Analysis of 77072 zip code using Facebook Prophet

Imports and loading csv

In [3]:
```
from fbprophet import Prophet
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
import warnings
warnings.filterwarnings('ignore')
#Supress default INFO logging
import logging
logger = logging.getLogger()
logger.setLevel(logging.CRITICAL)
import logging, sys
logging.disable(sys.maxsize)
from fbprophet.diagnostics import cross_validation
```

In [4]:
```
df=pd.read_csv('df_zillow_77072_prepped_fbprophet.csv')
```

In [5]:
```
df.head()
```

Out[5]:

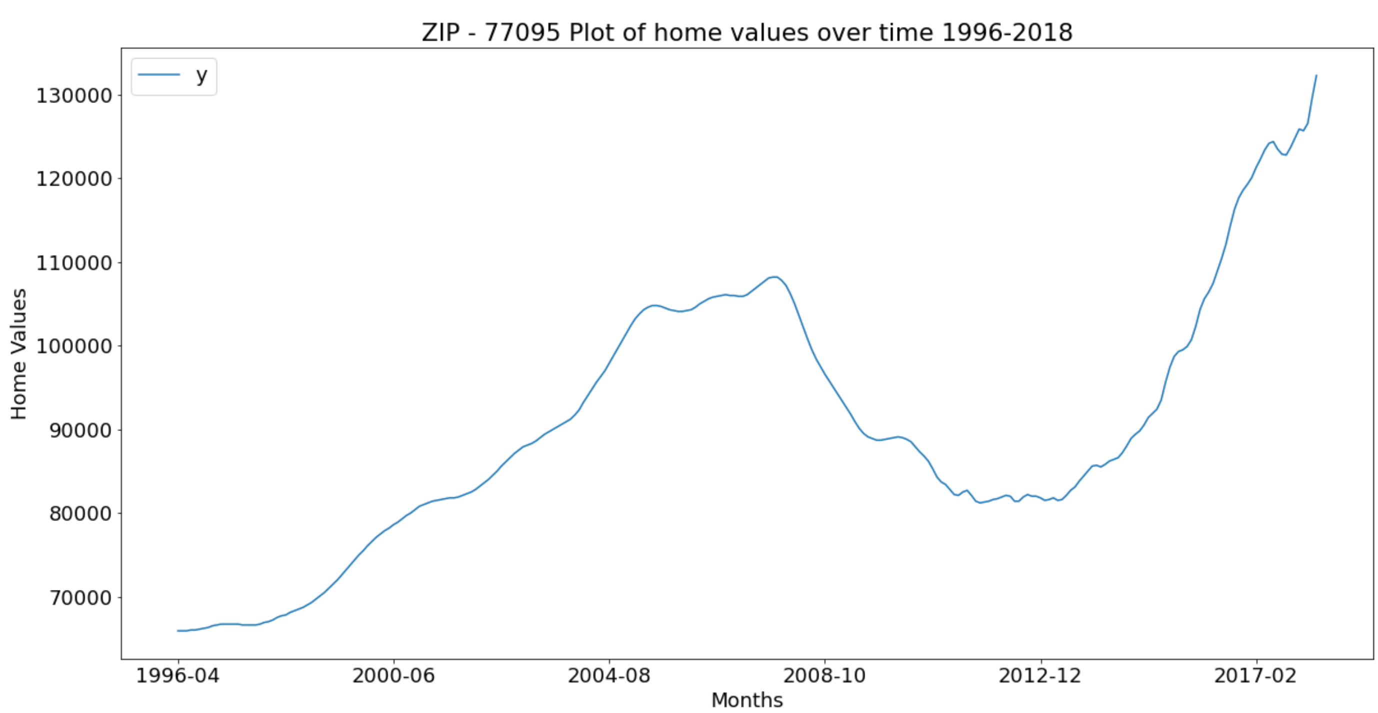
	ds	y
0	1996-04	65900.0
1	1996-05	65900.0
2	1996-06	65900.0
3	1996-07	66000.0
4	1996-08	66000.0

