Monte Carlo Simulation to determine Project Contingency

- Projected Costs
- Schedule
- Iterations
- Contingency

In [69]:
```
# Import Libraries
import numpy as np
import matplotlib.pyplot as plt
```

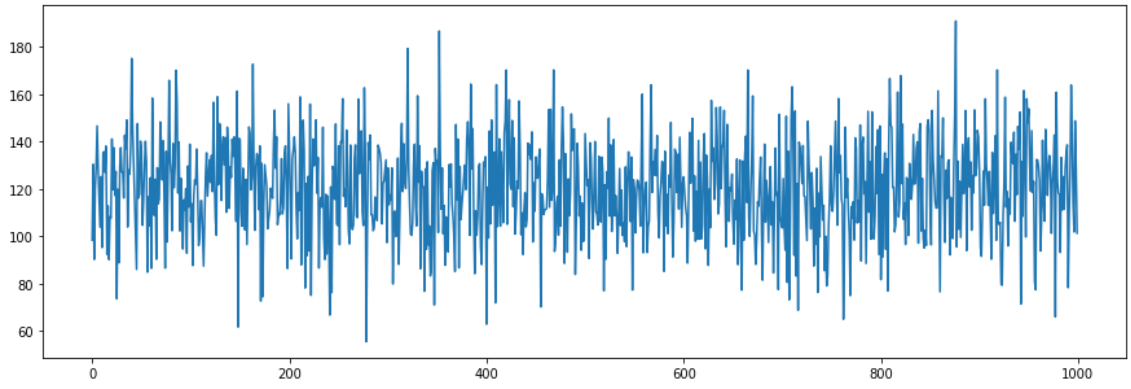
In [70]:
```
#Project Projected Costs

"""
Project costs based upon Work Breakdown Schedule (WBS) i.e:
1.0 Conceptual Engineering
2.0 Detailed Engineering
3.0 Procurement
4.0 Construction
5.0 Starup
"""
#In millions
projected_cost = 120

#Schedule Developement
"""
One example of project schedule levels (AACE, PMI Client organizations and EPC's
may
have different definitions) Some models have level 1 as detailed and 4 - 5 as
pre-feasability study.

Level 1 & 2 pre-feasibility studies
Level 3 High level detail Starting CPM (Critical Path method)
It should include major elements of design, engineering, procurement, constructio
n,
testing, commissioning and/or start-up
Level 4 Execution Schedule, also called a Project Working Level Schedule.
Level 5 Detailed Schedule.

Standard deviation
Level 1 & 2 %40 to %80 cost overrun (some industries this can be as high as 200%)
Recommend add 40% and run 40% Standard deviation
Level 3 20%
Level 4 10%
Level 5 5%

"""
# Use information to develope stdev example will use a level 3 schedule
schedule_stdev = 20

#Iterations number of simulated costs (use 500 as a minimum)
iterations = 1000
```

In [71]:
```
project = np.random.normal(projected_cost, schedule_stdev, iterations)
project
```

