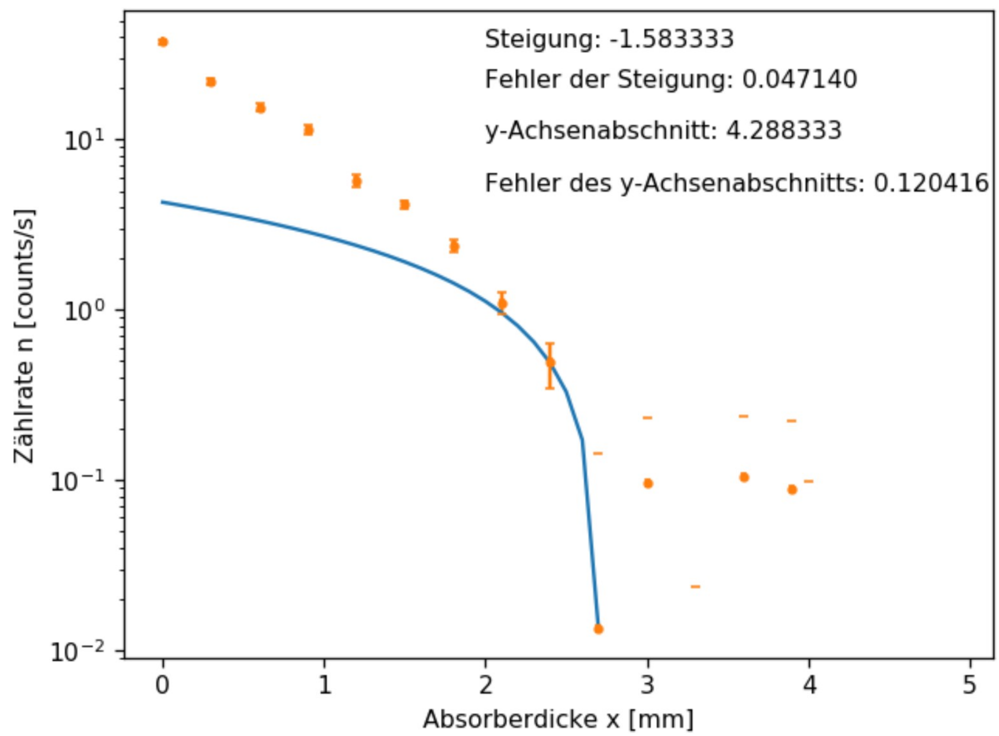
In [54]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
%matplotlib notebook

t1=np.array([30, 30, 30, 30, 30, 120, 120, 120, 120, 120, 120, 120, 120, 120, 300])
x=np.array([0.0, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3.0, 3.3, 3.6, 3.9, 4
.0])
dx=np.array([0.0, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01,
0.01, 0.01, 0.01])
N1=np.array([1172, 701, 508, 386, 217, 673, 462, 307, 233, 176, 186, 162, 187, 185,
436])
dN1=(N1)**(1/2)
n1=N1/t1
dn1=dN1/t1
print("n1=",n1)
print("dn1=",dn1)

n0B=n1[-1]
dn0B=dn1[-1]

diff=n1-n0B
ddiff=(dn1**2+dn0B**2)**(1/2)
print("diff=",diff)
print("ddiff=",ddiff)

A=np.arange(0,5,0.1)
def linear (x, a, b):
    return a*x+b

popt1, pcov1=curve_fit(linear, xdata=x[8:-5], ydata=diff[8:-5], sigma=dx[8:-5], abs
olute_sigma=True)
plt.plot(A,linear(A, *popt1), '-')
plt.errorbar(x, diff, yerr=ddiff, xerr=dx, fmt='.', capsize=2)
plt.yscale('log')
plt.text(2,35,'Steigung: %f' % popt1[0] )
plt.text(2,20,'Fehler der Steigung: %f' % np.sqrt(pcov1[0][0]))
plt.text(2,10,'y-Achsenabschnitt: %f' % popt1[1] )
plt.text(2,5,'Fehler des y-Achsenabschnitts: %f' % np.sqrt(pcov1[1][1]))
plt.xlabel('Absorberdicke x [mm]')
plt.ylabel('Zählrate n [counts/s]')
plt.show()

