

Using Static Analyses For Continuous Quality Improvement

27. Juni 2019 – Applied Static Analysis (APSA) – TU Darmstadt



Praxis

Software-**Audits**

Kontinuierliche **Qualitäts-**
und **Testkontrolle**

 **Teamscale**

CQSE GmbH



Forschung

17+ **Promotionen** in
Software Engineering

Eigene **Forschung**

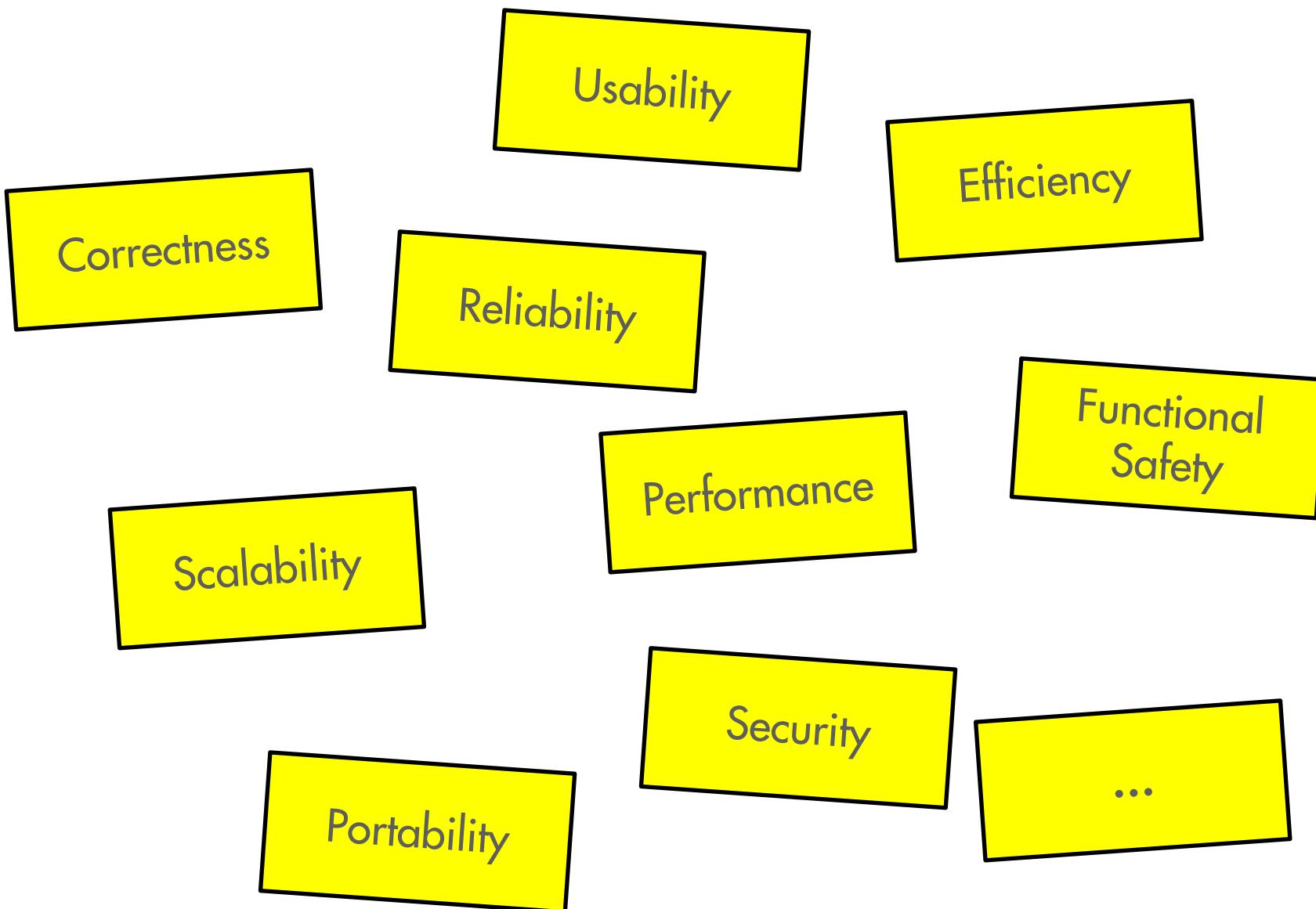
Enger Kontakt zu
Universitäten

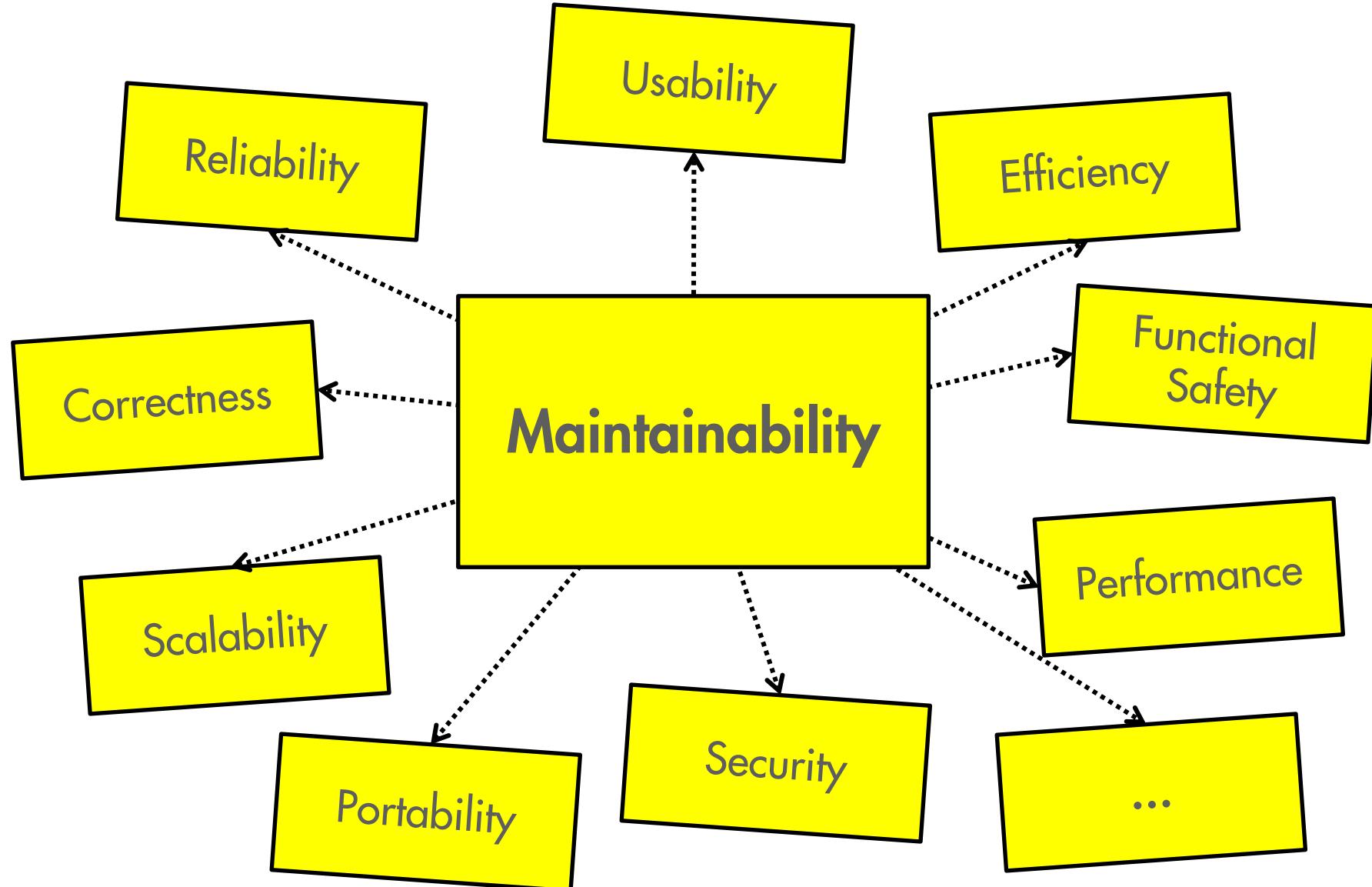
Agenda

- ▶ **11:40** **Introduction**
 - 11:45 Static Analysis for Quality Improvement
 - 12:05 Continuous Quality Improvement
 - 12:50 Innovation & Improvement
 - 13:10 Q&A
 - 13:20 End of Lecture

Agenda

- 11:40 Introduction
- 11:45 Static Analysis for Quality Improvement**
- 12:05 Continuous Quality Improvement
- 12:50 Innovation & Improvement
- 13:10 Q&A
- 13:20 End of Lecture



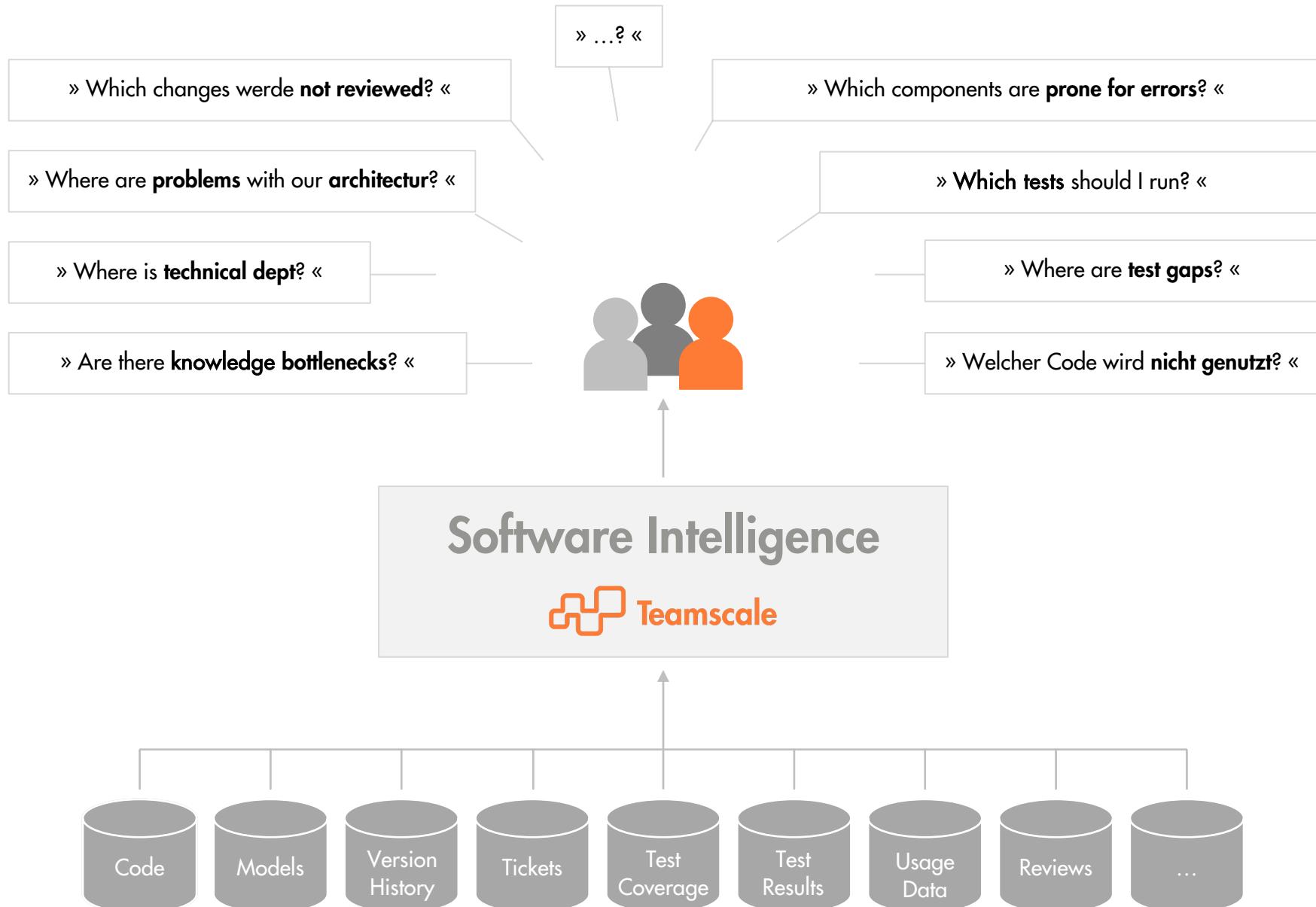


» All developers in a team

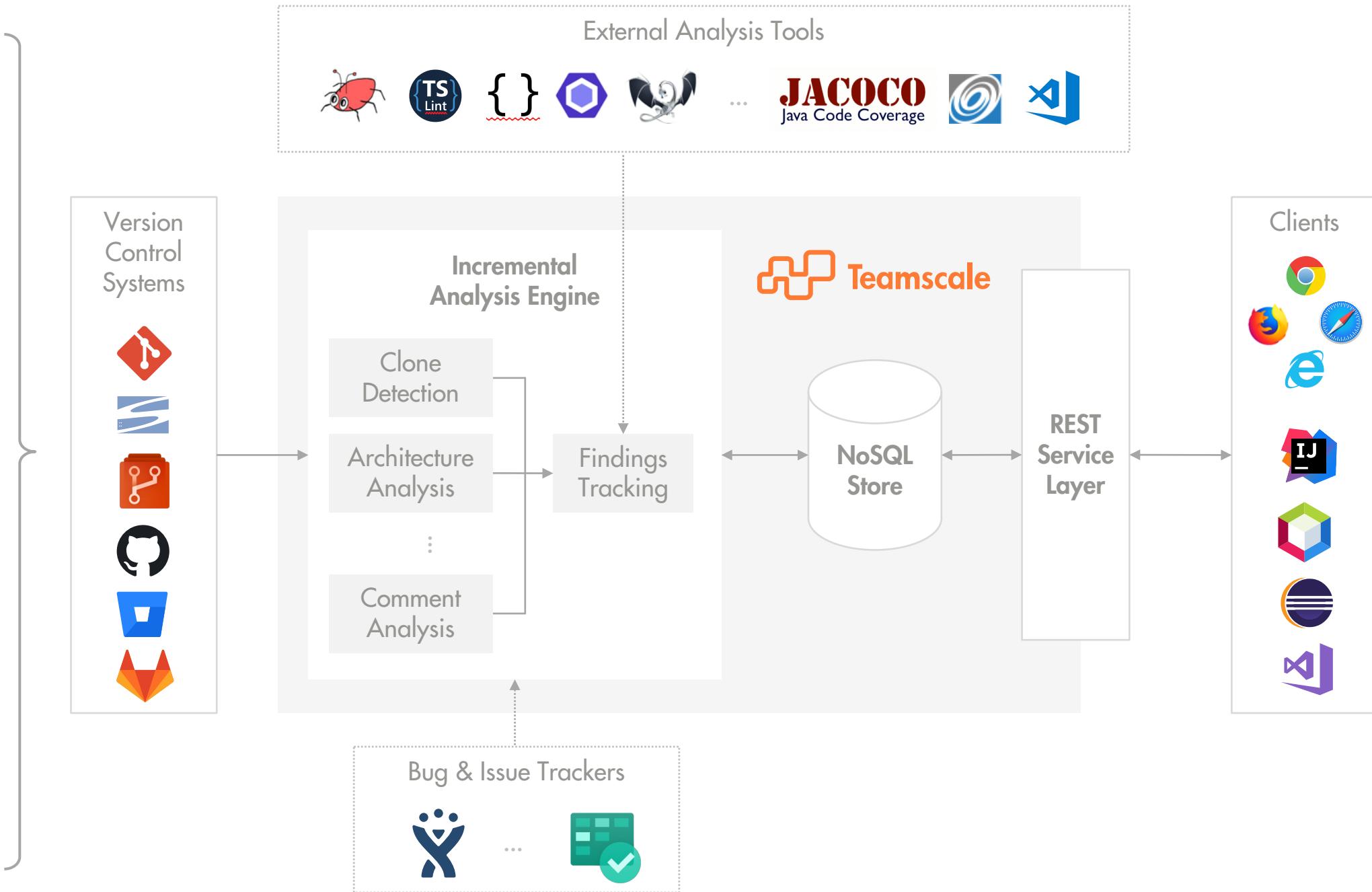
(newbies as well as veterans)

