Analyse der Webanwendung "PetClinic"

Priorisierung von Umbauarbeiten nach Nutzungsgrad

Technische Vorbereitung: Laden der Analysewerkzeuge

In [10]:
```
import pandas as pd
import py2neo
graph = py2neo.Graph(password="password")
```

Aggregation der Messwerte nach Subdomänen

In [11]:
```
query = """
MATCH
  (t:Type)-[:BELONGS_TO]->(s:Subdomain),
  (t)-[:HAS_CHANGE]->(ch:Change),
  (t)-[:HAS_MEASURE]->(co:Coverage)
OPTIONAL MATCH
  (t)-[:HAS_BUG]->(b:BugInstance)
RETURN
  s.name as ASubdomain,
  COUNT(DISTINCT t) as Types,
  COUNT(DISTINCT ch) as Changes,
  AVG(co.ratio) as Coverage,
  COUNT(DISTINCT b) as Bugs,
  SUM(DISTINCT t.lastMethodLineNumber) as Lines
ORDER BY Coverage ASC, Bugs DESC
"""
```

Ergebnisse nach Subdomänen

In [12]:
```
result = pd.DataFrame(graph.data(query))
result
```

Out[12]:

	ASubdomain	Bugs	Changes	Coverage	Lines	Types
0	Vet	0	75	0.170000	313	5
1	Visit	0	90	0.368056	472	6
2	Pet	1	167	0.490069	746	11
3	Owner	3	94	0.506932	531	4
4	crossfunctional	2	53	0.589231	268	5
5	Clinic	0	26	0.888889	110	1
6	Person	0	5	1.000000	53	1
7	Specialty	0	4	1.000000	28	1

Umbenennung nach geläufigen Begriffen

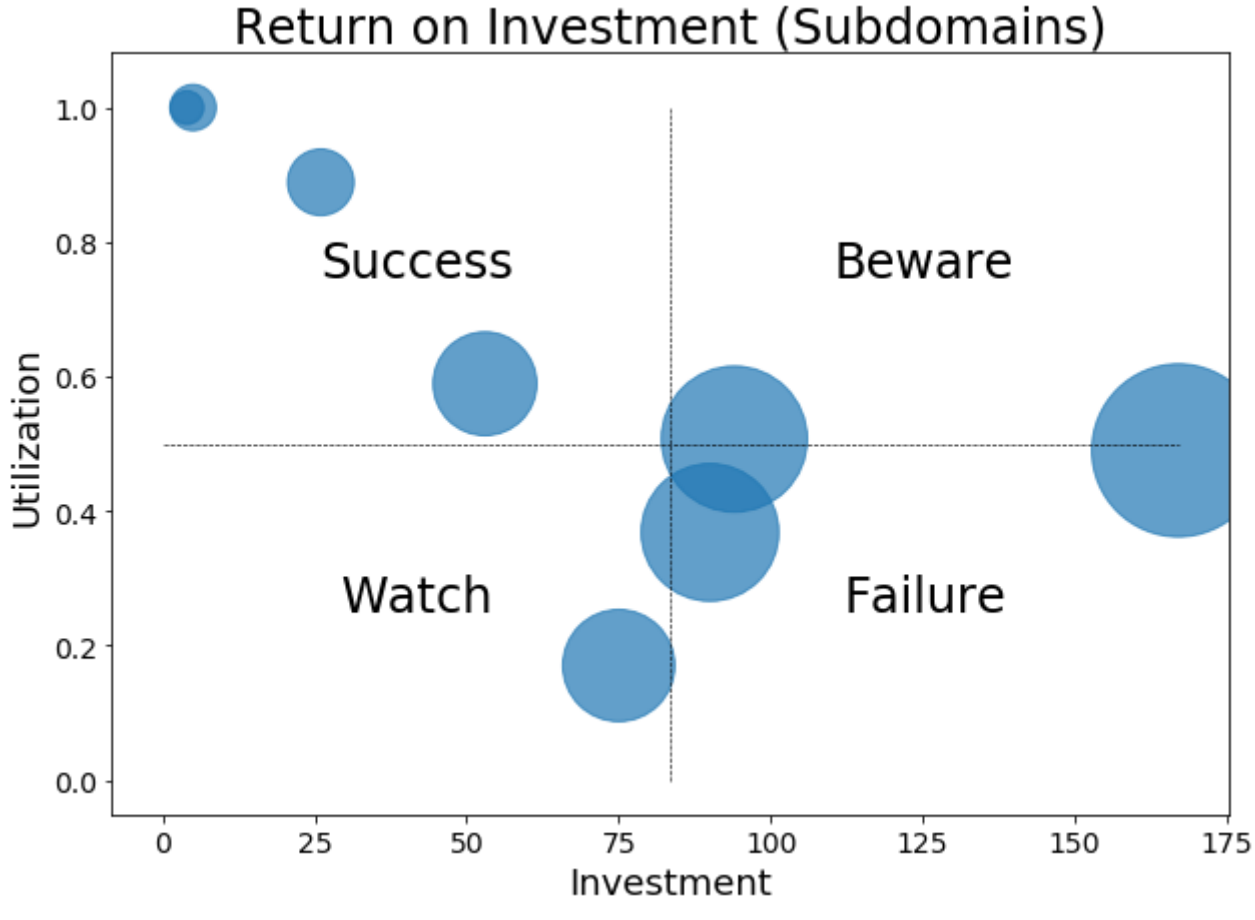
In [13]:
```
plot_data = result.copy()
plot_data = plot_data.rename(
    columns= {
        "Changes" : "Investment",
        "Coverage" : "Utilization",
        "Lines" : "Size"})
plot_data
```

Out[13]:

	ASubdomain	Bugs	Investment	Utilization	Size	Types
0	Vet	0	75	0.170000	313	5
1	Visit	0	90	0.368056	472	6
2	Pet	1	167	0.490069	746	11
3	Owner	3	94	0.506932	531	4
4	crossfunctional	2	53	0.589231	268	5
5	Clinic	0	26	0.888889	110	1
6	Person	0	5	1.000000	53	1
7	Specialty	0	4	1.000000	28	1

Vier-Felder-Matrix zur Priorisierung nach Subdomänen

In [15]:
```
plot_portfolio_diagramm(plot_data, "Subdomains")
```



Aggregation der Messwerte nach technischen Aspekten

In [16]:
```
query = """
MATCH
  (t:Type)-[:IS_A]->(ta:TechnicalAspect),
  (t)-[:HAS_CHANGE]->(ch:Change),
  (t)-[:HAS_MEASURE]->(co:Coverage)
OPTIONAL MATCH
  (t)-[:HAS_BUG]->(b:BugInstance)
RETURN
  ta.name as ATechnicalAspect,
  COUNT(DISTINCT t) as Types,
  COUNT(DISTINCT ch) as Investment,
  AVG(co.ratio) as Utilization,
  COUNT(DISTINCT b) as Bugs,
  SUM(DISTINCT t.lastMethodLineNumber) as Size
ORDER BY Utilization ASC, Bugs DESC
"""
```

Ergebnisse nach technischen Aspekten

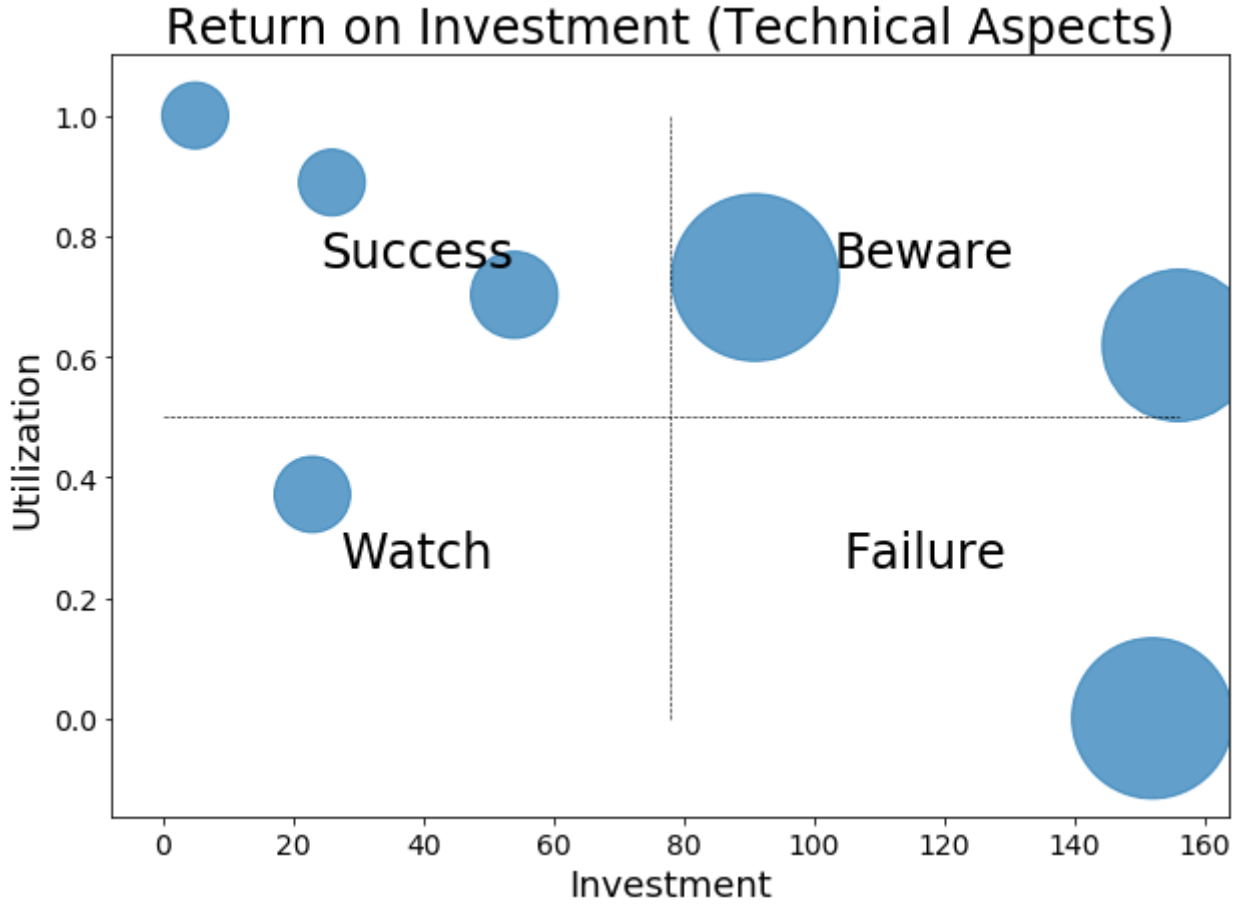
In [17]:
```
result = pd.DataFrame(graph.data(query))
result
```

Out[17]:

	ATechnicalAspect	Bugs	Investment	Size	Types	Utilization
0	jdbc	1	152	644	8	0.000000
1	util	2	23	144	2	0.371429
2	web	2	156	576	7	0.618619
3	jpa	0	54	188	4	0.702501
4	model	1	91	696	10	0.731240
5	service	0	26	110	1	0.888889
6	petclinic	0	5	110	1	1.000000

Vier-Felder-Matrix zur Priorisierung nach technischen Aspekten

In [18]:
```
plot_portfolio_diagramm(result, "Technical Aspects")
```



Ende Demo