In [1]:
```
#Load Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
print("Done")
```

```
Done
```

In [2]:
```
#Load Data
leuanalysisNewdata = pd.read_csv('./leuanalysisNew.csv')
leuanalysisNewdata.head()
```

Out[2]:

	CELL	SMEAR	INFIL	LI	BLAST	TEMP	REMISS
0	0.8	0.83	0.66	1.9	1.10	1.00	1
1	0.9	0.36	0.32	1.4	0.74	0.99	1
2	0.8	0.88	0.70	0.8	0.18	0.98	0
3	1.0	0.87	0.87	0.7	1.05	0.99	0
4	0.9	0.75	0.68	1.3	0.52	0.98	1

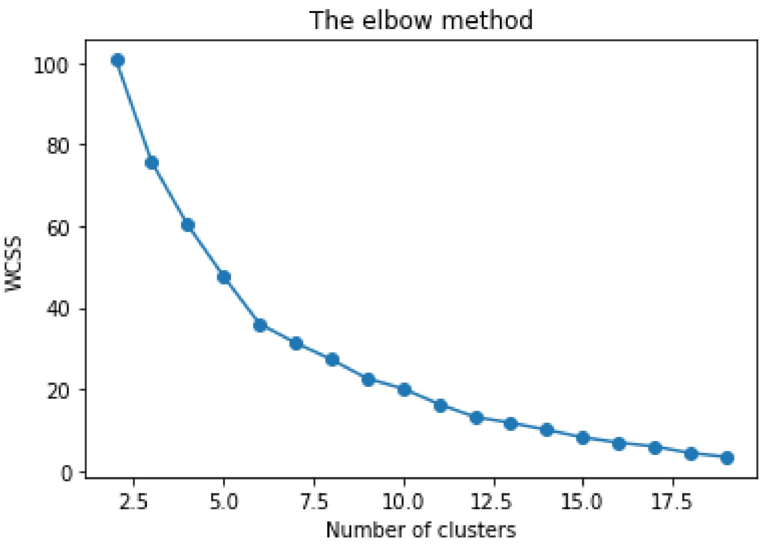
In [3]:
```
#Create x (we ignore the y variable)
x = leuanalysisNewdata.drop('REMISS', axis=1).to_numpy()

#Scale the Data
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
data_transformed=sc.fit_transform(x)
print("Done")
```

```
Done
```

In [15]:
```
#Determine optimum number of clusters
wcss = []
for i in range(2, 20):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10, r
    kmeans.fit(data_transformed)
    wcss.append(kmeans.inertia_)

#Plot Elbow Method
plt.plot(range(2,20), wcss,marker='o')
plt.title('The elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS') #within cluster sum of squares
plt.show()
```