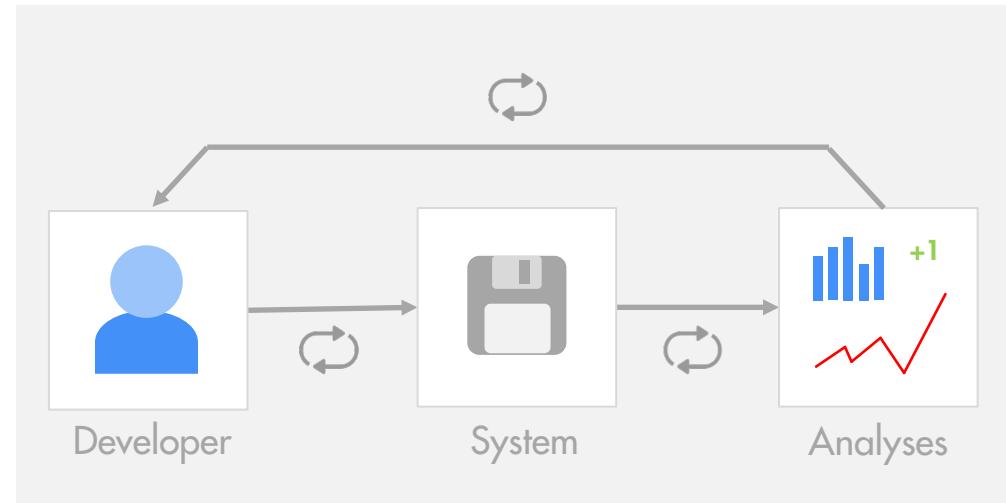
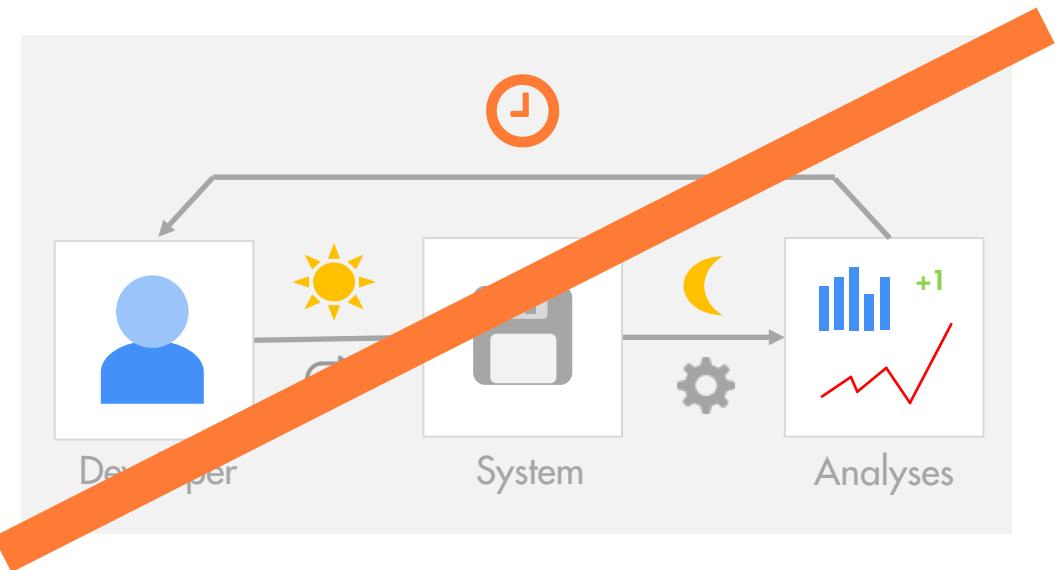
want to produce high-quality software,

i.e., write maintainable code. «



ABAP
Ada
C/C++
C#
COBOL
Delphi
Fortran
Gosu
Groovy
IEC 61131-3 ST
Java
JavaScript
Kotlin
Matlab
Objective-C
OpenCL
OScript
PHP
PL/SQL
Python
Rust
SAP HANA
Simulink
Swift
Visual Basic
XML
Xtend





Our Goal:

Measure What Is Relevant
(as opposed to what is easy to measure)

Cyclomatic Complexity (CycC) is a measure of the complexity of a program's control flow. It is calculated by counting the number of decision points (if statements, loops, etc.) in the code and adding one. A higher CycC value indicates a more complex program.

LCOM (Locality Coefficient of Maintenance) measures the degree of coupling between classes. It is calculated by dividing the number of identifiers shared between classes by the total number of identifiers in the system. A lower LCOM value indicates better cohesion and lower coupling.

No. Identifiers is a measure of the total number of unique identifiers (variables, functions, classes, etc.) in the codebase. It is often used as a proxy for the size of the system.

No. Classes is a measure of the number of distinct classes or objects defined in the codebase.

CBO (Cohesion Between Objects) measures the degree of coupling between objects. It is calculated by dividing the number of identifiers shared between objects by the total number of identifiers in the system. A lower CBO value indicates better cohesion and lower coupling.

Clone Coverage is a measure of the percentage of code that is covered by unit tests. It is calculated by dividing the number of unique test cases by the total number of test cases. A higher clone coverage value indicates that more of the codebase is being tested.

NOPM (No. Of Programming Methods) is a measure of the number of methods defined in the codebase.

Line Coverage is a measure of the percentage of code lines that are covered by unit tests. It is calculated by dividing the number of lines of code that have been executed by the total number of lines of code. A higher line coverage value indicates that more of the codebase is being tested.

Defect Density is a measure of the number of defects found per unit of code. It is calculated by dividing the number of defects by the total number of lines of code. A higher defect density value indicates a higher likelihood of bugs in the codebase.

Flow NOC (Flow Number of Components) is a measure of the number of components in a system. It is often used in the context of distributed systems to track the number of nodes in a network.

Abstractness is a measure of the level of abstraction of the code. It is calculated by dividing the number of abstract classes by the total number of classes. A higher abstractness value indicates a higher level of abstraction.

Branch Coverage is a measure of the percentage of branches in the codebase that are covered by unit tests. It is calculated by dividing the number of branches that have been executed by the total number of branches. A higher branch coverage value indicates that more of the codebase is being tested.

NOA (Number of Abstract Classes) is a measure of the number of abstract classes defined in the codebase.

LOC (Lines of Code) is a measure of the total number of lines of code in the codebase. It is often used as a proxy for the size of the system.

Stability is a measure of the stability of the codebase over time. It is calculated by tracking the number of changes made to the codebase and the impact of those changes on the system's behavior.

Fan Out is a measure of the number of outgoing edges from a single node in a control flow graph. It is often used to identify hotspots in the codebase where many paths converge.

WMC (Weighted Method Complexity) is a measure of the complexity of a method based on the number of statements, loops, and conditionals it contains. It is calculated by assigning a weight to each statement type and summing them up.

Fan In is a measure of the number of incoming edges to a single node in a control flow graph. It is often used to identify hotspots in the codebase where many paths converge.

NOAM (Number of Abstract Methods) is a measure of the number of abstract methods defined in the codebase.

MCDC (Multiple Condition Decision Coverage) is a testing technique that ensures that every possible combination of conditions in a decision point is exercised at least once.

RFSS (Reactive Function Statement Selection) is a testing technique that selects test cases based on the reactive function of the code, ensuring that all possible states are explored.

Statement Coverage is a measure of the percentage of statements in the codebase that are covered by unit tests. It is calculated by dividing the number of statements that have been executed by the total number of statements. A higher statement coverage value indicates that more of the codebase is being tested.

DIT (Depth of Inheritance Tree) is a measure of the depth of inheritance in the codebase. It is calculated by tracking the number of inheritance levels between the root class and the leaf classes.

Halstead Coupling Blow Up is a measure of the coupling between objects. It is calculated by dividing the number of identifiers shared between objects by the total number of identifiers in the system. A lower Halstead coupling blow up value indicates better cohesion and lower coupling.

RFC (RFC) is a measure of the number of requirements in the codebase. It is often used as a proxy for the size of the system.

$$171 - 5.2 \cdot \ln(\text{avgHV}) - 0.23 \cdot \text{avgCC}(g') - \\ 16.2 \cdot \ln(\text{avgLOC}) + 50 \cdot \sin(\sqrt{2.4 \cdot \text{perCM}})$$

HV: Halstead Volume

LOC: lines of code

CC:

perCM: % Comment Lines

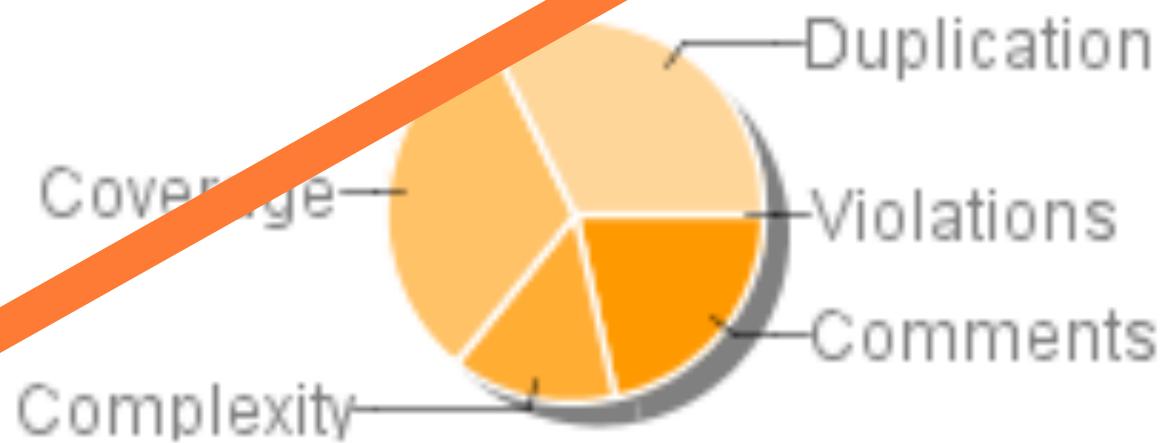
Cyclomatic Complexity

?

?

?

Technical Debt
\$ 14'605
29 man days



Best Practice: Goal-Oriented Analyses

- Objective
- Impact on the code is clear
- Actionable
- Clear relation to maintainability

```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg != null && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg != null && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```







Study



- Over 100 bugs in produktive Software



- 52% of all unintentional differences buggy

Juergens, Deissenboeck et al: *Do Code Clones Matter?* ICSE 2009

1659 LOC / 1301 SLOC

