Analysis of 77077 zip code using Facebook Prophet

Imports and loading csv

In [4]:
```
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 [5]:
```
df=pd.read_csv('df_zillow_77077_prepped_fbprophet.csv')
```

In [6]:
```
df.head()
```

Out[6]:

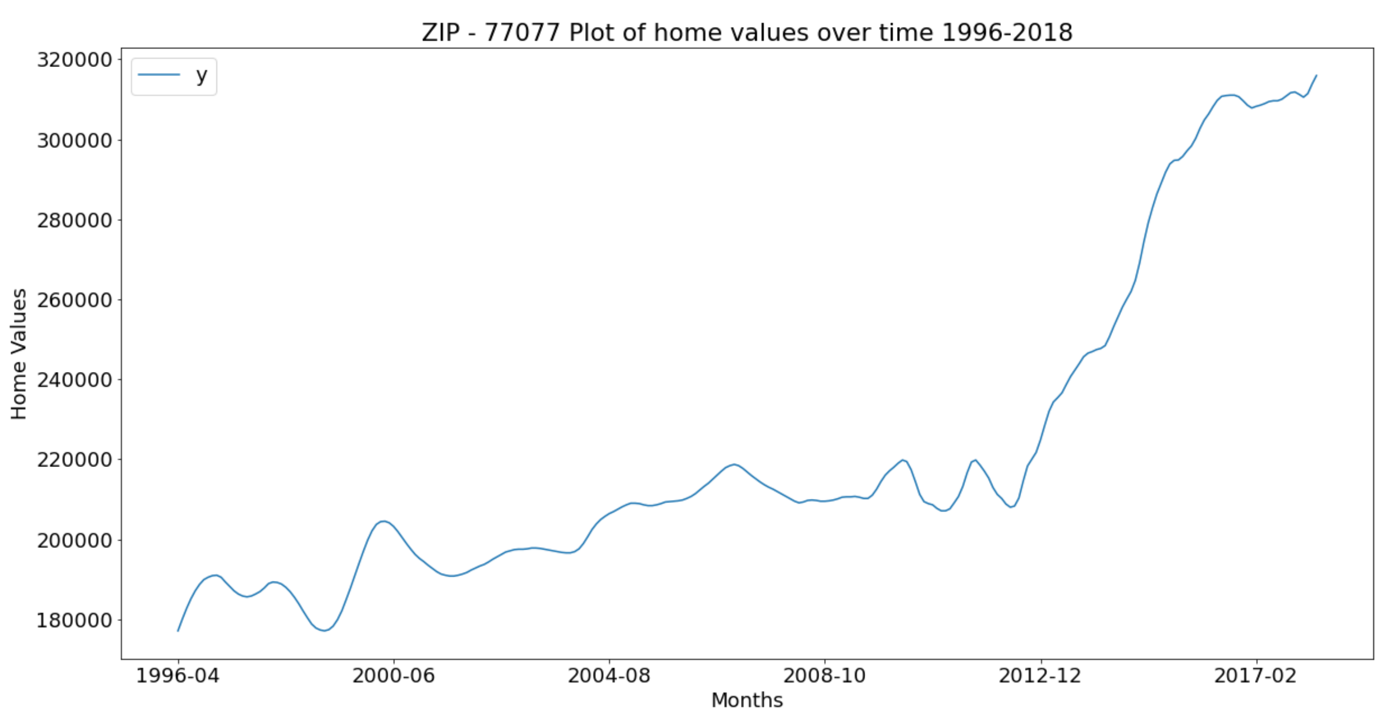
	ds	y
0	1996-04	177100.0
1	1996-05	180000.0
2	1996-06	182700.0
3	1996-07	185100.0
4	1996-08	187100.0

Plotting the specific zip code data from csv

In [7]:
```
plt.figure()
plt.rcParams.update({'font.size': 18})
ax = df.plot(title='ZIP - 77077 Plot of home values over time 1996-2018', figsize=(20,10),
