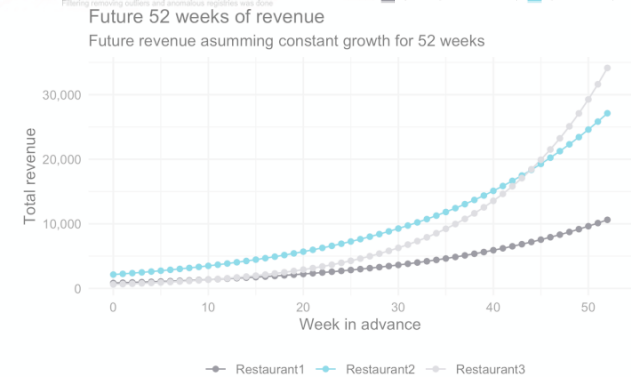
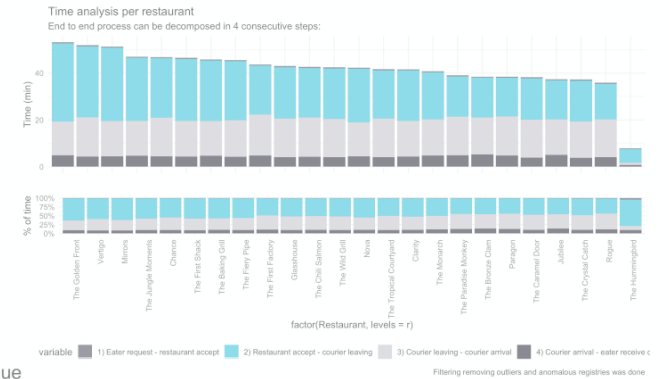
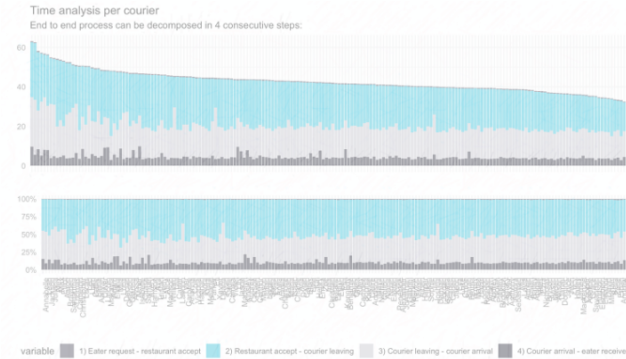


