Case Study: Pizza Delivered Quickly

It has been decided that the first few cycles of the Pizza Delivered Quickly (PDQ) APF project will build an iconic prototype of the entire system. The project team includes the PDQ supervisors and their key staff as well as the development team. The development team includes Pepe and his staff and a contractor team which will do the systems development. Pepe, the co-manager of the client team, and the manager of the contractor's development team thought this was the best approach to the entire project for several reasons.

- PDQ supervisors and staff were not technically savvy and probably could not relate to how technology could play an integral role in the new business model. They needed to come to a basic understanding of where technology could improve their way of doing business.
- Whatever form the solution took, it was not at all obvious to the development team what the underlying business rules should or could be. More structure was needed.
- The new business model would have a lot of moving parts, and how those parts interacted operationally was not at all obvious.

Note

This may appear to be a pedestrian approach, but consider the reasoning. There is no reason to believe that the client group has any grasp of the business process or technical complexity of the as-yet undefined solution. PDQ is a family-owned business that has not leveraged technology to any meaningful degree. For all intents, PDQ is technically challenged. For any APF project to have a chance at success, everyone must be on the same page. A prototype serves that purpose in this case. Not only will it be useful in getting everyone on the same page, but it will also give the project team some insight into the requirements of the solution and the underlying business rules.

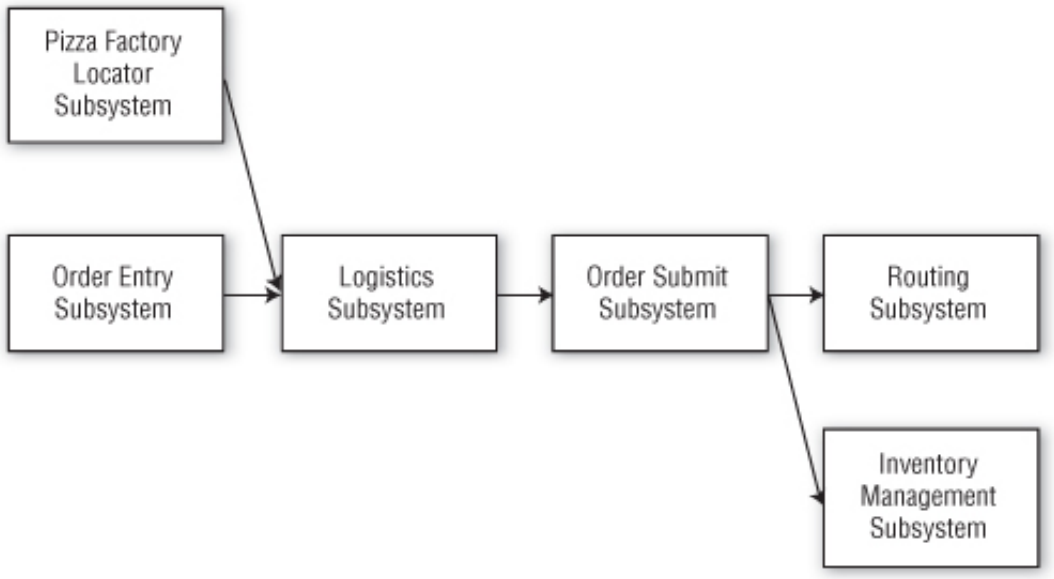
The solution will be very complex and consist of a number of dependent parts. It is hoped that the prototype will help answer several operational questions, such as:

- What factors will be used to evaluate a potential pizza-factory location?
- How will the PDQ market be defined and organized?
- What factors determine the production facility (store, factory, or van) that will be used to fulfill a specific order?
- How will these factors be used to make the decision as to which production facility to use?
- What factors determine which pizza van will deliver the order?
- How will these factors be used to make the decision as to which pizza van will deliver the order?

- What performance metrics can be used to measure solution effectiveness?

Figure 3.9 shows a high-level process flow of the six subsystems that make up the solution.

Figure 3.9. PDQ Subsystem Process Flow



The prototype proved to be valuable in helping the client gain an understanding of what the solution might include. Following that exercise, the RBS was generated. It follows. Based on what you now know about the solution, you will next map out your strategy for approaching the cycles for this systems-development effort. This same strategy applies to any APF project.

Subsystem #1: Pizza Factory Locator

This subsystem takes into account the current PDQ production and delivery facilities and primary, secondary, and tertiary markets to locate new sites for pizza factories. The model must take into account the impact of a new site on the success criteria.