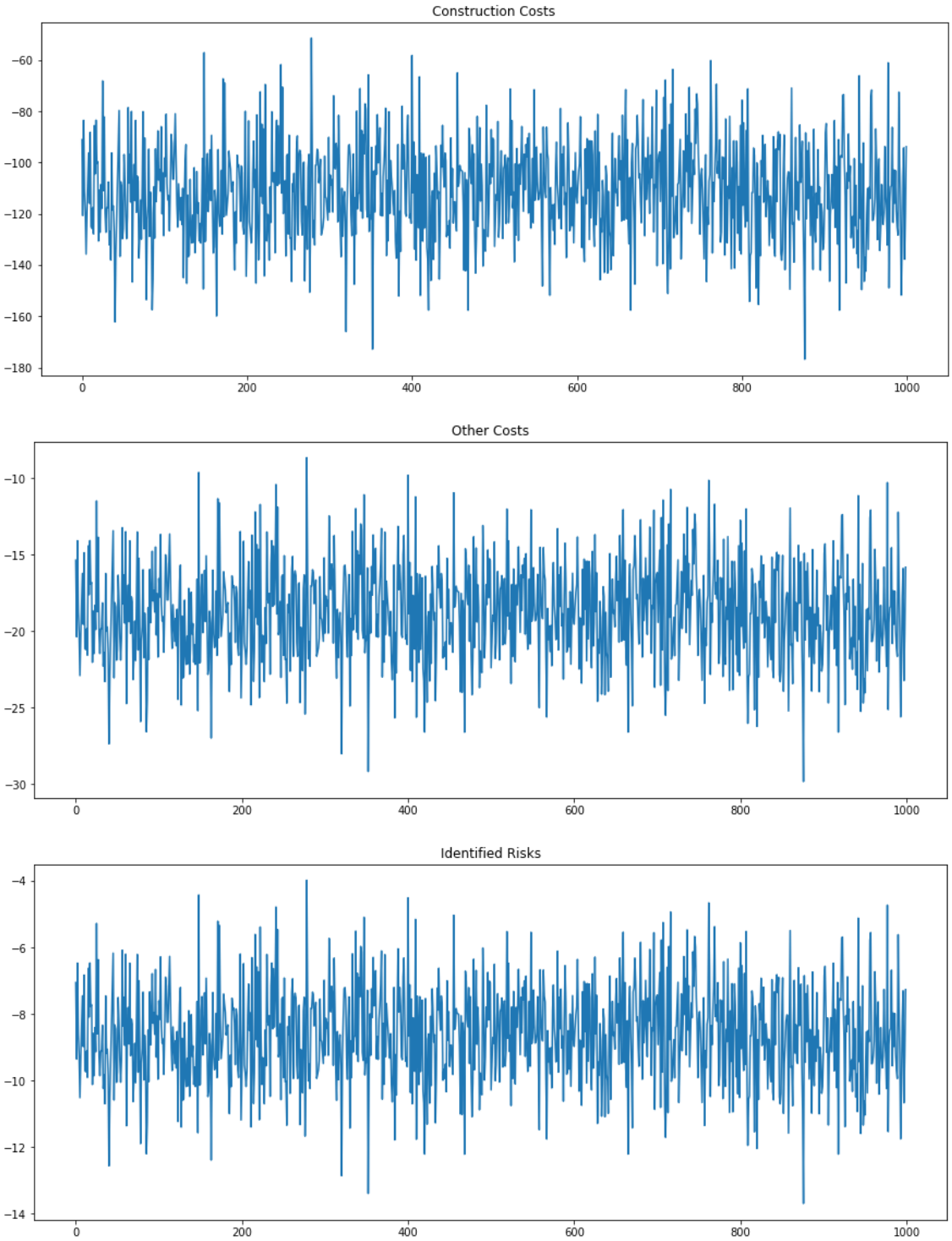
Out[71]:
```
array([ 98.40943426, 130.43430934,  90.27247915, 120.15827419,
       129.20180604, 146.65478592, 129.24671775, 116.95701805,
       103.9642008 , 125.24682218,  95.27151892, 135.0340705 ,
       135.7947325 , 127.07418117, 138.13258803,  92.4492642 ,
       112.8476701 ,  90.21851688, 108.36091333, 107.69270942,
       141.12600302, 119.71055157, 137.43579967, 117.30458033,
       127.4125919 ,  73.62916872, 119.89548904,  88.84272301,
       129.08003461, 137.44163994, 127.15921089, 127.14900155,
       116.23860966, 142.71025888, 137.08602464, 149.14306489,
       103.98562935, 128.00441184, 126.34744381, 137.28902172,
       175.19428181, 146.44013427, 130.8379454 , 117.07279885,
        98.42206873,  86.08098008, 147.62023321, 116.1496153 ,
       117.9101435 , 140.21945851, 135.88963335, 104.65500158,
       115.41535373, 130.25791142, 140.16877076, 130.98512602,
        84.85965252, 116.70010571, 104.54451015, 124.71892809,
        86.48461352, 158.39141011, 120.07500767, 108.85477551,
       124.14397334,  90.31737362, 129.01999466, 113.76554284,
       115.8916276 , 148.32768436, 139.42126454, 123.8571011 ,
       140.34491375, 117.2359125 ,  86.5964191 , 135.96340561,
        97.64094956, 126.87395221, 165.89909327, 132.00522652,
       126.18959289, 102.3841641 , 139.1973898 , 120.55252843,
       129.40289376, 170.1341779 , 157.04054544, 118.31319053,
       139.90718382, 102.29528864, 124.53196394, 112.87354928,
        94.69112789, 115.44740346, 111.08869074, 120.12475702,
        92.91246502, 129.68536147, 112.42240838, 138.91797763,
       106.1251598 , 113.84725902,  87.65638734, 116.81143993,
       123.82889725, 122.04141229, 136.88346692, 117.96937952,
        96.16973247,  99.79541306, 115.06054971, 115.03413965,
       100.41354502,  87.45412244, 107.04560536, 125.8283038 ,
       135.29338317, 117.00940772, 125.13693176, 132.34874956,
       122.81605097, 134.18573098, 118.88929383, 156.65200371,
       123.71038575, 107.29699214, 100.48385589, 158.96731749,