# Operations Manager | Uber Eats Exercise

Salvador Garcia

09/09/2019



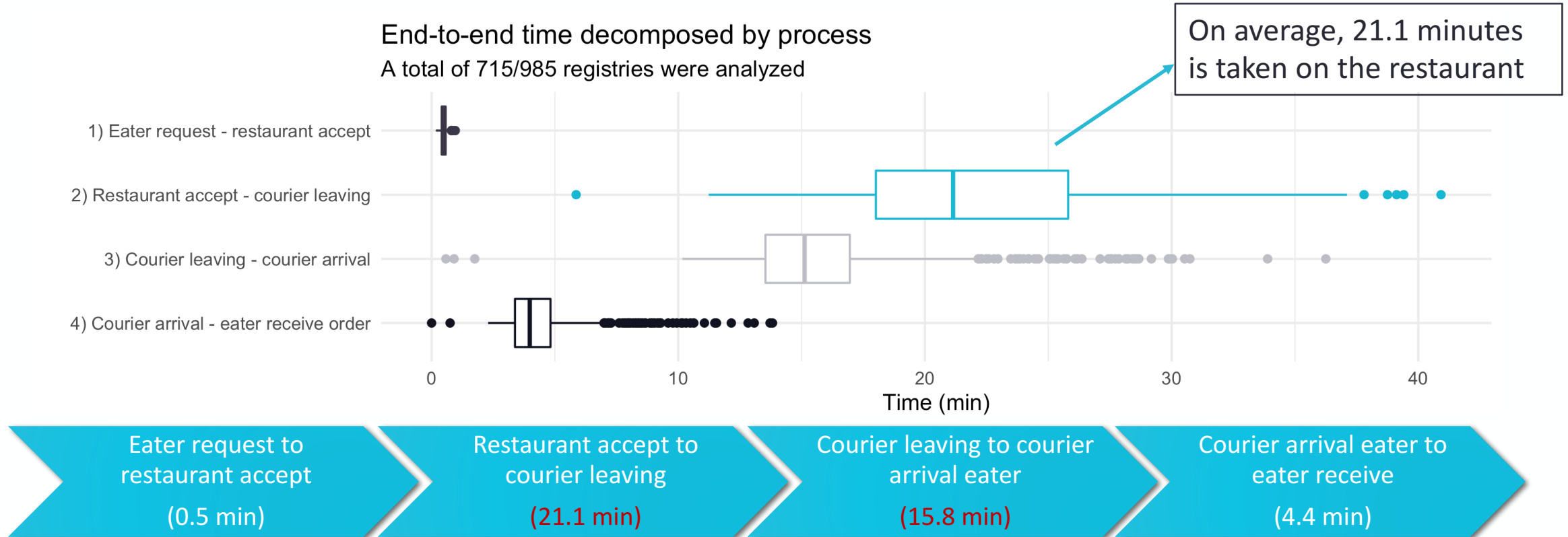
Uber Eats

Getting to the goal of 35 minutes per order

IMPROVING END TO END TIME PERFORMANCE

# Total E2E time can be decomposed in 4 consecutive steps

From eater request to eater receive order, it is possible to detect 4 consecutive steps. A wide dispersion it is found on two steps that can be managed by Uber Eats team. Currently the average E2E time is 40.7 minutes.



In terms of time, an effort on improving and standardizing times in steps 2) and 3) can provide an improvement that leads to the goal of 35 minutes

# The wider dispersion (2) can be divided in 4 further steps

In an optimal scenario, the only extra time after the restaurant accepting the order should be the preparation time (on average 14.7 min), but 21.1 minutes are spend on the restaurant. As a consequence, **6.4 min on average can be saved**.

Time to begin trip (21.1 min)

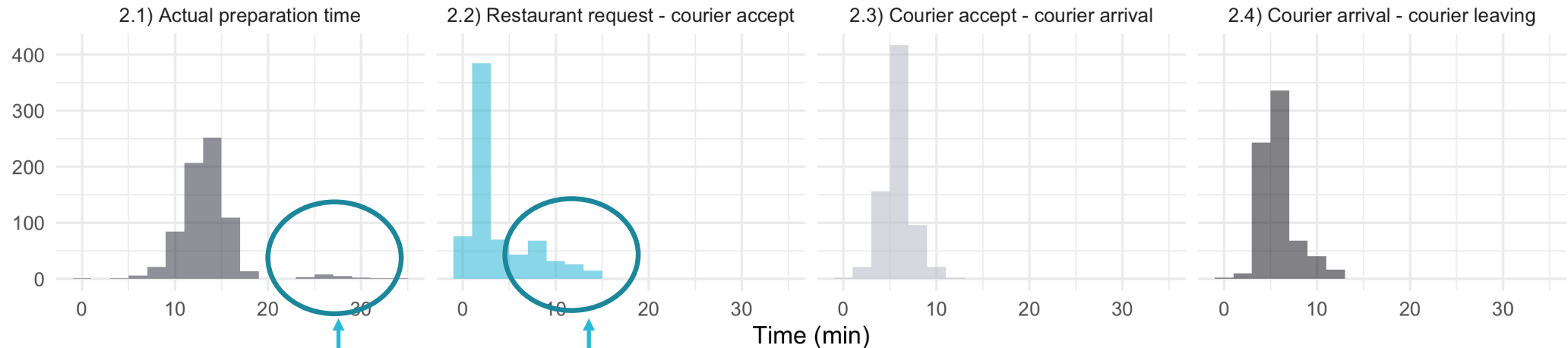
Preparation time (14.7 min)

