In [3]:
```
import pandas as pd
df2 = pd.read_csv("C:\\Users\\ABC\\Downloads\\DS+-+Part3+-+CompanyX_EU (1).csv")
df2.head()
```

Out[3]:

	Startup	Product	Funding	Event	Result	OperatingState
0	2600Hz	2600hz.com	NaN	Disrupt SF 2013	Contestant	Operating
1	3DLT	3dlt.com	$630K	Disrupt NYC 2013	Contestant	Closed
2	3DPrinterOS	3dprinteros.com	NaN	Disrupt SF 2016	Contestant	Operating
3	3Dprintler	3dprintler.com	$1M	Disrupt NY 2016	Audience choice	Operating
4	42 Technologies	42technologies.com	NaN	Disrupt NYC 2013	Contestant	Operating

In [4]:
```
df2.dtypes
```

Out[4]:
```
Startup           object
Product           object
Funding           object
Event             object
Result            object
OperatingState    object
dtype: object
```

In [5]:
```
df2.isnull().sum()
```

Out[5]:
```
Startup             0
Product             6
Funding           214
Event               0
Result              0
OperatingState      0
dtype: int64
```

In [6]:
```
df3 = df2.dropna()
df3.head()
```

Out[6]:

	Startup	Product	Funding	Event	Result	OperatingState
1	3DLT	3dlt.com	$630K	Disrupt NYC 2013	Contestant	Closed
3	3Dprintler	3dprintler.com	$1M	Disrupt NY 2016	Audience choice	Operating
5	5to1	5to1.com	$19.3M	TC50 2009	Contestant	Acquired
6	8 Securities	8securities.com	$29M	Disrupt Beijing 2011	Finalist	Operating
10	AdhereTech	adheretech.com	$1.8M	Hardware Battlefield 2014	Contestant	Operating

In [7]:
```
df3.loc[:,'Funds_in_million'] = df3['Funding'].apply(lambda x: float(x[1:-1])/1000 if
df3.head()
```

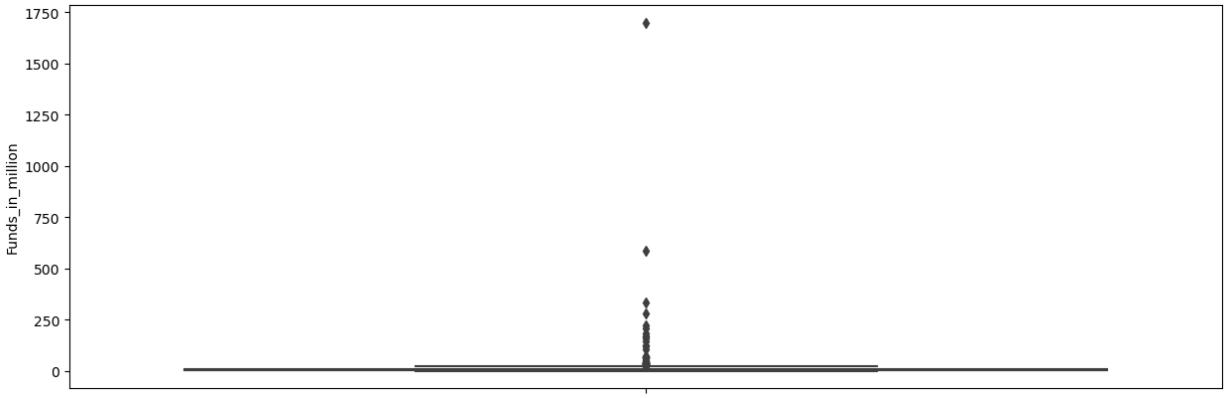
```
C:\Users\ABC\AppData\Local\Temp\ipykernel_8692\3442445927.py:1: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df3.loc[:,'Funds_in_million'] = df3['Funding'].apply(lambda x: float(x[1:-1])/1000
if x[-1] == 'K' else (float(x[1:-1])*1000 if x[-1] == 'B' else float(x[1:-1])));
```

Out[7]:

	Startup	Product	Funding	Event	Result	OperatingState	Funds_in_million
1	3DLT	3dlt.com	$630K	Disrupt NYC 2013	Contestant	Closed	0.63
3	3Dprintler	3dprintler.com	$1M	Disrupt NY 2016	Audience choice	Operating	1.00
5	5to1	5to1.com	$19.3M	TC50 2009	Contestant	Acquired	19.30
6	8 Securities	8securities.com	$29M	Disrupt Beijing 2011	Finalist	Operating	29.00
10	AdhereTech	adheretech.com	$1.8M	Hardware Battlefield 2014	Contestant	Operating	1.80

