Optimized Manufacturing:

A Recipe for Success

Abby Garrett & Caroline Daugherty

May 2023

Introductions

ABBY GARRETT

Lead Data Translator
MBAn from MIT
B.A. in Computer Science &
B.S. in Mathematics

CAROLINE DAUGHERTY

Data Translator
MBAn from MIT
B.A. in Mathematics &
Statistics, B.A. in Economics

Agenda

Problem Statement
- *Get medicines to patients faster by optimizing the manufacturing schedule*

Process Explanation
- *Pharmaceutical manufacturing for the lay man*

Optimization Formulation
- *How does the process translate to an optimization*

Objective Selection
- *Objective iterations based on business needs*

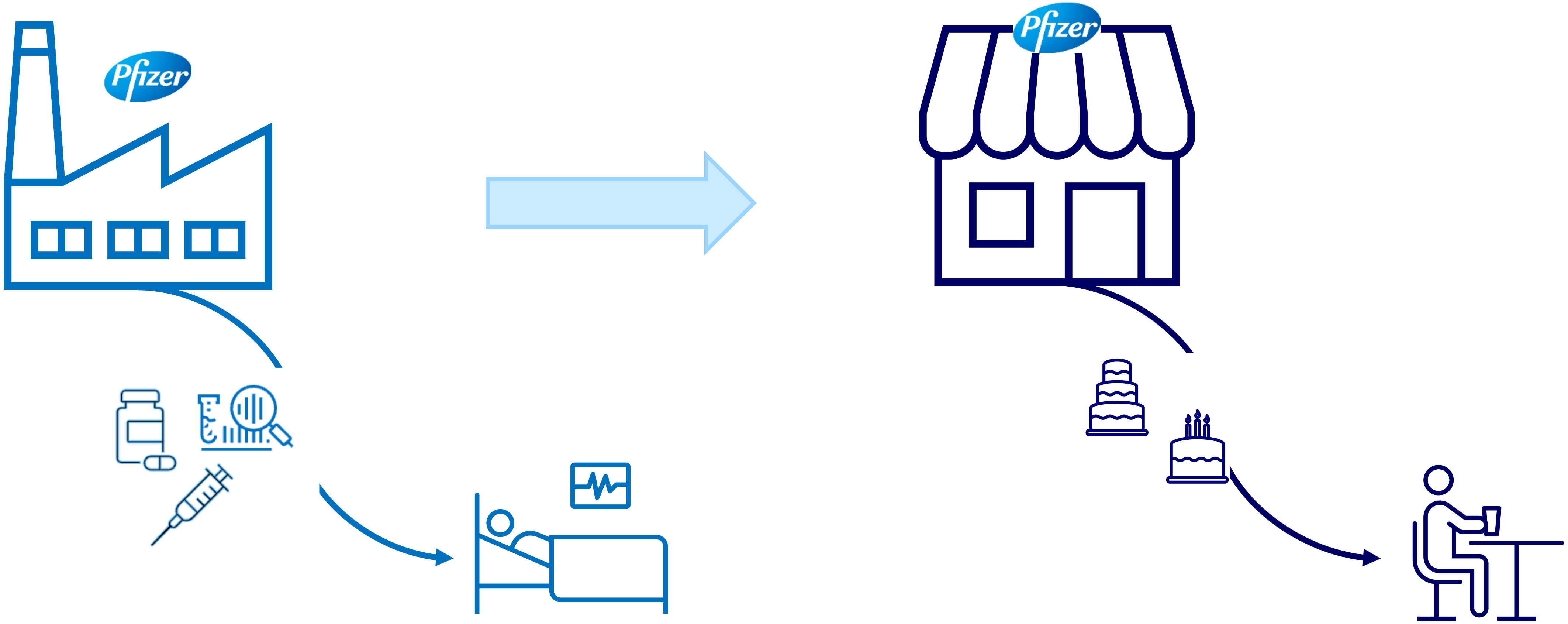
Results
- *Estimated 18% increase in throughput*

Technical Considerations
- *Improving the formulation to make it realistic*

Implementation
- *Change management and stages to implementation*

Problem Statement

Pfizer Global Supply looks to get more medicine to patients faster. The pharmaceutical manufacturing process has parallels to the process of baking a cake.