Plotting the specific zip code data from csv

In [6]:
```
plt.figure()
plt.rcParams.update({'font.size': 18})
ax = df.plot(title='ZIP - 77072 Plot of home values over time 1996-2018', figsize=(20,10),
ax.set_xlabel('Months')
ax.set_ylabel('Home Values')
```

Out[6]:
```
Text(0, 0.5, 'Home Values')

<Figure size 432x288 with 0 Axes>
```



Fitting and forecasting the model

```
1. The length of the forecast will be 5 years into the future.
2. Periods = 60 with freq = M
    - 60 months / 12 months per year  = 5 years
3. seasonality_mode = multiplicative
    - This is because additive would mean our graph will have a STEADY upward clim
b
    - This is not the case. There is a HUGE upward climb. Thus, multiplicative was
used instead of additive.
```

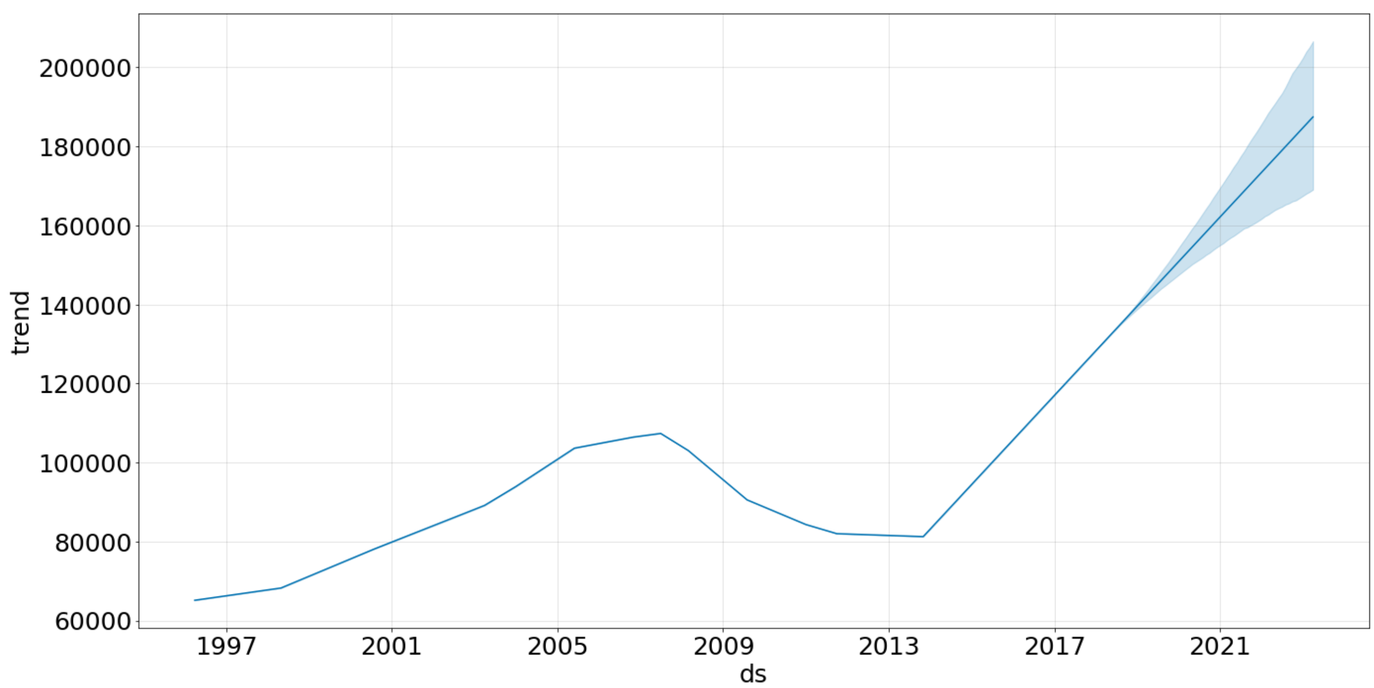
Observations:

