



DYNAMO

Einsatz von visueller Programmierung in BIM Workflows

Lejla Secerbegovic

Arch. Dipl.-Ing.
Autodesk | Technical Specialist

 @archistar

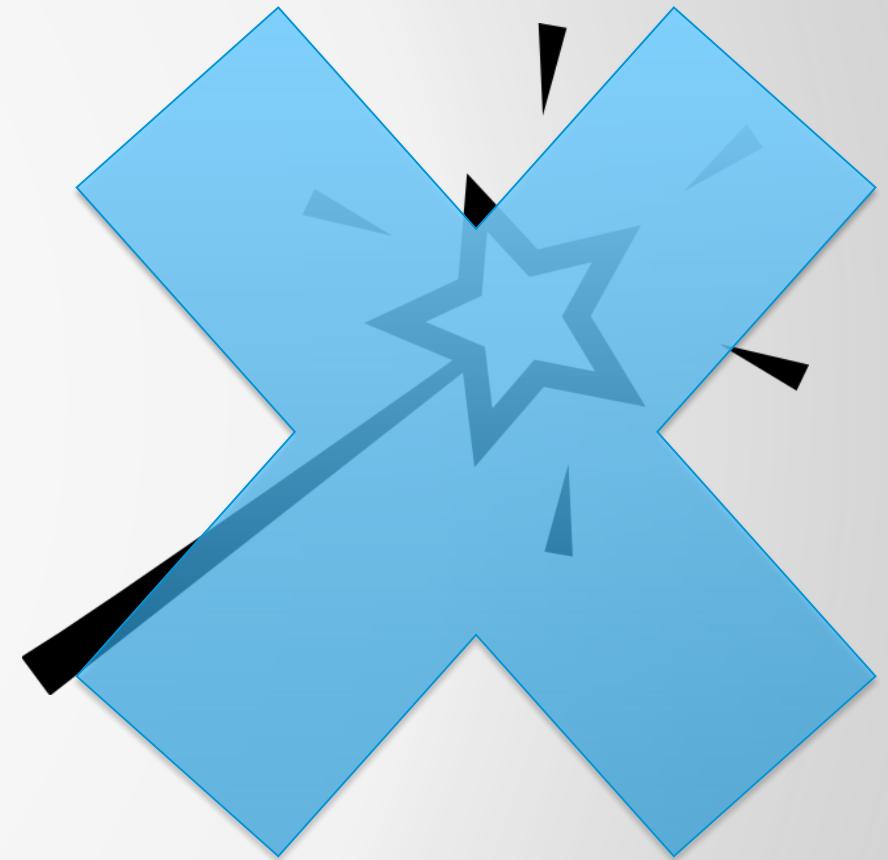
Tim Hoffeller

Dipl.- Ing. (FH)
BIM Consultant
CAD-Development
@thoffeller

Peter Kompolschek

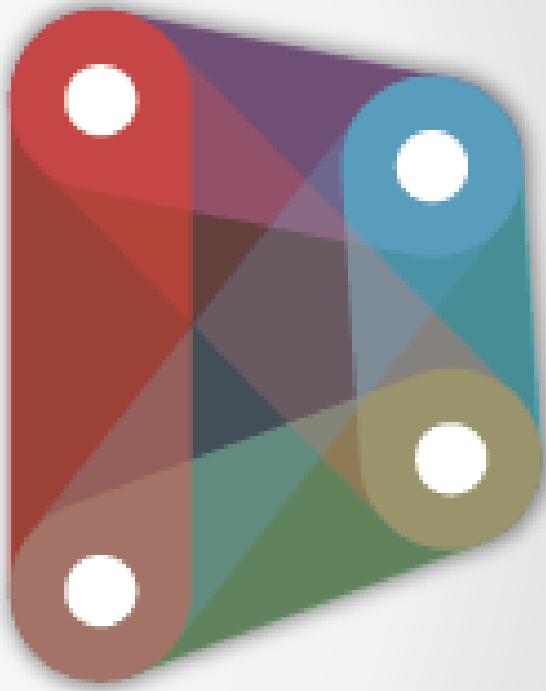
Arch. Dipl. Ing.
Architekt

Was ist Dynamo?

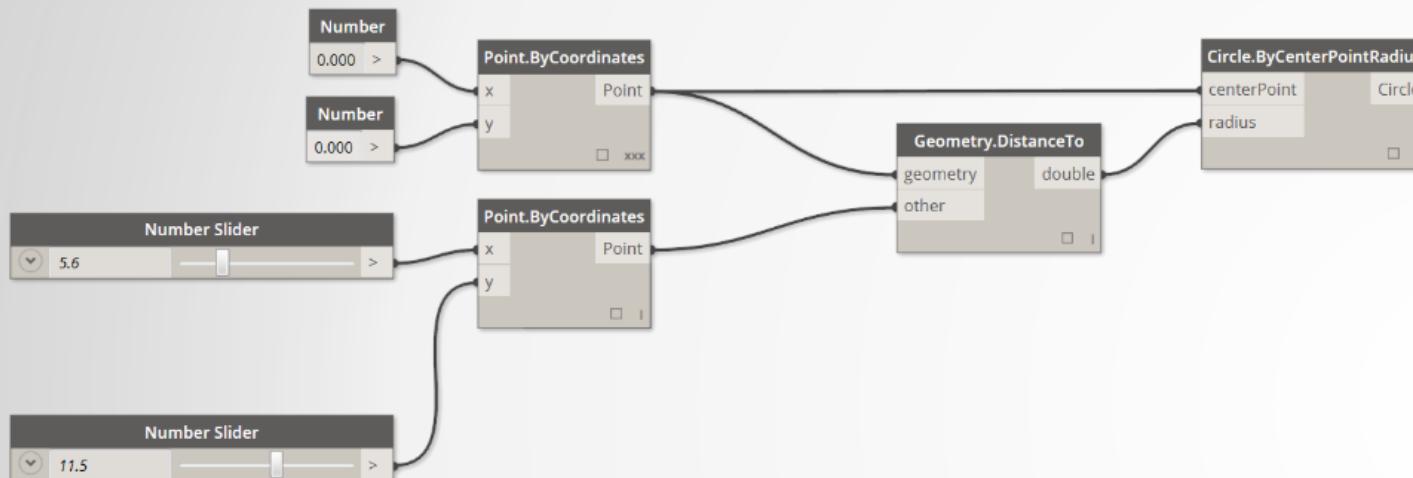


Was ist Dynamo für Revit?

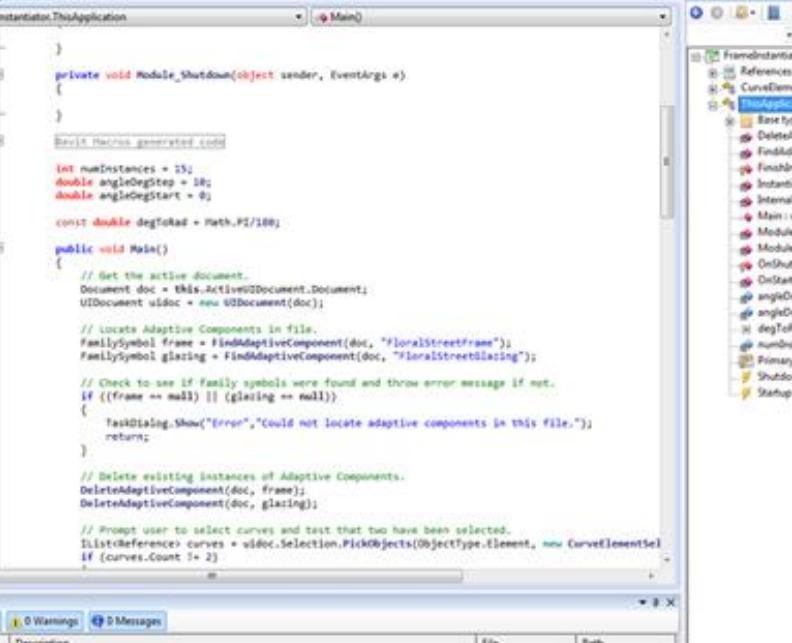
- openSource Revit-Plugin für Visual Scripting
- Keine Programmiererfahrung nötig
- Ermöglicht die Erstellung und Manipulation von Daten und Geometrie in Revit, die andernfalls gar nicht oder nur schwer möglich wäre