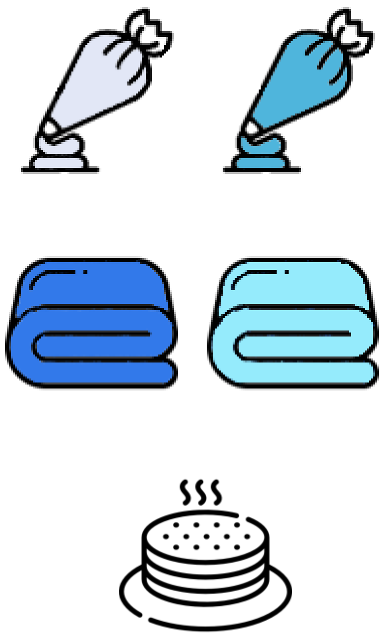


Production Process

01 Ingredients

02 Semi-Finished Components

03 Finished Components

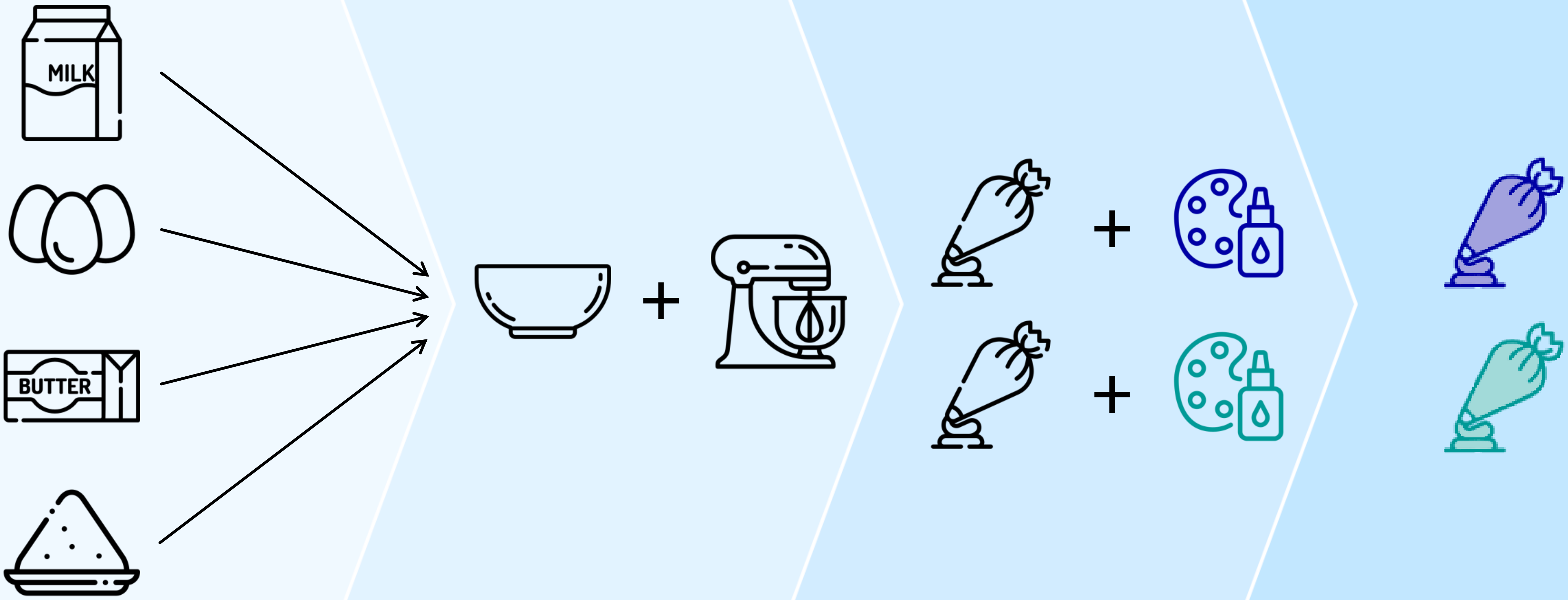


04 Finished Products