```
336: 419: 501: 553: 655: 749: 531: 514: 996: 1078: 1181: 1243: 1325: 1407: 1490: 1574: 1654: 1735: 1818: 1901: 1954: 1. B. der Parameter e_display richtig gesetzt sind
337: 420: 502: 554: 656: 750: 532: 515: 997: 1079: 1182: 1244: 1326: 1408: 1491: 1575: 1655: 1736: 1820: 1902: 1955:
338: 421: 503: 555: 657: 751: 533: 516: 998: 1080: 1183: 1245: 1327: 1409: 1492: 1576: 1656: 1737: 1821: 1903: 1956:
339: 422: 504: 556: 658: 752: 534: 517: 999: 1081: 1184: 1246: 1328: 1410: 1493: 1577: 1657: 1738: 1822: 1904: 1957:
340: 423: 505: 557: 659: 753: 535: 518: 1000: 1082: 1185: 1247: 1329: 1411: 1494: 1578: 1658: 1739: 1823: 1905: 1958:
341: 424: 506: 558: 660: 754: 536: 519: 1001: 1083: 1186: 1248: 1330: 1412: 1495: 1579: 1659: 1740: 1824: 1906: 1959:
342: 425: 507: 559: 661: 755: 537: 520: 1002: 1084: 1187: 1249: 1331: 1413: 1496: 1580: 1660: 1741: 1825: 1907: 1960:
343: 426: 508: 560: 662: 756: 538: 521: 1003: 1085: 1188: 1250: 1332: 1414: 1497: 1581: 1661: 1742: 1826: 1908: 1961:
344: 427: 509: 561: 663: 757: 539: 522: 1004: 1086: 1189: 1251: 1333: 1415: 1498: 1582: 1662: 1743: 1827: 1909: 1962:
345: 428: 510: 562: 664: 758: 540: 523: 1005: 1087: 1190: 1252: 1334: 1416: 1499: 1583: 1663: 1744: 1828: 1910: 1963:
346: 429: 511: 563: 665: 759: 541: 524: 1006: 1088: 1191: 1253: 1335: 1417: 1500: 1584: 1664: 1745: 1829: 1911: 1964:
347: 430: 512: 564: 666: 760: 542: 525: 1007: 1089: 1192: 1254: 1336: 1418: 1501: 1585: 1665: 1746: 1830: 1912: 1965:
348: 431: 513: 565: 667: 761: 543: 526: 1008: 1090: 1193: 1255: 1337: 1419: 1502: 1586: 1666: 1747: 1831: 1913: 1966:
349: 432: 514: 566: 668: 762: 544: 527: 1009: 1091: 1194: 1256: 1338: 1420: 1503: 1587: 1667: 1751: 1832: 1914: 1967:
350: 433: 515: 567: 669: 763: 545: 528: 1010: 1092: 1195: 1257: 1339: 1421: 1504: 1588: 1668: 1752: 1833: 1915: 1968:
351: 434: 516: 568: 670: 764: 546: 529: 1011: 1093: 1196: 1258: 1340: 1422: 1505: 1589: 1669: 1753: 1834: 1916: 1969:
352: 435: 517: 569: 671: 765: 547: 530: 1012: 1094: 1197: 1259: 1341: 1423: 1506: 1590: 1670: 1754: 1835: 1917: 1970:
353: 436: 518: 600: 672: 766: 548: 531: 1013: 1095: 1198: 1260: 1342: 1424: 1507: 1591: 1671: 1755: 1836: 1918: 1971:
354: 437: 519: 601: 673: 767: 549: 532: 1014: 1096: 1199: 1261: 1343: 1425: 1508: 1592: 1672: 1756: 1837: 1919: 1972:
355: 438: 520: 602: 674: 768: 550: 533: 1015: 1097: 1200: 1262: 1344: 1426: 1509: 1593: 1673: 1757: 1838: 1920: 1973:
356: 439: 521: 603: 675: 769: 551: 534: 1016: 1098: 1201: 1263: 1345: 1427: 1510: 1594: 1674: 1758: 1839: 1921: 1974:
357: 440: 522: 604: 676: 770: 552: 535: 1017: 1099: 1202: 1264: 1346: 1428: 1511: 1595: 1675: 1759: 1840: 1922: 1975:
358: 441: 523: 605: 677: 771: 553: 536: 1018: 1100: 1203: 1265: 1347: 1429: 1512: 1596: 1676: 1760: 1841: 1923: 1976:
359: 442: 524: 606: 678: 772: 554: 537: 1019: 1101: 1204: 1266: 1348: 1430: 1513: 1597: 1677: 1761: 1842: 1924: 1977:
360: 443: 525: 607: 679: 773: 555: 538: 1020: 1102: 1205: 1267: 1349: 1431: 1514: 1598: 1678: 1762: 1843: 1925: 1978:
361: 444: 526: 608: 680: 774: 556: 539: 1021: 1103: 1206: 1268: 1350: 1432: 1515: 1599: 1679: 1763: 1844: 1926: 1979:
362: 445: 527: 609: 681: 775: 557: 540: 1022: 1104: 1207: 1269: 1351: 1433: 1516: 1600: 1680: 1770: 1845: 1927: 1980:
363: 446: 528: 610: 682: 776: 558: 541: 1023: 1105: 1208: 1270: 1352: 1434: 1517: 1601: 1681: 1771: 1846: 1928: 1981:
364: 447: 529: 611: 683: 777: 559: 542: 1024: 1106: 1209: 1271: 1353: 1435: 1518: 1602: 1682: 1772: 1847: 1929: 1982:
365: 448: 530: 612: 684: 778: 560: 543: 1025: 1107: 1210: 1272: 1354: 1436: 1519: 1603: 1683: 1773: 1848: 1930: 1983:
366: 449: 531: 613: 685: 779: 561: 544: 1026: 1108: 1211: 1273: 1355: 1437: 1520: 1604: 1684: 1774: 1849: 1931: 1984:
367: 450: 532: 614: 686: 780: 562: 545: 1027: 1109: 1212: 1274: 1356: 1438: 1521: 1605: 1685: 1775: 1850: 1932: 1985:
368: 451: 533: 615: 687: 781: 563: 546: 1028: 1110: 1213: 1275: 1357: 1439: 1522: 1606: 1686: 1776: 1851: 1933: 1986:
369: 452: 534: 616: 688: 782: 564: 547: 1029: 1111: 1214: 1276: 1358: 1440: 1523: 1607: 1687: 1777: 1852: 1934: 1987:
370: 453: 535: 617: 689: 783: 565: 548: 1030: 1112: 1215: 1277: 1359: 1441: 1524: 1608: 1688: 1778: 1853: 1935: 1988:
371: 454: 536: 618: 700: 784: 566: 549: 1031: 1113: 1216: 1278: 1360: 1442: 1525: 1609: 1689: 1779: 1854: 1936: 1989:
372: 455: 537: 619: 701: 785: 567: 550: 1032: 1114: 1217: 1279: 1361: 1443: 1526: 1610: 1690: 1780: 1855: 1937: 1990:
373: 456: 538: 620: 702: 786: 568: 551: 1033: 1115: 1218: 1280: 1362: 1444: 1527: 1611: 1691: 1781: 1856: 1938: 1991:
374: 457: 539: 621: 703: 787: 569: 552: 1034: 1116: 1219: 1281: 1363: 1445: 1528: 1612: 1692: 1782: 1857: 1939: 1992:
375: 458: 540: 622: 704: 788: 570: 553: 1035: 1117: 1220: 1282: 1364: 1446: 1529: 1613: 1693: 1783: 1858: 1940: 1993:
376: 459: 541: 623: 705: 789: 571: 554: 1036: 1118: 1221: 1283: 1365: 1447: 1530: 1614: 1694: 1784: 1859: 1941: 1994:
377: 460: 542: 624: 706: 790: 572: 555: 1037: 1119: 1222: 1284: 1366: 1448: 1531: 1615: 1695: 1785: 1860: 1942: 1995:
378: 461: 543: 625: 707: 791: 573: 556: 1038: 1120: 1223: 1285: 1367: 1449: 1532: 1616: 1696: 1786: 1861: 1943: 1996:
379: 462: 544: 626: 708: 792: 574: 557: 1039: 1121: 1224: 1286: 1368: 1450: 1533: 1617: 1697: 1787: 1862: 1944: 1997:
380: 463: 545: 627: 709: 793: 575: 558: 1040: 1122: 1225: 1287: 1369: 1451: 1534: 1618: 1698: 1788: 1863: 1945: 1998:
381: 464: 546: 628: 710: 794: 576: 559: 1041: 1123: 1226: 1288: 1370: 1452: 1535: 1619: 1699: 1789: 1864: 1946: 1999:
382: 465: 547: 629: 711: 795: 577: 560: 1042: 1124: 1227: 1289: 1371: 1453: 1536: 1620: 1700: 1790: 1870: 1950: 1990:
383: 466: 548: 630: 712: 796: 578: 561: 1043: 1125: 1228: 1290: 1372: 1454: 1537: 1621: 1701: 1791: 1871: 1951: 1991:
384: 467: 549: 631: 713: 797: 579: 562: 1044: 1126: 1229: 1291: 1373: 1455: 1538: 1622: 1702: 1792: 1872: 1952: 1992:
385: 468: 550: 632: 714: 798: 580: 563: 1045: 1127: 1230: 1292: 1374: 1456: 1539: 1623: 1703: 1793: 1873: 1953: 1993:
386: 469: 551: 633: 715: 799: 581: 564: 1046: 1128: 1231: 1293: 1375: 1457: 1540: 1624: 1704: 1794: 1874: 1954: 1994:
387: 470: 552: 634: 716: 800: 582: 565: 1047: 1129: 1232: 1294: 1376: 1458: 1541: 1625: 1705: 1795: 1875: 1955: 1995:
388: 471: 553: 635: 717: 801: 583: 566: 1048: 1130: 1233: 1295: 1377: 1459: 1542: 1626: 1706: 1796: 1876: 1956: 1996:
389: 472: 554: 636: 718: 802: 584: 567: 1049: 1131: 1234: 1296: 1378: 1460: 1543: 1627: 1707: 1797: 1877: 1957: 1997:
390: 473: 555: 637: 720: 803: 585: 568: 1050: 1132: 1235: 1297: 1379: 1461: 1544: 1628: 1708: 1798: 1878: 1958: 1998:
391: 474: 556: 638: 721: 804: 586: 569: 1051: 1133: 1236: 1298: 1380: 1462: 1545: 1629: 1709: 1799: 1879: 1959: 1999:
392: 475: 557: 639: 722: 805: 587: 570: 1052: 1134: 1237: 1299: 1381: 1463: 1546: 1630: 1710: 1800: 1780: 1960: 1990:
393: 476: 558: 640: 723: 806: 588: 571: 1053: 1135: 1238: 1300: 1382: 1464: 1547: 1631: 1711: 1801: 1781: 1961: 1991:
394: 477: 559: 641: 724: 807: 589: 572: 1054: 1136: 1239: 1301: 1383: 1465: 1548: 1632: 1712: 1802: 1782: 1962: 1992:
395: 478: 560: 642: 725: 808: 590: 573: 1055: 1137: 1240: 1302: 1384: 1466: 1549: 1633: 1713: 1803: 1783: 1963: 1993:
396: 479: 561: 643: 726: 809: 591: 574: 1056: 1138: 1241: 1303: 1385: 1467: 1550: 1634: 1714: 1804: 1784: 1964: 1994:
397: 480: 562: 644: 727: 810: 592: 575: 1057: 1139: 1242: 1304: 1386: 1468: 1551: 1635: 1715: 1805: 1785: 1965: 1995:
398: 481: 563: 645: 728: 811: 593: 576: 1058: 1140: 1243: 1305: 1387: 1469: 1552: 1636: 1716: 1806: 1786: 1966: 1996:
399: 482: 564: 646: 729: 812: 594: 577: 1059: 1141: 1244: 1306: 1388: 1470: 1553: 1637: 1717: 1807: 1787: 1967: 1997:
400: 483: 565: 647: 730: 813: 595: 578: 1060: 1142: 1245: 1307: 1389: 1471: 1554: 1638: 1718: 1808: 1788: 1968: 1998:
401: 484: 566: 648: 731: 814: 596: 579: 1061: 1143: 1246: 1308: 1390: 1472: 1555: 1639: 1719: 1809: 1789: 1969: 1999:
402: 485: 567: 649: 732: 815: 597: 580: 1062: 1144: 1247: 1309: 1391: 1473: 1556: 1640: 1720: 1810: 1790: 1970: 1990:
403: 486: 568: 650: 733: 816: 598: 581: 1063: 1145: 1248: 1310: 1392: 1474: 1557: 1641: 1721: 1811: 1791: 1971: 1991:
404: 487: 569: 651: 734: 817: 599: 582: 1064: 1146: 1249: 1311: 1393: 1475: 1558: 1642: 1722: 1812: 1792: 1972: 1992:
405: 488: 570: 652: 735: 818: 600: 583: 1065: 1147: 1250: 1312: 1394: 1476: 1559: 1643: 1723: 1813: 1793: 1973: 1993:
406: 489: 571: 653: 736: 819: 601: 584: 1066: 1148: 1251: 1313: 1395: 1477: 1560: 1644: 1724: 1814: 1794: 1974: 1994:
407: 490: 572: 654: 737: 820: 602: 585: 1067: 1149: 1252: 1314: 1396: 1478: 1561: 1645: 1725: 1815: 1795: 1975: 1995:
408: 491: 573: 655: 738: 821: 603: 586: 1068: 1150: 1253: 1315: 1397: 1479: 1562: 1646: 1726: 1816: 1796: 1976: 1996:
409: 492: 574: 656: 739: 822: 604: 587: 1069: 1151: 1254: 1316: 1398: 1480: 1563: 1647: 1727: 1817: 1797: 1977: 1997:
410: 493: 575: 657: 740: 823: 605: 588: 1070: 1152: 1255: 1317: 1399: 1481: 1564: 1648: 1728: 1818: 1798: 1978: 1998:
411: 494: 576: 658: 741: 824: 606: 589: 1071: 1153: 1256: 1318: 1400: 1482: 1565: 1649: 1729: 1819: 1799: 1979: 1999:
412: 495: 577: 659: 742: 825: 607: 590: 1072: 1154: 1257: 1319: 1401: 1483: 1566: 1650: 1730: 1820: 1800: 1790: 1980: 1990:
413: 496: 578: 660: 743: 826: 608: 591: 1073: 1155: 1258: 1320: 1402: 1484: 1567: 1651: 1731: 1821: 1801: 1791: 1981: 1991:
414: 497: 579: 661: 744: 827: 609: 592: 1074: 1156: 1259: 1321: 1403: 1485: 1568: 1652: 1732: 1822: 1802: 1792: 1982: 1992:
415: 498: 580: 662: 745: 828: 610: 593: 1075: 1157: 1260: 1322: 1404: 1486: 1569: 1653: 1733: 1823: 1803: 1793: 1983: 1993:
416: 499: 581: 663: 746: 829: 611: 594: 1076: 1158: 1261: 1323: 1405: 1487: 1570: 1654: 1734: 1824: 1804: 1794: 1984: 1994:
417: 500: 582: 664: 747: 830: 612: 595: 1077: 1159: 1262: 1324: 1406: 1488: 1571: 1655: 1735: 1825: 1805: 1795: 1985: 1995:
418: 501: 583: 665: 748: 831: 613: 596: 1078: 1160: 1263: 1325: 1407: 1489: 1572: 1656: 1736: 1826: 1806: 1796: 1986: 1996:
```

```

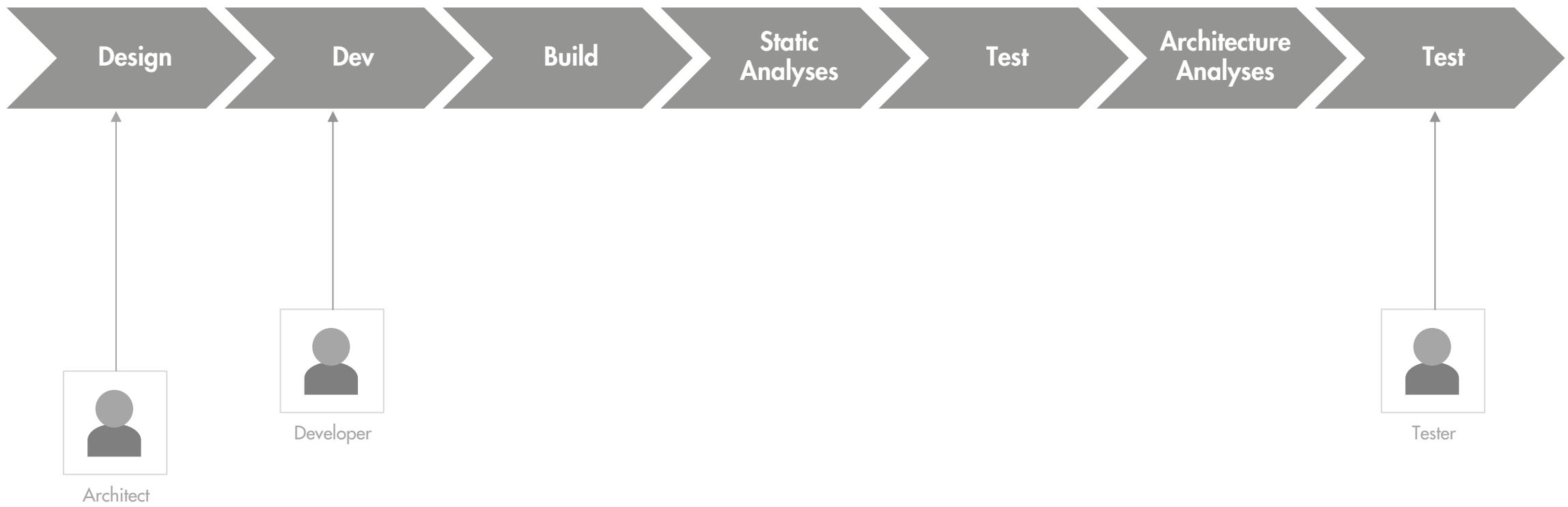
1418: 1 IF NOT lt_zeile_num IS INITIAL.
1419:   Anzahl = 1
1420: 2 IF l_anz = 1.
1421:   LOOP AT lt_zeile_num INTO ls_zeile_num.
1422:     READ TABLE gt_gui_liste_angebot_0500
1423:       INTO ls_gui_liste_angebot INDEX ls_zeile_num
1424:
1425: 4 IF ls_gui_liste_angebot-aufotyp EQ con_angebot_
1426:   OR p_vbsto EQ con_ein.
1427: * geändertes Verschoben bis Datum in Datenbanktabelle :
1428:   CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
1429:     EXPORTING
1430:       input = ls_gui_liste_angebot-vbeln
1431:     IMPORTING
1432:       output = l_vbeln.
1433:
1434:   SELECT * FROM /guigra INTO TABLE lt_
1435:     WHERE vbeln EQ l_vbeln
1436:     AND aktiv EQ con_ein.
1437:
1438: 5 IF sy-subrc IS INITIAL.
1439:   LOOP AT lt_guigra ASSIGNING <ls_guigra>
1440:     WHERE NOT verdat IS INITIAL.
1441: * Es ist ein "Verschoben bis Datum" vorhanden
1442:   EXIT.
1443: ENDLOOP.
1444:
1445: 6 IF sy-subrc IS INITIAL.
1446: * Es ist ein "Verschoben bis Datum" vorhanden
1447:   ls_fields-tabname = '/GUIGRA'.
1448:   ls_fields-fieldname = 'VERDAT'.
1449:   ls_fields-value = sy-datum.
1450:
1451:   APPEND ls_fields TO lt_fields.
1452: * Bei Änderung Verschoben bis Datum Eingabepopup aufmachen
1453:   CALL FUNCTION 'POPUP_GET_VALUES_USER_CHECK_DATE'
1454:     EXPORTING
1455:       formname      = 'CHECK_DATE_VERDAT'
1456:       popup_title   = text-ver
1457:       programname   = '/GUI_LISTE_ANGEBOT'
1458:
1459:     IMPORTING
1460:       returncode    = l_returncode
1461:     TABLES
1462:       fields        = lt_fields
1463:     EXCEPTIONS
1464:       error_in_fields = 1
1465:       OTHERS         = 2.
1466:
1467: 7 IF sy-subrc IS INITIAL AND l_returncode NE con_a.
1468:   READ TABLE lt_fields INTO ls_fields INDEX 1.
1469:
1470: 8 IF sy-subrc IS INITIAL.
1471:   LOOP AT lt_guigra ASSIGNING <ls_guigra>
1472:     WHERE NOT verdat IS INITIAL.
1473:
1474: 9 IF ls_fields-value EQ space.
1475:   CLEAR <ls_guigra>-verdat.
1476:
1477: 10 ELSE.
1478:   <ls_guigra>-verdat = ls_fields-value.
1479:
1480: ENDIF.
1481: ENDLOOP.
1482: IF sy-subrc IS INITIAL.
1483:   UPDATE /guigra FROM TABLE lt_guigra.
1484: ENDIF.
1485: IF sy-subrc IS INITIAL.
1486:   COMMIT WORK.
1487: ENDIF.
1488:
1489: ELSE.
1490:   * keine Änderungen vorgenommen
1491:   MESSAGE i255(/) /40_reports DISPLAY LIKE 'W'.
1492: ENDIF.
1493:
1494: ELSE.
1495:   MESSAGE e258(/) /40_reports DISPLAY LIKE 'E'.
1496: ENDIF.
1497:
1498: ELSE.
1499:   MESSAGE e256(/) /40_reports DISPLAY LIKE 'E'.
1500: ENDIF.
1501:
1502: ELSE.
1503:   MESSAGE e253(/) /40_reports DISPLAY LIKE 'E'.
1504: ENDIF.
1505: ELSE.
1506:   MESSAGE e028(/) /40_reports DISPLAY LIKE 'E'.
1507: ENDIF.