ax.set_xlabel('Months')
ax.set_ylabel('Home Values')
```

Out[7]:
```
Text(0, 0.5, 'Home Values')
```

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



Fitting and forecasting the model

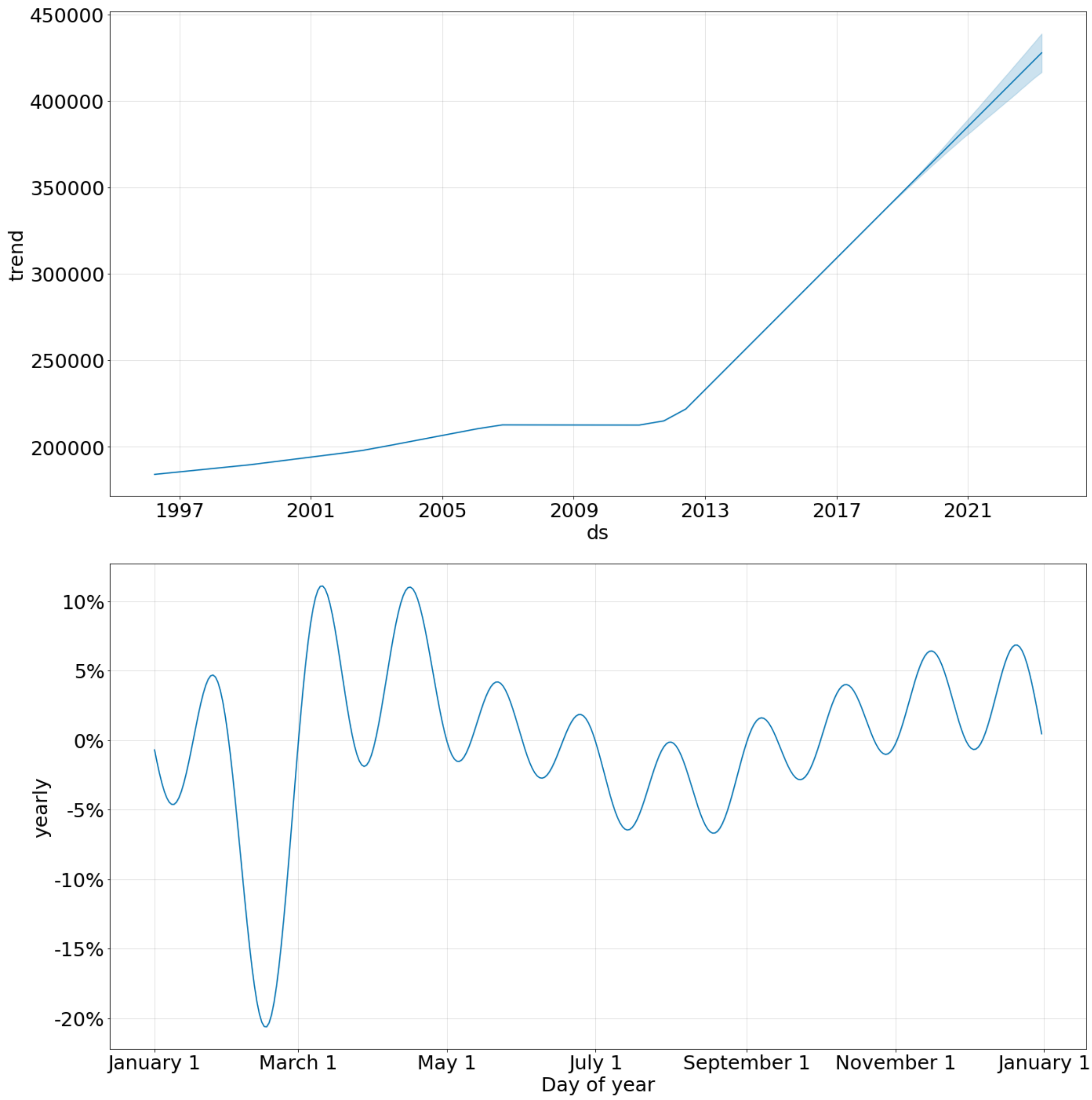
```
1. The length of the forecast will be 5 years into the future.
2. Periods = 60 with freq = M
    - 60 months / 12 months per year  = 5 years
3. seasonality_mode = multiplicative
    - This is because additive would mean our graph will have a STEADY upward climb
    - This is not the case. There is 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 area.