Optimization leads to +18%

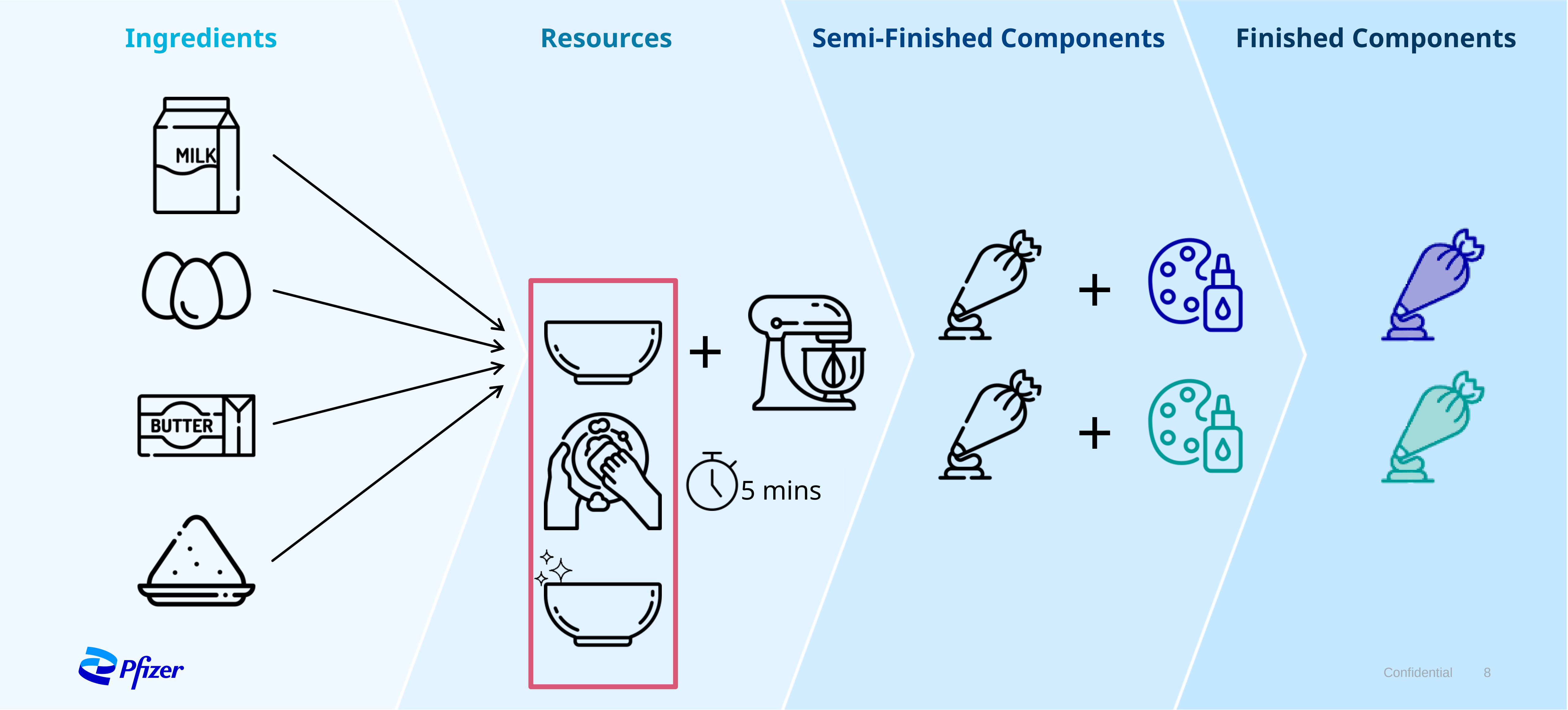
Multi-Step Processes

Ingredients | **Resources** | **Semi-Finished Components** | **Finished Components**