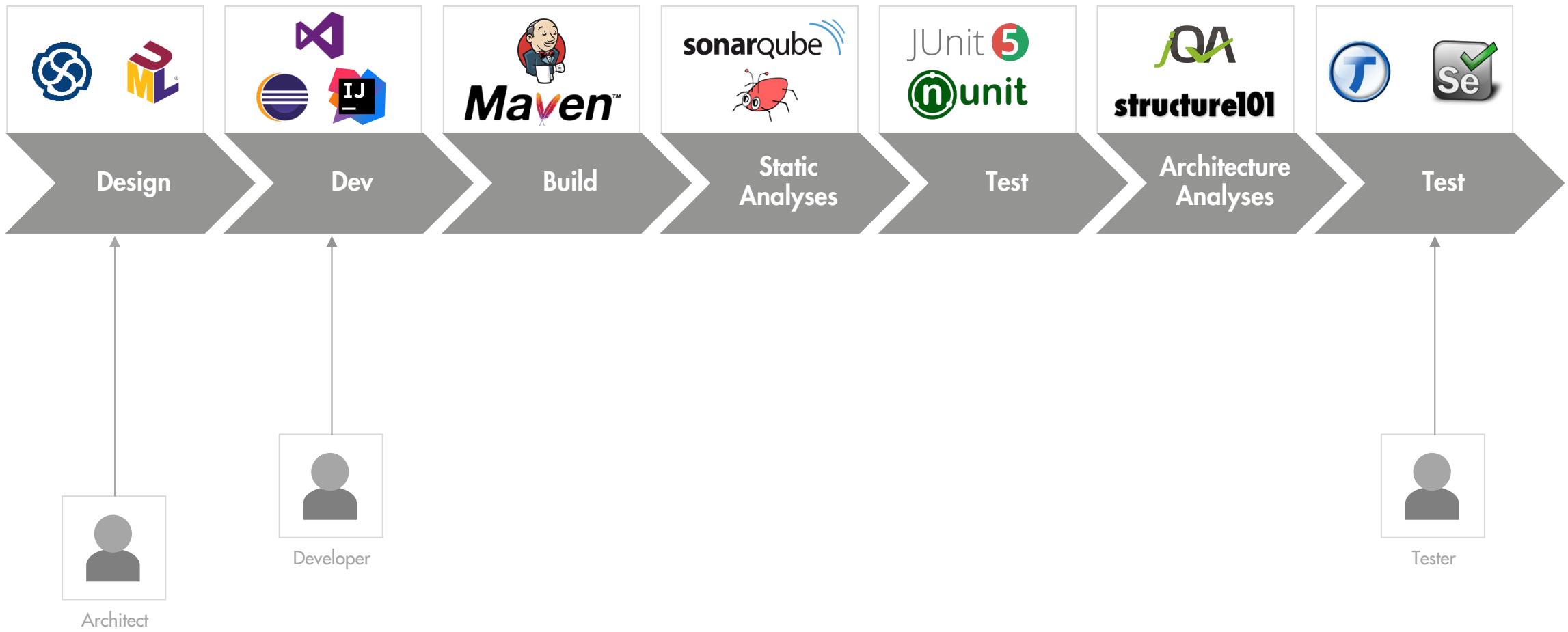
```

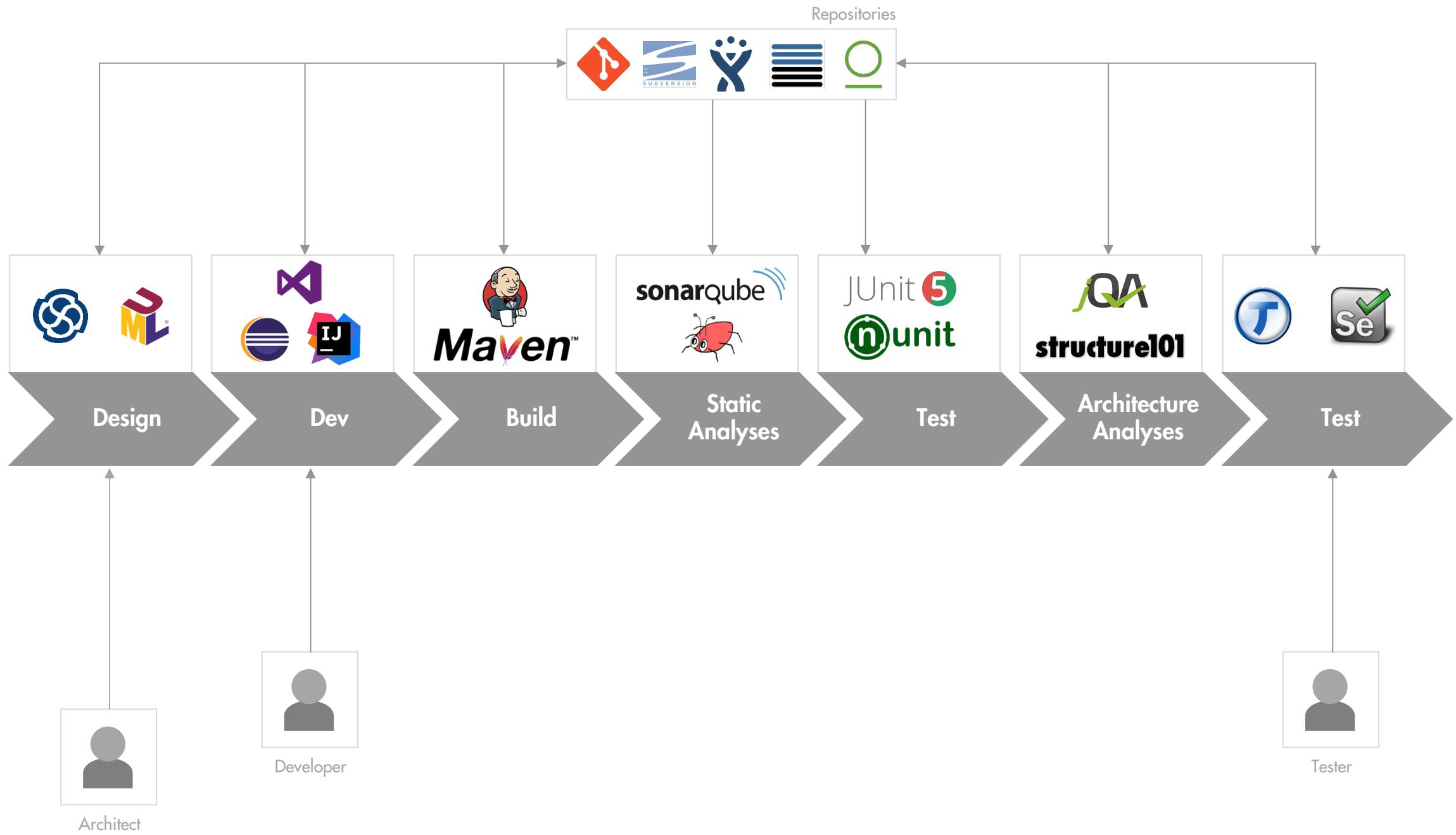

Agenda

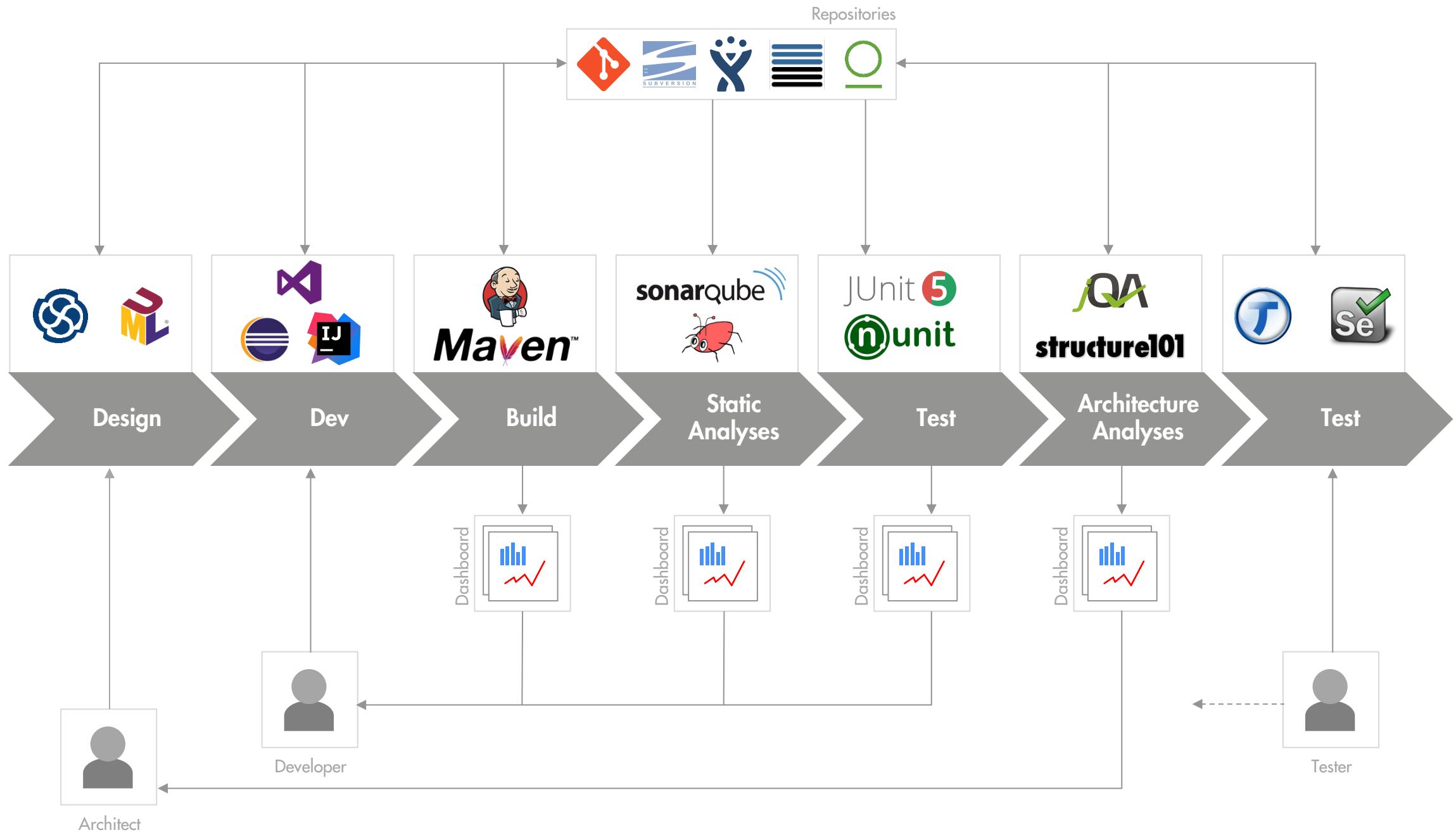
- 11:40 Introduction
- 11:45 Static Analysis for Quality Improvement
- ▶ **12:05 Continuous Quality Improvement**
- 12:50 Innovation & Improvement
- 13:10 Q&A
- 13:20 End of Lecture

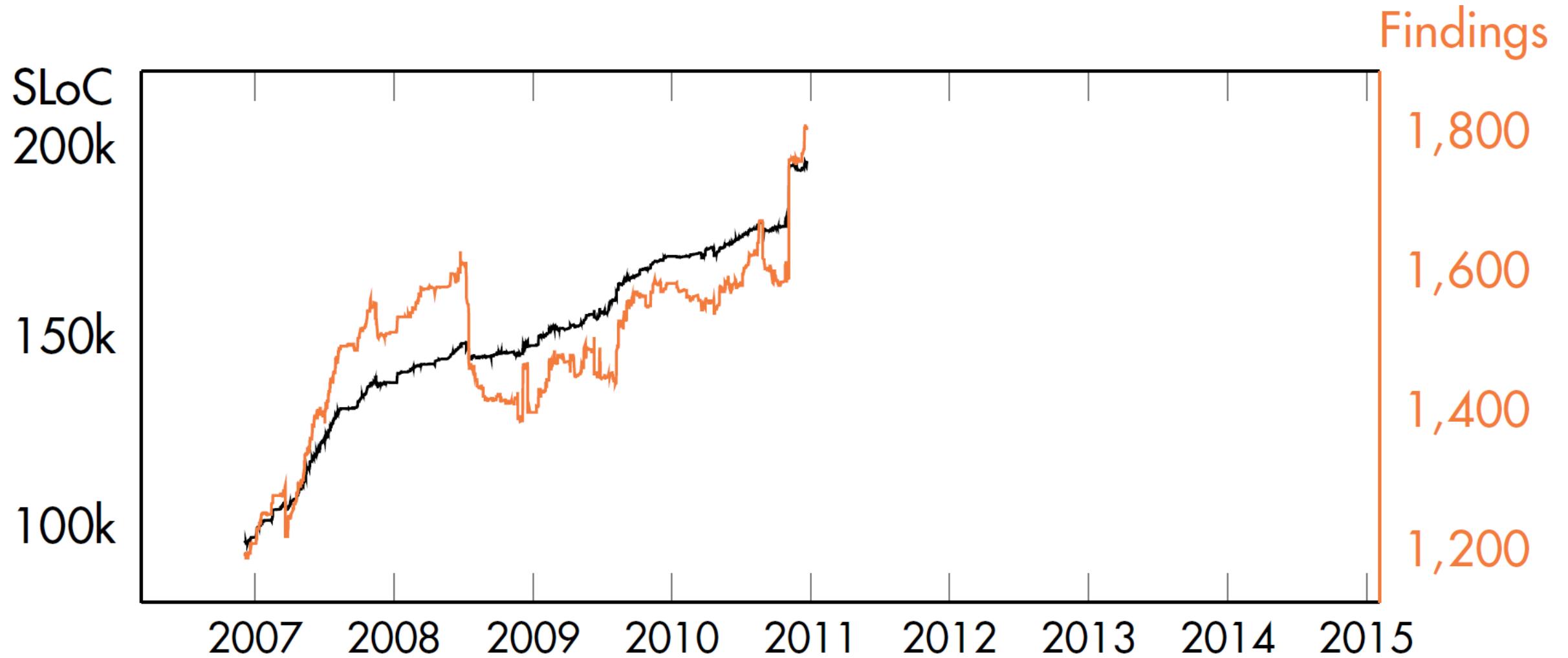


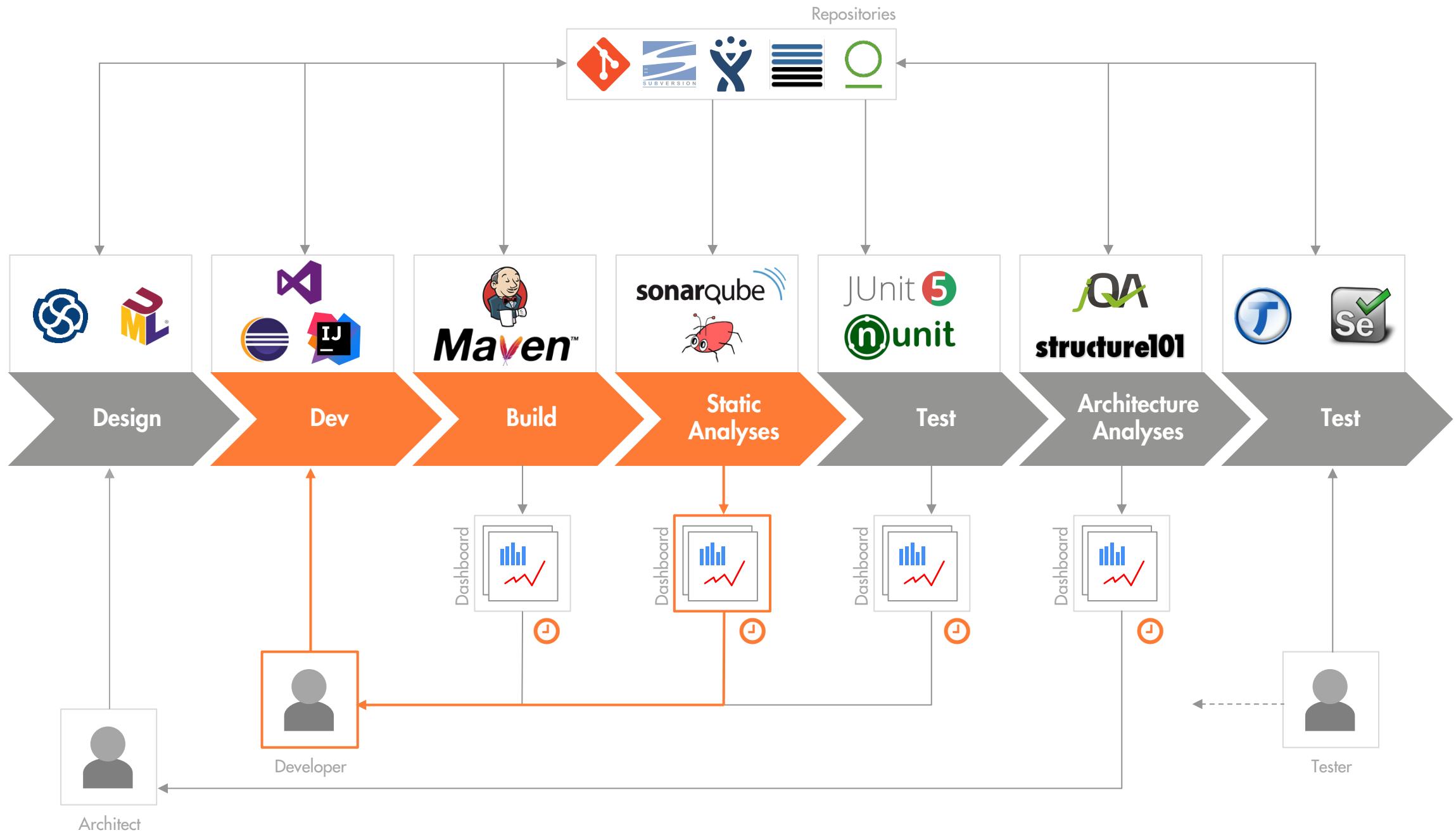


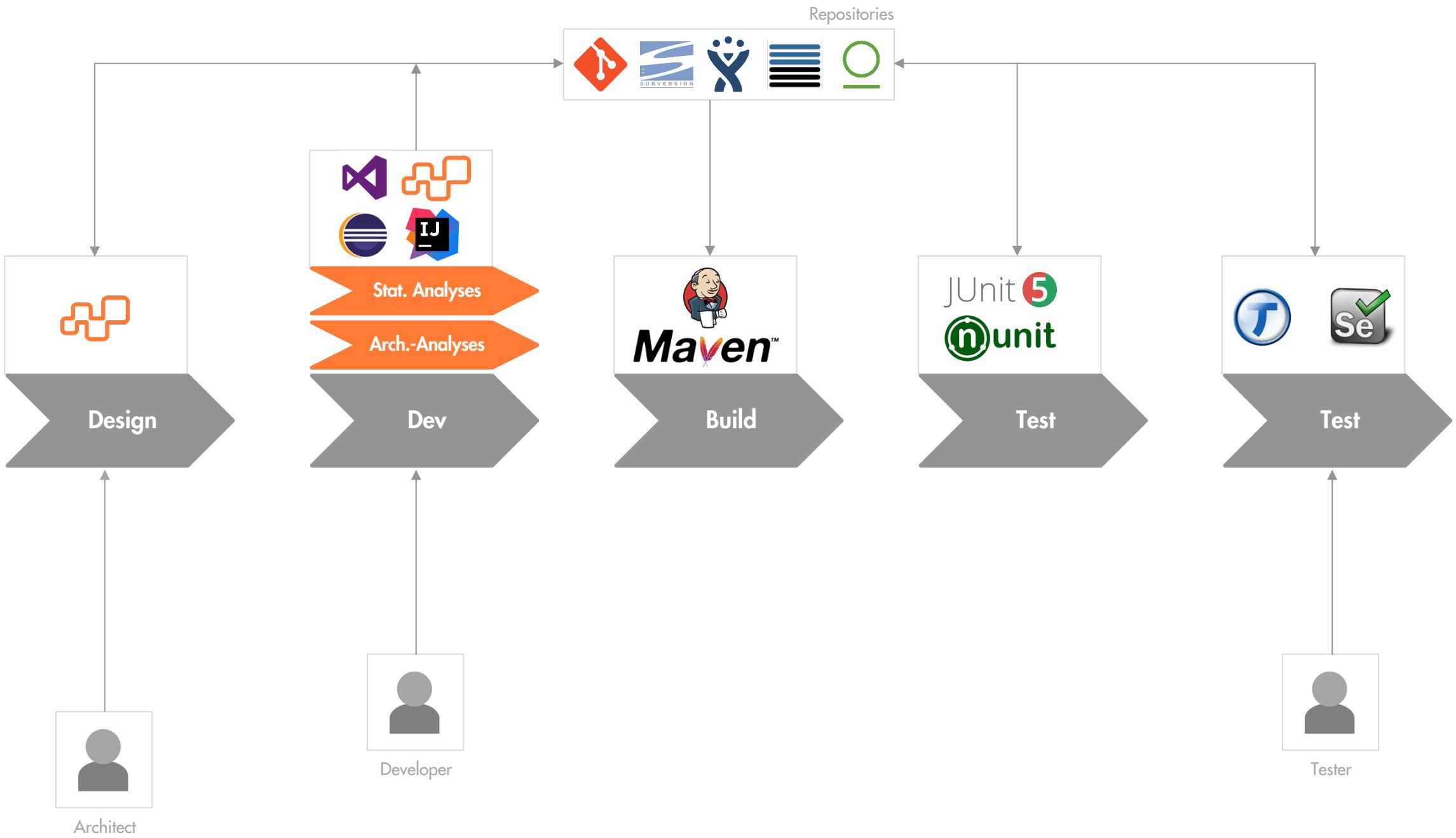


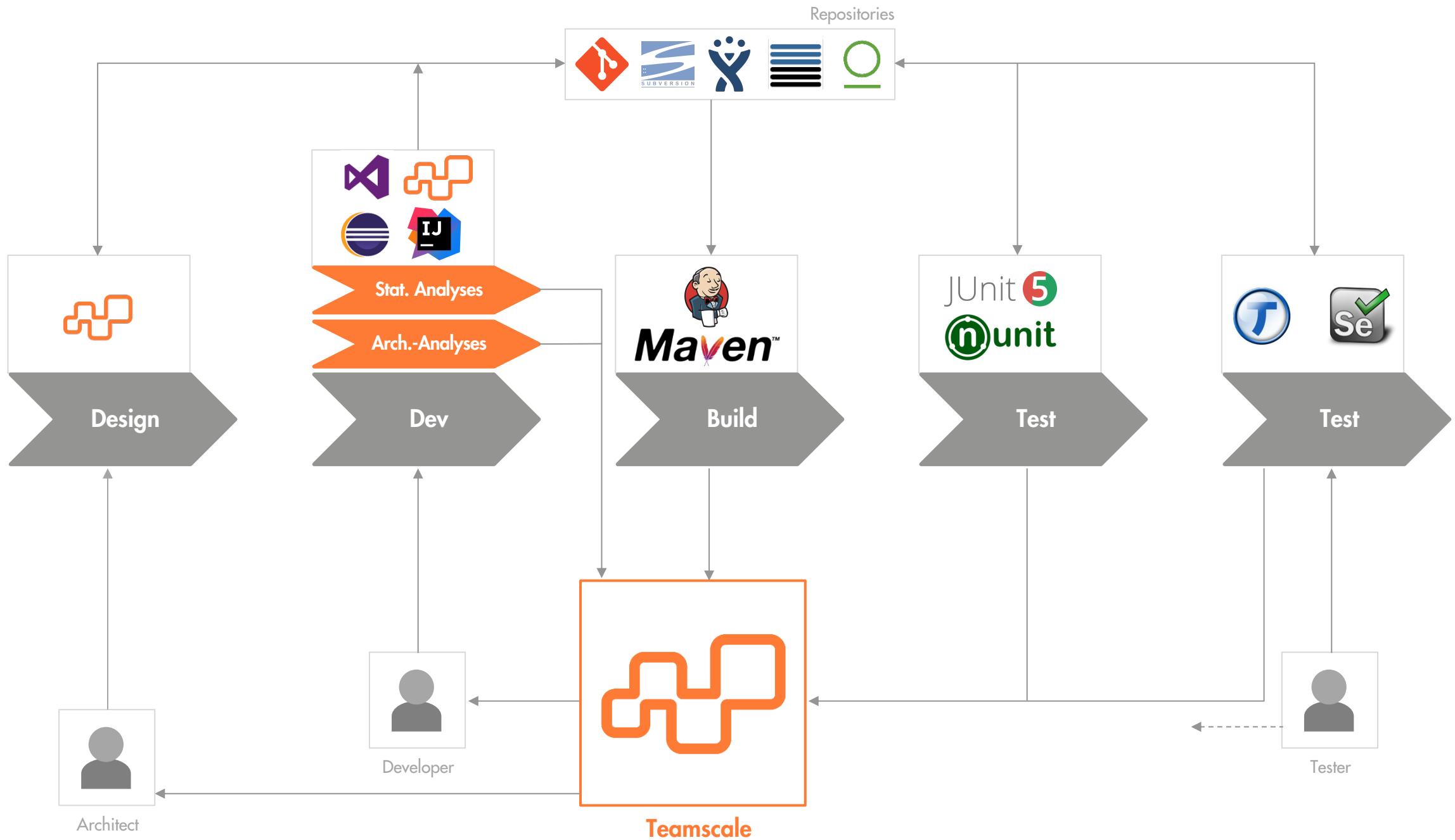