In [14]:
```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
fig, ax = plt.subplots(figsize=(15,5))
sns.boxplot(data=df3, y='Funds_in_million');
```



In [9]:
```
Q1 = df3['Funds_in_million'].quantile(0.25)
Q3 = df3['Funds_in_million'].quantile(0.75)
IQR = Q3 - Q1
print("The IQR of attribute Funds_in_million is:",IQR)
print('The number of outliers greater than the upper fence is:',(df3['Funds_in_millior
```

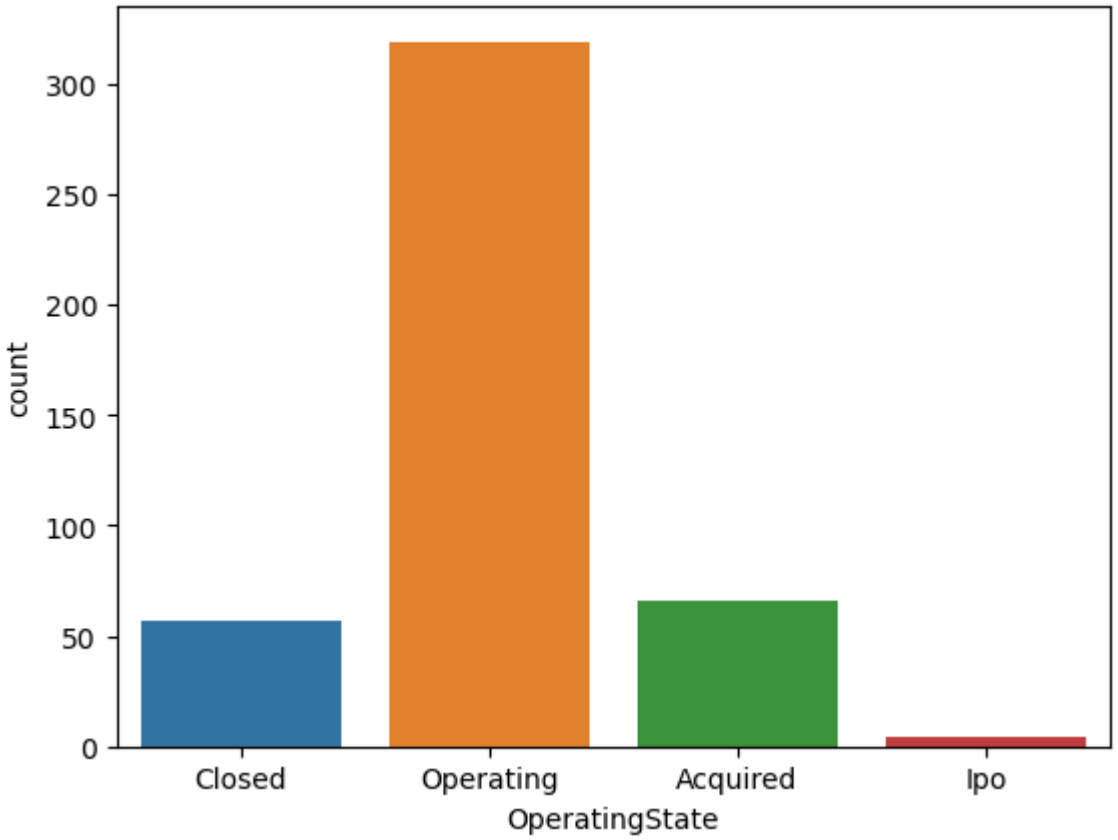
```
The IQR of attribute Funds_in_million is: 8.72975
The number of outliers greater than the upper fence is: 60
```

In [10]:
```
df3['OperatingState'].value_counts()
```

Out[10]:
```
Operating    319
Acquired      66
Closed        57
Ipo            4
Name: OperatingState, dtype: int64
```

In [15]:
```
sns.countplot(x='OperatingState', data=df3)
```

Out[15]:
```
<Axes: xlabel='OperatingState', ylabel='count'>
```



In [16]:
```
df3.groupby(['OperatingState']).sum()
```

```
C:\Users\ABC\AppData\Local\Temp\ipykernel_8692\2167407984.py:1: FutureWarning: The de
fault value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future versio
n, numeric_only will default to False. Either specify numeric_only or select only col
umns which should be valid for the function.
  df3.groupby(['OperatingState']).sum()
```