Processing Time

Changeover Time

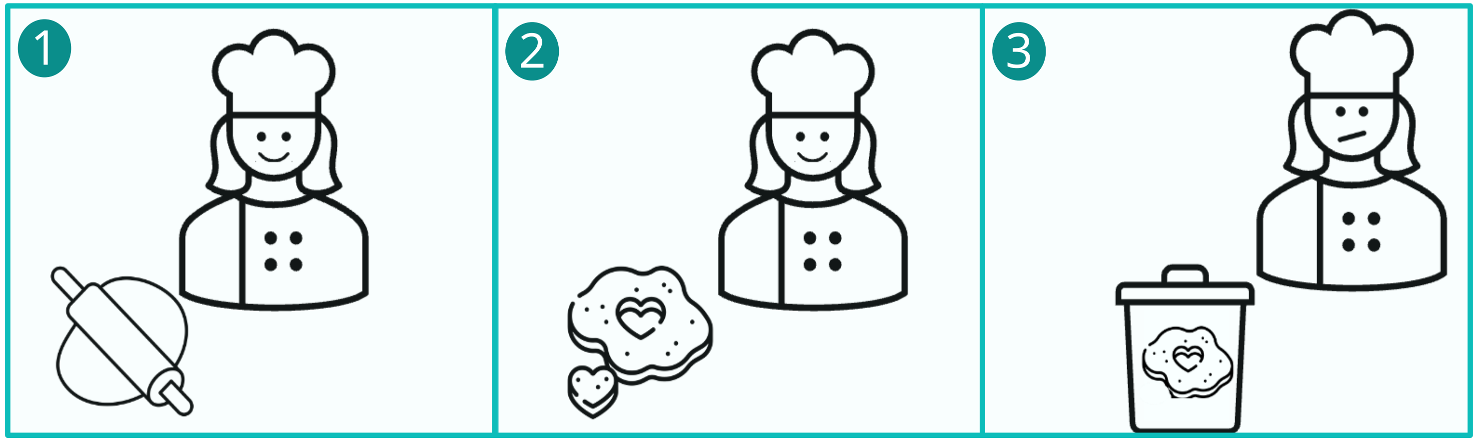


Scrapping

Plans must account for batches to be partially or fully scrapped

Fully scrap

Partially scrap

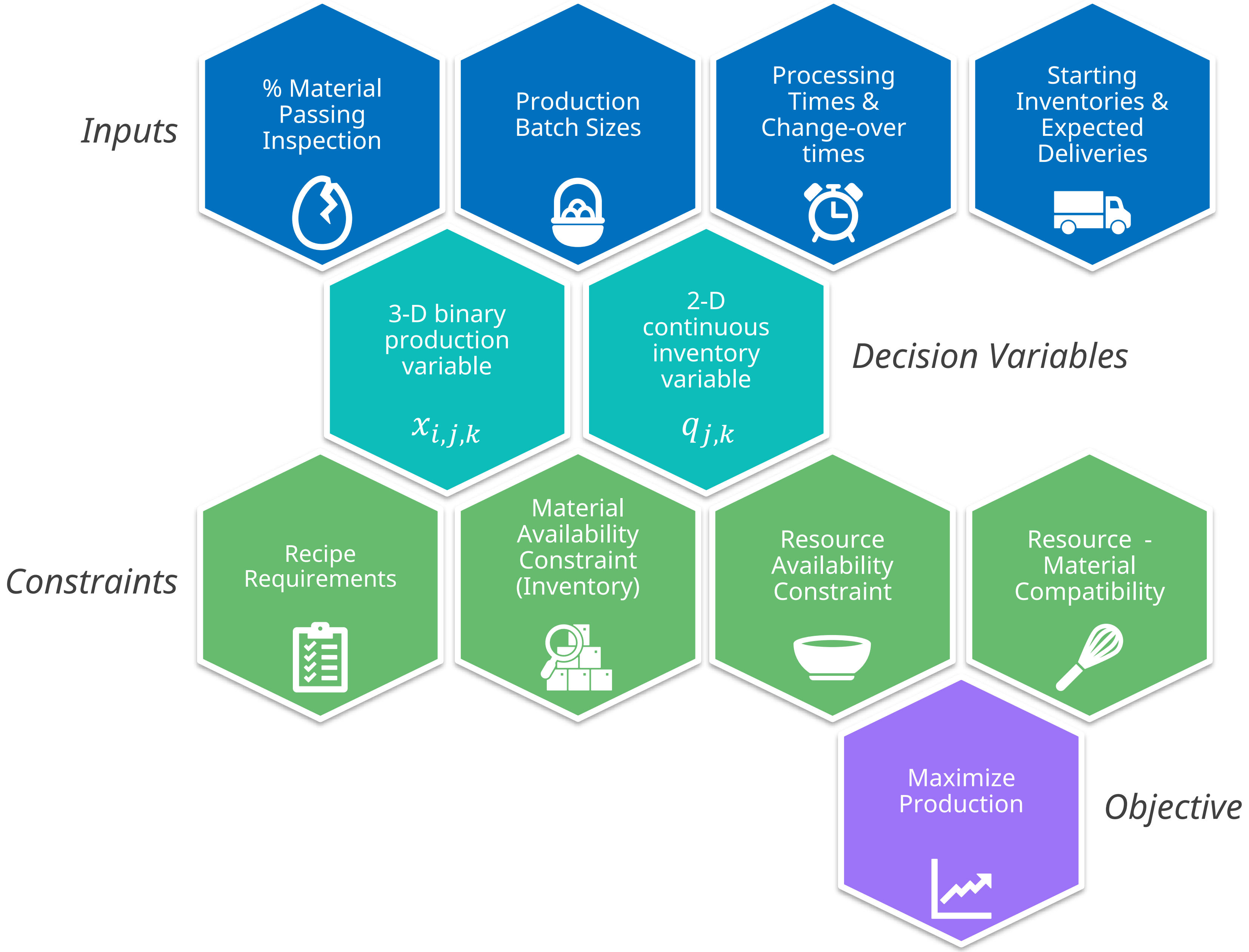


Competing Requirements

The components compete for ingredients and resources. Producing more of one could mean producing less of another, especially considering each has a different batch size

			SUGAR	MILK	BUTTER						Batch size
		x3	x6	x3	x2			x2	x1	x1	5 cups
		x1		x1		x6	x2	x3	x1	x1	3 cups
	x2	x3	x6	x2	x2				x1	x1	8 cups
Available	x10	x12	x18	x24	x8	x12	x6	x12	x1	x1	

Putting it all Together: Optimization Formulation



Objective Iterations

Total # of Finished Component Batches

$$\max \sum_{i,j,k} x_{i,j,k}$$

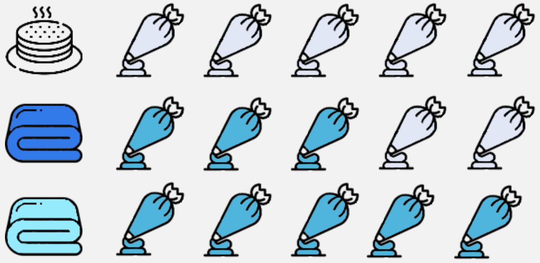
Will prioritize simple recipes & those with inventory on hand