2. The 2008 crash is reflected in the dip in home prices. This should not be confused for a cyclical occurence.
    - The dip is not as prevalent in this area. Could be an indicator that home values were stable here.
3. We cannot say much about seasonality. There is a huge upward trend.
    - Future work - maybe find stronger seasonality in daily data instead of monthly.
```

In [8]:
```python
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 [9]:
```python
fcst[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

Out[9]:

	ds	yhat	yhat_lower	yhat_upper
320	2022-11-30	421674.851219	408580.115091	434387.045153
321	2022-12-31	426114.121121	412937.563800	438915.486983
322	2023-01-31	429668.213718	415688.494836	443958.043533
323	2023-02-28	411920.816903	398207.950950	425575.473524
324	2023-03-31	422231.313175	408408.780050	436050.057120

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 [10]:
```
cv_results = cross_validation( model = m, initial = pd.to_timedelta(5475, unit="d"),period
```

```
HBox(children=(HTML(value=''), FloatProgress(value=0.0, max=5.0), HTML(value='')))
```

In [11]:
```
cv_results.head()
```

Out[11]:

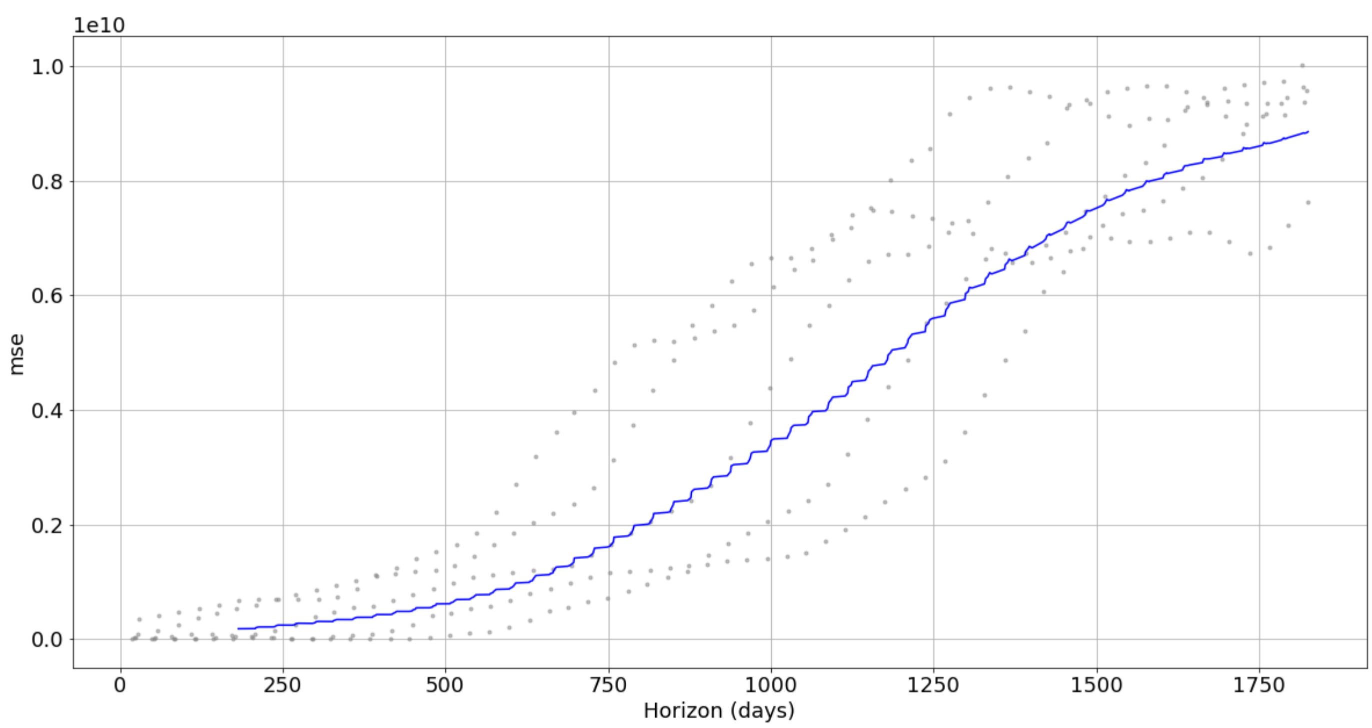
	ds	yhat	yhat_lower	yhat_upper	y	cutoff
0	2011-05-01	211473.646606	206283.459969	217456.786542	210700.0	2011-04-13
1	2011-06-01	211145.590216	205416.526644	216711.139457	213200.0	2011-04-13
2	2011-07-01	210843.114533	205173.956235	216199.993541	216600.0	2011-04-13
3	2011-08-01	210488.981128	204904.977643	215800.346538	219300.0	2011-04-13