Visual Scripting = Visuelle Programmierung



Dynamo-Script

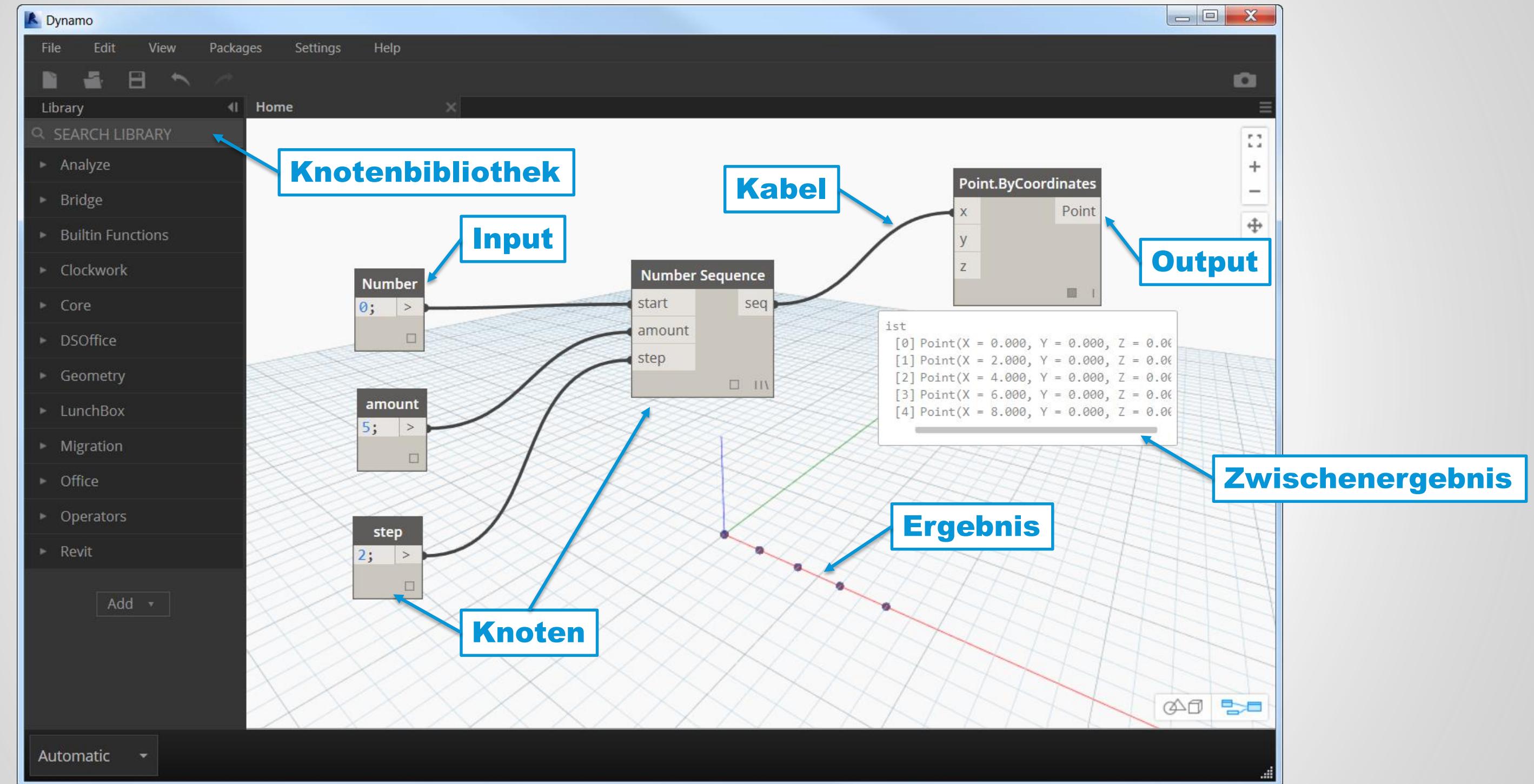


```
24
25
26
27     private void Module_Shutdown(object sender, EventArgs e)
28     {
29
30     }
31
32     [System.Diagnostics.DebuggerStepThrough()]
33
34     int numInstances = 15;
35     double angleDegStep = 8;
36     double angleDegStart = 0;
37
38     const double degToRad = Math.PI/180;
39
40     public void Main()
41     {
42         // Get the active document.
43         Document doc = this.ActiveUIDocument.Document;
44         UIDocument uidoc = new UIDocument(doc);
45
46         // Locate Adaptive Components in file.
47         FamilySymbol frame = FindAdaptiveComponent(doc, "FloralStreetFrame");
48         FamilySymbol glazing = FindAdaptiveComponent(doc, "FloralStreetGlazing");
49
50         // Check to see if Family symbols were found and throw error message if not.
51         if ((frame == null) || (glazing == null))
52         {
53             TaskDialog.Show("Error", "Could not locate adaptive components in this file.");
54             return;
55         }
56
57         // Delete existing instances of Adaptive Components.
58         DeleteAdaptiveComponent(doc, frame);
59         DeleteAdaptiveComponent(doc, glazing);
60
61         // Prompt user to select curves and test that two have been selected.
62         IList<Reference> curves = uidoc.Selection.PickObjects(ObjectType.Element, new CurveElementSelectionFilter());
63         if (curves.Count != 2)
64         {
65             TaskDialog.Show("Error", "Please select two curves to use this application.");
66             return;
67         }
68
69     }
70
71     [System.Diagnostics.DebuggerStepThrough()]
72
73     void DeleteAdaptiveComponent(Document doc, FamilySymbol symbol)
74     {
75         foreach (FamilyInstance fi in doc.FamilyInstances)
76         {
77             if (fi.Symbol == symbol)
78             {
79                 doc.FamilyInstances.Remove(fi);
80             }
81         }
82     }
83
84     void FindAdaptiveComponent(Document doc, string name)
85     {
86         foreach (FamilySymbol symbol in doc.FamilySymbols)
87         {
88             if (symbol.Name == name)
89             {
90                 return symbol;
91             }
92         }
93
94         return null;
95     }
96
97     void Main()
98     {
99         // ...
100    }
101
102    void Module_Shutdown()
103    {
104        // ...
105    }
106
107    void OnShutdown()
108    {
109        // ...
110    }
111
112    void OnStartup()
113    {
114        // ...
115    }
116
117    void InternalStartup()
118    {
119        // ...
120    }
121
122    void Main()
123    {
124        // ...
125    }
126
127    void Module_Shutdown()
128    {
129        // ...
130    }
131
132    void Module_Startup()
133    {
134        // ...
135    }
136
137    void OnShutdown()
138    {
139        // ...
140    }
141
142    void OnStartup()
143    {
144        // ...
145    }
146
147    void angleDegStart()
148    {
149        // ...
150    }
151
152    void angleDegStep()
153    {
154        // ...
155    }
156
157    void numInstances()
158    {
159        // ...
160    }
161
162    void degToRad()
163    {
164        // ...
165    }
166
167    void Shutdown()
168    {
169        // ...
170    }
171
172    void Startup()
173    {
174        // ...
175    }
176
177    void PrimaryCookie()
178    {
179        // ...
180    }
181
182    void DeleteAdaptiveComponent()
183    {
184        // ...
185    }
186
187    void FindAdaptiveComponent()
188    {
189        // ...
190    }
191
192    void FindAdaptiveComponent()
193    {
194        // ...
195    }
196
197    void InstantiateComponents()
198    {
199        // ...
200    }
201
202    void InternalStartup()
203    {
204        // ...
205    }
206
207    void Main()
208    {
209        // ...
210    }
211
212    void Module_Shutdown()
213    {
214        // ...
215    }
216
217    void Module_Startup()
218    {
219        // ...
220    }
221
222    void OnShutdown()
223    {
224        // ...
225    }
226
227    void OnStartup()
228    {
229        // ...
230    }
231
232    void InternalStartup()
233    {
234        // ...
235    }
236
237    void Main()
238    {
239        // ...
240    }
241
242    void Module_Shutdown()
243    {
244        // ...
245    }
246
247    void Module_Startup()
248    {
249        // ...
250    }
251
252    void OnShutdown()
253    {
254        // ...
255    }
256
257    void OnStartup()
258    {
259        // ...
260    }
261
262    void InternalStartup()
263    {
264        // ...
265    }
266
267    void Main()
268    {
269        // ...
270    }
271
272    void Module_Shutdown()
273    {
274        // ...
275    }
276
277    void Module_Startup()
278    {
279        // ...
280    }
281
282    void OnShutdown()
283    {
284        // ...
285    }
286
287    void OnStartup()
288    {
289        // ...
290    }
291
292    void InternalStartup()
293    {
294        // ...
295    }
296
297    void Main()
298    {
299        // ...
300    }
301
302    void Module_Shutdown()
303    {
304        // ...
305    }
306
307    void Module_Startup()
308    {
309        // ...
310    }
311
312    void OnShutdown()
313    {
314        // ...
315    }
316
317    void OnStartup()
318    {
319        // ...
320    }
321
322    void InternalStartup()
323    {
324        // ...
325    }
326
327    void Main()
328    {
329        // ...
330    }
331
332    void Module_Shutdown()
333    {
334        // ...
335    }
336
337    void Module_Startup()
338    {
339        // ...
340    }
341
342    void OnShutdown()
343    {
344        // ...
345    }
346
347    void OnStartup()
348    {
349        // ...
350    }
351
352    void InternalStartup()
353    {
354        // ...
355    }
356
357    void Main()
358    {
359        // ...
360    }
361
362    void Module_Shutdown()
363    {
364        // ...
365    }
366
367    void Module_Startup()
368    {
369        // ...
370    }
371
372    void OnShutdown()
373    {
374        // ...
375    }
376
377    void OnStartup()
378    {
379        // ...
380    }
381
382    void InternalStartup()
383    {
384        // ...
385    }
386
387    void Main()
388    {
389        // ...
390    }
391
392    void Module_Shutdown()
393    {
394        // ...
395    }
396
397    void Module_Startup()
398    {
399        // ...
400    }
401
402    void OnShutdown()
403    {
404        // ...
405    }
406
407    void OnStartup()
408    {
409        // ...
410    }
411
412    void InternalStartup()
413    {
414        // ...
415    }
416
417    void Main()
418    {
419        // ...
420    }
421
422    void Module_Shutdown()
423    {
424        // ...
425    }
426
427    void Module_Startup()
428    {
429        // ...
430    }
431
432    void OnShutdown()
433    {
434        // ...
435    }
436
437    void OnStartup()
438    {
439        // ...
440    }
441
442    void InternalStartup()
443    {
444        // ...
445    }
446
447    void Main()
448    {
449        // ...
450    }
451
452    void Module_Shutdown()
453    {
454        // ...
455    }
456
457    void Module_Startup()
458    {
459        // ...
460    }
461
462    void OnShutdown()
463    {
464        // ...
465    }
466
467    void OnStartup()
468    {
469        // ...
470    }
471
472    void InternalStartup()
473    {
474        // ...
475    }
476
477    void Main()
478    {
479        // ...
480    }
481
482    void Module_Shutdown()
483    {
484        // ...
485    }
486
487    void Module_Startup()
488    {
489        // ...
490    }
491
492    void OnShutdown()
493    {
494        // ...
495    }
496
497    void OnStartup()
498    {
499        // ...
500    }
501
502    void InternalStartup()
503    {
504        // ...
505    }
506
507    void Main()
508    {
509        // ...
510    }
511
512    void Module_Shutdown()
513    {
514        // ...
515    }
516
517    void Module_Startup()
518    {
519        // ...
520    }
521
522    void OnShutdown()
523    {
524        // ...
525    }
526
527    void OnStartup()
528    {
529        // ...
530    }
531
532    void InternalStartup()
533    {
534        // ...
535    }
536
537    void Main()
538    {
539        // ...
540    }
541
542    void Module_Shutdown()
543    {
544        // ...
545    }
546
547    void Module_Startup()
548    {
549        // ...
550    }
551
552    void OnShutdown()
553    {
554        // ...
555    }
556
557    void OnStartup()
558    {
559        // ...
560    }
561
562    void InternalStartup()
563    {
564        // ...
565    }
566
567    void Main()
568    {
569        // ...
570    }
571
572    void Module_Shutdown()
573    {
574        // ...
575    }
576
577    void Module_Startup()
578    {
579        // ...
580    }
581
582    void OnShutdown()
583    {
584        // ...
585    }
586
587    void OnStartup()
588    {
589        // ...
590    }
591
592    void InternalStartup()
593    {
594        // ...
595    }
596
597    void Main()
598    {
599        // ...
600    }
601
602    void Module_Shutdown()
603    {
604        // ...
605    }
606
607    void Module_Startup()
608    {
609        // ...
610    }
611
612    void OnShutdown()
613    {
614        // ...
615    }
616
617    void OnStartup()
618    {
619        // ...
620    }
621
622    void InternalStartup()
623    {
624        // ...
625    }
626
627    void Main()
628    {
629        // ...
630    }
631
632    void Module_Shutdown()
633    {
634        // ...
635    }
636
637    void Module_Startup()
638    {
639        // ...
640    }
641
642    void OnShutdown()
643    {
644        // ...
645    }
646
647    void OnStartup()
648    {
649        // ...
650    }
651
652    void InternalStartup()
653    {
654        // ...
655    }
656
657    void Main()
658    {
659        // ...
660    }
661
662    void Module_Shutdown()
663    {
664        // ...
665    }
666
667    void Module_Startup()
668    {
669        // ...
670    }
671
672    void OnShutdown()
673    {
674        // ...
675    }
676
677    void OnStartup()
678    {
679        // ...
680    }
681
682    void InternalStartup()
683    {
684        // ...
685    }
686
687    void Main()
688    {
689        // ...
690    }
691
692    void Module_Shutdown()
693    {
694        // ...
695    }
696
697    void Module_Startup()
698    {
699        // ...
700    }
701
702    void OnShutdown()
703    {
704        // ...
705    }
706
707    void OnStartup()
708    {
709        // ...
710    }
711
712    void InternalStartup()
713    {
714        // ...
715    }
716
717    void Main()
718    {
719        // ...
720    }
721
722    void Module_Shutdown()
723    {
724        // ...
725    }
726
727    void Module_Startup()
728    {
729        // ...
730    }
731
732    void OnShutdown()
733    {
734        // ...
735    }
736
737    void OnStartup()
738    {
739        // ...
740    }
741
742    void InternalStartup()
743    {
744        // ...
745    }
746
747    void Main()
748    {
749        // ...
750    }
751
752    void Module_Shutdown()
753    {
754        // ...
755    }
756
757    void Module_Startup()
758    {
759        // ...
760    }
761
762    void OnShutdown()
763    {
764        // ...
765    }
766
767    void OnStartup()
768    {
769        // ...
770    }
771
772    void InternalStartup()
773    {
774        // ...
775    }
776
777    void Main()
778    {
779        // ...
780    }
781
782    void Module_Shutdown()
783    {
784        // ...
785    }
786
787    void Module_Startup()
788    {
789        // ...
790    }
791
792    void OnShutdown()
793    {
794        // ...
795    }
796
797    void OnStartup()
798    {
799        // ...
800    }
801
802    void InternalStartup()
803    {
804        // ...
805    }
806
807    void Main()
808    {
809        // ...
810    }
811
812    void Module_Shutdown()
813    {
814        // ...
815    }
816
817    void Module_Startup()
818    {
819        // ...
820    }
821
822    void OnShutdown()
823    {
824        // ...
825    }
826
827    void OnStartup()
828    {
829        // ...
830    }
831
832    void InternalStartup()
833    {
834        // ...
835    }
836
837    void Main()
838    {
839        // ...
840    }
841
842    void Module_Shutdown()
843    {
844        // ...
845    }
846
847    void Module_Startup()
848    {
849        // ...
850    }
851
852    void OnShutdown()
853    {
854        // ...
855    }
856
857    void OnStartup()
858    {
859        // ...
860    }
861
862    void InternalStartup()
863    {
864        // ...
865    }
866
867    void Main()
868    {
869        // ...
870    }
871
872    void Module_Shutdown()
873    {
874        // ...
875    }
876
877    void Module_Startup()
878    {
879        // ...
880    }
881
882    void OnShutdown()
883    {
884        // ...
885    }
886
887    void OnStartup()
888    {
889        // ...
890    }
891
892    void InternalStartup()
893    {
894        // ...
895    }
896
897    void Main()
898    {
899        // ...
900    }
901
902    void Module_Shutdown()
903    {
904        // ...
905    }
906
907    void Module_Startup()
908    {
909        // ...
910    }
911
912    void OnShutdown()
913    {
914        // ...
915    }
916
917    void OnStartup()
918    {
919        // ...
920    }
921
922    void InternalStartup()
923    {
924        // ...
925    }
926
927    void Main()
928    {
929        // ...
930    }
931
932    void Module_Shutdown()
933    {
934        // ...
935    }
936
937    void Module_Startup()
938    {
939        // ...
940    }
941
942    void OnShutdown()
943    {
944        // ...
945    }
946
947    void OnStartup()
948    {
949        // ...
950    }
951
952    void InternalStartup()
953    {
954        // ...
955    }
956
957    void Main()
958    {
959        // ...
960    }
961
962    void Module_Shutdown()
963    {
964        // ...
965    }
966
967    void Module_Startup()
968    {
969        // ...
970    }
971
972    void OnShutdown()
973    {
974        // ...
975    }
976
977    void OnStartup()
978    {
979        // ...
980    }
981
982    void InternalStartup()
983    {
984        // ...
985    }
986
987    void Main()
988    {
989        // ...
990    }
991
992    void Module_Shutdown()