Find courier (3.2 min) + Courier arrival (5.7 min) + Courier leave (5.8 min)

On optimal scenario, the two process can be performed on parallel. Thus, taking only the maximum time amount of both of them

(Restaurant accept - courier leaving) decomposed by process

A total of 715/985 registries were analyzed



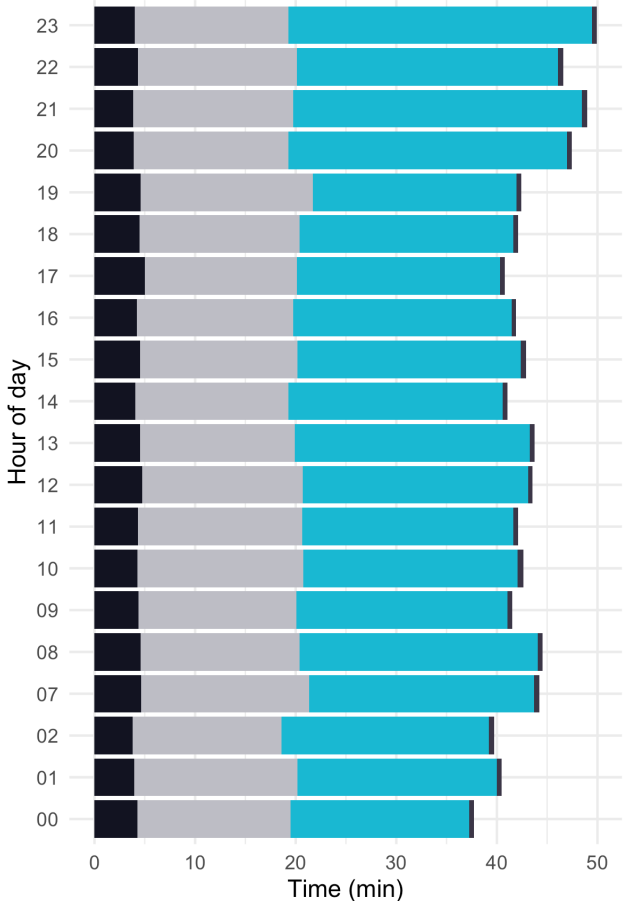
Dispersion on finding a courier is high, **some taking over 10 minutes**. In addition, an standardization of actual preparation time should be suggested, as some orders take over 30 minutes

# Step 2.2 exhibits long waiting time, causing order to be delayed

In some hours of the day the E2E time takes longer, in special the waiting time for courier. These hours are at 7,8,20,21,22,23 hours, so implementing a custom demand incentives for courier availability can be performed