       121.77522644, 147.67798194, 146.97848752, 115.13737534,
       137.47333152, 141.99963535, 130.17498235, 141.69127229,
       110.17463968, 146.07593485, 135.5883846 , 111.81475936,
       129.30820591, 124.93606729, 141.06532999, 133.82309689,
       142.00071543, 127.37892811, 106.34114474, 161.3312208 ,
        61.73145055, 141.3704838 , 140.98773556, 109.95823512,
       104.27865511, 128.7631831 , 119.79887721, 102.69948849,
       124.08992417,  96.57383584, 121.67991351, 146.18819539,
       143.57737713, 119.71060919, 124.77647982, 172.75028215,
       128.26141094, 102.57016251, 129.36894867, 134.93203449,
       133.77963607, 111.3117858 , 138.22462672,  72.72517106,
       130.86549631,  74.51192225, 113.75820513, 130.27337184,
       125.92440611, 121.18207956, 103.22885286, 107.46402528,
       110.87232579, 120.35020148, 118.01978469, 116.09761558,
       139.5258499 , 153.34419599, 128.80734217, 142.06373686,
       104.96953112, 109.80874737, 109.01336079, 115.52653903,
       132.81113352, 109.46496701, 112.25232712, 134.90728936,
       138.21087539, 120.7603639 ,  86.42768617, 155.99908075,
       137.2336286 , 118.30148739,  90.48999246, 133.02376222,
       136.40884497, 141.97284794, 135.63234268, 112.18248571,
        99.01606351, 118.51740415, 111.07705382, 158.89807681,
        87.95953818, 128.76138486, 149.06195209, 136.04288107,
        78.22129382, 122.31686076,  91.90088572, 125.30602483,
        93.87836054, 155.85944109,  75.12307969, 120.48947961,
       141.14667022, 129.85832936, 130.4421667 , 149.18994616,
       118.25363798, 133.30675368,  86.63081539, 114.07020387,
       116.31115333, 112.32052862, 146.0938616 ,  99.88429199,
        90.24702029, 117.76830778,  92.52526079, 117.04272061,
       101.94301067,  66.76192533, 120.93256913,  76.25786055,
       129.56262529, 127.66682989, 115.65418374, 147.4545987 ,
       105.25583135, 104.60288353, 138.15499143,  96.51093461,
       139.98343388, 139.78470826, 158.18855049, 116.95010994,
       120.11507458, 112.58493191, 144.87409478, 130.96751263,
```