s=popt1[0]
ds=np.sqrt(pcov1[0][0])
y0=popt1[1]
dy0=np.sqrt(pcov1[1][1])

x0=-y0/s
dx0=((dy0/s)**2+(y0/(s**2)*ds)**2)**(1/2)
print("x0=",x0)
print("dx0=",dx0)
```

```
n1= [ 39.06666667  23.36666667  16.93333333  12.86666667   7.23333333
   5.60833333   3.85         2.55833333   1.94166667   1.46666667   1.55
   1.35         1.55833333   1.54166667   1.45333333]
dn1= [ 1.14114952  0.88254682  0.75129518  0.65489609  0.49103066  0.21618536
  0.17911821  0.1460118   0.12720281  0.11055416  0.11365151  0.10606602
  0.11395662  0.11334559  0.06960204]
diff= [  3.76133333e+01   2.19133333e+01   1.54800000e+01   1.14133333e+01
   5.78000000e+00   4.15500000e+00   2.39666667e+00   1.10500000e+00
   4.88333333e-01   1.33333333e-02   9.66666667e-02  -1.03333333e-01
   1.05000000e-01   8.83333333e-02   0.00000000e+00]
ddiff= [ 1.14327016  0.88528715  0.75451235  0.65858434  0.49593906  0.22711353
  0.19216602  0.16175255  0.145       0.13063945  0.13327082  0.12686388
  0.1335311   0.13301002  0.09843215]
```



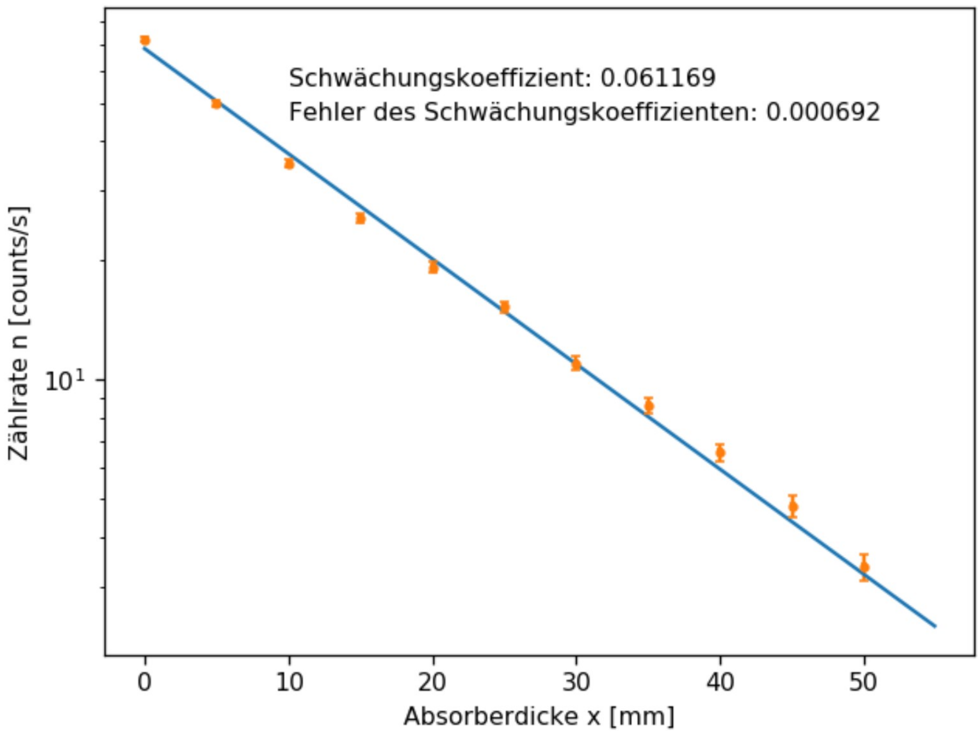