Out[16]:

OperatingState	Funds_in_million
Acquired	872.0510
Closed	185.7157
Ipo	551.1000
Operating	6080.8372

In [17]:
```
group1 = df3['OperatingState']=='Operating'
group1 = df3[group1]['Funds_in_million']
group2 = df3['OperatingState']=='Closed'
```

```
group2 = df3[group2]['Funds_in_million']
from scipy.stats import ttest_ind
t_statistic, p_value = ttest_ind(group1, group2)
print(t_statistic, p_value)
print ("two-sample t-test p-value=", p_value)
```

```
1.1382924515740138 0.25572701885629406
two-sample t-test p-value= 0.25572701885629406
```

In [18]:
```
df2.head()
```

Out[18]:

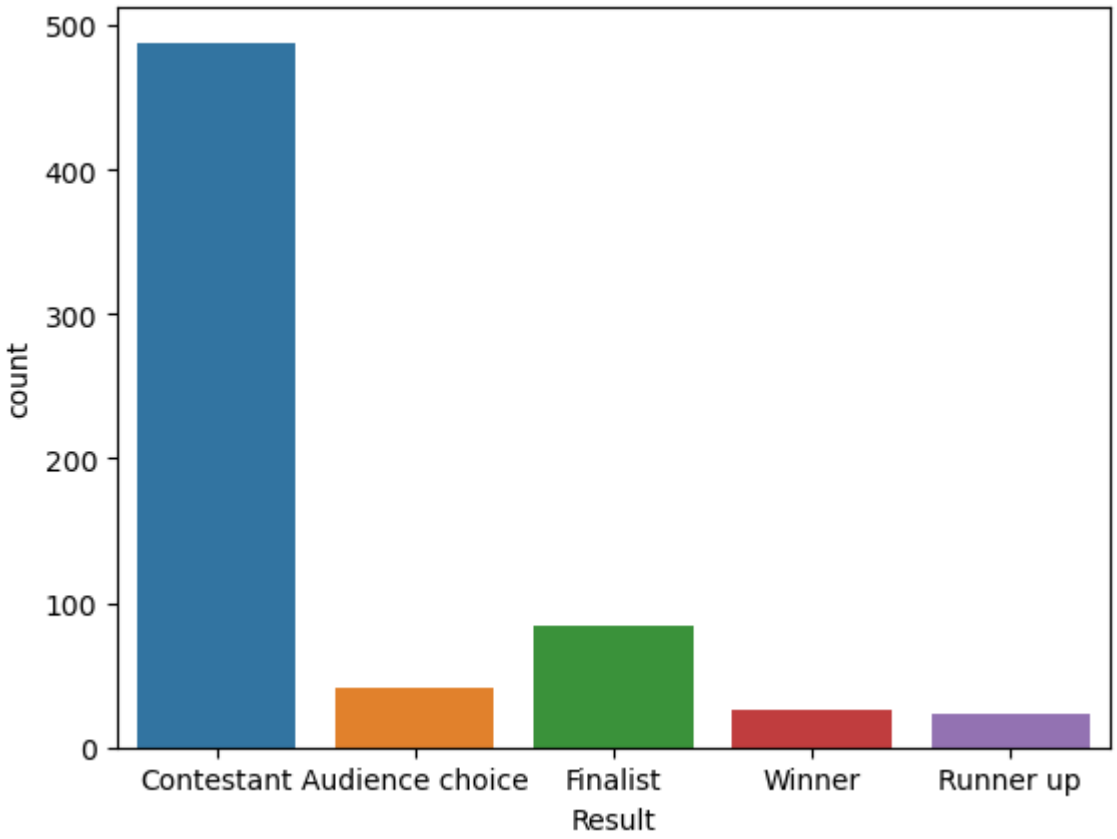
	Startup	Product	Funding	Event	Result	OperatingState
0	2600Hz	2600hz.com	NaN	Disrupt SF 2013	Contestant	Operating
1	3DLT	3dlt.com	$630K	Disrupt NYC 2013	Contestant	Closed
2	3DPrinterOS	3dprinteros.com	NaN	Disrupt SF 2016	Contestant	Operating
3	3Dprintler	3dprintler.com	$1M	Disrupt NY 2016	Audience choice	Operating
4	42 Technologies	42technologies.com	NaN	Disrupt NYC 2013	Contestant	Operating

