Food for Thought: What Restaurants Can Teach Us About Reliability

Alex Hidalgo (@ahidalgosre)
Nobl9
alex-hidalgo.com
Food Service is Very Old
Restaurants in China - 12th Century
Sobrino de Botín - 1725
People learn over time and we can learn from each other.
Alex Hidalgo
Director of Site Reliability Engineering

@ahidalgosre
alex-hidalgo.com
Alex Hidalgo

Former server, bartender, line cook, busser, runner, and restaurant manager

@ahidalgosre

alex-hidalgo.com
A Feast with Friends
What Can Go Wrong?
Wrong Order
Badly Cooked
Incorrect Bill
Good Restaurants are Reliable
Host + Maitre D'
Servers
Runners + Bussers
Bar
Cooks
Dishwashers + Prep
Prep
Complex Systems are Everywhere
Restaurants are complex systems made up of component complex systems.
The Floor

- Primary interaction for most customers
- Takes requests and fulfills them
- Has dependencies on the bar and the kitchen
The Bar

- Primary interaction for some customers
- Secondary interaction for others
- Often required for the floor to operate
- Soft dependency on the kitchen
The Kitchen

- No direct customer interaction
- Most important backing service
- Always required for the floor to operate
- Sometimes required for the bar to operate
How Things Fail at Restaurants
Floor
Bar
Kitchen
Failures on the Floor

- Food delivered late
- Badly cooked food
- Wrong food brought out
- Instructions not followed
Food Delivered Late

Problem:

- Food takes longer than customer requirements

Equivalence:

- Latency
Food Delivered Late

Solution:
- Hire more kitchen staff
- Hire runners
- Seat better number of customers per section
Food Delivered Late

Solution:

- Hire more kitchen staff (*resources*)
- Hire runners (*routing service*)
- Seat better number of customers per section (*load balancing*)
Badly Cooked Food

Problem:
- Food is not acceptable

Equivalence:
- Errors
Badly Cooked Food

Solution:

- Hire more staff
- Simplify recipes
- Remove difficult dishes
Badly Cooked Food

Solution:

- Hire more staff (resources)
- Simplify recipes (refactoring)
- Remove difficult dishes (deprecation)
Wrong Food Brought Out

Equivalence:
- Data correctness

Solution:
- Hire an expediter
Wrong Food Brought Out

Equivalence:
  ◦ Data correctness

Solution:
  ◦ Hire an expediter *(verification)*
Instructions Not Followed

Equivalence:
- Bug

Solution:
- Fix the dish ASAP
Instructions Not Followed

Equivalence:
- Bug

Solution:
- Fix the dish ASAP (quick feedback loops and releases)
Complexity Problems
The Bar vs The Floor

- Requests are ad-hoc
- Ambiguous load requirements
- Latency requirements stricter
- Primary data source
Failures at the Bar

- Drinks made late
- Wrong drink mixed
- Customers greeted/served late
- Customers not consulted correctly
Multiple User Classes

- Servers
- Direct Customers
Multiple Direct Customer Classes

- Beer/Wine
- Simple Mixed Drinks
- Advanced Mixed Drinks
- Non-alcoholic Drinks
Serving Complex Needs
Beer/Wine Drinkers

- Basic imbibers
- Experts
  - What kind of sours do you have?
  - What would you recommend to go with the fish special?
  - QoS is important
Mixed Drinks

- Knowledge of many drink types *(caches + databases)*
- Recipes for custom drinks *(runbooks)*
Non-alcoholic Drinks

- Cannot mess this up *(data correctness + integrity)*
An Incident Every Night
Incident Management

- Preparation
- Established Hierarchy
- Formalized Communications
Mise en Place - “everything in its place”

- Prepare what you need
- Make what you need accessible
- Communicate what you have
Mise en Place - “everything in its place”

- Noun
  - “Is your mise ready?”
- Verb
  - “Mise the mushrooms.”
- State of mind
Incident Preparedness

- “I am in my place.”
Established Hierarchy
Established Hierarchy

- Everyone knows who is in charge
- Everyone knows who is doing what
- Everyone knows what is expected
Formalized Communications
Kitchen Slang

- Is its own language
- Differs slightly from restaurant to restaurant
- But is always coherent
**Some Examples**

<table>
<thead>
<tr>
<th>Yes, chef!</th>
<th>Acknowledged!</th>
</tr>
</thead>
<tbody>
<tr>
<td>$THINGs all day!</td>
<td>In total I need $THINGs!</td>
</tr>
<tr>
<td>Fire $THING!</td>
<td>Do $THING now!</td>
</tr>
<tr>
<td>In the weeds.</td>
<td>Can’t keep up.</td>
</tr>
<tr>
<td>Behind/Hot/Sharp!</td>
<td>DON’T MOVE!</td>
</tr>
</tbody>
</table>
All day!

- Differentiates from working off of tickets
- Tickets describe concrete areas of work that need to be accomplished in order to satisfy the requirements of a single table
That might sound familiar.
Complex Systems are Everywhere
Host + Maitre D’

Bar

Servers

Runners + Bussers

Dishwashers + Prep

Cooks
Lessons Learned:

1. Everyone Needs Reliability
2. Look Everywhere for Help
3. Learn from each Other
Thank you!

Alex Hidalgo
Former server, bartender, line cook, busser, and runner

@ahidalgosre
alex-hidalgo.com

(All images used under license from Shutterstock.com)