```
x0= 2.70842105263
dx0= 0.110843833878
```

In [55]:
```
roh=2.6989 #Wikipedia
RES=0.130
Rges=roh*x0*0.1+RES
dRges=roh*dx0*0.1

print("Rges=",Rges)
print("dRges=",dRges)
```

```
Rges= 0.860975757895
dRges= 0.0299156423254
```

In [56]:
```
x2=np.array([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50])
dx2=np.array([0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1])
N2=np.array([4366, 3011, 2135, 1560, 1177, 939, 682, 539, 415, 311, 225])
dN2=N2**(1/2)
n2=N2/60
dn2=dN2/60
print("n2=",n2)
print("dn2=",dn2)
N0=115
dN0=N0**(1/2)
n0=N0/300
dn0=dN0/300
print("n0=",n0)
print("dn0=",dn0)

diff2=n2-n0
ddiff2=(dn2**2+dn0**2)**(1/2)
print("diff2=",diff2)
print("ddiff2=",ddiff2)
def exp(x, a, b):
    return b*np.exp(-a*x)

B=np.arange(0,55,0.1)
popt2, pcov2=curve_fit(exp, xdata=x2, ydata=diff2, p0=[0.5,72], sigma=ddiff2, absol
ute_sigma=True, maxfev=3000000)
plt.plot(B,exp(B, popt2[0], popt2[1]), '-')
plt.errorbar(x2, diff2, yerr=ddiff2, xerr=dx2, fmt='.', capsize=2)
plt.yscale('log')
plt.text(10,55,'Schwächungskoeffizient: %f' % popt2[0] )
plt.text(10,45,'Fehler des Schwächungskoeffizienten: %f' % np.sqrt(pcov2[0][0]))
plt.xlabel('Absorberdicke x [mm]')
plt.ylabel('Zählrate n [counts/s]')
plt.show()

mü=popt2[0]
dmü=np.sqrt(pcov2[0][0])
roh=11.342
V=mü*10/roh
dV=dmü*10/roh
print("V=",V)
print("dV=",dV)
```

```
n2= [ 72.76666667  50.18333333  35.58333333  26.          19.61666667  15.65
  11.36666667   8.98333333   6.91666667   5.18333333   3.75       ]
dn2= [ 1.1012619   0.91454299  0.770101    0.65828059  0.57179056  0.51071845
  0.43525216  0.38693956  0.33952581  0.29391987  0.25       ]
n0= 0.38333333333333336
dn0= 0.03574601764921203
diff2= [ 72.38333333  49.8         35.2         25.61666667  19.23333333
  15.26666667  10.98333333   8.6          6.53333333   4.8          3.36666667]
ddiff2= [ 1.10184189  0.91524132  0.77093017  0.65925042  0.57290682  0.51196788
  0.43671755  0.38858718  0.34140234  0.29608557  0.25254263]
```




```
V= 0.0539312375715
dV= 0.000609769446765
```

In [57]:
```
E=0.04
r=0.7
dr=0.01
d=np.array([5.0, 10.0, 20.0])
dd=np.array([0.2, 0.2, 0.2])
N3=np.array([45295, 12824, 3325])
dN3=N3**(1/2)
n3=N3/60
dn3=dN3/60
N0=115
dN0=N0**(1/2)
n0=N0/300
dn0=N0/300
n31=(n3-n0)
dn31=((dn3)**2+(dn0)**2)**(1/2)
print("n31=",n31)
print("dn31=",dn31)
n32=n31/2
dn32=dn31/2
print("n32=",n32)
print("dn32=",dn32)
A=4*n32*d**2/(E*r**2)
dA=((4*dn32*d**2/(E*r**2))**2+(8*n32*d*dd/(E*r**2))**2+(8*n32*d**2*dr/(E*r**3))**2)
**(1/2)
print("A=",A)
print("dA=",dA)
```