In [72]:
```
plt.figure(figsize = (15,5))
plt.plot(project)
```

Out[72]:
```
[<matplotlib.lines.Line2D at 0x11e087cf8>]
```



In [86]:
```
#Construction costs to mechanical completion 70% of cost with a standard deviatio
n of 20%
#you can break down costs indetail, risk etc. In this section

construction_costs = - (project * np.random.normal(.7,0.2))
other_costs = - (project * np.random.normal(.3,0.2))
# Use risk register for project amount (if risks are positive use + sign)
identified_risks = - (project * np.random.normal(.03,0.1))

plt.figure(figsize=(15, 6))
plt.plot(construction_costs)
plt.title('Construction Costs')
plt.show()

plt.figure(figsize=(15, 6))
plt.title('Other Costs')
plt.plot(other_costs)
plt.show()

plt.figure(figsize=(15, 6))
plt.title('Identified Risks')
plt.plot(identified_risks)
plt.show()
```



Mean and Standard Deviation

In [87]:
```
construction_costs.mean()
```

Out[87]:
```
-111.24979069844763
```

In [88]:
```
construction_costs.std()
```

Out[88]: 18.999284959101843

In [89]:
```
other_costs.mean()
```

Out[89]: -18.770474100282545

In [90]:
```
other_costs.std()
```

Out[90]: 3.205629278129377

In [91]:
```
identified_risks.mean()
```

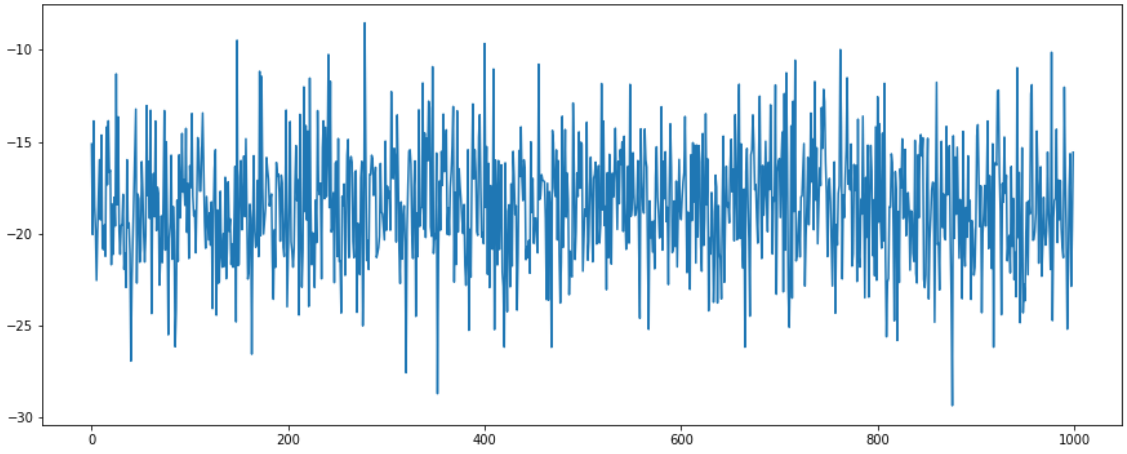
Out[91]: -8.61683459421484

In [92]:
```
identified_risks.std()
```

Out[92]: 1.4715865519665998

Contingency

In [93]:
```
contingency = project + (construction_costs + other_costs + identified_risks)
contingency

plt.figure(figsize=(15,6))
plt.plot(contingency)
plt.show()
```



In [94]:
```
max(contingency)
```

Out[94]: -8.541013090668798

In [95]:
```
min(contingency)
```

Out[95]: -29.348531545832458

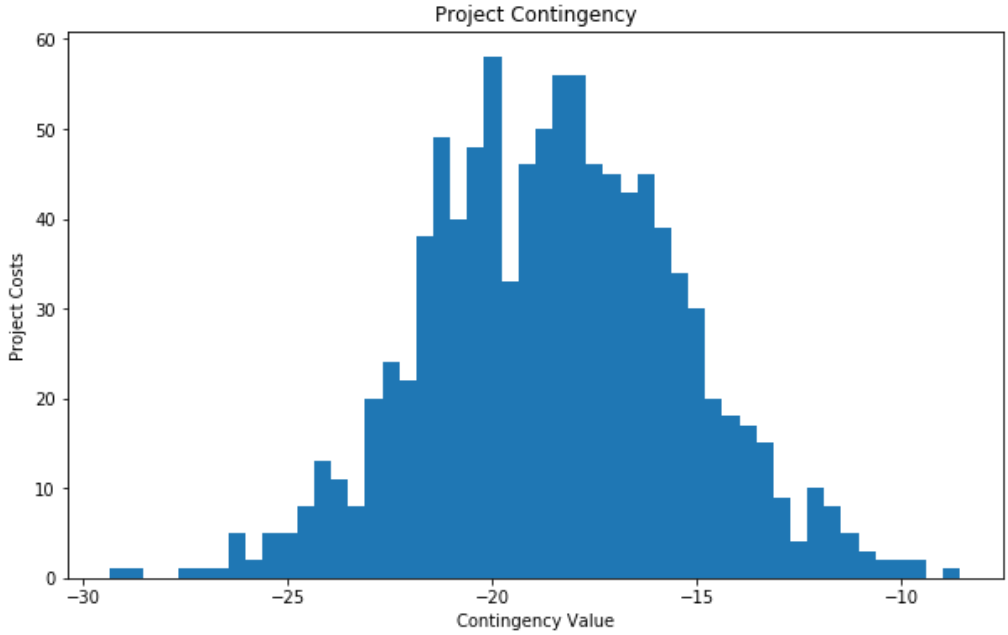
In [96]:
```
contingency.mean()
```

Out[96]: -18.467850452441787

In [97]:
```
contingency.std()
```

Out[97]:
```
3.153947087227334
```

In [101]:
```
plt.figure(figsize =(10,6))
plt.title('Project Contingency')
plt.xlabel('Contingency Value')
plt.ylabel('Project Costs')
plt.hist(contingency, bins =50);
plt.show()
```



Recomendations

Project Exposure

The project could expose the organization to a maximum project overrun of 29.34 million dollars.

Contingency Recomendation

Based upon the Monte Carlo simulation a recommendation of 18.46 million contingency be set up for this project.

In [ ]:
```
```