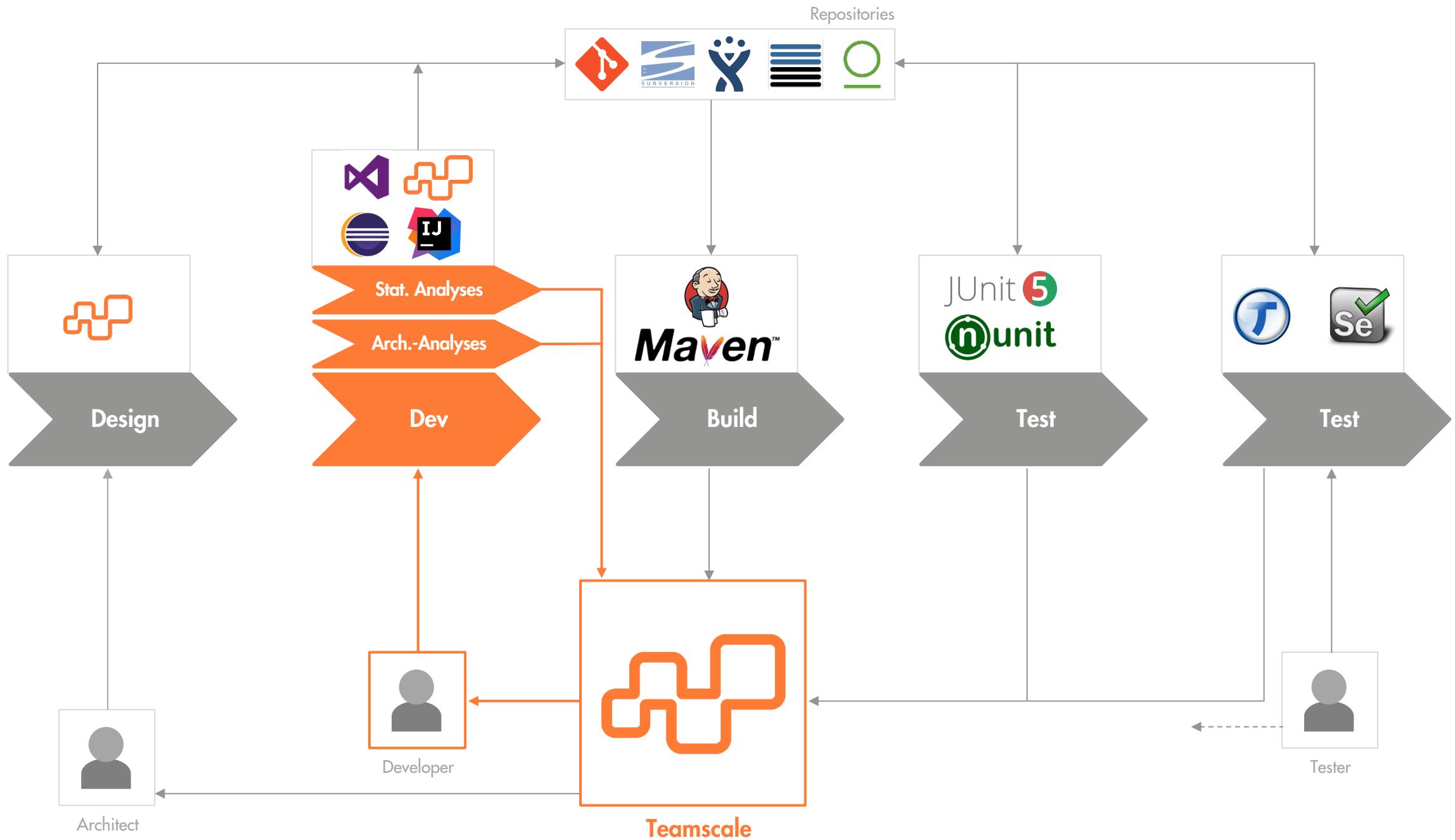








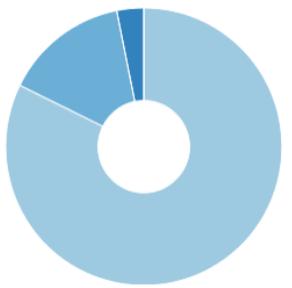




Demo

<https://demo.teamscale.com>

LOC Distribution for cqse-all



Clone Coverage for cqse-all

2.4%
+2.40%

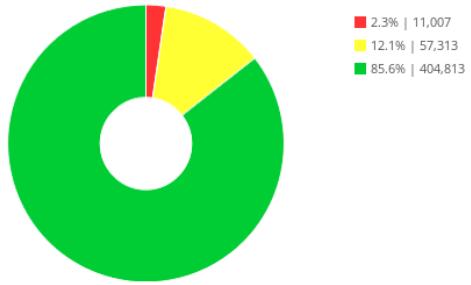
Findings Density for cqse-all

21.9
+21.9

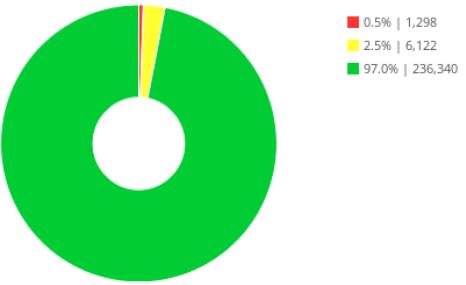
Test Coverage for cqse-all

71.8%
+0.05%

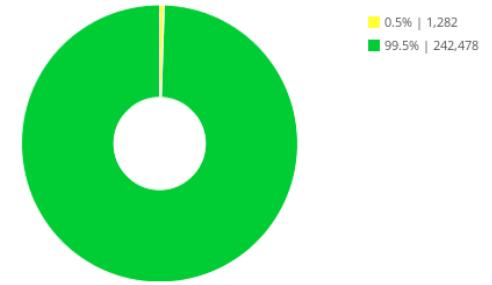
File Length for cqse-all



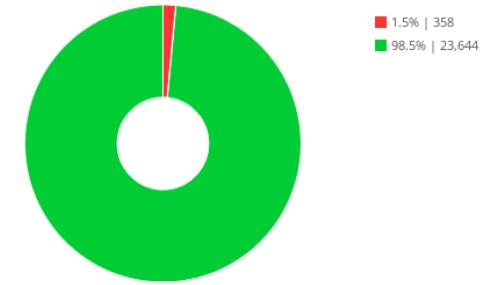
Method Length for cqse-all