Analysis of time per hour  
End to end process can be decomposed in 4 consec

Incentives to reduce  
Time in some hours



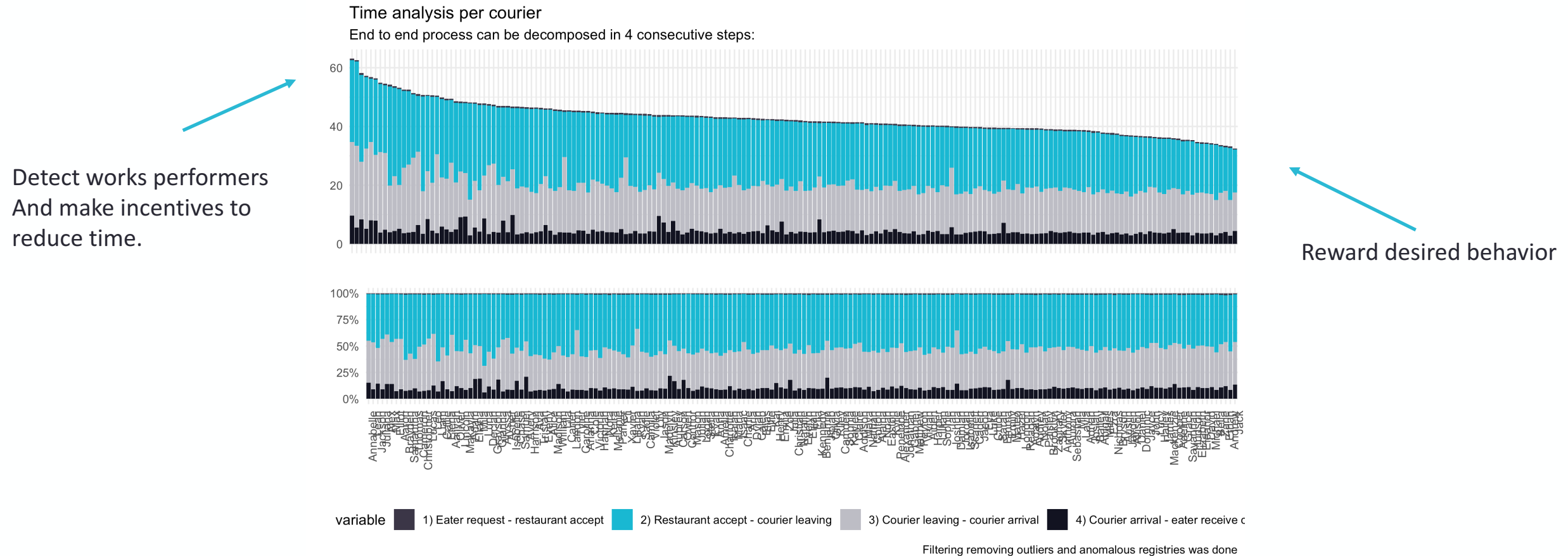
Hour	Rest request – courier accept
0	12.2
1	11.7
2	10.7
7	11.1
8	2.4
9	2.5
10	2.8
11	3.1
12	2.5
13	2.6
14	3.1
15	2.8
16	2.9
17	2.5
18	2.7
19	2.7
20	2.6
21	3.0
22	2.1
23	1.8

For orders in hour 0,1,2,7,8  
Up to 9 minutes can be reduced  