993    {
994        // ...
995    }
996
997    void Module_Startup()
998    {
999        // ...
1000    }
1001
1002    void OnShutdown()
1003    {
1004        // ...
1005    }
1006
1007    void OnStartup()
1008    {
1009        // ...
1010    }
1011
1012    void InternalStartup()
1013    {
1014        // ...
1015    }
1016
1017    void Main()
1018    {
1019        // ...
1020    }
1021
1022    void Module_Shutdown()
1023    {
1024        // ...
1025    }
1026
1027    void Module_Startup()
1028    {
1029        // ...
1030    }
1031
1032    void OnShutdown()
1033    {
1034        // ...
1035    }
1036
1037    void OnStartup()
1038    {
1039        // ...
1040    }
1041
1042    void InternalStartup()
1043    {
1044        // ...
1045    }
1046
1047    void Main()
1048    {
1049        // ...
1050    }
1051
1052    void Module_Shutdown()
1053    {
1054        // ...
1055    }
1056
1057    void Module_Startup()
1058    {
1059        // ...
1060    }
1061
1062    void OnShutdown()
1063    {
1064        // ...
1065    }
1066
1067    void OnStartup()
1068    {
1069        // ...
1070    }
1071
1072    void InternalStartup()
1073    {
1074        // ...
1075    }
1076
1077    void Main()
1078    {
1079        // ...
1080    }
1081
1082    void Module_Shutdown()
1083    {
1084        // ...
1085    }
1086
1087    void Module_Startup()
1088    {
1089        // ...
1090    }
1091
1092    void OnShutdown()
1093    {
1094        // ...
1095    }
1096
1097    void OnStartup()
1098    {
1099        // ...
1100    }
1101
1102    void InternalStartup()
1103    {
1104        // ...
1105    }
1106
1107    void Main()
1108    {
1109        // ...
1110    }
1111
1112    void Module_Shutdown()
1113    {
1114        // ...
1115    }
1116
1117    void Module_Startup()
1118    {
1119        // ...
1120    }
1121
1122    void OnShutdown()
1123    {
1124        // ...
1125    }
1126
1127    void OnStartup()
1128    {
1129        // ...
1130    }
1131
1132    void InternalStartup()
1133    {
1134        // ...
1135    }
1136
1137    void Main()
1138    {
1139        // ...
1140    }
1141
1142    void Module_Shutdown()
1143    {
1144        // ...
1145    }
1146
1147    void Module_Startup()
1148    {
1149        // ...
1150    }
1151
1152    void OnShutdown()
1153    {
1154        // ...
1155    }
1156
1157    void OnStartup()
1158    {
1159        // ...
1160    }
1161
1162    void InternalStartup()
1163    {
1164        // ...
1165    }
1166
1167    void Main()
1168    {
1169        // ...
1170    }
1171
1172    void Module_Shutdown()
1173    {
1174        // ...
1175    }
1176
1177    void Module_Startup()
1178    {
1179        // ...
1180    }
1181
1182    void OnShutdown()
1183    {
1184        // ...
1185    }
1186
1187    void OnStartup()
1188    {
1189        // ...
1190    }
1191
1192    void InternalStartup()
1193    {
1194        // ...
1195    }
1196
1197    void Main()
1198    {
1199        // ...
1200    }
1201
1202    void Module_Shutdown()
1203    {
1204        // ...
1205    }
1206
1207    void Module_Startup()
1208    {
1209        // ...
1210    }
1211
1212    void OnShutdown()
1213    {
1214        // ...
1215    }
1216
1217    void OnStartup()
1218    {
1219        // ...
1220    }
1221
1222    void InternalStartup()
1223    {
1224        // ...
1225    }
1226
1227    void Main()
1228    {
1229        // ...
1230    }
1231
1232    void Module_Shutdown()
1233    {
1234        // ...
1235    }
1236
1237    void Module_Startup()
1238    {
1239        // ...
1240    }
1241
1242    void OnShutdown()
1243    {
1244        // ...
1245    }
1246
1247    void OnStartup()
1248    {
1249        // ...
1250    }
1251
1252    void InternalStartup()
1253    {
1254        // ...
1255    }
1256
1257    void Main()
1258    {
1259        // ...
1260    }
1261
1262    void Module_Shutdown()
1263    {
1264        // ...
1265    }
1266
1267    void Module_Startup()
1268    {
1269        // ...
1270    }
1271
1272    void OnShutdown()
1273    {
1274        // ...
1275    }
1276
1277    void OnStartup()
1278    {
1279        // ...
1280    }
1281
1282    void InternalStartup()
1283    {
1284        // ...
1285    }
1286
1287    void Main()
1288    {
1289        // ...
1290    }
1291
1292    void Module_Shutdown()
1293    {
1294        // ...
1295    }
1296
1297    void Module_Startup()
1298    {
1299        // ...
1300    }
1301
1302    void OnShutdown()
1303    {
1304        // ...
1305    }
1306
1307    void OnStartup()
1308    {
1309        // ...
1310    }
1311
1312    void InternalStartup()
1313    {
1314        // ...
1315    }
1316
1317    void Main()
1318    {
1319        // ...
1320    }
1321
1322    void Module_Shutdown()
1323    {
1324        // ...
1325    }
1326
1327    void Module_Startup()
1328    {
1329        // ...
1330    }
1331
1332    void OnShutdown()
1333    {
1334        // ...
1335    }
1336
1337    void OnStartup()
1338    {
1339        // ...
1340    }
1341
1342    void InternalStartup()
1343    {
1344        // ...
1345    }
1346
1347    void Main()
1348    {
1349        // ...
1350    }
1351
1352    void Module_Shutdown()
1353    {
1354        // ...
1355    }
1356
1357    void Module_Startup()
1358    {
1359        // ...
1360    }
1361
1362    void OnShutdown()
1363    {
1364        // ...
1365    }
1366
1367    void OnStartup()
1368    {
1369        // ...
1370    }
1371
1372    void InternalStartup()
1373    {
1374        // ...
1375    }
1376
1377    void Main()
1378    {
1379        // ...
1380    }
1381
1382    void Module_Shutdown()
1383    {
1384        // ...
1385    }
1386
1387    void Module_Startup()
1388    {
1389        // ...
1390    }
1391
1392    void OnShutdown()
1393    {
1394        // ...
1395    }
1396
1397    void OnStartup()
1398    {
1399        // ...
1400    }
1401
1402    void InternalStartup()
1403    {
1404        // ...
1405    }
1406
1407    void Main()
1408    {
1409        // ...
1410    }
1411
1412    void Module_Shutdown()
1413    {
1414        // ...
1415    }
1416
1417    void Module_Startup()
1418    {
1419        // ...
1420    }
1421
1422    void OnShutdown()
1423    {
1424        // ...
1425    }
1426
1427    void OnStartup()
1428    {
1429        // ...
1430    }
1431
1432    void InternalStartup()
1433    {
1434        // ...
1435    }
1436
1437    void Main()
1438    {
1439        // ...
1440    }
1441
1442    void Module_Shutdown()
1443    {
1444        // ...
1445    }
1446
1447    void Module_Startup()
1448    {
1449        // ...
1450    }
1451
1452    void OnShutdown()
1453    {
1454        // ...
1455    }
1456
1457    void OnStartup()
1458    {
1459        // ...
1460    }
1461
1462    void InternalStartup()
1463    {
1464        // ...
1465    }
1466
1467    void Main()
1468    {
1469        // ...
1470    }
1471
1472    void Module_Shutdown()
1473    {
1474        // ...
1475    }
1476
1477    void Module_Startup()
1478    {
1479        // ...
1480    }
1481
1482    void OnShutdown()
1483    {
1484        // ...
1485    }
1486
1487    void OnStartup()
1488    {
1489        // ...
1490    }
1491
1492    void InternalStartup()
1493    {
1494        // ...
1495    }
1496
1497    void Main()
1498    {
1499        // ...
1500    }
1501
1502    void Module_Shutdown()
1503    {
1504        // ...
1505    }
1506
1507    void Module_Startup()
1508    {
1509        // ...
1510    }
1511
1512    void OnShutdown()
1513    {
1514        // ...
1515    }
1516
1517    void OnStartup()
1518    {
1519        // ...
1520    }
1521
1522    void InternalStartup()
1523    {
1524        // ...
1525    }
1526
1527    void Main()
1528    {
1529        // ...
1530    }
1531
1532    void Module_Shutdown()
1533    {
1534        // ...
1535    }
1536
1537    void Module_Startup()
1538    {
1539        // ...
1540    }
1541
1542    void OnShutdown()
1543    {
1544        // ...
1545    }
1546
1547    void OnStartup()
1548    {
1549        // ...
1550    }
1551
1552    void InternalStartup()
1553    {
1554        // ...
1555    }
1556
1557    void Main()
1558    {
1559        // ...
1560    }
1561
1562    void Module_Shutdown()
1563    {
1564        // ...
1565    }
1566
1567    void Module_Startup()
1568    {
1569        // ...
1570    }
1571
1572    void OnShutdown()
1573    {
1574        // ...
1575    }
1576
1577    void OnStartup()
1578    {
1579        // ...
1580    }
1581
1582    void InternalStartup()
1583    {
1584        // ...
1585    }
1586
1587    void Main()
1588    {
1589        // ...
1590    }
1591
1592    void Module_Shutdown()
1593    {
1594        // ...
1595    }
1596
1597    void Module_Startup()
1598    {
1599        // ...
1600    }
1601
1602    void OnShutdown()
1603    {
1604        // ...
1605    }
1606
1607    void OnStartup()
1608    {
1609        // ...
1610    }
1611
1612    void InternalStartup()
1613    {
1614        // ...
1615    }
1616
1617    void Main()
1618    {
1619        // ...
1620    }
1621
1622    void Module_Shutdown()
1623    {
1624        // ...
1625    }
1626
1627    void Module_Startup()
1628    {
1629        // ...
1630    }
1631
1632    void OnShutdown()
1633    {
1634        // ...
1635    }
1636
1637    void OnStartup()
1638    {
1639        // ...
1640    }
1641
1642    void InternalStartup()
1643    {
1644        // ...
1645    }
1646
1647    void Main()
1648    {
1649        // ...
1650    }
1651
1652    void Module_Shutdown()
1653    {
1654        // ...
1655    }
1656
1657    void Module_Startup()
1658    {
1659        // ...
1660    }
1661
1662    void OnShutdown()
1663    {
1664        // ...
1665    }
1666
1667    void OnStartup()
1668    {
1669        // ...
1670    }
1671
1672    void InternalStartup()
1673    {
1674        // ...
1675    }
1676
1677    void Main()
1678    {
1679        // ...
1680    }
1681
1682    void Module_Shutdown()
1683    {
1684        // ...
1685    }
1686
1687    void Module_Startup()
1688    {
1689        // ...
1690    }
1691
1692    void OnShutdown()
1693    {
1694        // ...
1695    }
1696
1697    void OnStartup()
1698    {
1699        // ...
1700    }
1701
1702    void InternalStartup()
1703    {
1704        // ...
1705    }
1706
1707    void Main()
1708    {
1709        // ...
1710    }
1711
1712    void Module_Shutdown()
1713    {
1714        // ...
1715    }
1716
1717    void Module_Startup()
1718    {
1719        // ...
1720    }
1721
1722    void OnShutdown()
1723    {
1724        // ...
1725    }
1726
1727    void OnStartup()
1728    {
1729        // ...
1730    }
1731
1732    void InternalStartup()
1733    {
1734        // ...
1735    }
1736
1737    void Main()
1738    {
1739        // ...
1740    }
1741
1742    void Module_Shutdown()
1743    {
1744        // ...
1745    }
1746
1747    void Module_Startup()
1748    {
1749        // ...
1750    }
1751
1752    void OnShutdown()
1753    {
1754        // ...
1755    }
1756
1757    void OnStartup()
1758    {
1759        // ...
1760    }
1761
1762    void InternalStartup()
1763    {
1764        // ...
1765    }
1766
1767    void Main()
1768    {
1769        // ...
1770    }
1771
1772    void Module_Shutdown()
1773    {
1774        // ...
1775    }
1776
1777    void Module_Startup()
1778    {
1779        // ...
1780    }
1781
1782    void OnShutdown()
1783    {
1784        // ...
1785    }
1786
1787    void OnStartup()
1788    {
1789        // ...
1790    }
1791
1792    void InternalStartup()
1793    {
1794        // ...
1795    }
1796
1797    void Main()
1798    {
1799        // ...
1800    }
1801
1802    void Module_Shutdown()
1803    {
1804        // ...
1805    }
1806
1807    void Module_Startup()
1808    {
1809        // ...
1810    }
1811
1812    void OnShutdown()
1813    {
1814        // ...
1815    }
1816
1817    void OnStartup()
1818    {
1819        // ...
1820    }
1821
1822    void InternalStartup()
1823    {
1824        // ...
1825    }
1826
1827    void Main()
1828    {
1829        // ...
1830    }
1831
1832    void Module_Shutdown()
1833    {
1834        // ...
1835    }
1836
1837    void Module_Startup()
1838    {
1839        // ...
1840    }
1841
1842    void OnShutdown()
1843    {
1844        // ...
1845    }
1846
1847    void OnStartup()
1848    {
1849        // ...
1850    }
1851
1852    void InternalStartup()
1853    {
1854        // ...
1855    }
1856
1857    void Main()
1858    {
1859        // ...
1860    }
1861
1862    void Module_Shutdown()
1863    {
1864        // ...
1865    }
1866
1867    void Module_Startup()
1868    {
1869        // ...
1870    }
1871
1872    void OnShutdown()
1873    {
1874        // ...
1875    }
1876
1877    void OnStartup()
1878    {
1879        // ...
1880    }
1881
1882    void InternalStartup()
1883    {
1884        // ...
1885    }
1886
1887    void Main()
1888    {
1889        // ...
1890    }
1891
1892    void Module_Shutdown()
1893    {
1894        // ...
1895    }
1896
1897    void Module_Startup()
1898    {
1899        // ...
1900    }
1901
1902    void OnShutdown()
1903    {
1904        // ...
1905    }
1906
1907    void OnStartup()
1908    {
1909        // ...
1910    }
1911
1912    void InternalStartup()
1913    {
1914        // ...
1915    }
1916
1917    void Main()
1918    {
1919        // ...
1920    }
1921
1922    void Module_Shutdown()
1923    {
1924        // ...
1925    }
1926
1927    void Module_Startup()
1928    {
1929        // ...
1930    }
1931
1932    void OnShutdown()
1933    {
1934        // ...
1935    }
1936
1937    void OnStartup()
1938    {
1939        // ...
1940    }
1941
1942    void InternalStartup()
1943    {
1944        // ...
1945    }
1946
1947    void Main()
1948    {
1949        // ...
1950    }
1951
1952    void Module_Shutdown()
1953    {
1954        // ...
1955    }
1956
1957    void Module_Startup()
1958    {
1959        // ...
1960    }
1961
1962    void OnShutdown()
1963    {
1964        // ...
1965    }
1966
1967    void OnStartup()
1968    {
1969        // ...
1970    }
1971
1972    void InternalStartup()
1973    {
1974        // ...
1975    }
1976
1977    void Main()
1978    {
1979        // ...
1980    }
1981
1982    void Module_Shutdown()
1983    {
1984        // ...
1985    }
1986
1987    void Module_Startup()
1988    {
1989        // ...
1990    }
1991
1992    void
```