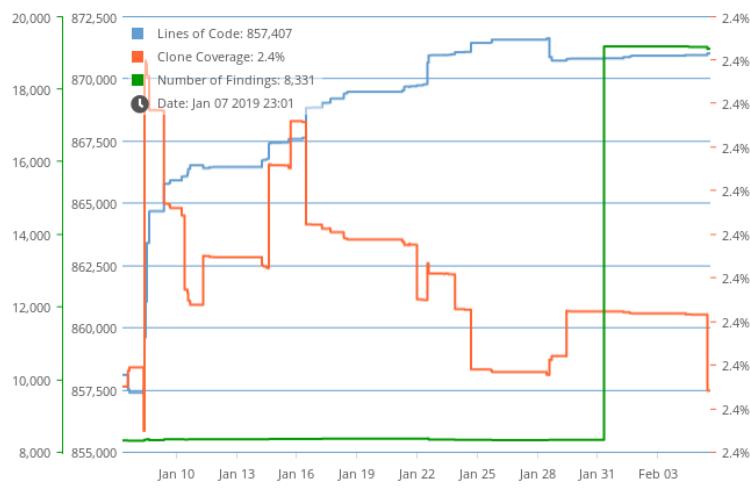
Nesting Depth for cqse-all



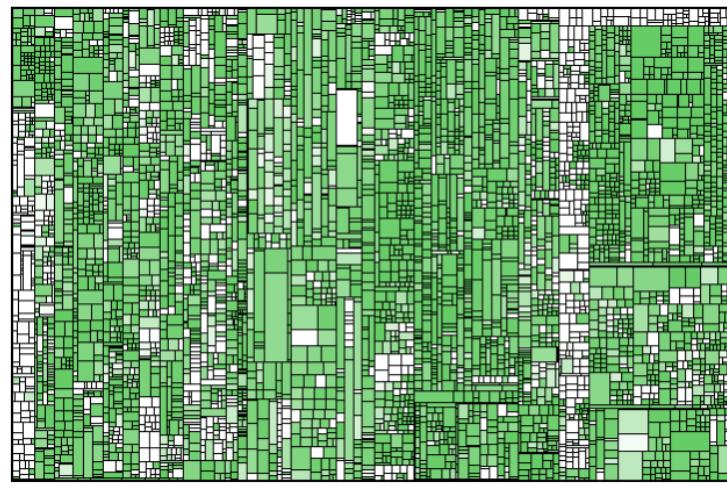
Comment Completeness for cqse-all



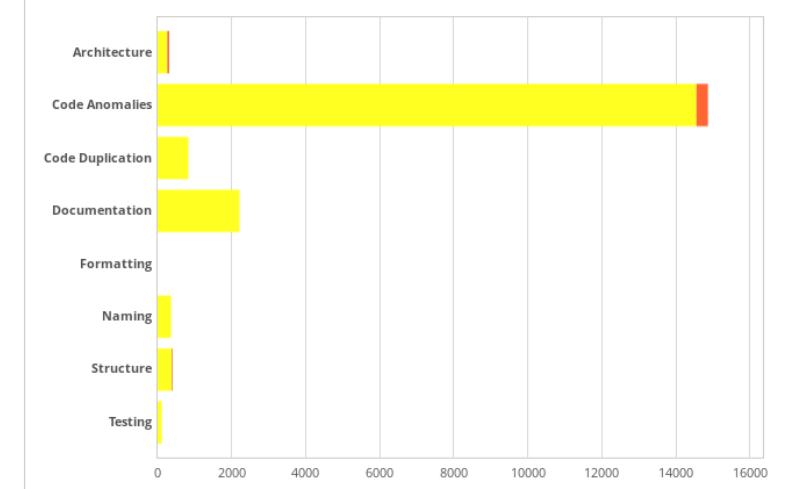
30 Day Trend for cqse-all

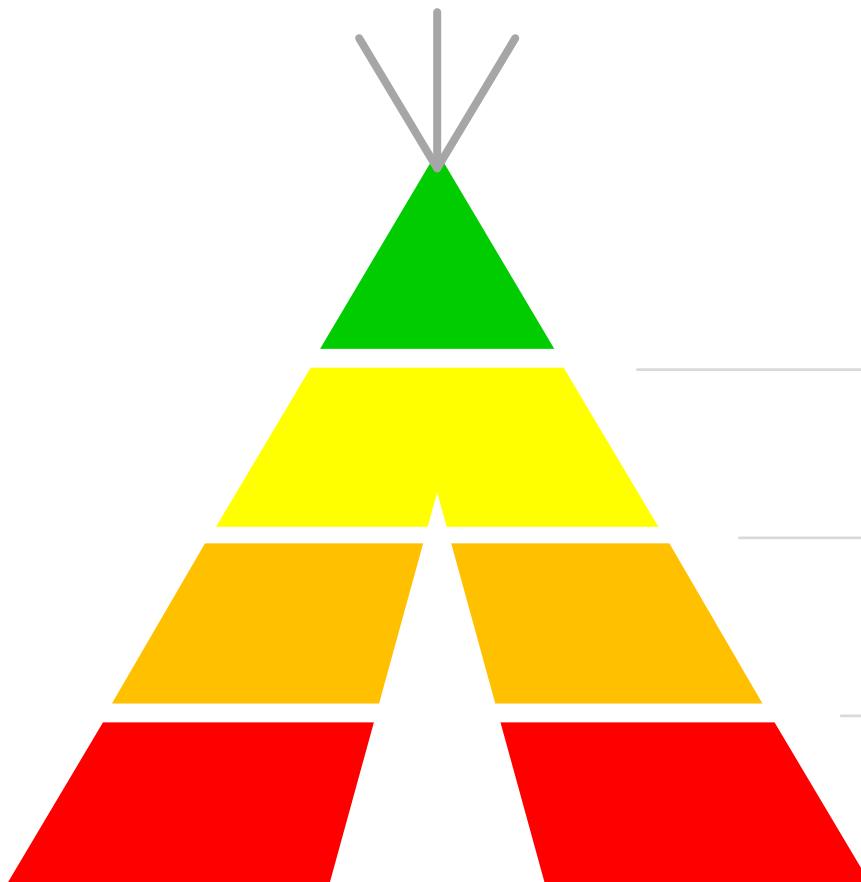


Test Coverage Treemap for cqse-all



Findings Summary Bar Chart for cqse-all



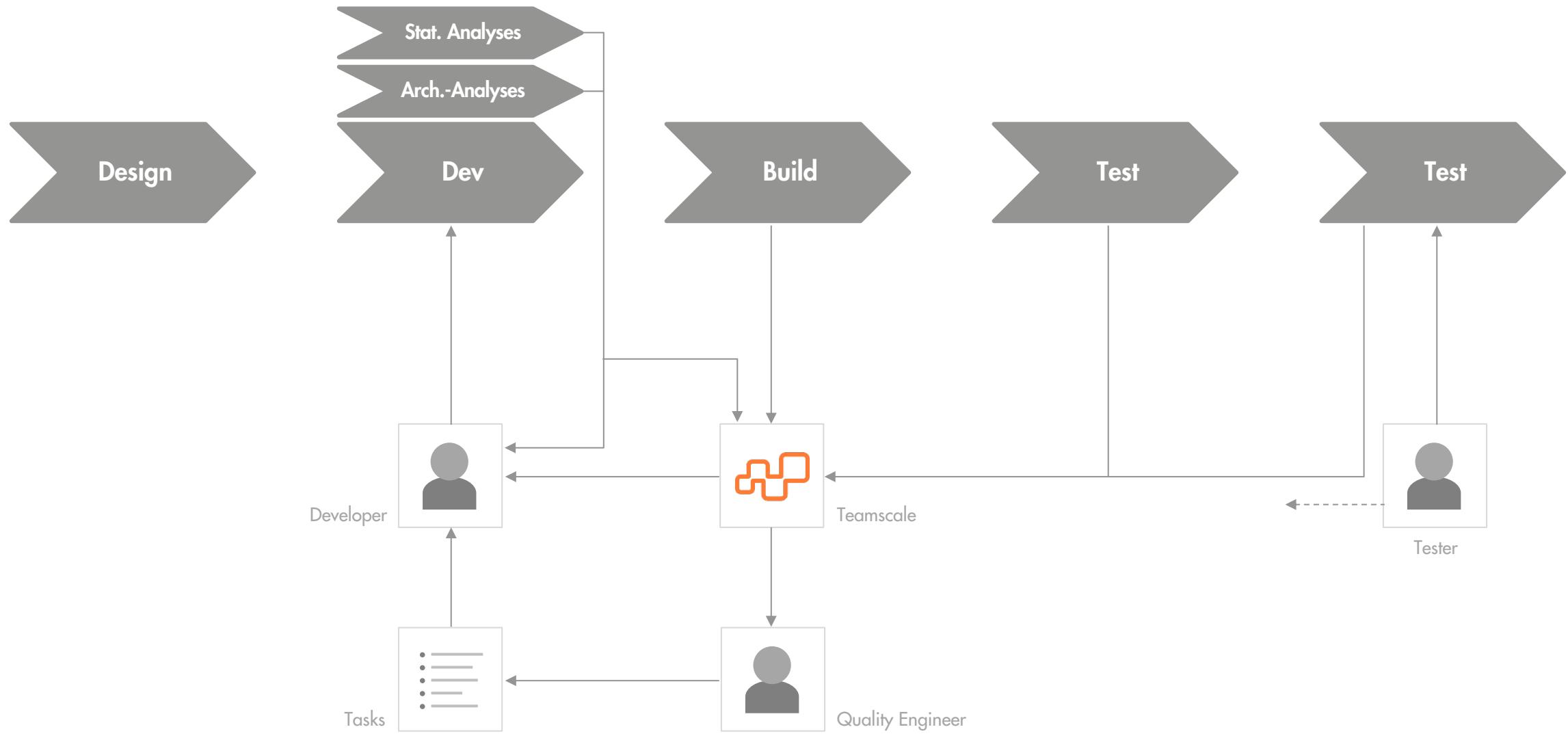


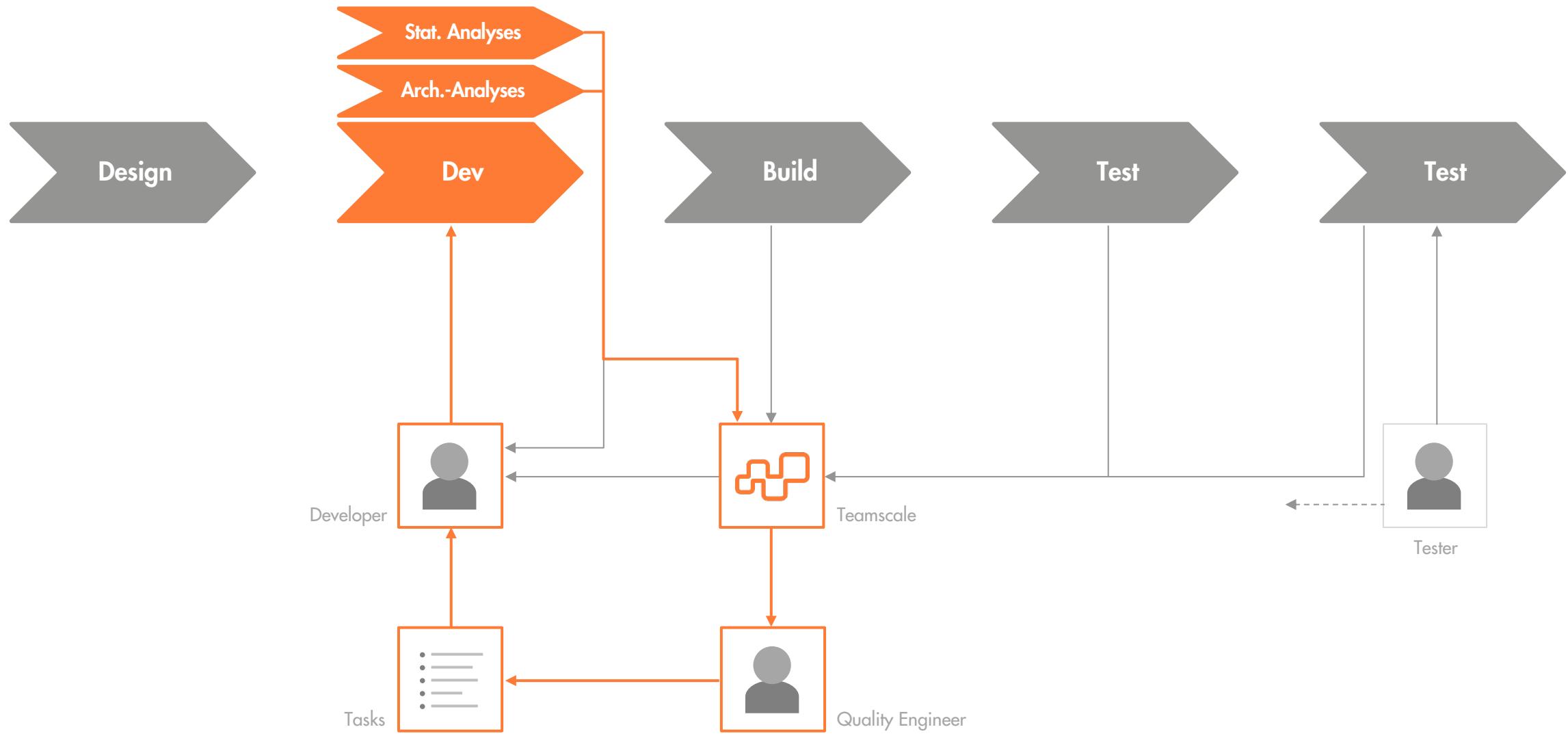
No Findings

No Findings in Changed Code

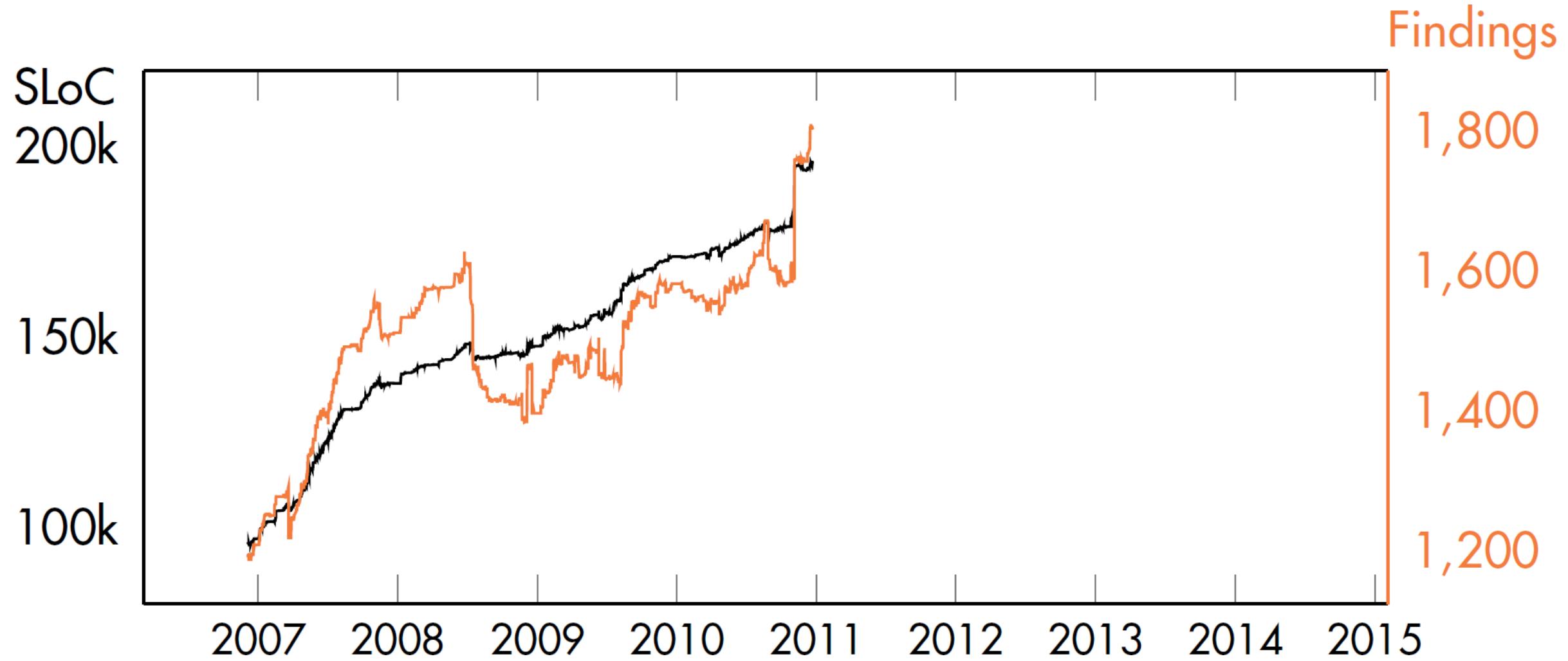