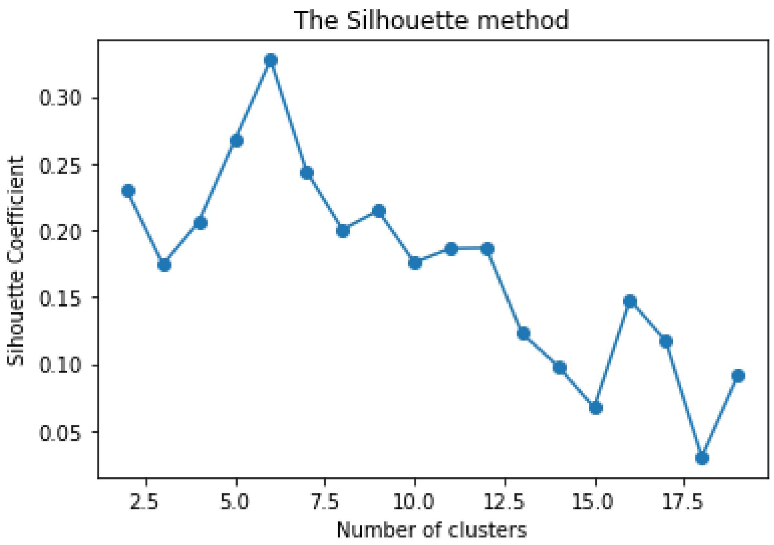


In [7]:

```
#Create Silhouette Coefficients
scores= []
for n_cluster in range(2, 20):
       kmeans = KMeans(n_clusters=n_cluster).fit(x)
       label = kmeans.labels_
       sil_coeff = silhouette_score(data_transformed, label, metric='euclidean')
       scores.append(sil_coeff)
       print('For n_clusters= {}, The Silhouette Coefficient is {}'.format(n_cluster,

#Plot Silhouette
plt.plot(range(2,20), scores, marker='o')
plt.title('The Silhouette method')
plt.xlabel('Number of clusters')
plt.ylabel('Sihouette Coefficient')
plt.show()
```

```
For n_clusters= 2, The Silhouette Coefficient is 0.22972085065748718
For n_clusters= 3, The Silhouette Coefficient is 0.1743850640966066
For n_clusters= 4, The Silhouette Coefficient is 0.20631955740350025
For n_clusters= 5, The Silhouette Coefficient is 0.26748741305184426
For n_clusters= 6, The Silhouette Coefficient is 0.32761133000558906
For n_clusters= 7, The Silhouette Coefficient is 0.24421391153398997
For n_clusters= 8, The Silhouette Coefficient is 0.20050259137980767
For n_clusters= 9, The Silhouette Coefficient is 0.21469198060593014
For n_clusters= 10, The Silhouette Coefficient is 0.1758985486861702
For n_clusters= 11, The Silhouette Coefficient is 0.18633903493419923
For n_clusters= 12, The Silhouette Coefficient is 0.18675754378291073
For n_clusters= 13, The Silhouette Coefficient is 0.12278317007865666
For n_clusters= 14, The Silhouette Coefficient is 0.09842876593399202
For n_clusters= 15, The Silhouette Coefficient is 0.06748255197358485
For n_clusters= 16, The Silhouette Coefficient is 0.14791053652001043
For n_clusters= 17, The Silhouette Coefficient is 0.11685274974125172
For n_clusters= 18, The Silhouette Coefficient is 0.03039553247699224
For n_clusters= 19, The Silhouette Coefficient is 0.09132129773532019
```



In [8]:

```
#Create KMeans with 2 Clusters

#Apply KMeans clustering
kmeans = KMeans(n_clusters = 2, init = 'k-means++', max_iter = 300, n_init = 10, rando
y_kmeans = kmeans.fit_predict(data_transformed)

#Add Prediction column to dataset
dataset2= leuanalysisNewdata.drop('REMISS', axis=1)
dataset2['New Cluster']=kmeans.labels_
dataset2['New Cluster'] = dataset2['New Cluster'].map({0:'Cluster 1', 1:'Cluster 2'})
dataset2.head()
```

Out[8]:

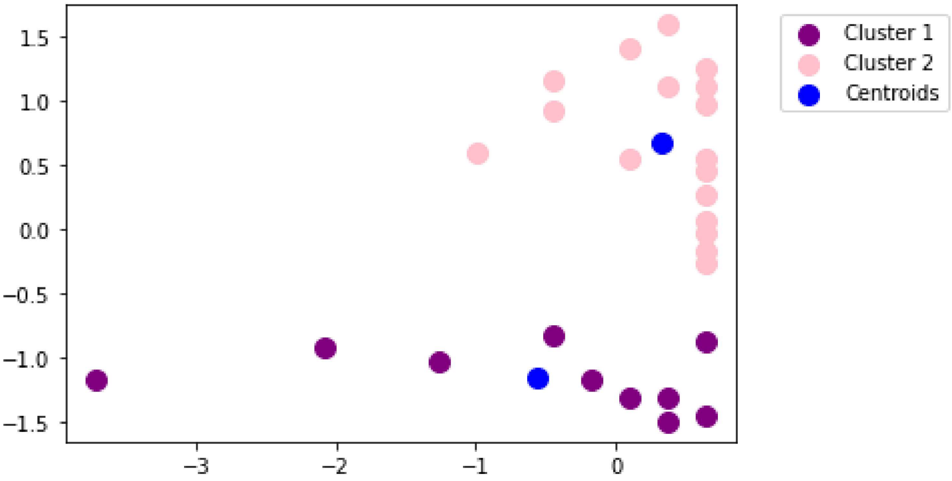
	CELL	SMEAR	INFIL	LI	BLAST	TEMP	New Cluster
0	0.8	0.83	0.66	1.9	1.10	1.00	Cluster 2
1	0.9	0.36	0.32	1.4	0.74	0.99	Cluster 1
2	0.8	0.88	0.70	0.8	0.18	0.98	Cluster 2
3	1.0	0.87	0.87	0.7	1.05	0.99	Cluster 2
4	0.9	0.75	0.68	1.3	0.52	0.98	Cluster 2

In [10]:

```
#Plot of 2 Clusters
plt.scatter(data_transformed[y_kmeans == 0, 0], data_transformed[y_kmeans == 0, 1], s
plt.scatter(data_transformed[y_kmeans == 1, 0], data_transformed[y_kmeans == 1, 1], s
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,1], s = 100, c =
#Legend
plt.legend(bbox_to_anchor=(1.05, 1), loc=2)
```

Out[10]:

```
<matplotlib.legend.Legend at 0x161b879b400>
```


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−3
−2
−1
0
Cluster 1
Cluster 2
Centroids