4	2011-09-01	210478.927157	205022.188927	216162.107204	219800.0	2011-04-13

MSE observation:

1. MSE starts to increase rapidly after 500-750 days.
2. This reflects higher uncertainty the farther into the horizon

In [12]:
```
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))
```

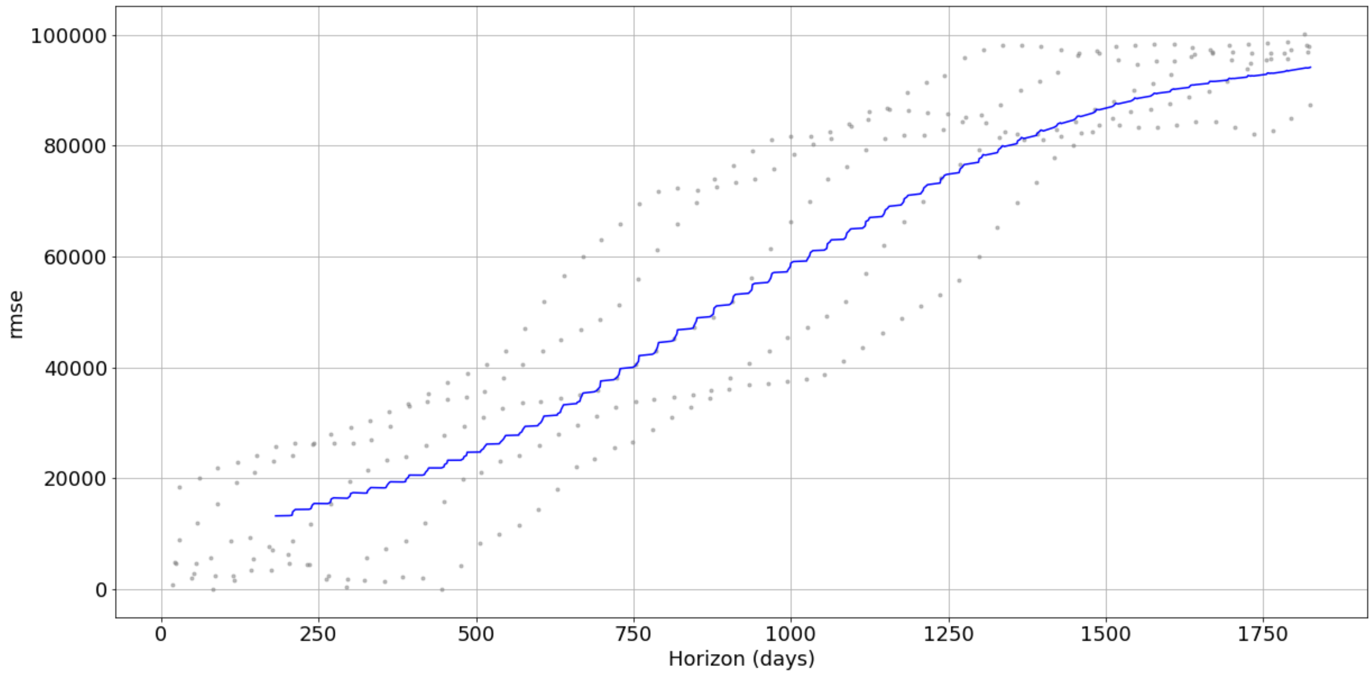
```
<Figure size 432x288 with 0 Axes>
```



RMSE Observation:

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

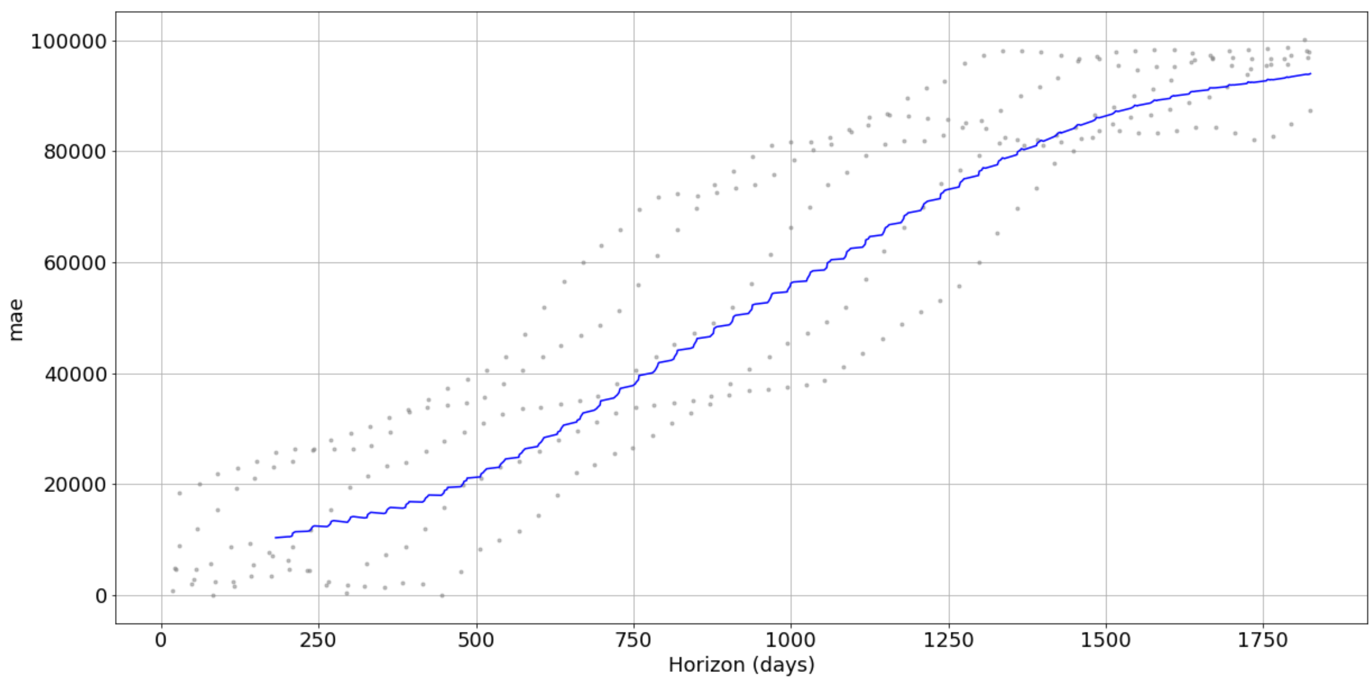
```
In [13]: fig = plot_cross_validation_metric(cv_results, metric='rmse', figsize=(20,10))
```



MAE - Mean Absolute Error Observation:

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

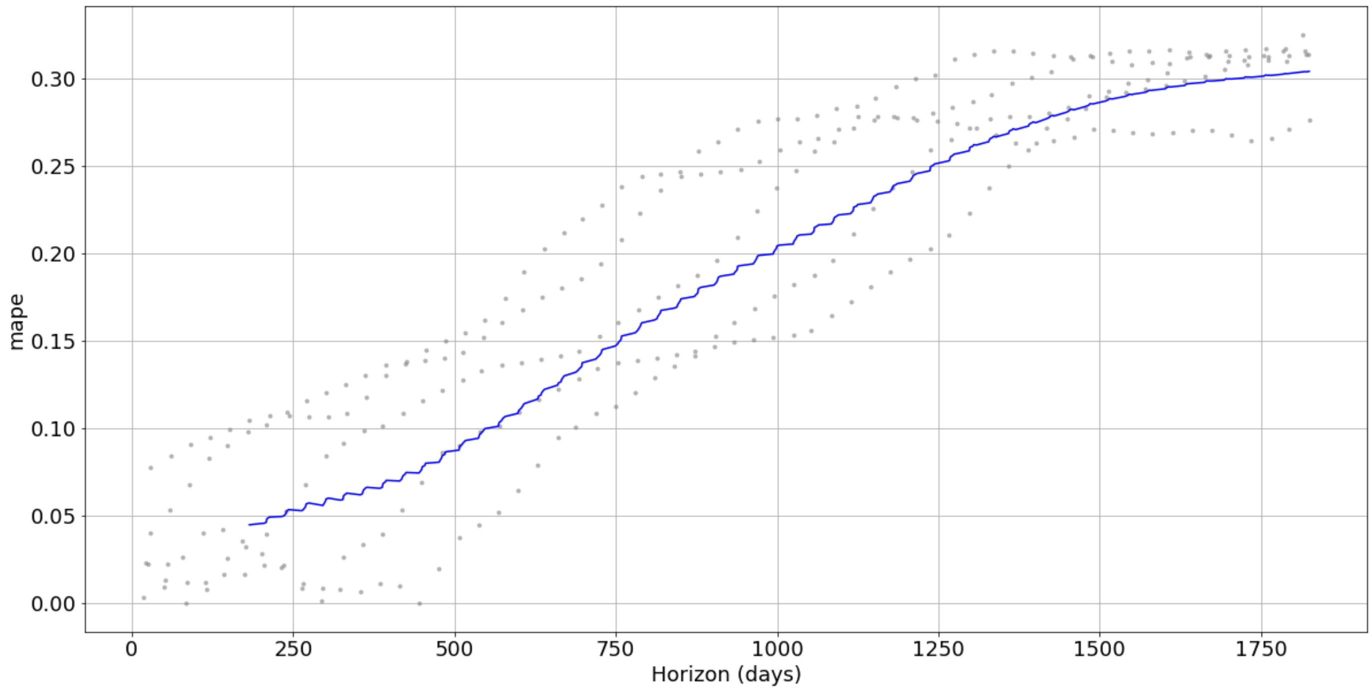
```
In [14]: fig = plot_cross_validation_metric(cv_results, metric='mae', figsize=(20,10))
```



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 1000 days
3. We will focus on MAPE as our main diagnostic metric.
    - Shows the model was about 80% accurate at 1000 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 [15]:
```
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 assumption is probably not true, so you should not expect to get accurate coverage on these uncertainty intervals.
2. After about 600 days, there won't be any true values in the estimated range.
3. Similar to other zip codes, coverage goes to zero after some time.

In [16]:
```
fig = plot_cross_validation_metric(cv_results, metric='coverage', figsize=(20,10))
```