No New Findings

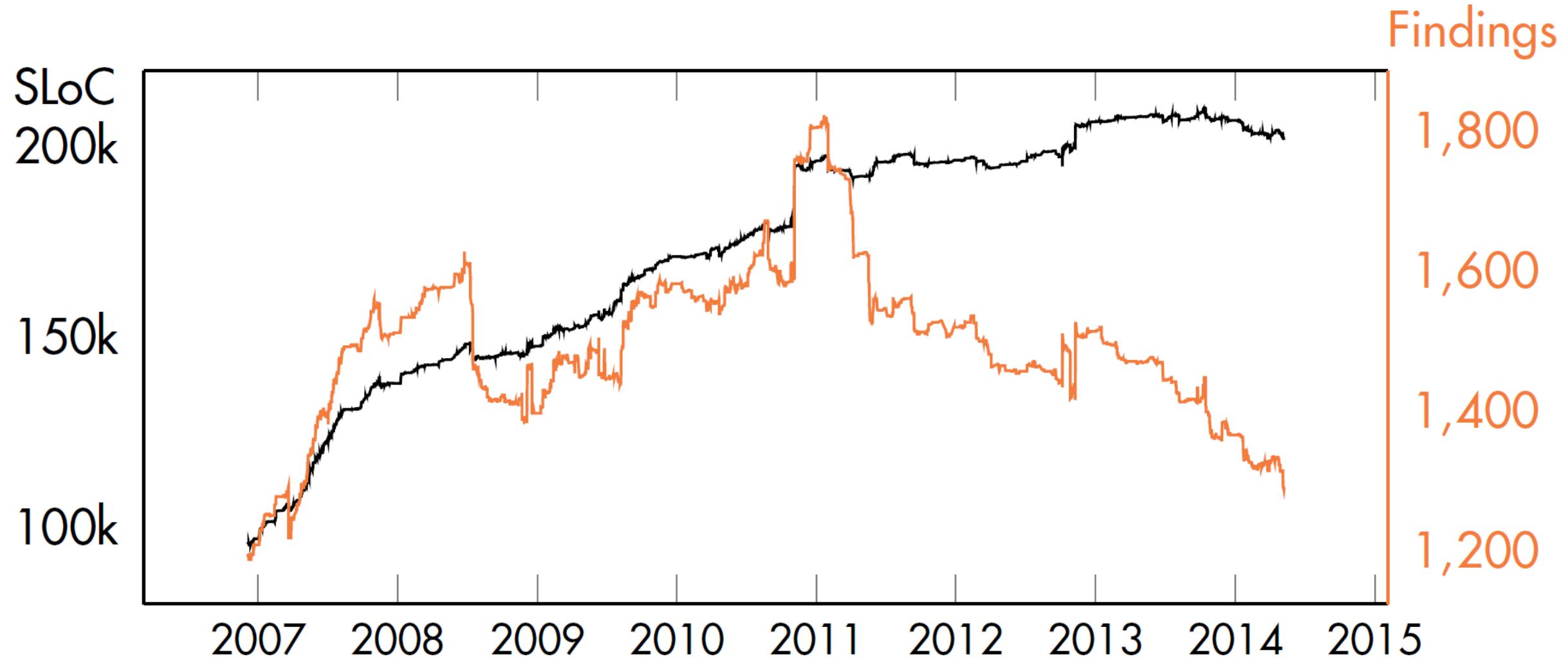
I don't care

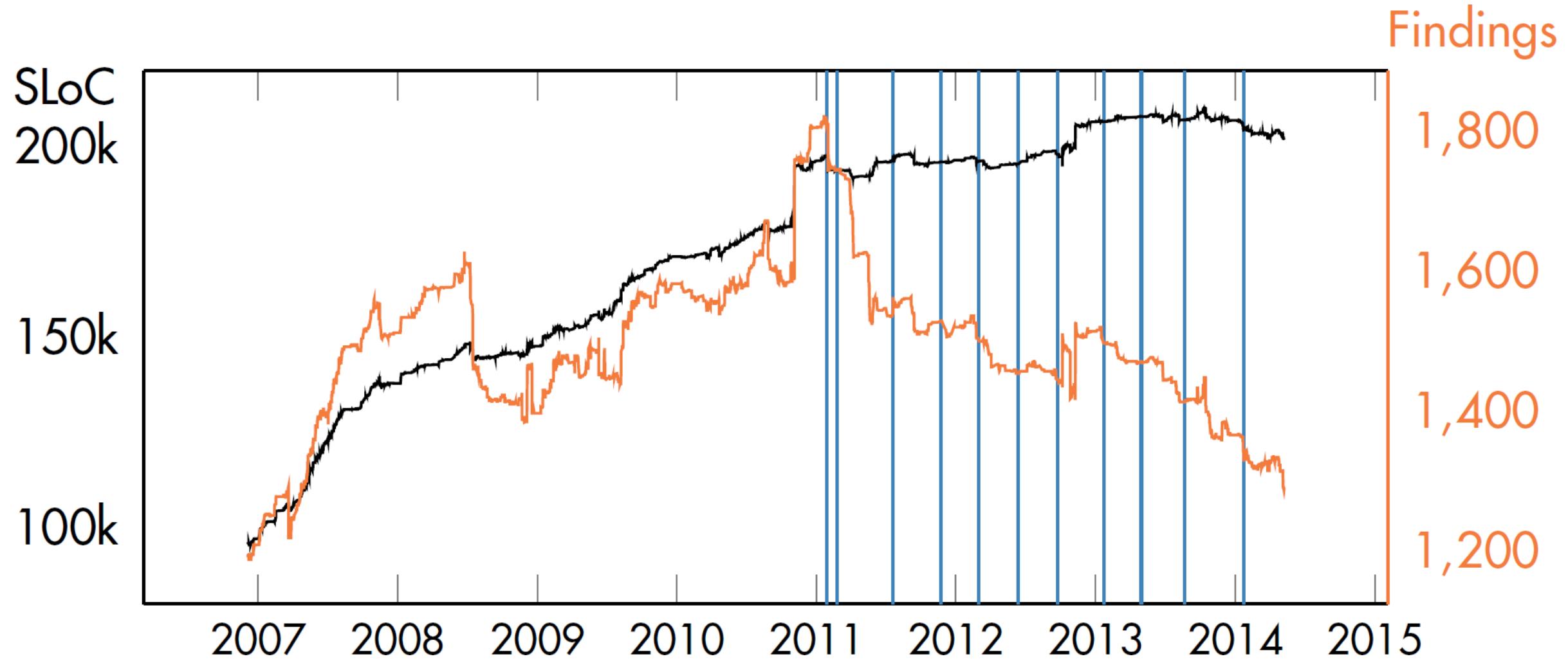




Demo







Agenda

- 11:40 Introduction
- 11:45 Static Analysis for Quality Improvement
- 12:05 Continuous Quality Improvement
- 12:50 Innovation & Improvement**
- 13:10 Q&A
- 13:20 End of Lecture



www.thesisguide.org

THEESIS GUIDE

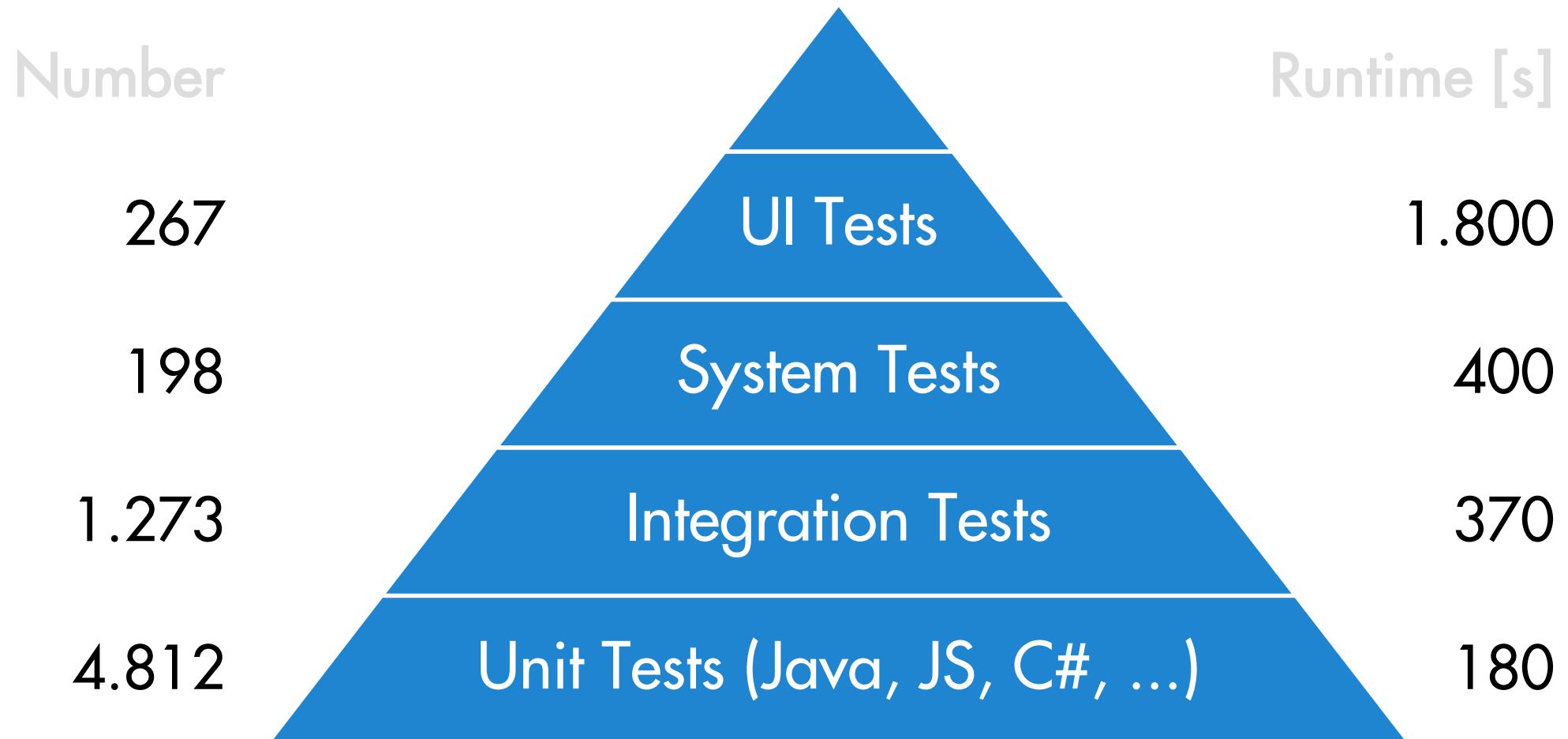
[START HERE](#) [PREFACE](#) [CONTENT](#) [CONTRIBUTE](#) [ABOUT ME](#)