(but just a small amount of orders)

At some hours, acceptance  
time is longer, degrading  
service performance

# Standardization of times per courier and restaurant

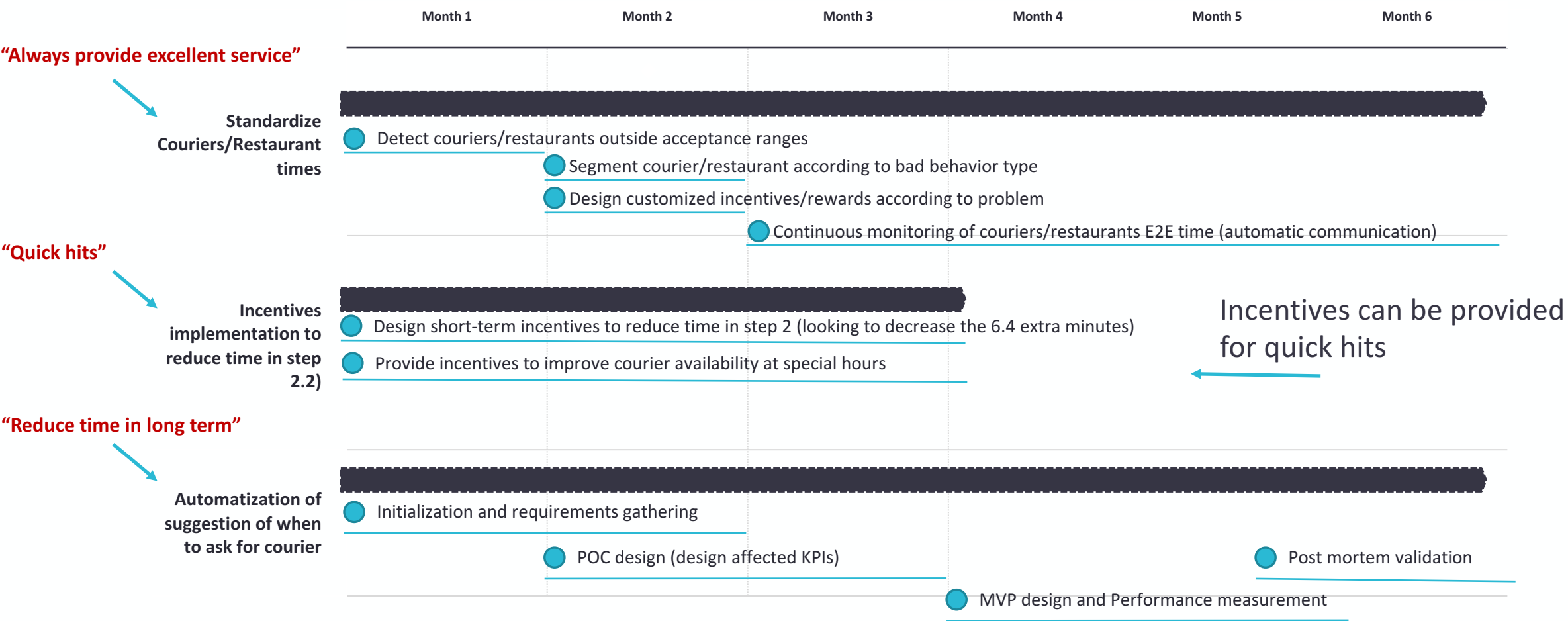
Data exhibits a time difference taking into consideration restaurants and couriers. Some difference can be justified by travel time and order complexity, but top/bottom performers can be identified