Traditioneller Programmiercode (Visual Studio)

GLEICHES ERGEBNIS!

Dynamo Benutzeroberfläche



Dynamo Arbeitsweisen



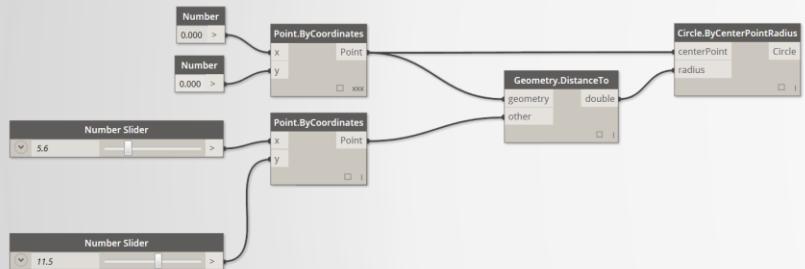
Anfänger



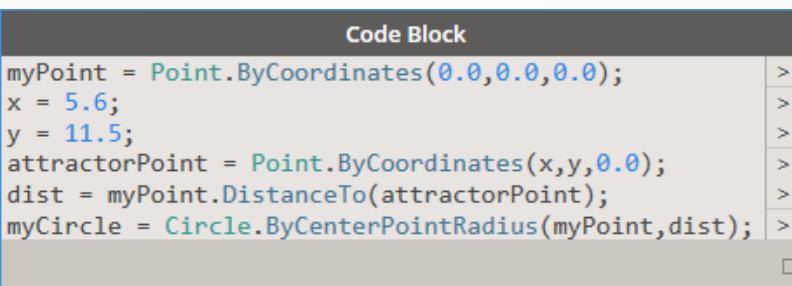
Profi



Nerd



Blöcke (Nodes)



DesignScript

```
# Default imports
import clr
clr.AddReference('RevitAPI')
clr.AddReference('RevitAPIUI')
from Autodesk.Revit.DB import *
import Autodesk
import sys
import clr
path = r'C:\Autodesk\Revit\Addins\2.19.0\PythonAPI'
exec_path = r'C:\Autodesk\Revit\Addins\2.19.0\PythonAPI\bin'
sys.path.append(path)
sys.path.append(exec_path)
clr.AddReference('LibGNet')
from Autodesk.LibG import *
clr.AddReference('DynamoPython')
clr.AddReference('DynamoCore')
clr.AddReference("DynamoUtilities")
clr.AddReference("DynamoRevit")
import Dynamo
#The input to this node will be stored in the IN variable.
dataEnteringNode = IN

#Revit.Application.Documents
x = Dynamo.Utilities.dynRevitSettings.Revit.Application.Documents

docsSet = []
for doc in x:
    docsSet.append(doc)

#Assign your output to the OUT variable
OUT = docsSet
```

Python

Einsatzmöglichkeiten

GEOMETRIE

DATENMANIPULATION

FUNKTIONSERWEITERUNG
FÜR REVIT

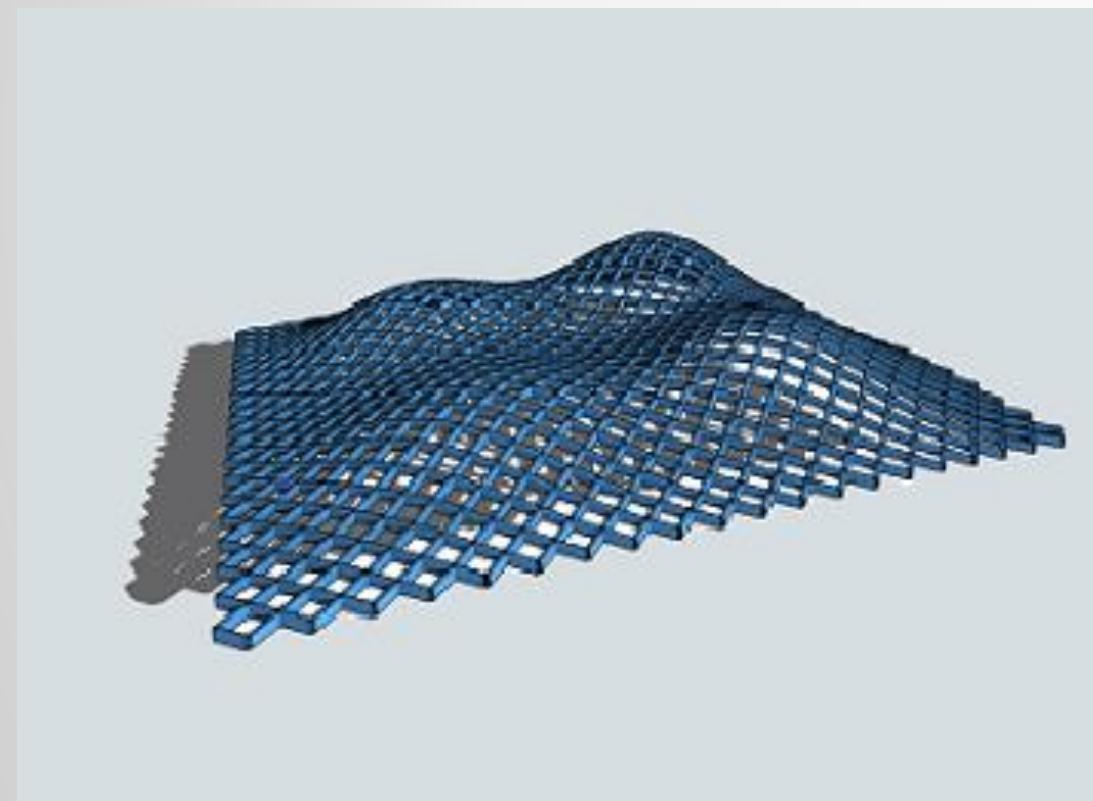
Einsatzmöglichkeiten

GEOMETRIE

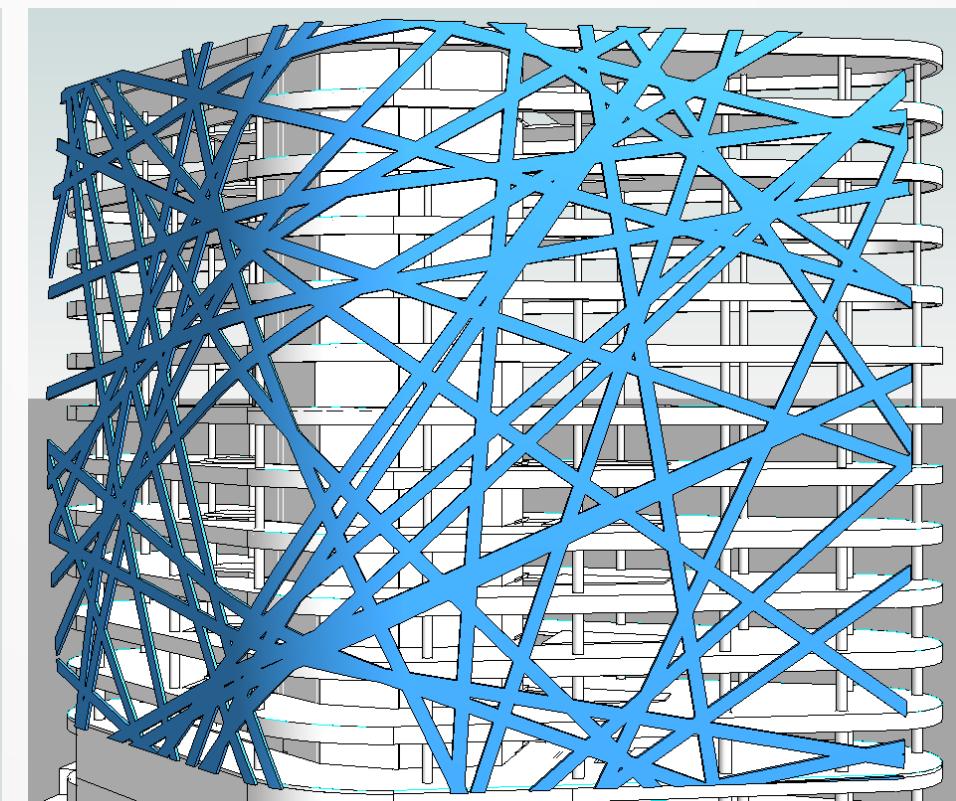
DATENMANIPULATION

FUNKTIONSERWEITERUNG FÜR REVIT

Erzeugung generativer / parametrischer Geometrie und komplexer Strukturen

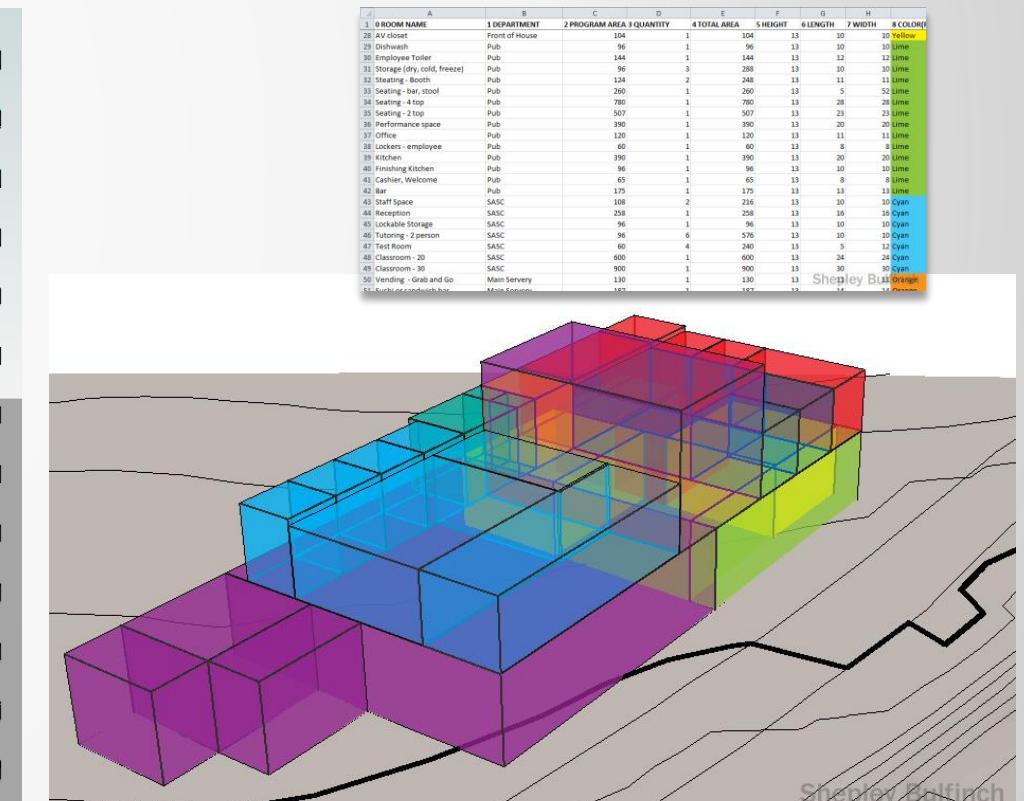


Generatives Design



Erstellung komplexer Tragwerke / Systeme

1 ROOM NAME	1 DEPARTMENT	2 PROGRAM AREA	3 QUANTITY	4 TOTAL AREA	5 HEIGHT	6 LENGTH	7 WIDTH	8 COLOR
28 AV closet	Front of House		1	104	13	10	10	Yellow
29 Dishwash	Pub		1	96	13	10	10	lime
30 Employee Teller	Pub		1	144	13	12	12	lime
31 Fridge - dry, cold, freeze	Pub		3	360	13	20	20	lime
32 Seating - Booth	Pub		2	248	13	11	11	lime
33 Seating - bar, stool	Pub		1	260	13	5	52	lime
34 Seating - 4 top	Pub		1	780	13	28	28	lime
35 Seating - 2 top	Pub		1	567	13	23	23	lime
36 Seating - entrance space	Pub		1	390	13	20	20	lime
37 Office	Pub		1	120	13	11	11	lime
38 Lockers - employee	Pub		1	60	13	8	8	lime
39 Kitchen	Pub		1	390	13	20	20	lime
40 Kitchen - prep kitchen	Pub		1	90	13	20	20	lime
41 Cashier, Welcome	Pub		1	65	13	8	8	lime
42 Bar	Pub		1	175	13	13	13	lime
43 Staff Room	SASC		1	216	13	10	10	cyan
44 Staff Room	SASC		1	250	13	16	16	cyan
45 Lockable storage	SASC		1	96	13	10	10	cyan
46 Tutoring - 2 person	SASC		6	576	13	10	10	cyan
47 Test Room	SASC		4	240	13	5	5	cyan
48 Classroom - 20	SASC		1	600	13	24	24	cyan
49 Classroom - 10	SASC		1	300	13	12	12	cyan
50 Vending - Grab and Go	Main Service		1	110	13	14	14	lime
51 Fuchi - one cond bar	Main Service		1	187	13	14	14	lime



