

The Systemic Event Discovery Approach Workshop - SEDA

BurgerBest is a fast-food restaurant, located in downtown California, with a traditional ordering system. Customers can place orders by phone, in person at the shop, or online. However, cancellations and delivery services are not available.

System Characteristics

- Monolithic Architecture
 - I. A single, tightly coupled application that handles all functionalities, including the website, order processing, payment handling, and reporting.
 - II. Runs as a single executable on a local server inside the shop.
- On-Premise Deployment
 - I. Hosted entirely on a local server located inside the shop.
 - II. No cloud services, meaning no external hosting or scalability features.

Key Features

- ◆ In-Shop Order Processing
- ◆ Online Ordering (No Delivery)
- ◆ Basic Payment Handling
- ◆ Manual Order Reporting

Limitations

Not Scalable: Adding new stores or remote access would require significant effort since everything is tied to a single local system.

No Modern UI/UX: The website is simple, non-responsive, and lacks interactive features.

Limited Access: No mobile app, and online ordering is restricted to customers within the shop's network range due to local hosting.

No Automation: Inventory and payment handling require manual intervention.

High Maintenance: The system relies on an old-school on-premise setup, requiring manual updates, backups, and potential hardware failures.

Your Task

We aim to transform and modernize the ordering system for **BurgerBest**. Currently, as mentioned, customers have two options for placing orders:

1. In-shop
2. Online (No Delivery, Just pickup)

The new ordering system must be designed to accommodate modern payment methods, including credit cards and other solutions. Customers must have the option to receive their orders at their doorstep, free of charge.

Step 1: Understanding the Current State of a System Through Holistic Analysis

- Analyze the **events**, **patterns**, **system architecture**, and **mental models** shaping the current ecosystem. Explore Conway's Law to identify potential inefficiencies caused by excessive handoffs.

Step 2: Redesign Based on Domains and Implement a Bottom-Up Approach

- Identify key domain events and bounded contexts through an EventStorming workshop. Pinpoint core domains as strategic leverage points during the session.
- Define Team Topologies based on bounded contexts and system boundaries.

Step 3: Definition and Adoption of System Dynamics

This step consists of two key parts:

1. Identifying the system's emergent behavior via Thinking in System
2. Defining feedback loops (Negative or Positive) in relation to these emergent behaviors.