Function 1.1: Aggregate sales data and order entry to order fulfillment time by census block

Function 1.2: Calculate market penetration by census block

Function 1.3: Analyze market area by census block

Function 1.4: Identify census blocks that are underserved

Function 1.5: Identify potential pizza-factory sites

Function 1.6: Analyze impact on success criteria of each potential pizza-factory site

Function 1.7: Update pizza factory locations

Subsystem #2: Order Entry

The customer enters an order into the system. This can be done by phone, Internet, or store visit.

Function 2.1: Identify customer

Feature 2.1.1	Customer history
Feature 2.1.2	Name, address, etc.

Function 2.2: Get order

Feature 2.2.1	Products requested
Feature 2.2.2	Quantity ordered
Feature 2.2.3	Options selected
Feature 2.2.4	Baked or unbaked
Feature 2.2.5	Update order history

Function 2.3: Get delivery instructions

Feature 2.3.1	Delivery location
Feature 2.3.2	Delivery options (pick-up, delivery, and so forth)
Feature 2.3.3	Delivery time requested

Function 2.4: Price order

Feature 2.4.1	Apply promotions
Feature 2.4.2	Calculate price
Feature 2.4.3	Maintain pricing table

Function 2.5: Confirm order

Feature 2.5.1	Payment type
Feature 2.5.2	Display order
Feature 2.5.3	Accept, cancel, or modify

Function 2.6: Submit order details to Logistics Subsystem

Feature 2.6.1	Submit order
Feature 2.6.2	Confirm order acceptance
Feature 2.6.3	*Unknown*

Subsystem #3: Logistics

This subsystem continuously monitors workloads across all preparation locations and all delivery alternatives. Based on current workloads, the system decides where to prepare the pizza and which van will deliver the order.

Function 3.1: Choose prep location

Feature 3.1.1	Get prep-location workload
Feature 3.1.2	Get next-order data
Feature 3.1.3	Estimate production and delivery time from each prep location
Feature 3.1.4	Assign prep location (business rule not clear)

Function 3.2: Transmit order to prep location

Feature 3.2.1	Submit order to prep location
Feature 3.2.2	Update prep-location workload

Function 3.3: Choose delivery van

Feature 3.3.1	Estimate location of each delivery van when order is ready to be delivered
Feature 3.3.2	Estimate pick-up and delivery time for each van to each location
Feature 3.3.3	Choose delivery van (business rule not clear)

Subsystem #4: Order Submission

Based on output from the Logistics Subsystem, the prep location receives the order for scheduling.

Function 4.1: Receive order detail from Logistics Subsystem

Feature 4.1.2	Retrieve order detail

Function 4.2: Schedule order for prep

Feature 4.2.1	Prioritize new order
Feature 4.2.2	Get into prep queue

Function 4.3: Check order status

Feature 4.3.1	When will order be ready?
Feature 4.3.2	When will order be delivered?
Feature 4.3.3	Notify delivery van driver of order pick-up time
Feature 4.3.4	Notify delivery van driver that order is complete
Feature 4.3.5	Update Inventory Management Subsystem

Function 4.4: Schedule order pick-up and delivery

Feature 4.4.1	Delivery-van driver confirms pick-up time
Feature 4.4.2	Get route from Routing Subsystem
Feature 4.4.3	Confirm order delivery
Feature 4.4.4	Confirm payment receipt
Feature 4.4.5	Update delivery-van location and time

Subsystem #5: Routing

This subsystem gives the driver the route from the prep location to the customer location. There should be a COTS application for this subsystem.

Function 5.1: Upload delivery-van location

Feature 5.1.1	*Unknown*

Function 5.2: Upload customer delivery address

Feature 5.2.1	*Unknown*

Function 5.3: Download driving directions

Feature 5.3.1	*Unknown*

Function 5.4: Confirm order delivery

Feature 5.4.1	*Unknown*

Subsystem #6: Inventory Management

This subsystem monitors real-time inventory levels at all preparation locations, automatically issues replenishment orders to the trucks to replenish preparation-location inventories, and automatically re-orders inventory from the vendor.

Function 6.1: Get inventory-use data

Feature 6.1.1	Collect inventory-depletion data by prep location
Feature 6.1.2	Update on-hand inventory by product

Function 6.2: Identify reorder needs

Feature 6.2.1	Send reorder automatically to vendor
Feature 6.2.2	Inform each prep location of reorder quantity and date for each product

Function 6.3: Receive reorder

Feature 6.3.1	Inspect and accept reorder shipment
Feature 6.3.2	Package orders for each prep location
Feature 6.3.3	Ship orders to each prep location