```
n31= [ 754.53333333  213.35         55.03333333]
dn31= [ 3.56775684  1.92591969  1.03467655]
n32= [ 377.26666667  106.675        27.51666667]
dn32= [ 1.78387842  0.96295985  0.51733827]
A= [ 1924829.93197279  2177040.81632653  2246258.50340136]
dA= [ 163765.41481169  108804.44120743   88998.355315  ]
```

In [58]:
```
l=4
Akorr=4*n32*(d+l/2)**2/(E*r**2)
dAkorr=((4*dn32*(d+l/2)**2/(E*r**2))**2+(8*n32*(d+l/2)*dd/(E*r**2))**2+(8*n32*(d+l/
2)**2*dr/(E*r**3))**2)**(1/2)
k1=Akorr/A
dk1=((dAkorr/A)**2+(Akorr/(A**2)*dA)**2)**(1/2)
print("Akorr=",Akorr)
print("dAkorr=",dAkorr)
print("k1=",k1)
print("dk1=",dk1)
```

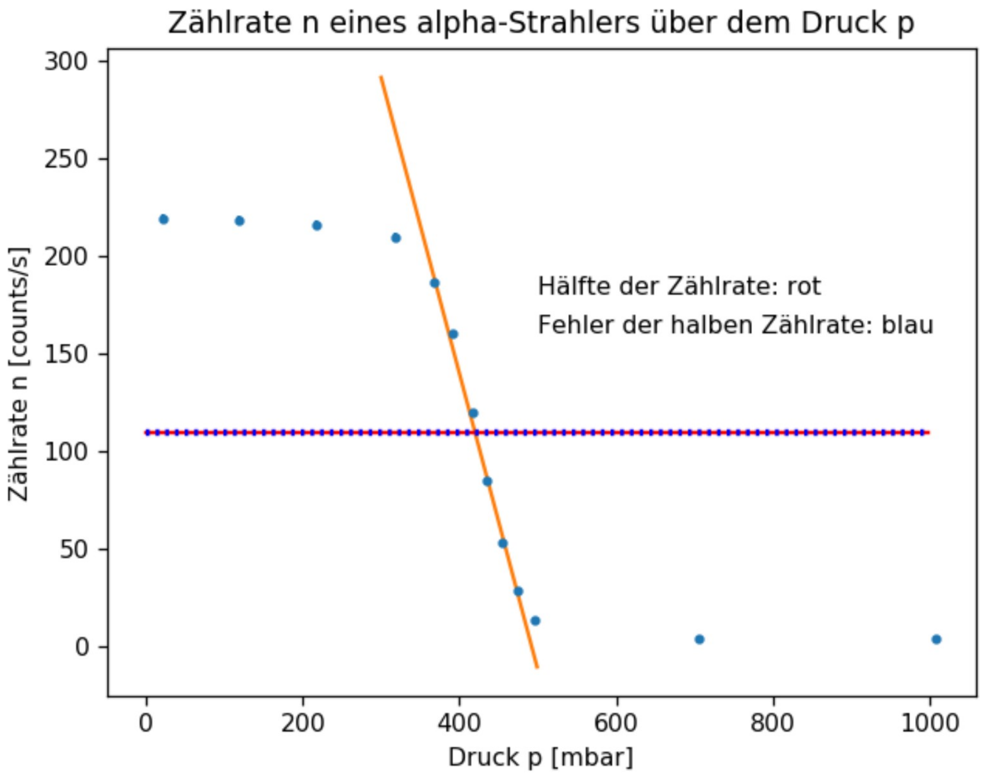
```
Akorr= [ 3772666.66666667  3134938.7755102   2717972.78911565]
dAkorr= [ 241686.06912495  140511.19968723  105279.93562787]
k1= [ 1.96  1.44  1.21]
dk1= [ 0.20874391  0.09667044  0.06704514]
```

In [59]:
```
mü2=V*7.9
dmü2=dV*7.9
Akorr2=Akorr*np.exp(mü2*0.14)
dAkorr2=((dAkorr*np.exp(mü2*0.14))**2+(0.14*Akorr*dmü2*np.exp(mü2*0.14))**2)**(1/2)
k=np.exp(mü2*0.14)
dk=0.14*dmü2*np.exp(-mü2*0.14)
print("mü2=",mü2)
print("dmü2=",dmü2)
print("Akorr2=",Akorr2)
print("dAkorr2=",dAkorr2)
print("k=",k)
print("dk=",dk)
```