Modellierung anhand von vorgegebenen Daten

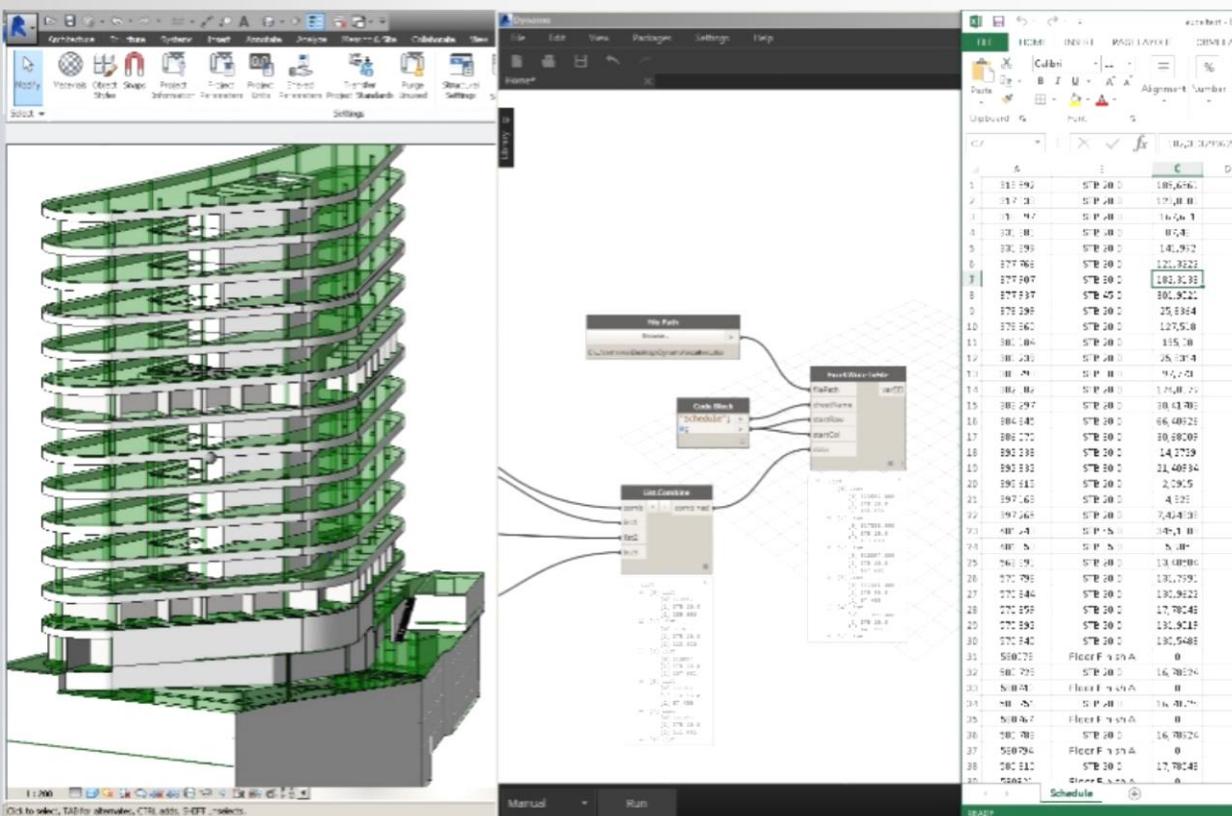
Einsatzmöglichkeiten

GEOMETRIE

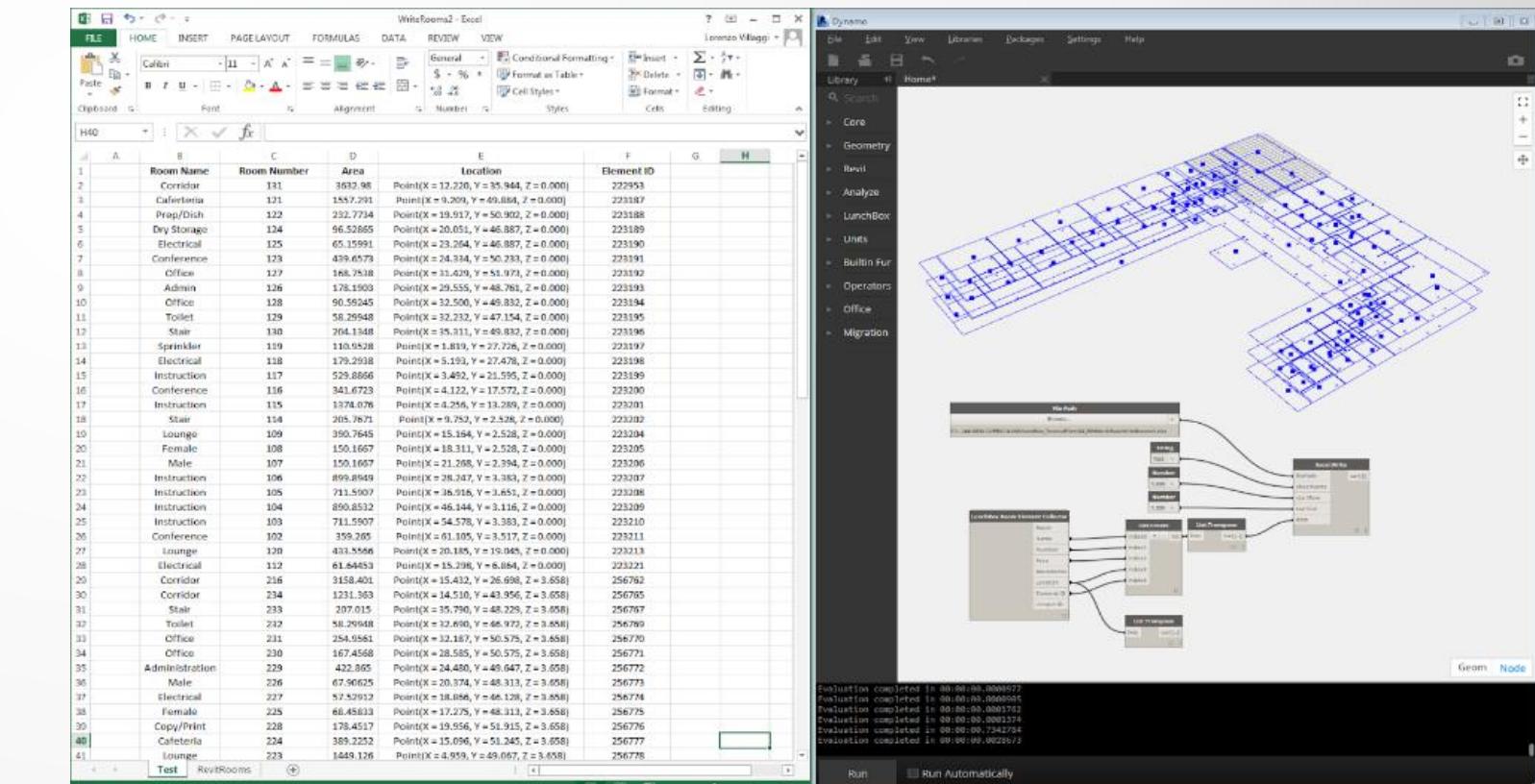
DATENMANIPLUATION

FUNKTIONSERWEITERUNG FÜR REVIT

Auslesen, Auswertung, Manipulation, Befüllen von BIM Attributen / Excel-Anbindung



Mengenermittlung



Raumlisten

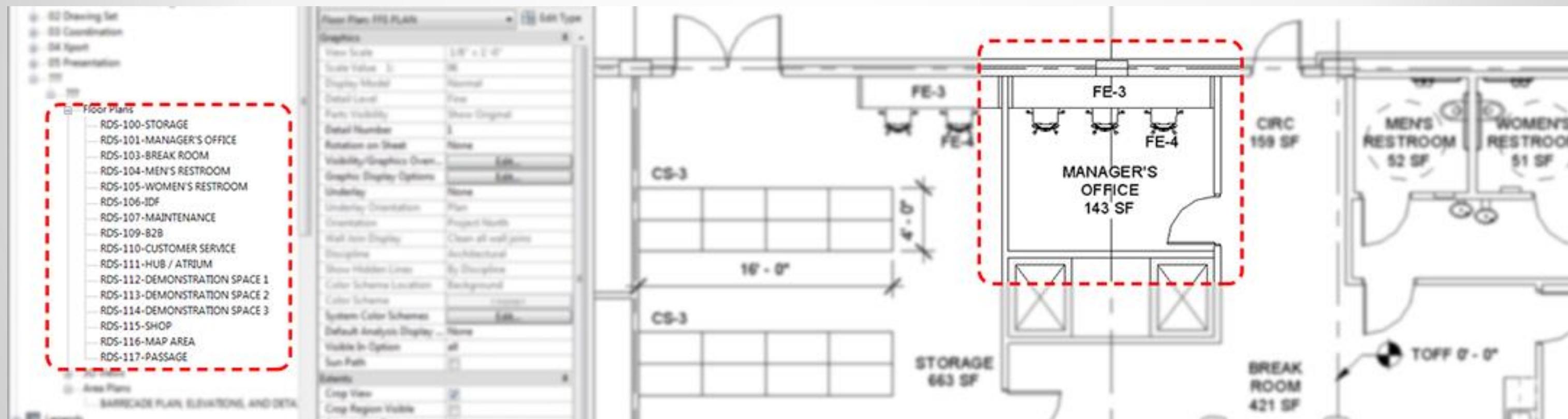
Einsatzmöglichkeiten

GEOMETRIE

DATENMANIPULATION

FUNKTIONSERWEITERUNG
FÜR REVIT

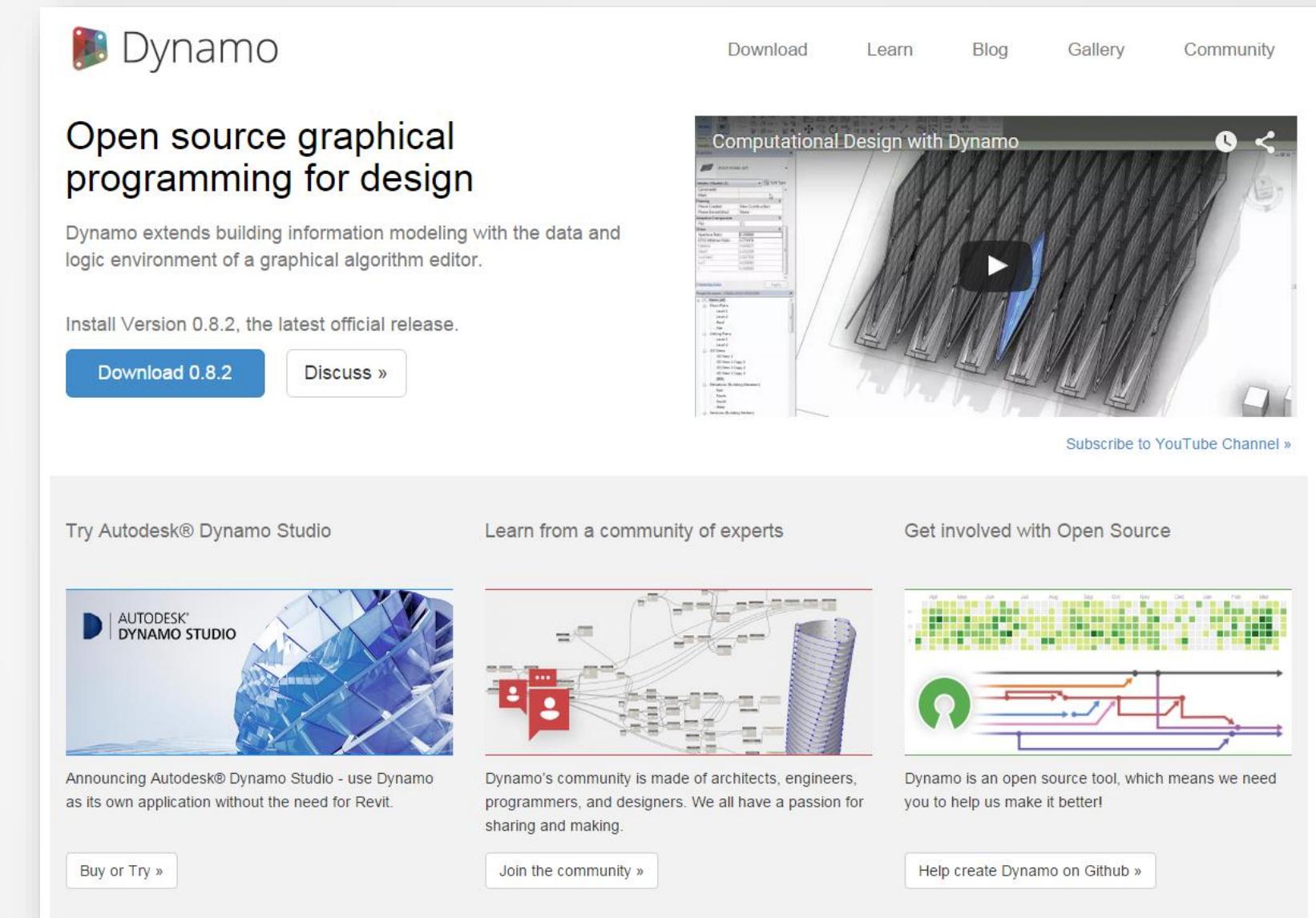
Direkter Zugriff auf alle Funktionen über die Revit-Programmierschnittstelle



Automatische Erstellung von Planansichten für alle Räume

Weitere Infos: Weltweite Dynamo Community

dynamobim.com



The screenshot shows the homepage of dynamobim.com. At the top, there is a navigation bar with links for 'Download', 'Learn', 'Blog', 'Gallery', and 'Community'. The main content area features a large image of a 3D model with a play button, indicating a video. Below this, there is a section for 'Autodesk® Dynamo Studio' with a 'Buy or Try' button. To the right, there are sections for 'Learn from a community of experts' and 'Get involved with Open Source', each with a 'Join the community' and 'Help create Dynamo on Github' button respectively.

Dynamo

Open source graphical programming for design

Dynamo extends building information modeling with the data and logic environment of a graphical algorithm editor.

Install Version 0.8.2, the latest official release.

[Download 0.8.2](#) [Discuss »](#)

Computational Design with Dynamo

Subscribe to YouTube Channel »

Try Autodesk® Dynamo Studio

Announcing Autodesk® Dynamo Studio - use Dynamo as its own application without the need for Revit.