Best performer couriers exhibit in average half the time vs worse performers. The same applies to restaurants

# A plan consisting in 3 pillars is suggested

The first and most important pillar is the automatization of automatic suggestion of when to ask for a courier to reduce E2E time in long term. This has a potential of reducing up to 6.4 minutes avg the E2E time. Next is the always provide excellent service



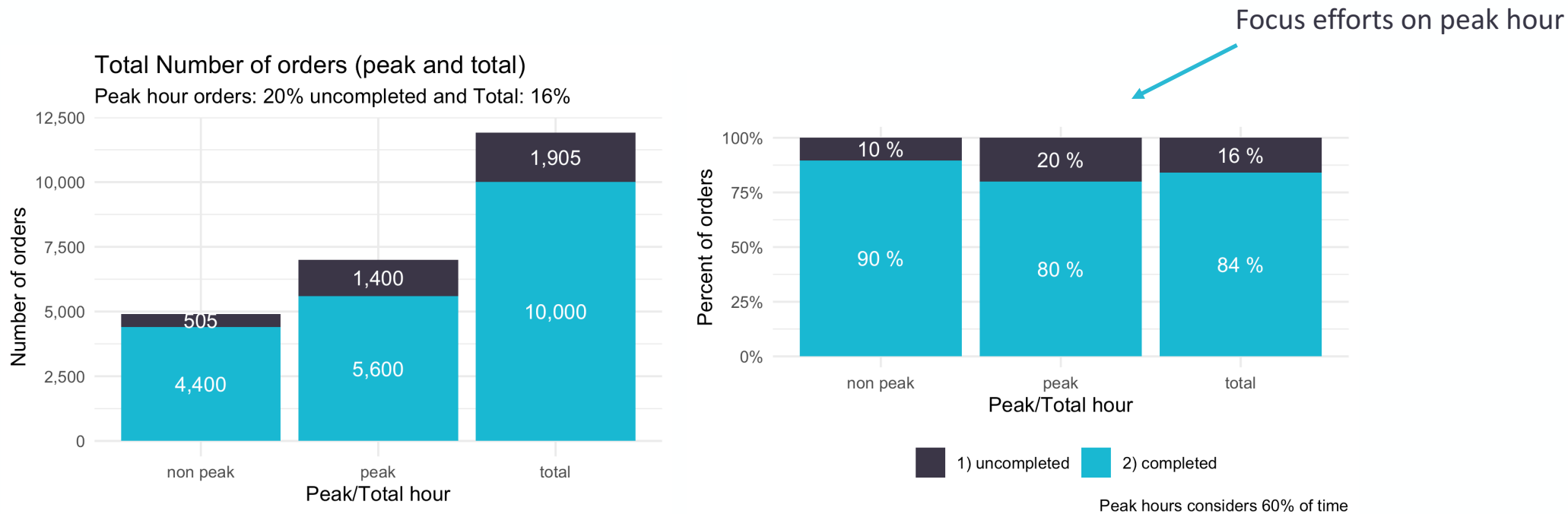
# Uber availability improvement

DETECTING SERVICE GAPS FOR COURIERS



# Peak hours exhibits a larger amount of incomplete orders

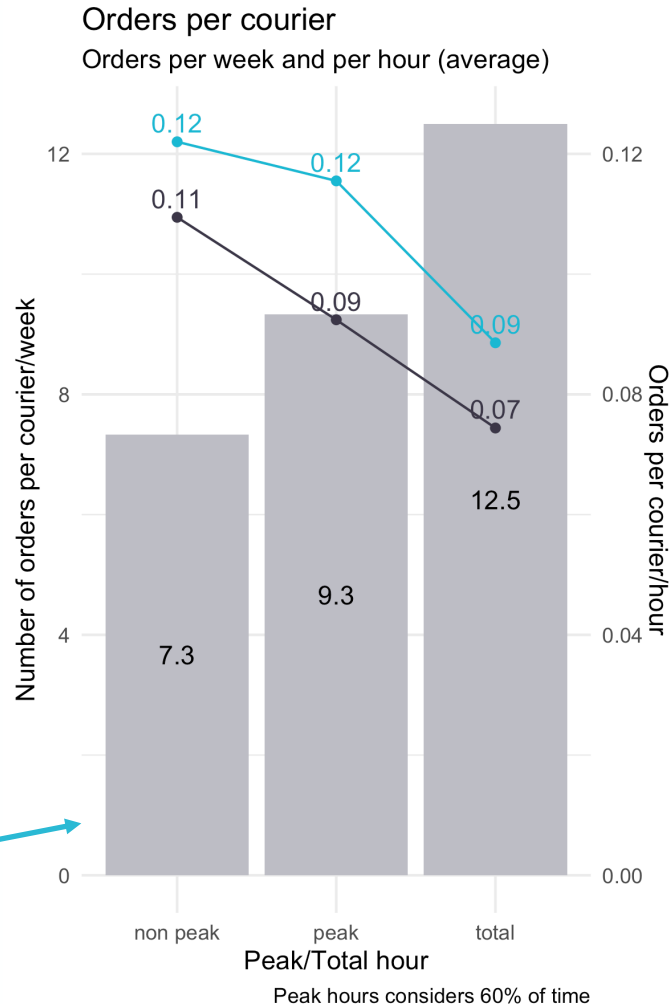
In average for peak hours 20% of orders are nor provided. This represents 1400/1905 (73%) of total orders not provided in 60% of the total time.



If reduced peak incomplete orders to levels of non-peak hours, the total incomplete orders would be reduced up to 6 percent points (700 complete orders per week, 6% more of orders per week)

# The total number of courier for the market seems to be adequate

When normalizing by courier and by courier/hour, only 12.5 orders per week per courier are accountable. This means that per hour (assuming 8\*5 working hours), only 0.09 orders per hour in average is presented (low courier share of time)



Assuming 600  
couriers on not  
peak hours

This can be caused because of the couriers work hours (not full time). Thus an strategy as follows can be implemented:

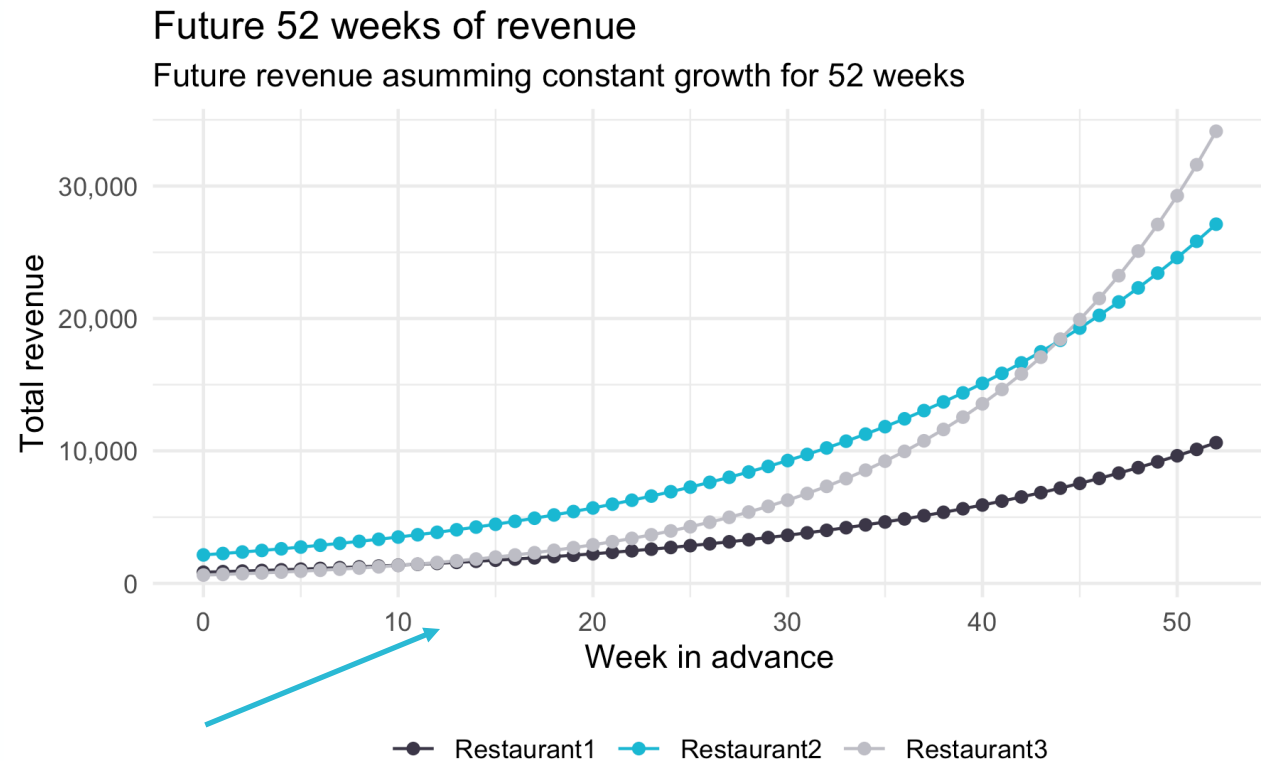
- Increase courier database at start (quick hit)
- Create incentives to increase total time spent on the app (quick hit)
- Segment courier database to reward good behave couriers (long term strategy)
- Try to maintain a healthy database where full time courier are procured (long term strategy, low cost, more experience couriers)

# Revisiting fee strategy

BEST RESTAURANT SELECTION

# With a forecast of revenue, the best option at long term is found

When considering a time window of 52 weeks (assuming constant growth) in terms of revenue, the restaurant 3 is the most promising option. If only a time window of 26 weeks is considered, then Restaurant 2 is the best option



In time window of 26 weeks, restaurant 2 provide more total revenue than other options

Future information for 52 weeks

When considering a time window of 26 weeks, The restaurant 2 is a promising option. This restaurant is a safe option because it has a good yelp review, so It is very likely that users like the meal (thus, good image for Uber)

For the **unitary revenue**, the fee %, the avg basket size and the delivery cost is considered

For the **total revenue**, the total number of eaters, the number of weekly orders and the unitary revenue is considered