```
mü2= 0.426056776815
dmü2= 0.00481717862944
Akorr2= [ 4004545.29124993  3327620.86371078  2885026.98384834]
dAkorr2= [ 256554.98467912  149164.29283702  111767.67102697]
k= 1.06146279146
dk= 0.000635354355845
```

In [60]:
```
%matplotlib notebook
p=np.array([21, 118, 218, 319, 368, 391, 416, 435, 455, 474, 496, 705, 1008])
dp=np.array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
N4=np.array([13161, 13100, 12965, 12580, 11194, 9594, 7186, 5078, 3187, 1738, 821,
235, 258])
dN4=N4**(1/2)
n4=N4/60
dn4=dN4/60
n4halb=n4[0]/2
dn4halb=dn4[0]/2
print("n4=",n4)
print("dn4=",dn4)
print("n4halb=",n4halb)
print("dn4halb=",dn4halb)

def Linear(x, a, b):
    return a*x+b
M=np.arange(300,500,1)
Z=np.arange(1000)
z=np.ones(1000)*n4[0]/2
z1=np.ones(1000)*(n4[0]+dn4[0])/2
z2=np.ones(1000)*(n4[0]-dn4[0])/2
plt.errorbar(x=p, y=n4, yerr=dn4, xerr=dp, fmt='.')
popt1, pcov1 = curve_fit(Linear,p[6:10] ,n4[6:10] , sigma=dn4[6:10], absolute_sigma
=True)
Corr1 = plt.plot(M,Linear(M, *popt1), '-')
plt.plot(Z,z, color='r')
plt.plot(Z,z1,':', color='b')
plt.plot(Z,z2,':', color='b')
plt.title("Zählrate n eines alpha-Strahlers über dem Druck p")
plt.xlabel("Druck p [mbar]")
plt.ylabel("Zählrate n [counts/s]")
plt.text(500,180,"Hälfte der Zählrate: rot")
plt.text(500,160,"Fehler der halben Zählrate: blau")
plt.show()

p=419.8
dp=0.6
p0=1013
s0=3.95
ds0=0.05
rohgl=2.25
s1=p/p0*s0
ds1=((dp/p0*s0)**2+(p/p0*ds0)**2)**(1/2)
s2=rohgl/1.43
s3=0.68
sges=s1+s2+s3
dsges=ds1
print("s1=",s1)
print("ds1=",ds1)
print("s2=",s2)
print("sges=",sges)
print("dsges=",dsges)
```

```
n4= [ 219.35        218.33333333  216.08333333  209.66666667  186.56666667
  159.9         119.76666667   84.63333333   53.11666667   28.96666667
   13.68333333    3.91666667    4.3        ]
dn4= [ 1.91202336  1.90758719  1.89773257  1.86934332  1.76336169  1.63248277
  1.41283796  1.18766812  0.94089201  0.69482212  0.47755163  0.25549516
  0.26770631]
n4halb= 109.675
dn4halb= 0.956011680542
```




```
s1= 1.6369299111549853
ds1= 0.020852295831388996
s2= 1.5734265734265735
sges= 3.890356484581559
dsges= 0.020852295831388996
```