[Buy or Try »](#)

Learn from a community of experts

Dynamo's community is made of architects, engineers, programmers, and designers. We all have a passion for sharing and making.

[Join the community »](#)

Get involved with Open Source

Dynamo is an open source tool, which means we need you to help us make it better!

[Help create Dynamo on Github »](#)

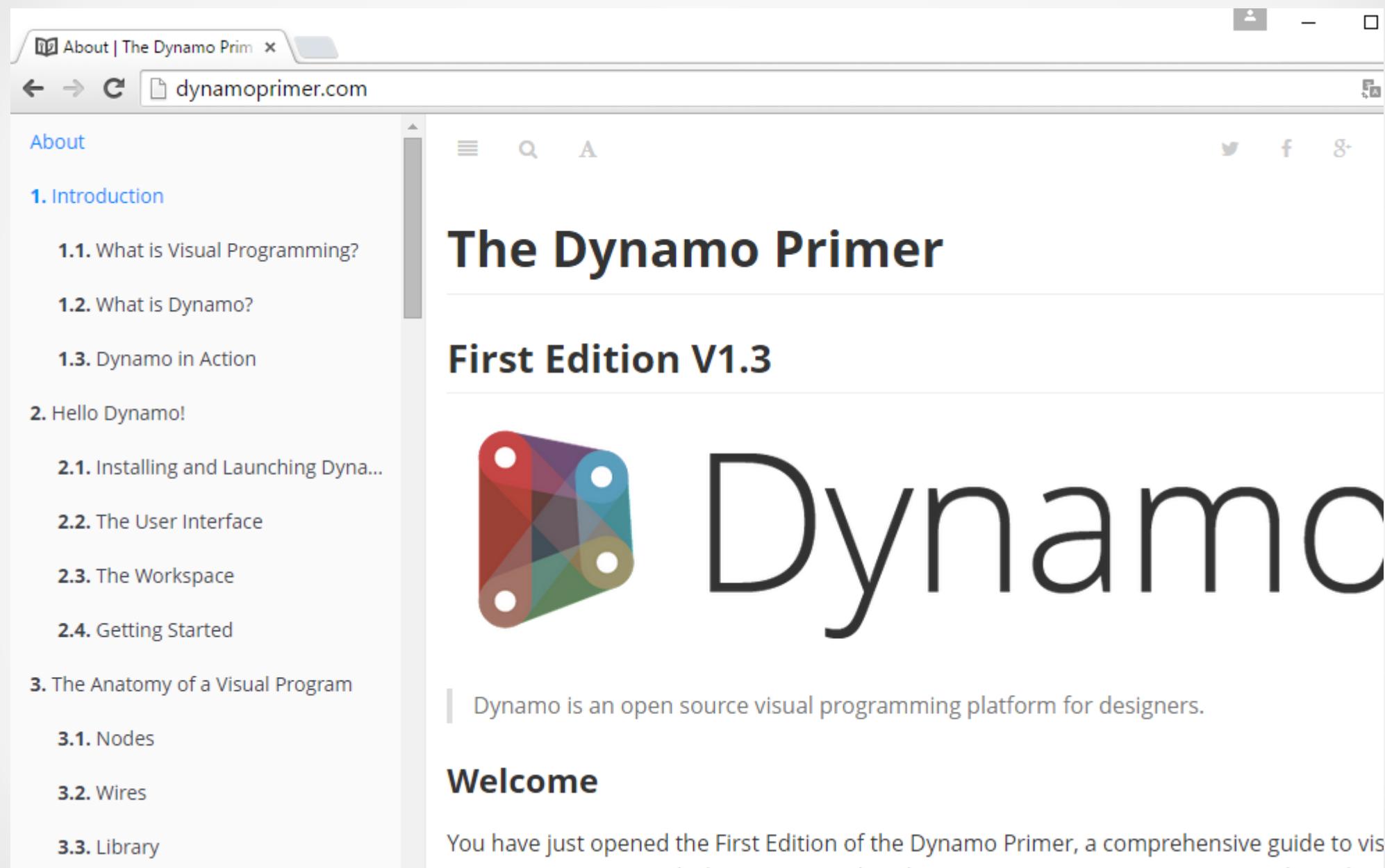
Weitere Infos: deutschsprachige Informationen

dynamobim.de

The background of the image is a complex, abstract 3D rendering. It features a large, translucent blue sphere on the right side, with a grid-like pattern of white lines forming its surface. To the left of the sphere, there are several other geometric shapes, including a large, faceted blue cube and a smaller, more complex blue and white structure. The overall aesthetic is modern and futuristic, with a focus on light, shadow, and geometric form.

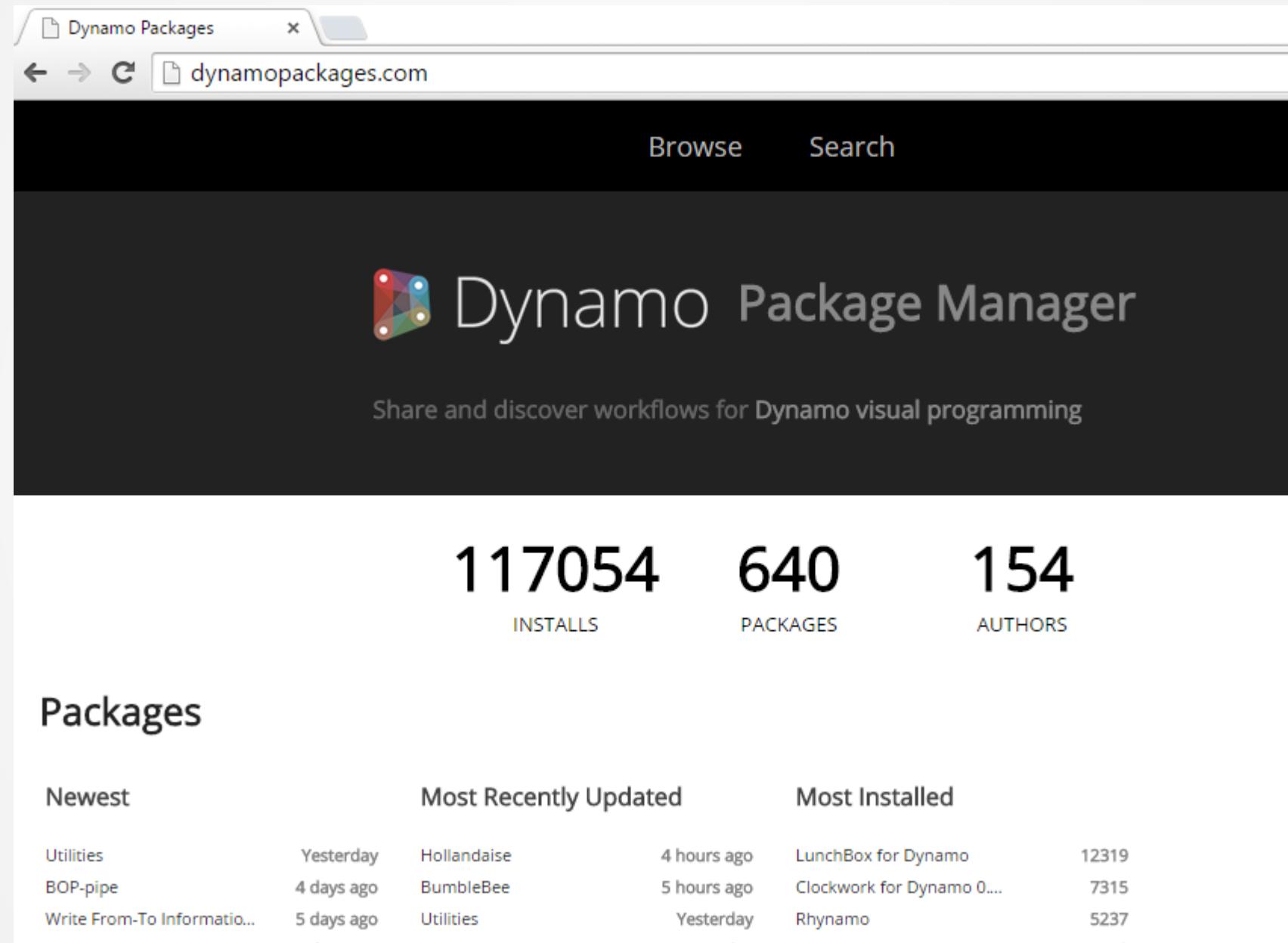
No Panic ;-)

Dynamo Hilfe und Beschreibung



The screenshot shows a web browser displaying the 'About | The Dynamo Primer' page from dynamoprimer.com. The page title is 'The Dynamo Primer' and the subtitle is 'First Edition V1.3'. It features a large 'Dynamo' logo with a colorful, geometric node icon. The text 'Dynamo is an open source visual programming platform for designers.' is displayed. The 'Welcome' section states, 'You have just opened the First Edition of the Dynamo Primer, a comprehensive guide to vis...'. On the left, a sidebar lists navigation links: 'About', '1. Introduction' (with sub-links 1.1, 1.2, 1.3), '2. Hello Dynamo!' (with sub-links 2.1, 2.2, 2.3, 2.4), and '3. The Anatomy of a Visual Program' (with sub-links 3.1, 3.2, 3.3). The browser's address bar shows 'dynamoprimer.com'.

Wo bekomme ich Zusatzpakete?

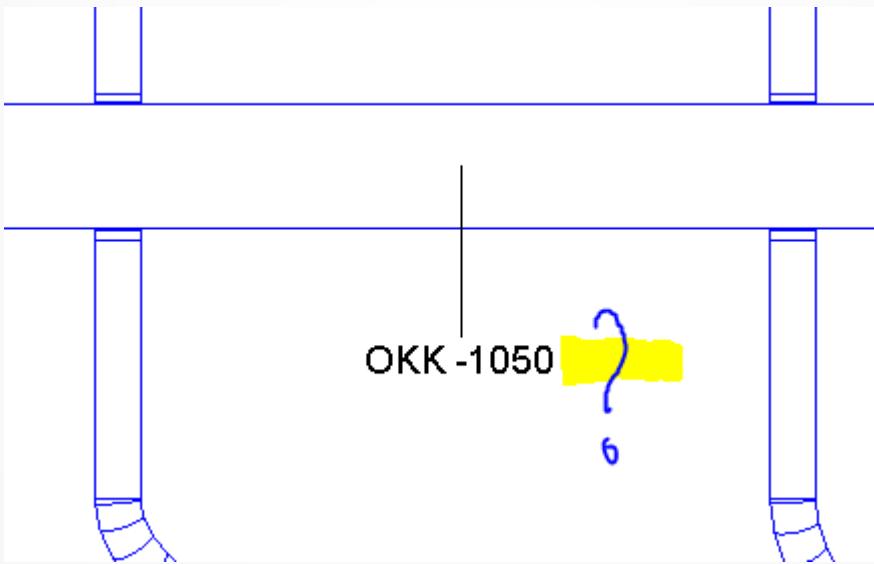


The screenshot shows a web browser window with the title 'Dynamo Packages' and the URL 'dynamopackages.com'. The page has a dark header with 'Browse' and 'Search' buttons. The main content area features the 'Dynamo Package Manager' logo and the tagline 'Share and discover workflows for Dynamo visual programming'. Below this, three large numbers are displayed: 117054 (INSTALS), 640 (PACKAGES), and 154 (AUTHORS). The 'PACKAGES' section is titled 'Packages' and lists packages in three categories: 'Newest', 'Most Recently Updated', and 'Most Installed'. The table shows the following data:

Category	Item	Details	Value			
Newest	Utilities	Yesterday	Hollandaise	4 hours ago	LunchBox for Dynamo	12319
	BOP-pipe	4 days ago	BumbleBee	5 hours ago	Clockwork for Dynamo 0....	7315
	Write From-To Informatio...	5 days ago	Utilities	Yesterday	Rhynamo	5237
...	