Objective Iterations

Total # of Finished Component Batches

$$\max \sum_{i,j,k} x_{i,j,k}$$

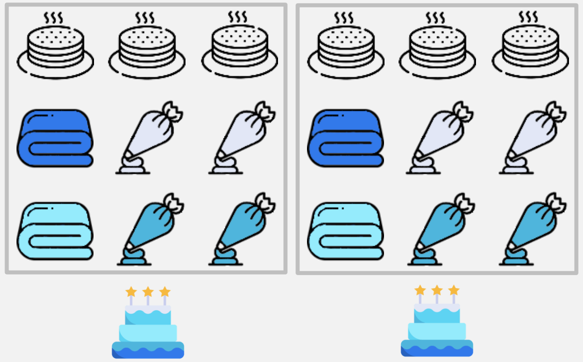
Will prioritize simple recipes & those with inventory on hand



Equal Quantities of Finished Components

$$\max(\min_{j} \sum_{i,k} b_j * x_{i,j,k})$$

Only as good as weakest link, provides more realistic outcomes



Objective Iterations

Total # of Finished Component Batches

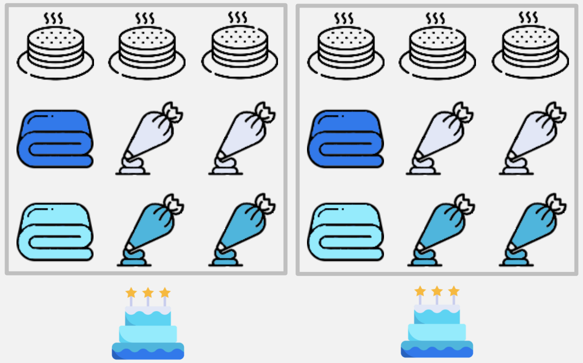
$$\max \sum_{i,j,k} x_{i,j,k}$$

Will prioritize simple recipes & those with inventory on hand

Equal Quantities of Finished Components

$$\max(\min_{j} \sum_{i,k} b_j * x_{i,j,k})$$

Only as good as weakest link, provides more realistic outcomes



Match or Exceed the Proposed Plan

$$\max \sum_{j} \left(\sum_{i,k} x_{i,j,k} - planned_j \right)$$

Meets minimum targets set by planners & identifies opportunities for additional output

Objective Iterations

Total # of Finished Component Batches

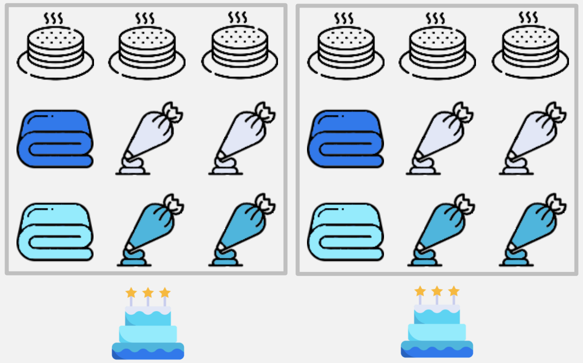
$$\max \sum_{i,j,k} x_{i,j,k}$$

Will prioritize simple recipes & those with inventory on hand

Equal Quantities of Finished Components

$$\max(\min_{j} \sum_{i,k} b_j * x_{i,j,k})$$

Only as good as weakest link, provides more realistic outcomes



Match or Exceed the Proposed Plan

$$\max \sum_{j} \left(\sum_{i,k} x_{i,j,k} - planned_j \right)$$

Meets minimum targets set by planners & identifies opportunities for additional output

More Realistic, More Computational Complexity

Objective Iterations

Total # of Finished Component Batches

$$\max \sum_{i,j,k} x_{i,j,k}$$

Will prioritize simple recipes & those with inventory on hand

Results:

15% increase in number of batches produced

Equal Quantities of Finished Components

$$\max(\min_{j} \sum_{i,k} b_j * x_{i,j,k})$$

Only as good as weakest link, provides more realistic outcomes

Results:

33% reduction in days to minimum requirements

Match or Exceed the Proposed Plan

$$\max \sum_{j} \left(\sum_{i,k} x_{i,j,k} - planned_j \right)$$

Meets minimum targets set by planners & identifies opportunities for additional output

Results:

