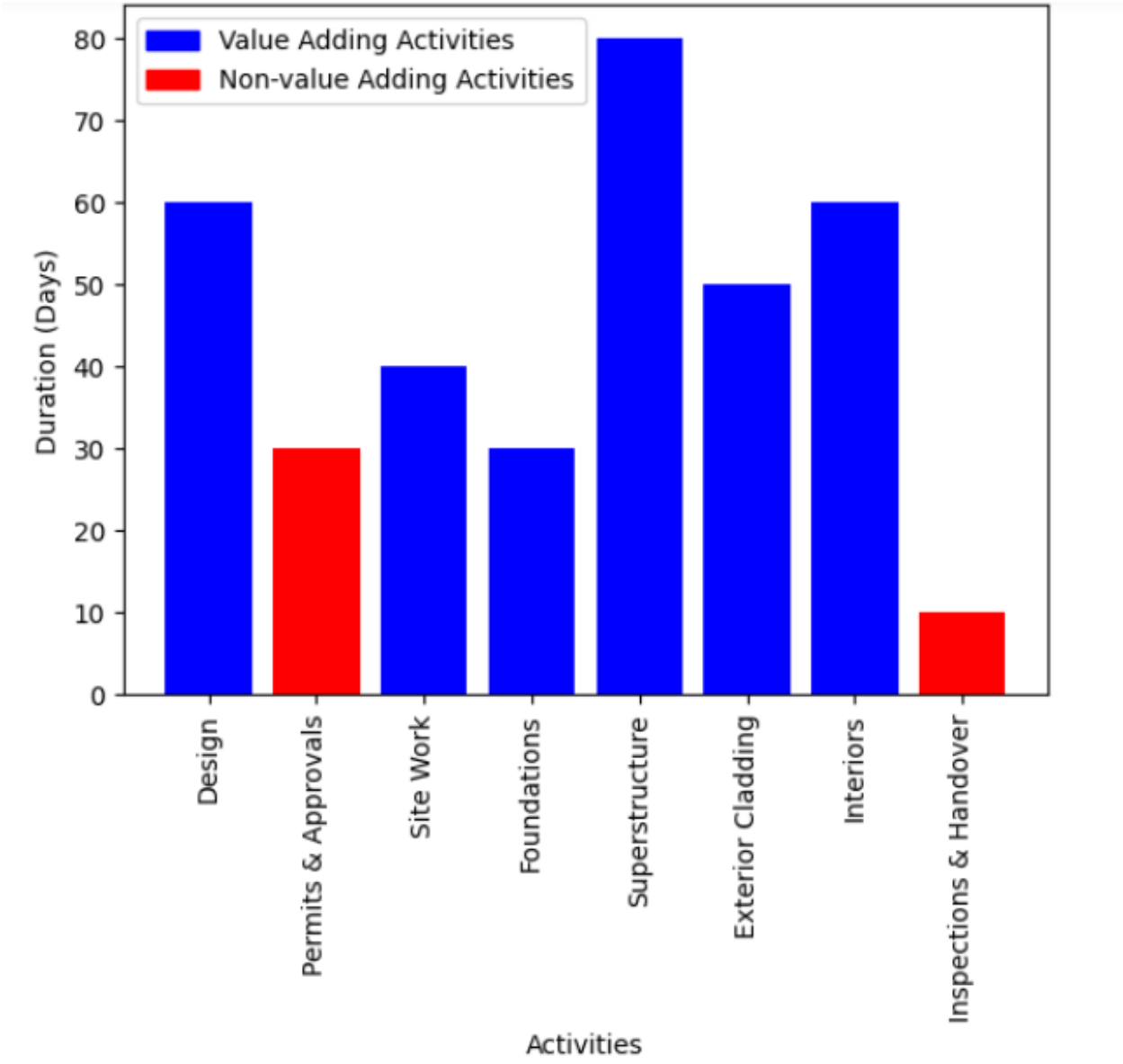

Figure 5: Identification of non-value adding activities.

In Figure 5, the red bars represent non-value-adding activities, while the blue bars represent value-adding activities. With this visualization, project managers can better understand where time is spent and identify opportunities for process improvement to reduce waste.

It's important to note that the classification of activities into 'value-adding' and 'non-value-adding' can be subjective and may vary based on the specific circumstances of a project.

Also, keep in mind that in practice, not all non-value-adding activities can be eliminated (like "Permits & Approvals" and "Inspections & Handover" in this example), as they might be essential for legal or safety reasons. However, the Lean methodology encourages us to minimize non-value-adding activities and continuously improve efficiency.

The above example is a starting point for more sophisticated data analysis and visualization efforts as you delve deeper into your Lean journey.

References

Six Sigma Handbook by Thomas and Paul Keller

This is Lean: Resolving the Efficiency Paradox by Modig, Niklas, Ahlstrom, Par (2012) Paperback

Business Improvement- Operational Excellence with Lean Six Sigma and Project Management Principles and Practices, by Francis Dakubo (October 2023, www.Amazon .com)