BSP 1 - Parameter setzen

- Revit bietet nicht alle Parameter für Beschriftungen an...
 - ... dann holen wir uns den Wert eben ;-)

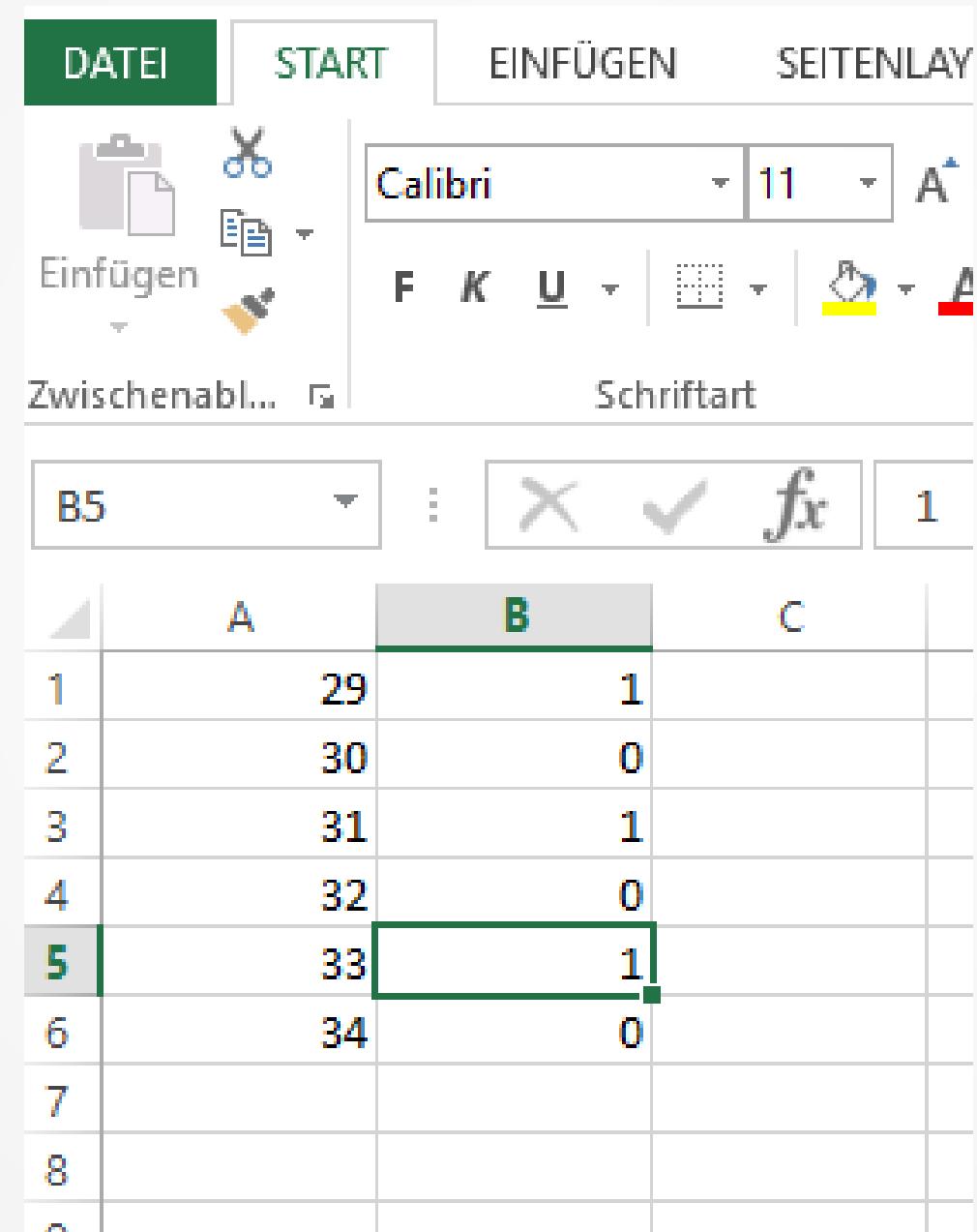


BSP 2 - Berechnungen

- Werte aus vorhandenen Parametern lesen und berechnen.

<MEP-Raumliste>					
A	B	C	D	E	F
Nummer	Name	Volumen	RLT_Raum_Luftwe	Angegebener Zulu	ZUL_Berechnet
1	b1	370.00 m ³	1	0 m ³ /h	370.00 m ³
3	b3	1286.82 m ³	1	0 m ³ /h	1286.82 m ³
4	b4	84.54 m ³	1	0 m ³ /h	84.54 m ³
5	b5	553.62 m ³	1	0 m ³ /h	553.62 m ³

Excel Ex- und Import



DATEI START EINFÜGEN SEITENLAY

Einfügen

Zwischenab... Schriftart

B5

	A	B	C
1	29	1	
2	30	0	
3	31	1	
4	32	0	
5	33	1	
6	34	0	
7			
8			
9			

Wer hat Angst vor Dynamo? Lösungen mit Dynamo

Norbert ZSIROS / WALDHAUSER+HERMANN AG

Dipl. HLK-Ingenieur FH

norbert.zsiros@waldhauser-hermann.ch



Diskutieren Sie mit auf twitter.com #AUD2015 #DynamoBIM

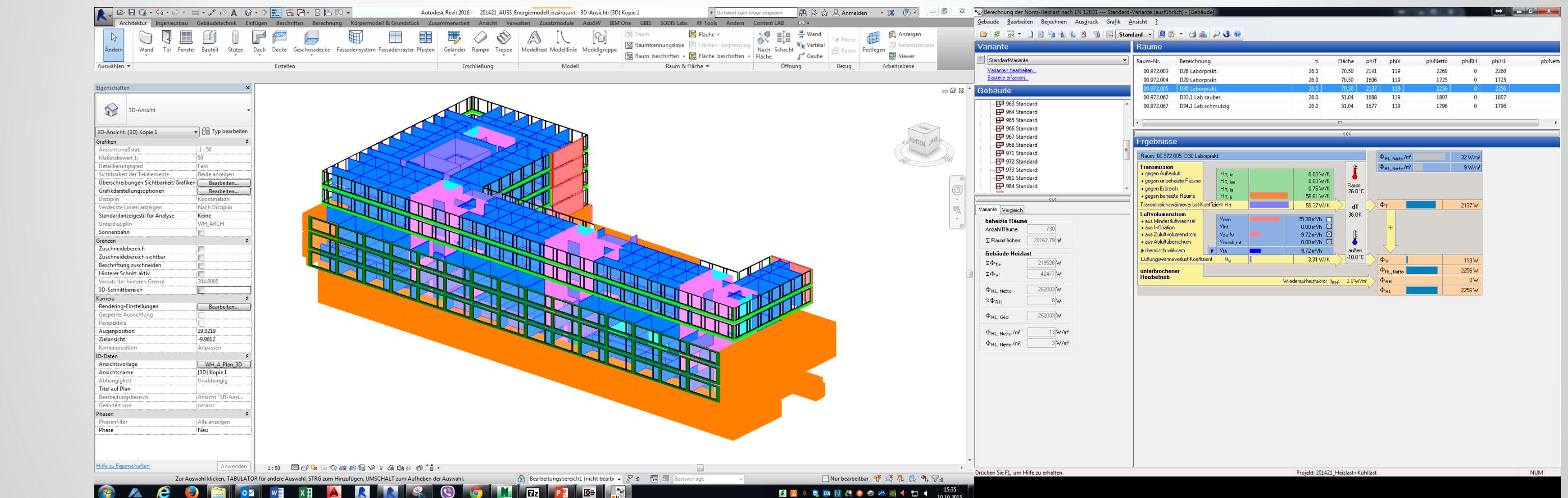
Projekt: Neubau ETH Forschungsgebäude GLC

- **Standort:** Zürich
- **Leistungen:** Planung Heizung / Lüftung / Klima /Kälte / Fachkoordination
- **Geschossfläche:** 22.500 m²
- **Geschosse:** 8
- **Räume:** 741 in 48 Nutzungszonen
- **Aktuelle Phase:** Ausschreibung



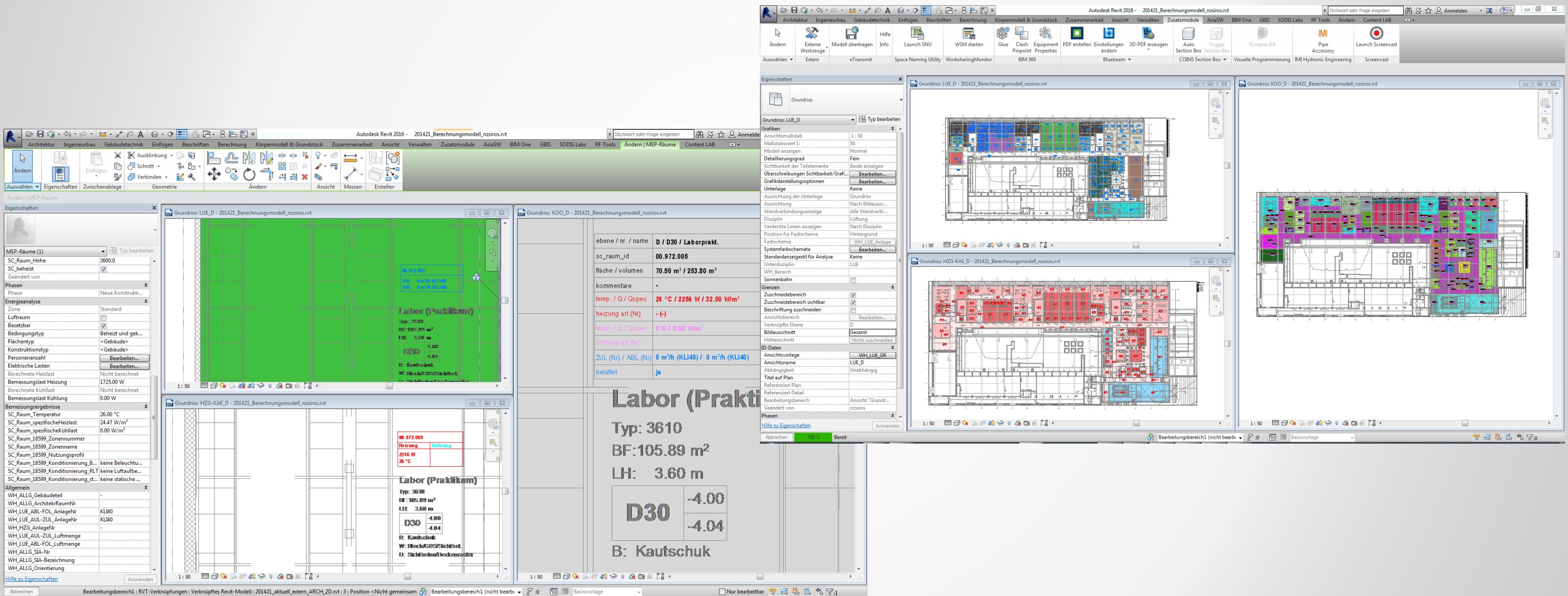
Revit: Energiemodell

- Energiemodell mit Räumen für die Berechnungen
- Heizlast – Kühllast Berechnungen nach SIA mit SolarComputer