18% improvement on proposed plans

More Realistic, More Computational Complexity

Technical Problem Solving with Gurobi

Warm Start

- Results in solution being reached
- Allows us to run a more complex model over a longer time period
- Took 3 approaches to warm starts

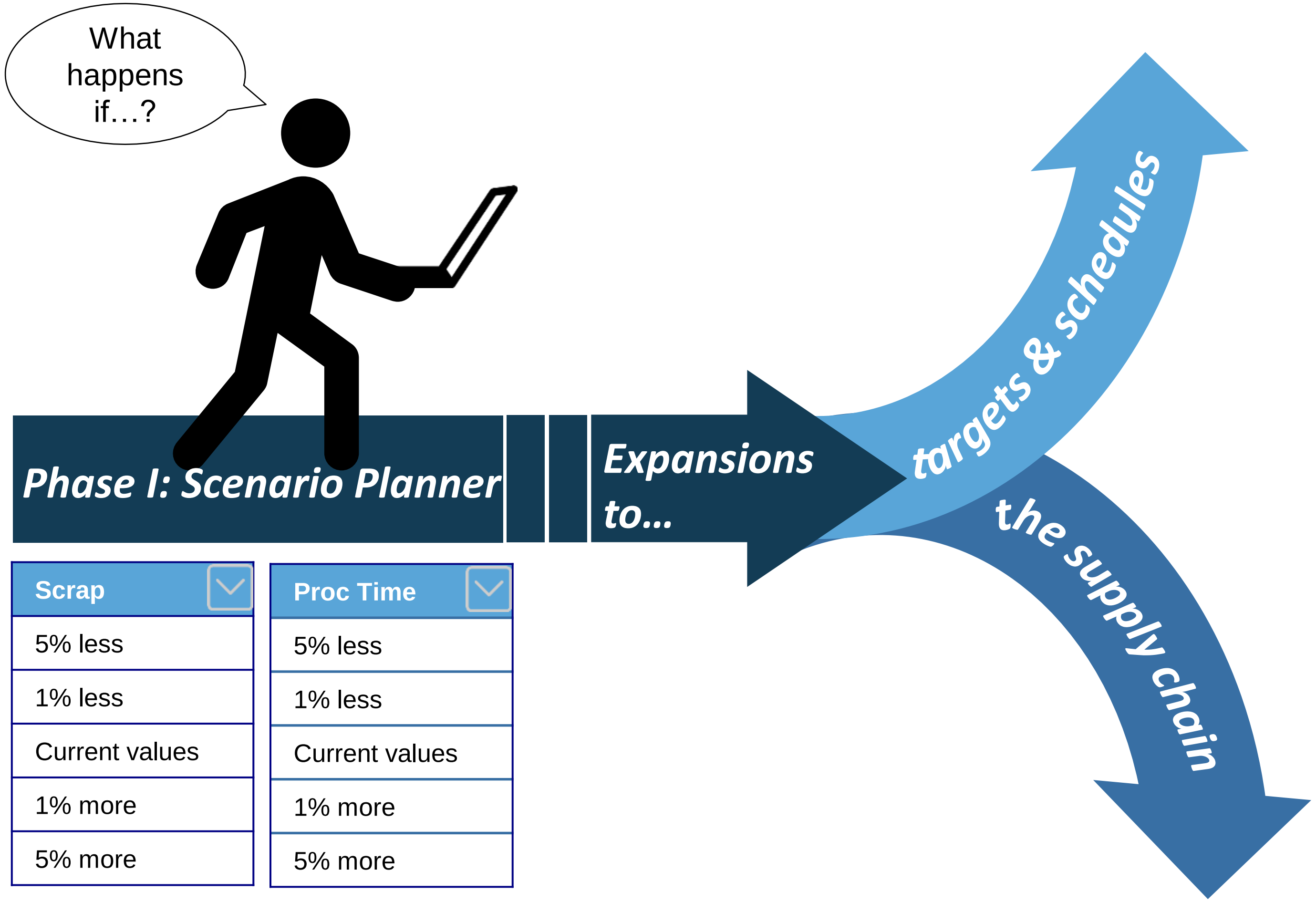
Robustness

- Applied to processing times
- Allows us to account for the “worst case scenario”
- Ensures output schedule will not be too sensitive to a single delay or hiccup