In [19]:
```
df2['Result'].value_counts()
```

Out[19]:
```
Contestant         488
Finalist            84
Audience choice     41
Winner              26
Runner up           23
Name: Result, dtype: int64
```

In [20]:
```
sns.countplot(x='Result', data=df2)
```

Out[20]:
```
<Axes: xlabel='Result', ylabel='count'>
```



In [21]:
```
pd.crosstab(df2['OperatingState'],df2['Result'])
```

Out[21]:

OperatingState	Audience choice	Contestant	Finalist	Runner up	Winner
Acquired	0	62	15	2	7
Closed	9	90	7	0	0
Ipo	0	4	1	0	0
Operating	32	332	61	21	19

In [22]:
```
winners = 26
winner_opr = 19
print('Percentage of winners that are still operating:',round((winner_opr/winners)*100
```

```
Percentage of winners that are still operating: 73.08
```

In [23]:
```
contestant = 488
contestant_opr = 332
print('Percentage of contestants that are still operating:',round((contestant_opr/cont
```

```
Percentage of contestants that are still operating: 68.03
```

In [24]:
```
'Winner' and 'Contestant' are two categories in the attribute Result. We want to see i

Percentage of winners that are still operating: 73.08 %

Percentage of contestants that are still operating: 68.03 %

The proportions are different but are they statistically significant?
```

```
Null Hypothesis (H0) = The proportion of companies that are operating between winners

Alternative Hypothesis (Ha) = The proportion of companies that are operating between w
```

```
  Cell In[24], line 1
    'Winner' and 'Contestant' are two categories in the attribute Result. We want to
see if the proportion of operating companies in the Winner category is significantly
less than it is in the Contestant category.
                                           ^
SyntaxError: invalid syntax
```

In [25]:
```
from statsmodels.stats.proportion import proportions_ztest

stat, pval = proportions_ztest([winner_opr, contestant_opr] , [winners, contestant])

if pval < 0.05:
    print(f'With a p-value of {round(pval,4)} the difference is significant. aka |We r
else:
    print(f'With a p-value of {round(pval,4)} the difference is not significant. aka |
```

```
With a p-value of 0.5902 the difference is not significant. aka |We fail to reject th
e null|
```

In [26]:
```
df2 = df2[df2['Event'].str.contains('Disrupt')]
df2['EventYear'] = df2['Event'].str[-4:]
df2
```

Out[26]:

	Startup	Product	Funding	Event	Result	OperatingState	EventYear
0	2600Hz	2600hz.com	NaN	Disrupt SF 2013	Contestant	Operating	2013
1	3DLT	3dlt.com	$630K	Disrupt NYC 2013	Contestant	Closed	2013
2	3DPrinterOS	3dprinteros.com	NaN	Disrupt SF 2016	Contestant	Operating	2016
3	3Dprintler	3dprintler.com	$1M	Disrupt NY 2016	Audience choice	Operating	2016
4	42 Technologies	42technologies.com	NaN	Disrupt NYC 2013	Contestant	Operating	2013
...	...	...	...	...	...	...	...
653	ZAP!	zapreklam.com/	NaN	Disrupt EU 2014	Audience choice	Operating	2014
654	ZEFR	zefr.com	$62.1M	Disrupt NYC 2010	Contestant	Operating	2010
656	Zenefits	zenefits.com	$583.6M	Disrupt NYC 2013	Finalist	Operating	2013
660	Zula	zulaapp.com	$3.4M	Disrupt SF 2013	Audience choice	Operating	2013
661	Zumper	zumper.com	$31.5M	Disrupt SF 2012	Finalist	Operating	2012

465 rows × 7 columns

In [27]:
```
df2['EventYear'] = df2['EventYear'].apply(pd.to_numeric)
year_filter=df2['EventYear']>=2013
year_filter
```

Out[27]:
```
0       True
1       True
2       True
3       True
4       True
       ...
653     True
654    False
656     True
660     True
661    False
Name: EventYear, Length: 465, dtype: bool
```

In [28]:
```
year_filter = df2[year_filter]['Event']
year_filter
```

Out[28]:
```
0          Disrupt SF 2013
1         Disrupt NYC 2013
2          Disrupt SF 2016
3          Disrupt NY 2016
4         Disrupt NYC 2013
              ...
646    Disrupt London 2015
648    Disrupt London 2015
653        Disrupt EU 2014
656       Disrupt NYC 2013
660        Disrupt SF 2013
Name: Event, Length: 276, dtype: object
```