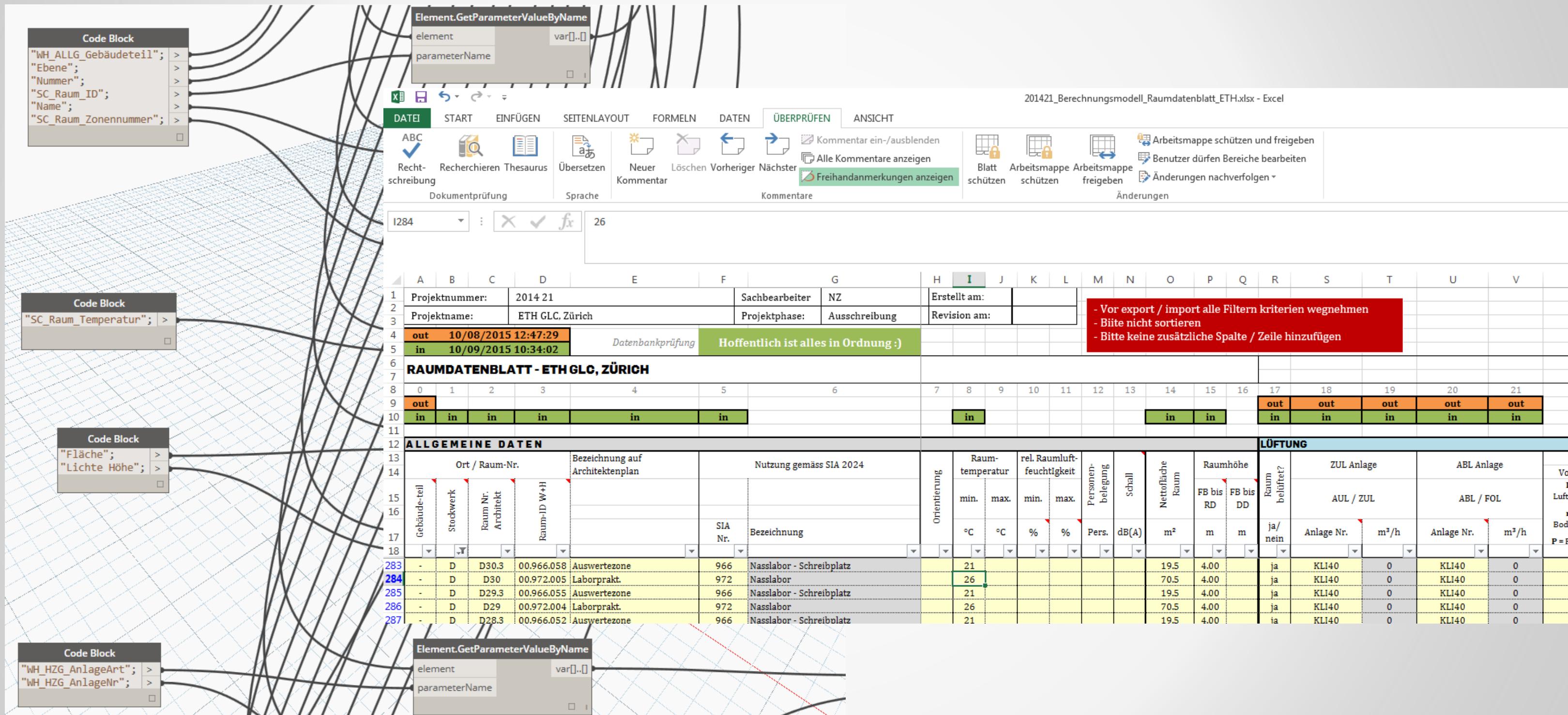


Revit: Flächenpläne und Raumstempel

- Shared Parameters für die Beschriftungen und für die Flächenpläne

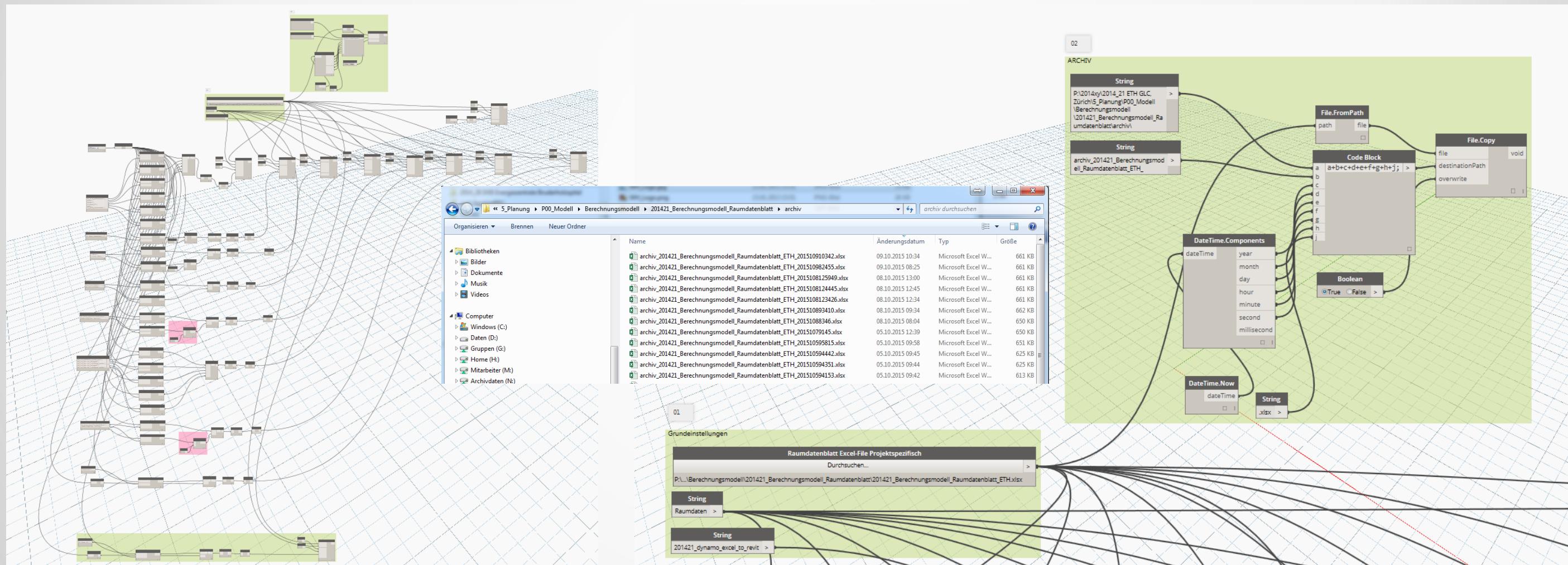


Dynamo: Revit – Excel Verbindung

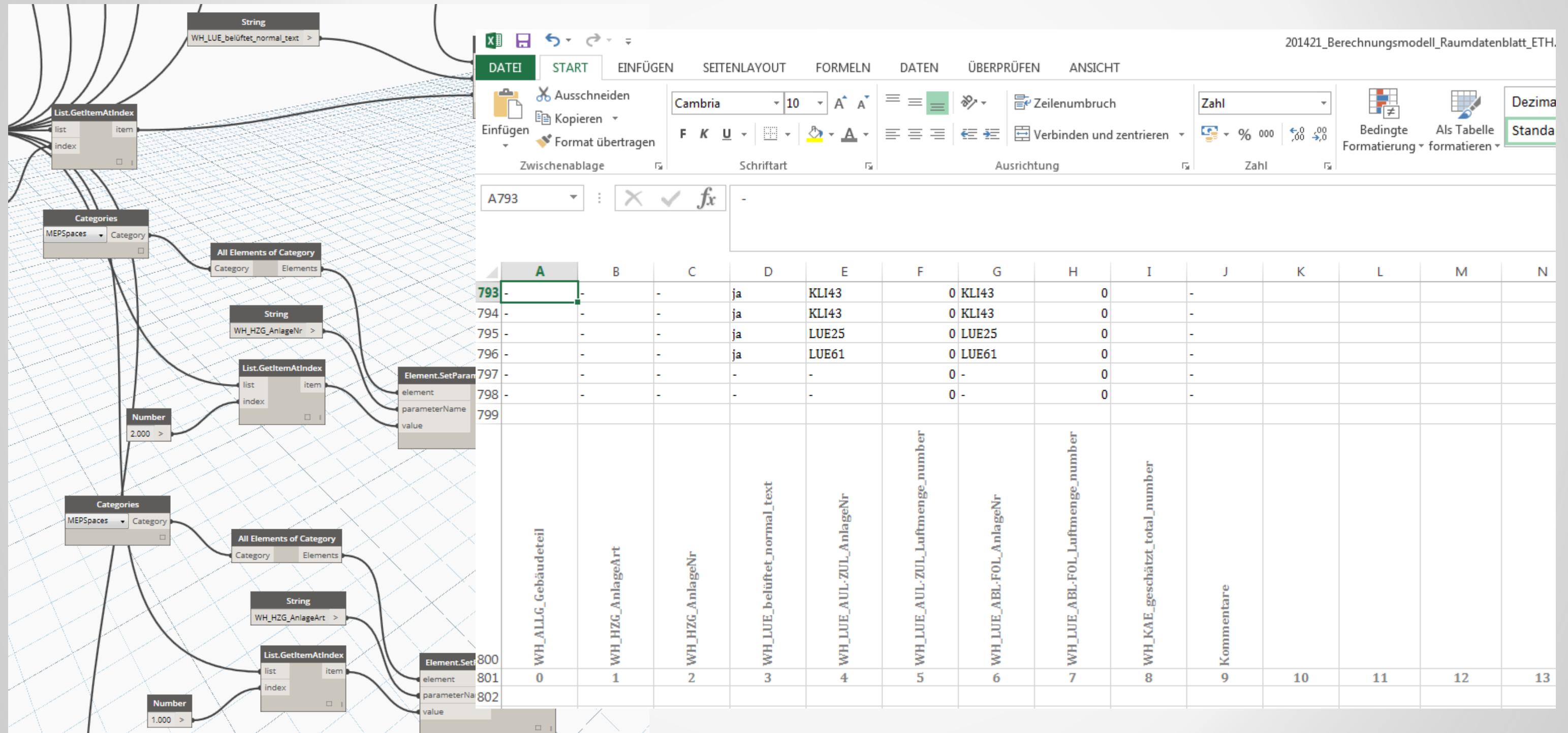


Dynamo: Revit – Excel Verbindung

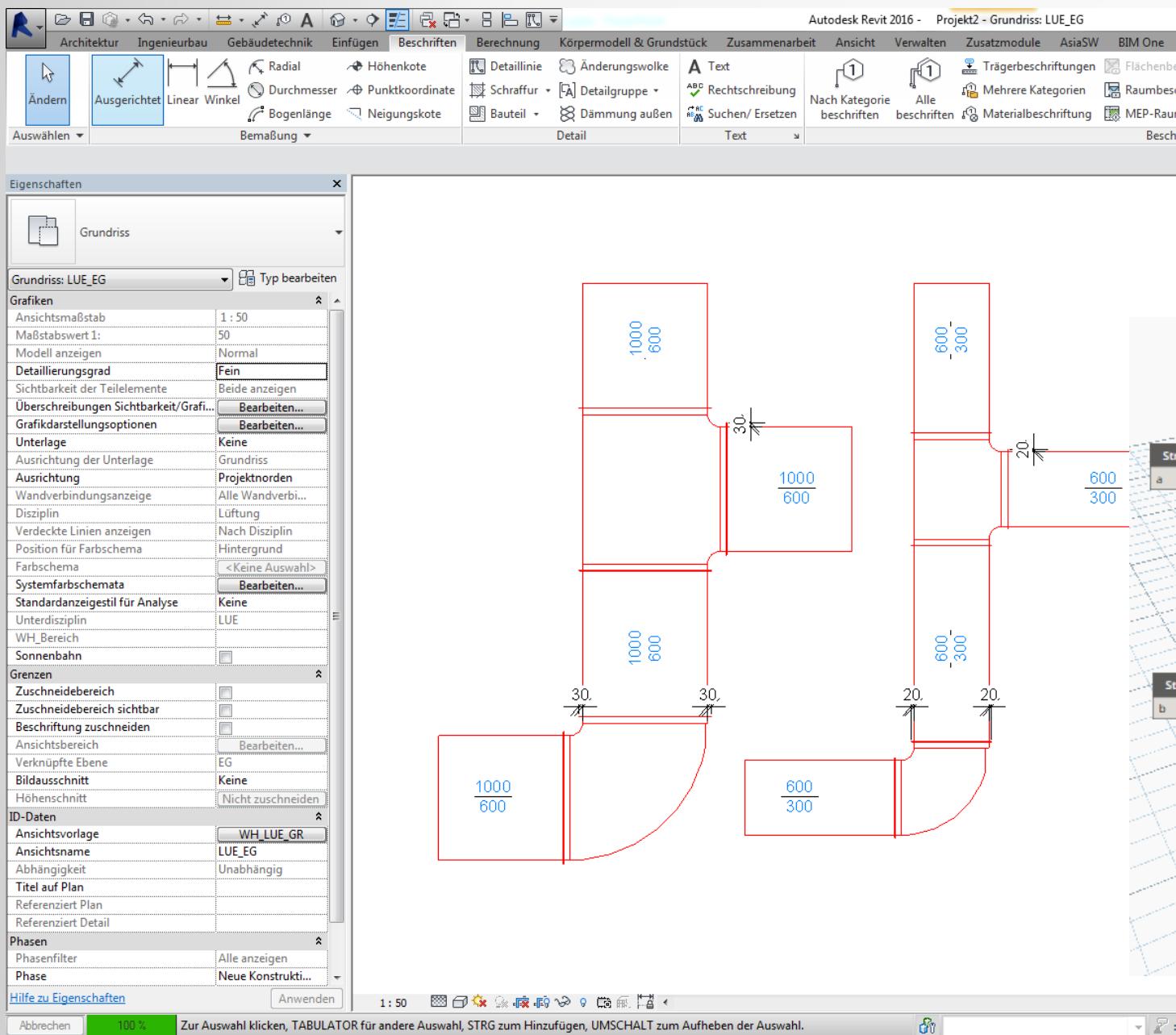
- Sicherheitskopie inkl. Zeitstempel mit Dynamo



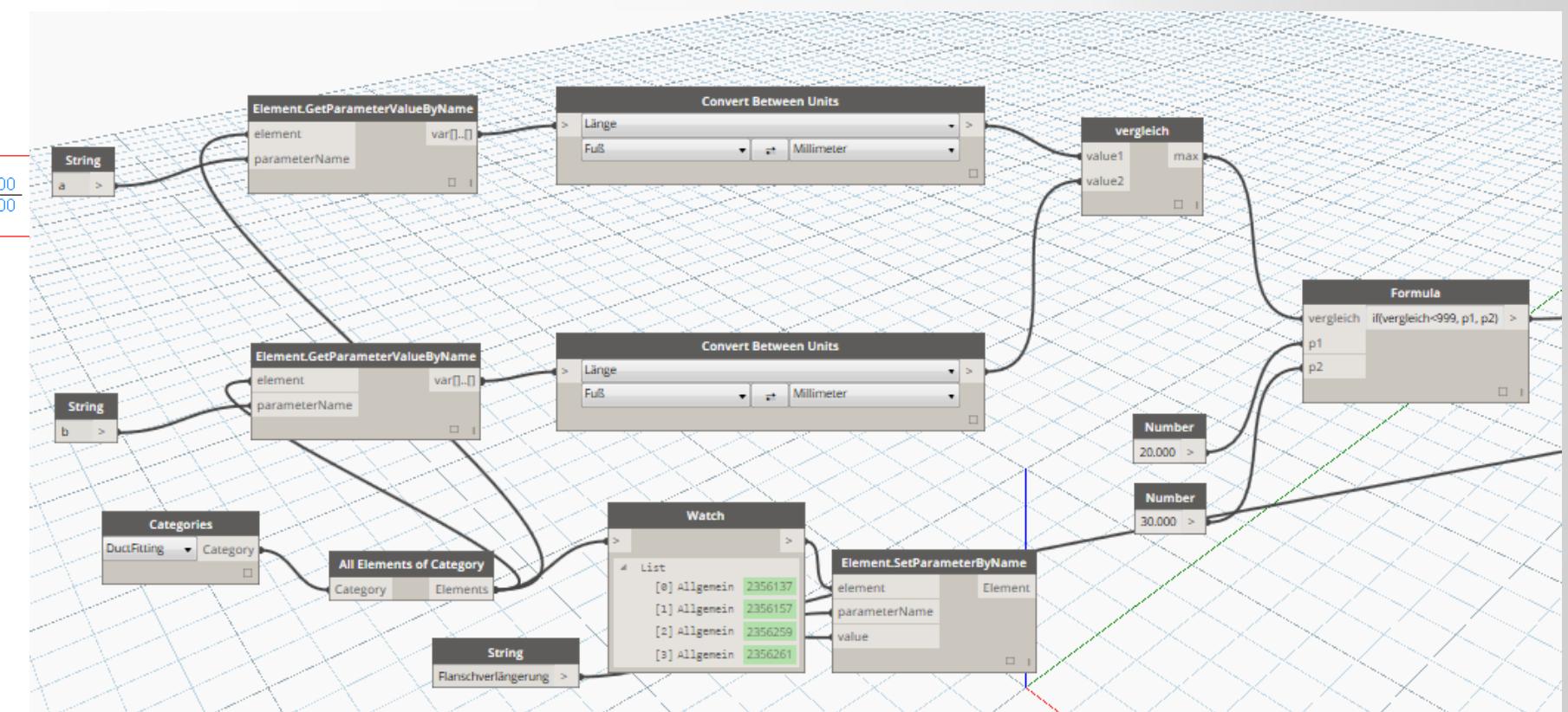
Dynamo: Excel – Revit Verbindung



Dynamo: Lüftungskanal Flanschverlängerung Einstellungen



- in der Schweiz
- Unter 1000 mm ist 20 mm
- Über 1000 mm ist 30 mm



Beispiele oda