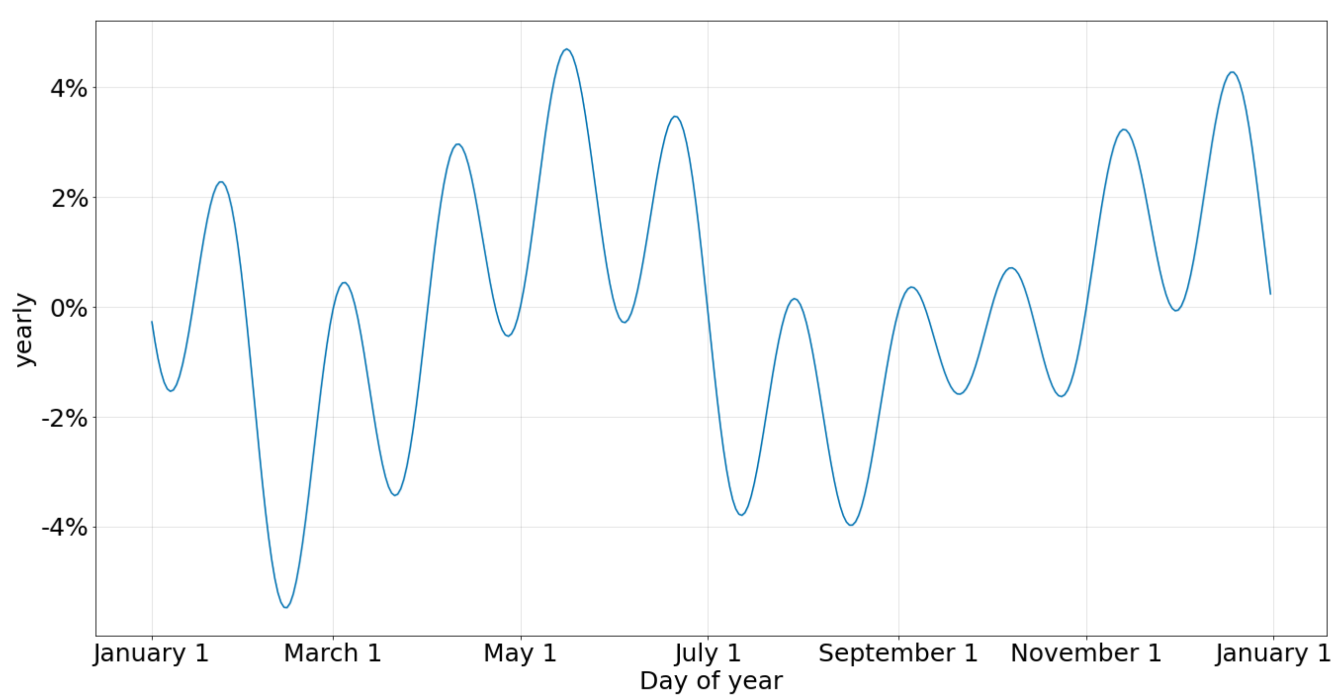
```
1. The trend shows promise, reflects the growth and demand of properties in the ar
ea.
2. The 2008 crash is reflected in the dip in home prices. This should not be confu
sed for a cyclical occurence.
3. We cannot say much about seasonality. There is a huge upward trend.
    - Future work - maybe find stronger seasonality in daily data instead of month
ly.
```

In [7]:

```
m = Prophet(seasonality_mode='multiplicative').fit(df)
future = m.make_future_dataframe(periods=60, freq='M')
fcst = m.predict(future)
plt.figure()
plt.rcParams.update({'font.size': 25})
fig = m.plot_components(fcst, figsize=(20,20))
```

```
<Figure size 432x288 with 0 Axes>
```





In [8]:
```
fcst[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

Out[8]:

	ds	yhat	yhat_lower	yhat_upper
320	2022-11-30	183594.939898	166874.496955	200744.292388
321	2022-12-31	185309.246263	168216.656758	203440.359730
322	2023-01-31	186433.334023	168491.991571	204973.858758
323	2023-02-28	185588.481847	167689.532483	204701.663838
324	2023-03-31	186255.714592	167542.065894	205576.010058

Forecast Model Diagnostics

Here I will check the accuracy of the model using cross validation

Cross validation parameters are as follows:

```
1. Model will be "m" from above fitted by Prophet() method
2. The initial training lengh parameter will be 5475 days or 15 years (365*15 = 54
75)
    - This means cutoff will be after 15 years (1996 - 2011)
3. The horizon will be 1825 days or 5 years (365 * 5 = 1825)
    - from 2012 - 2017
4. The period is set to 180 days
    - Means it will make a prediction roughly every 6 months
```

In [9]:
```
cv_results = cross_validation( model = m, initial = pd.to_timedelta(5475, unit="d"),period
```

A Jupyter widget could not be displayed because the widget state could not be found. This could happen if the kernel storing the widget is no longer available, or if the widget state was not saved in the notebook. You may be able to create the widget by running the appropriate cells.

In [10]:
```
cv_results.head()
```

Out[10]:

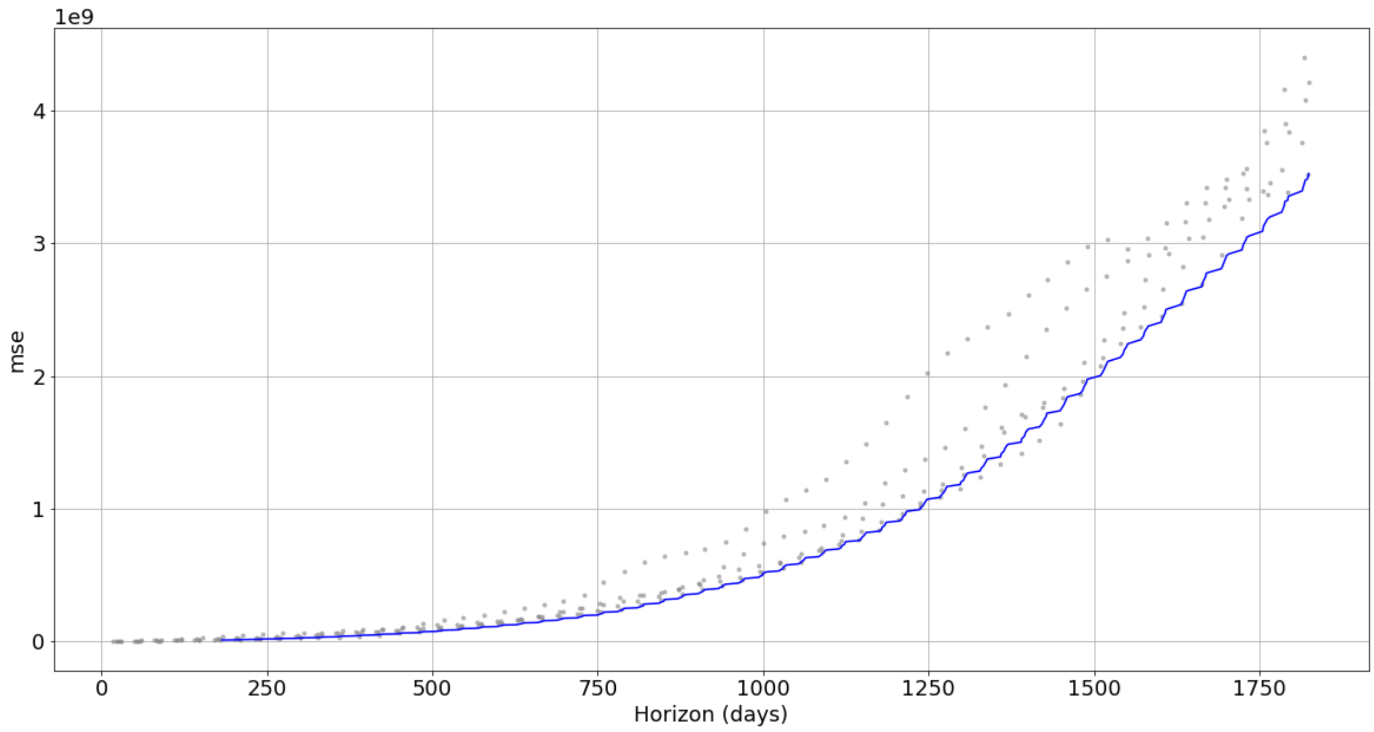
	ds	yhat	yhat_lower	yhat_upper	y	cutoff
0	2011-05-01	80319.686726	78684.123620	81888.530878	82100.0	2011-04-13
1	2011-06-01	79817.043714	78197.831499	81378.179651	82500.0	2011-04-13
2	2011-07-01	79241.061996	77664.427581	80774.559699	82700.0	2011-04-13
3	2011-08-01	78550.756301	76916.471687	80106.044208	82100.0	2011-04-13
4	2011-09-01	77850.868834	76210.391966	79493.785972	81400.0	2011-04-13

MSE observation:

```
1. MSE starts to increase rapidly after 1000 days.
2. This reflects higher uncertainty the farther into the horizon
```

```
In [11]: from fbprophet.plot import plot_cross_validation_metric
plt.figure()
plt.rcParams.update({'font.size': 18})
fig = plot_cross_validation_metric(cv_results, metric='mse', figsize=(20,10))
```

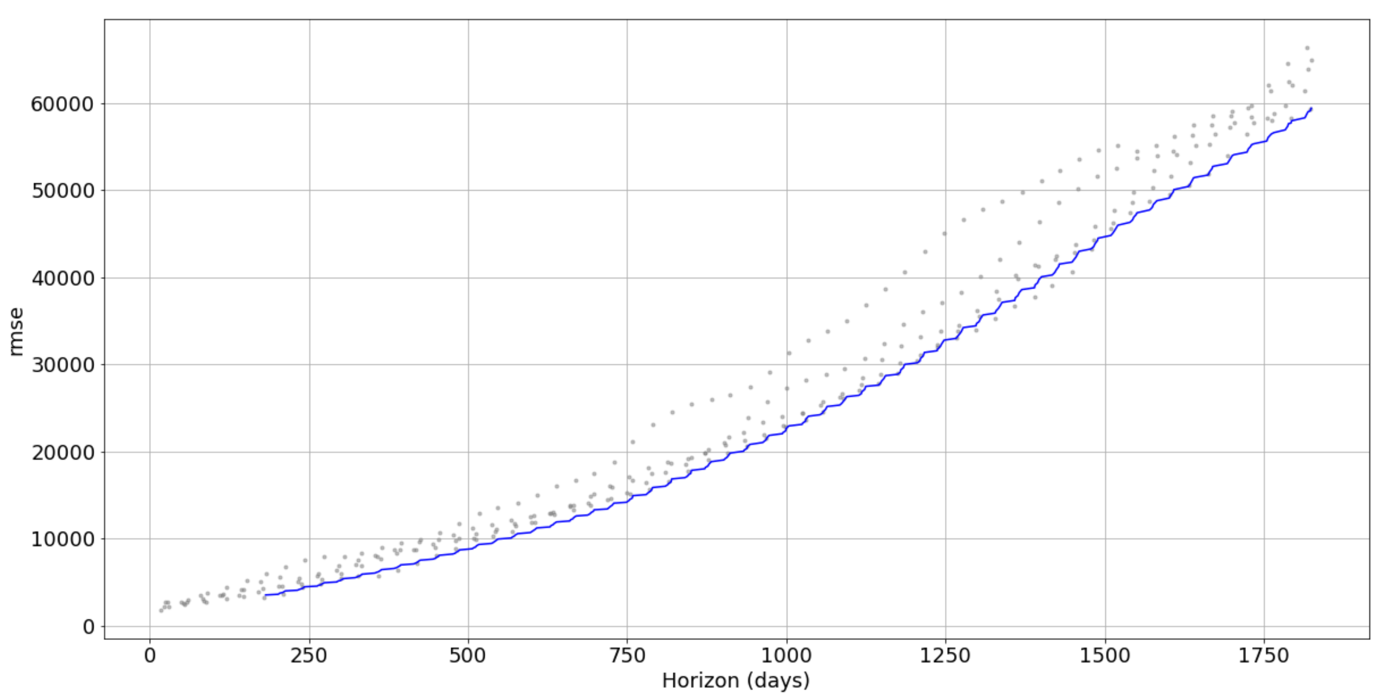
<Figure size 432x288 with 0 Axes>



RMSE Observation:

1. Similar to MSE, the error increases with longer time into horizon

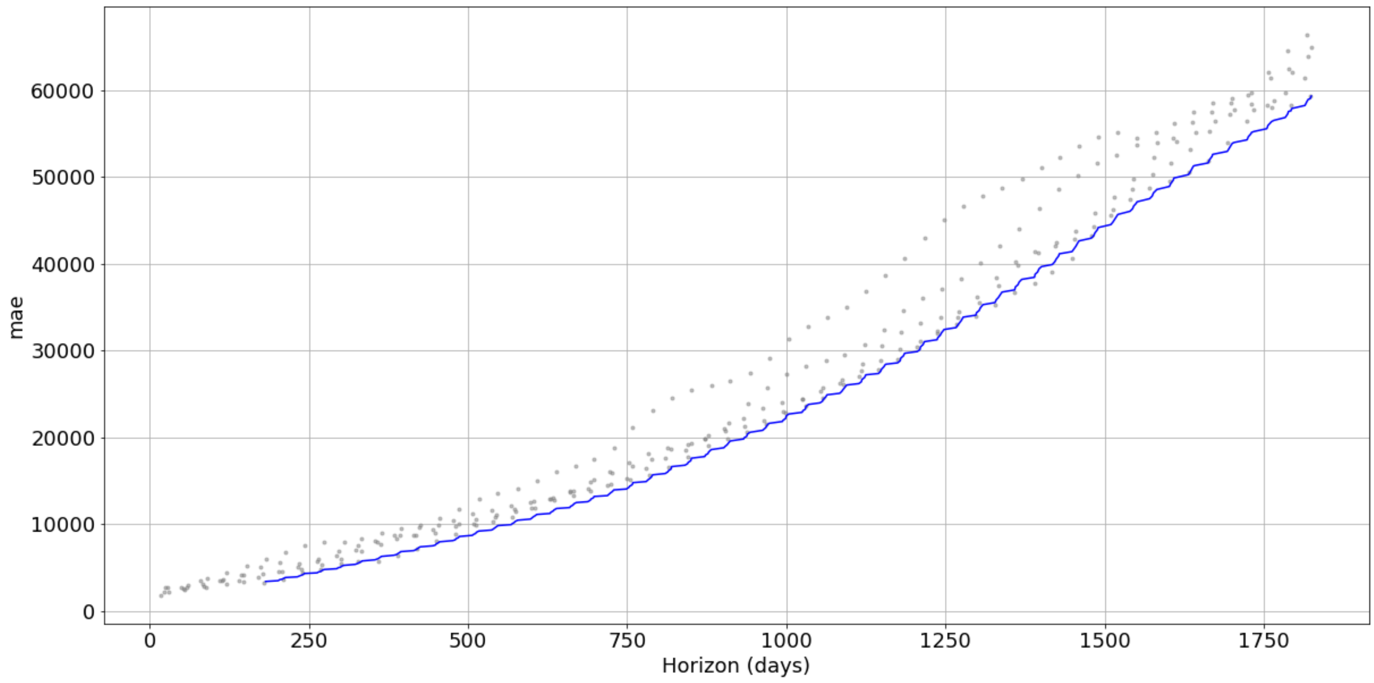
```
In [12]: fig = plot_cross_validation_metric(cv_results, metric='rmse', figsize=(20,10))
```



MAE - Mean Absolute Error Observation:

1. Similar to MSE and RMSE, the difference starts increasing as the number of obse rvations are increasing

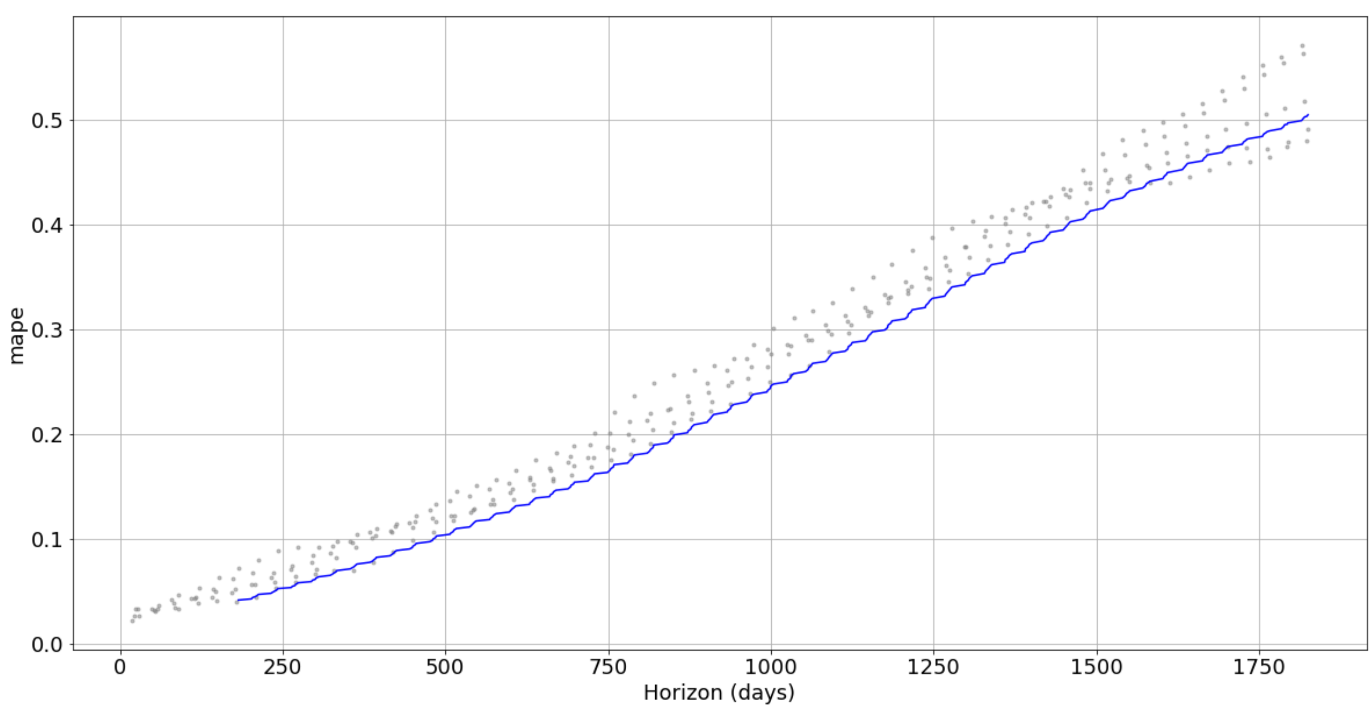
In [13]:
```
fig = plot_cross_validation_metric(cv_results, metric='mae', figsize=(20,10))
```



MAPE (Mean Average Percent Error) - Observation:

```
1. We see that MAPE increases over time
2. I am willing to tolerate MAPE of 0.1 to 0.2
    - This gets exceeded after about 800 days
3. We will focus on MAPE as our main diagnostic metric.
    - Shows the model was about 80% accurate at 800 days
    - Bullish prediction for the next 2-3 years
    - Supports the high upward trend we saw in the graph of all the data points fo
r the zip code
```

In [14]:
```
fig = plot_cross_validation_metric(cv_results, metric='mape', figsize=(20,10))
```



Coverage Observation:

```
1. From the fbprophet document on coverage: These intervals assume that the future
will see the same frequency and magnitude of rate changes as the past. This assump
tion is probably not true, so you should not expect to get accurate coverage on th
ese uncertainty intervals.
2. Given that coverage is zero here, we can assume that it is due to the upward tr
end.
3. Similar to other zip codes, coverage goes to zero after some time.
4. This means the true value does not lie in the range of the estimated values
```

In [15]:
```
fig = plot_cross_validation_metric(cv_results, metric='coverage', figsize=(20,10))
```