Compatibility

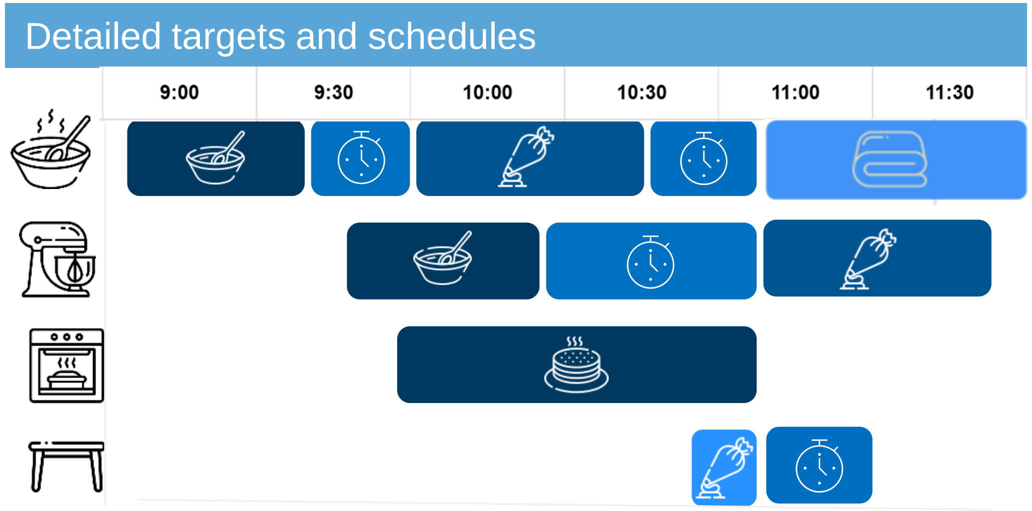
- Compatibility with both Python and Julia
- Easy to connect to and monitor jobs on the Compute Server

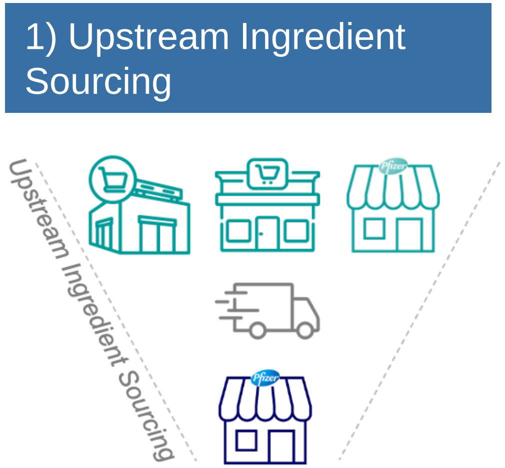
Implementation

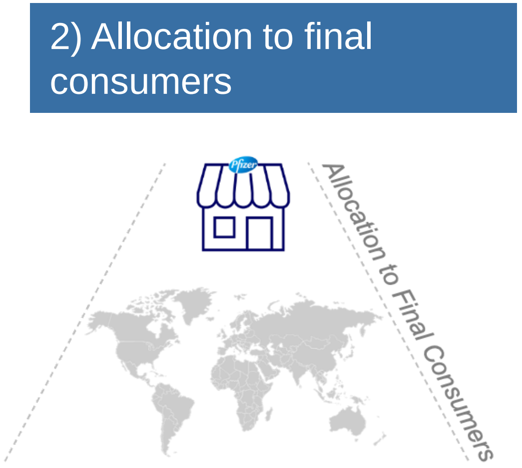


Scrap
5% less
1% less
Current values
1% more
5% more

Proc Time
5% less
1% less
Current values
1% more
5% more







Thank you!
Any questions?