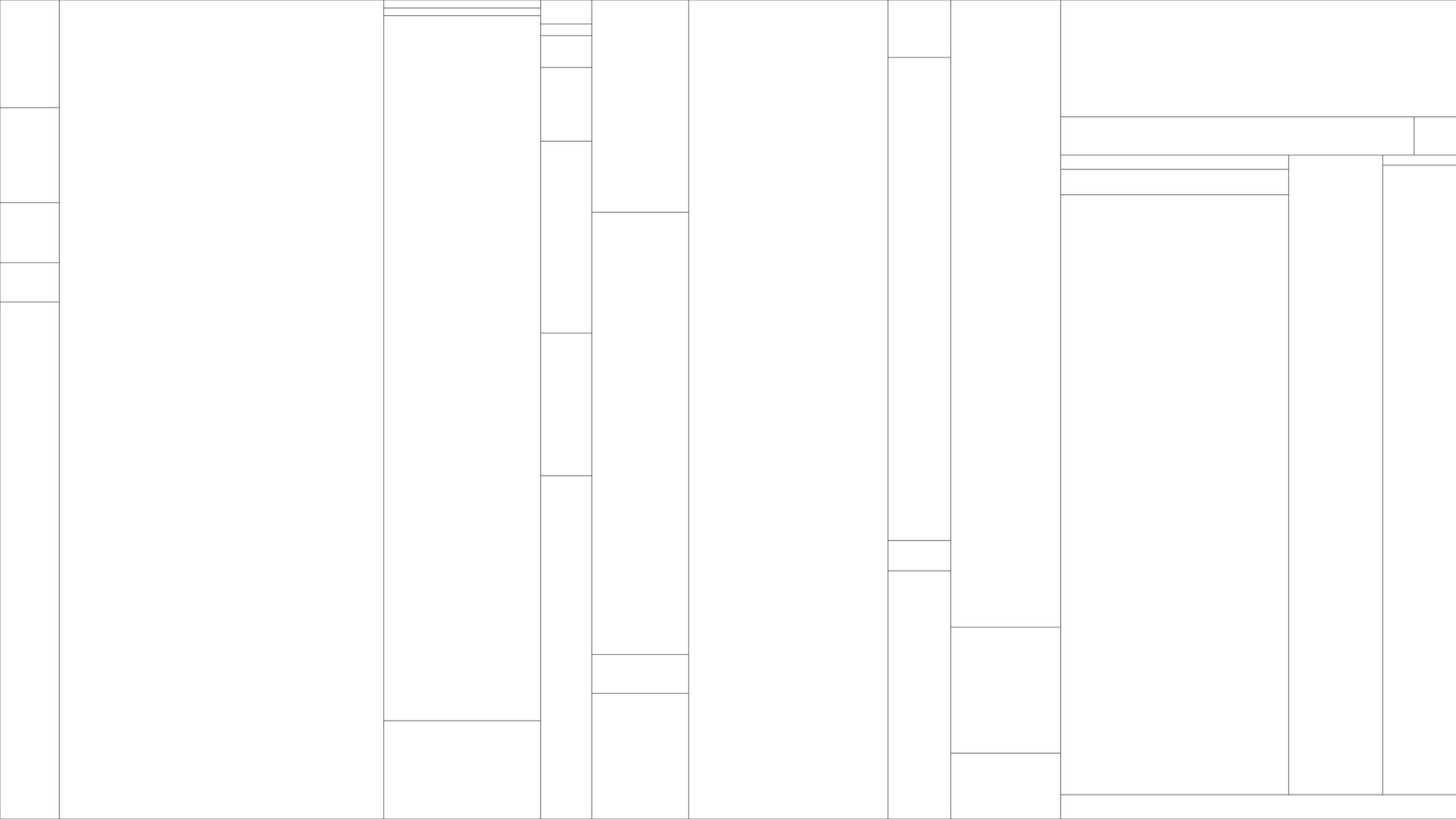


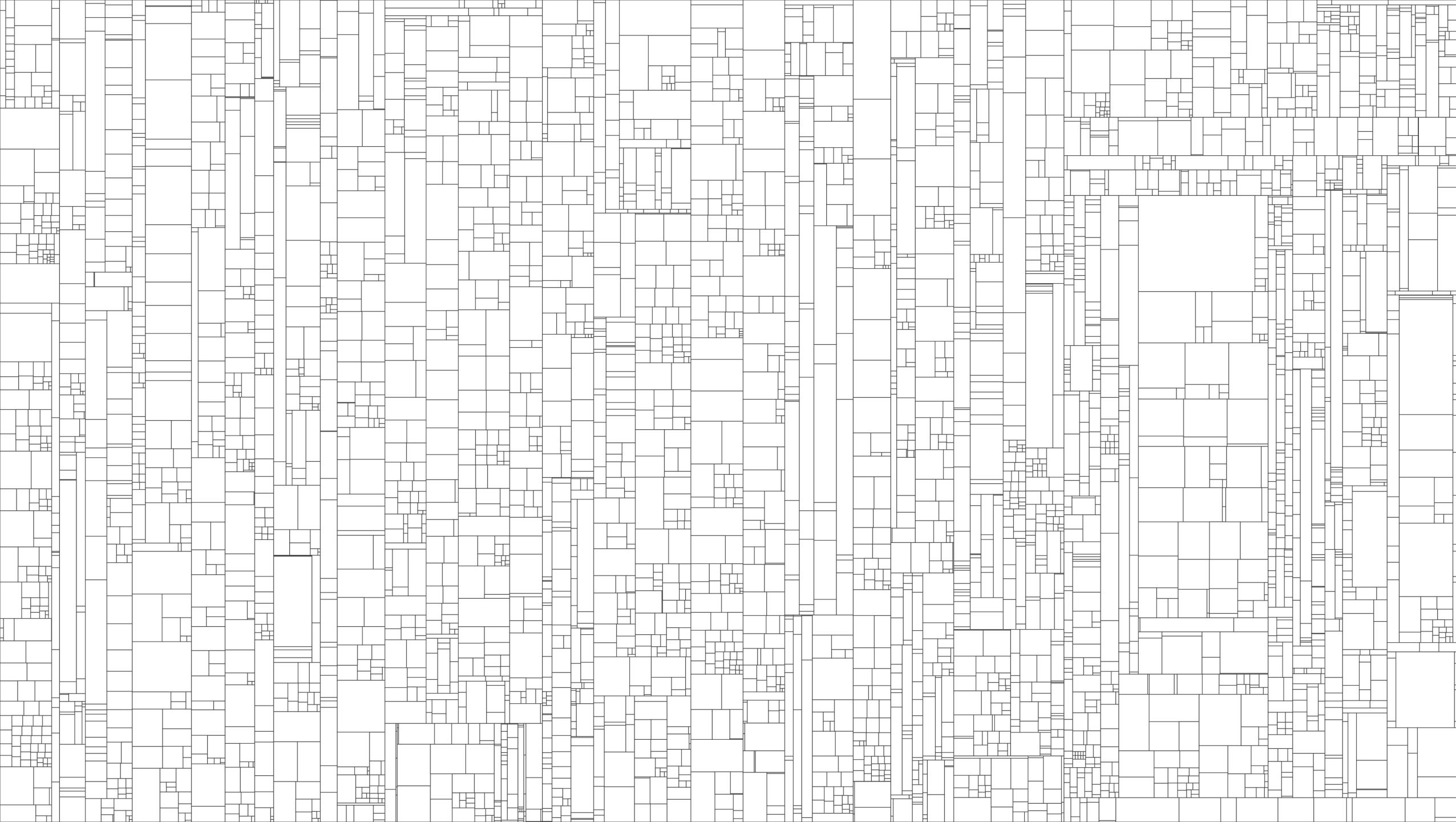
Test-Impact Analysis

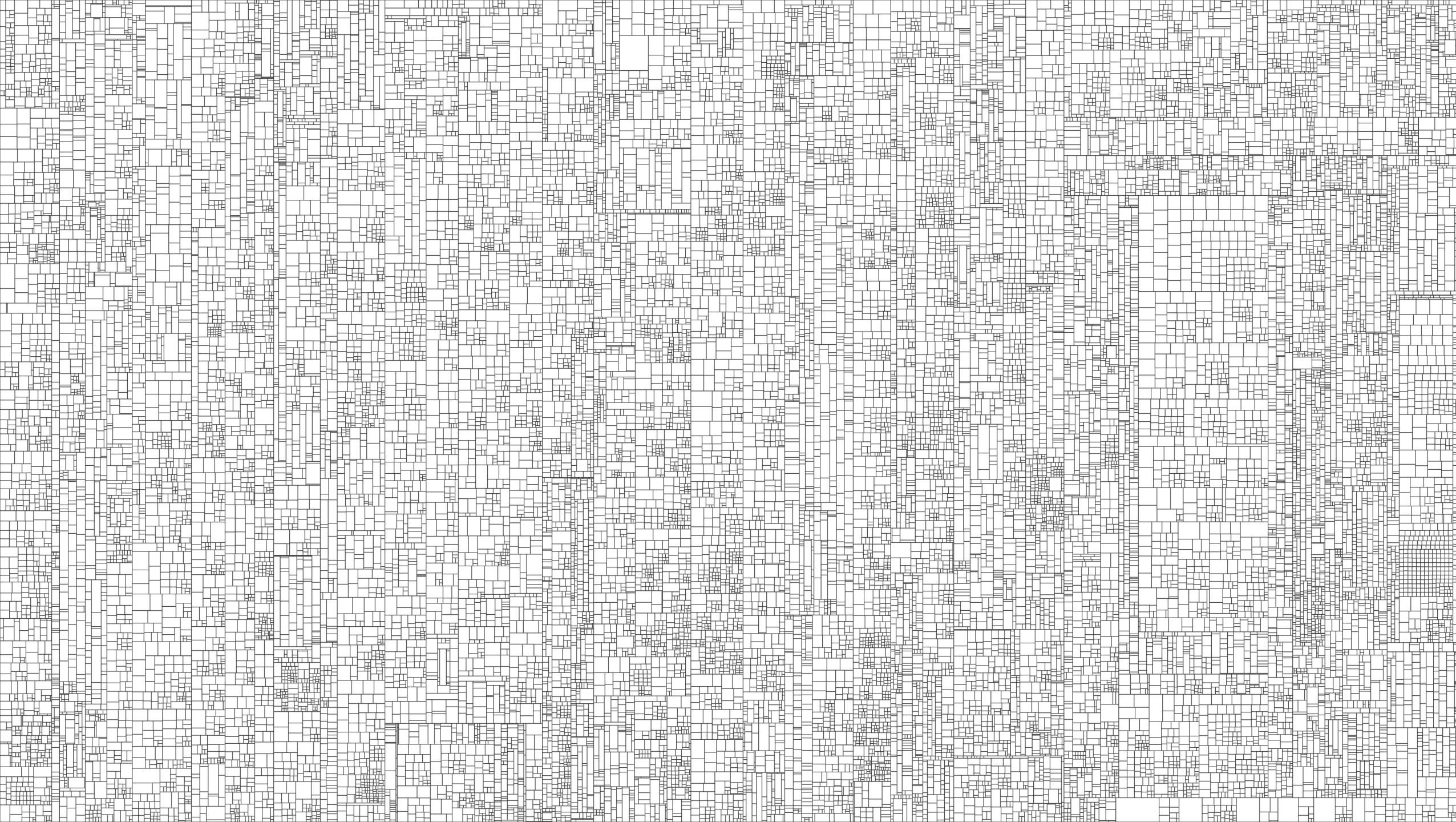
Master Thesis by Jakob Rott

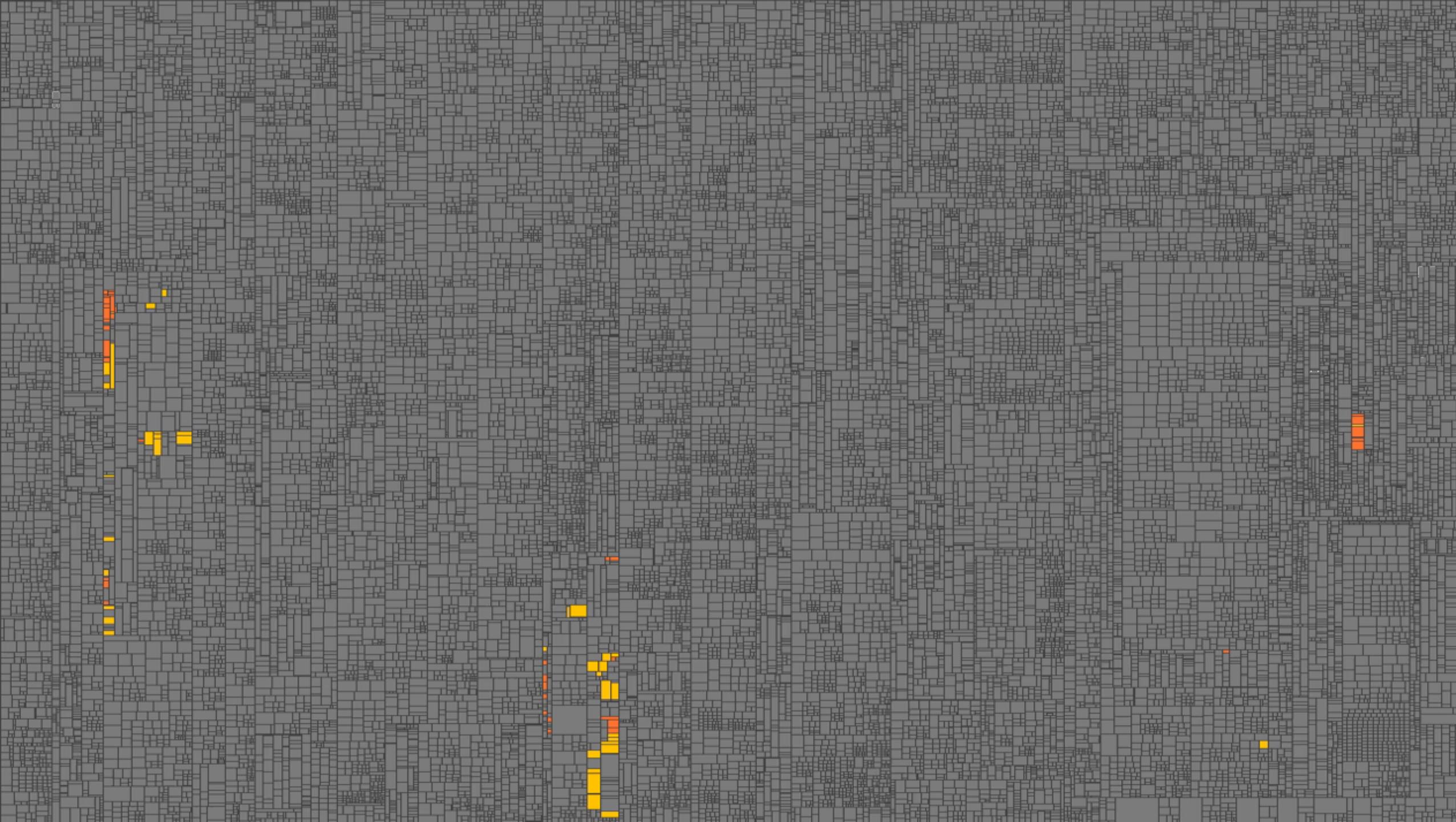
How We Test Teamscale







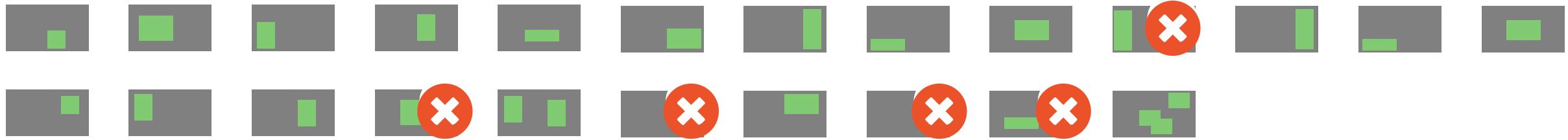




Schritt 1: Selektion betroffener Testfälle

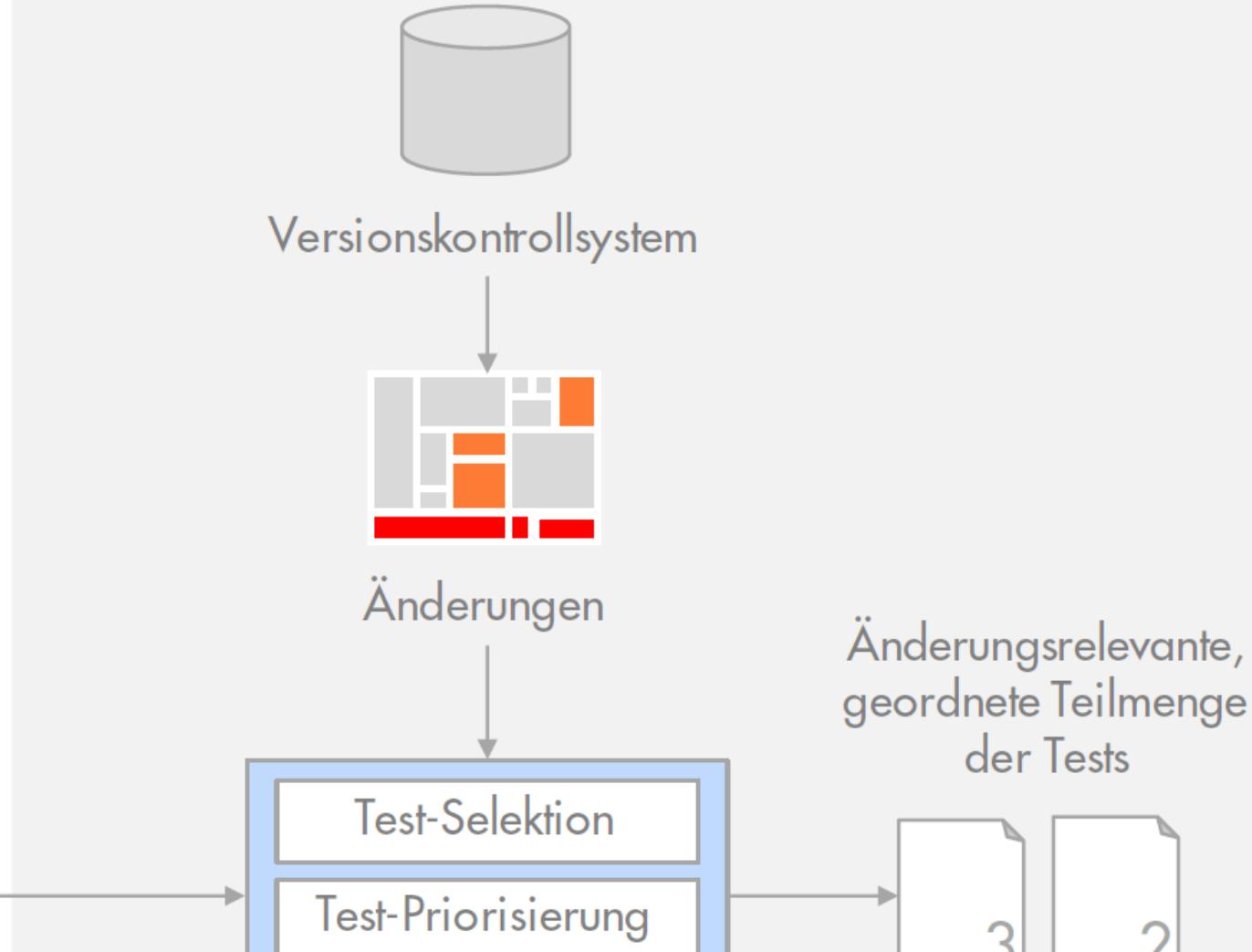
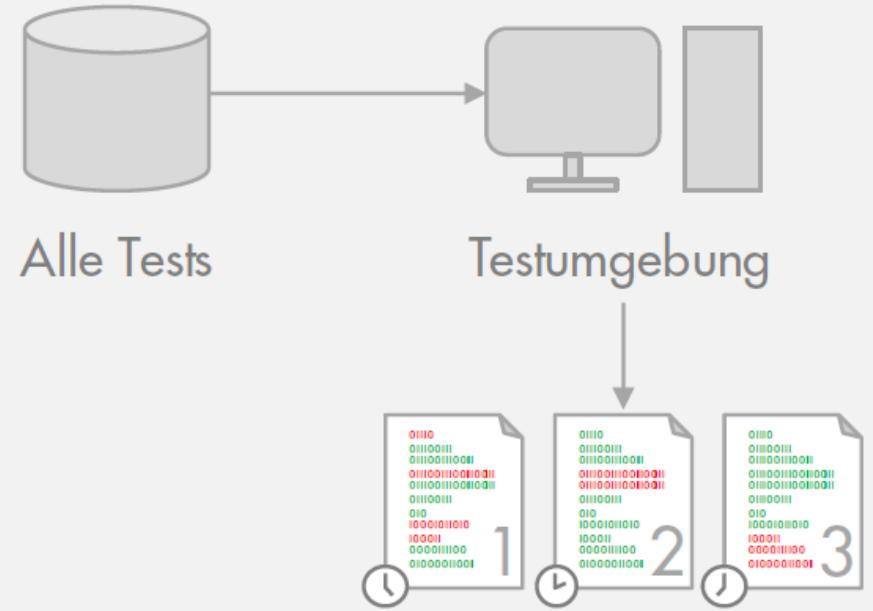


Schritt 2: Priorisierung selektierter Testfälle



Schritt 2: Priorisierung selektierter Testfälle

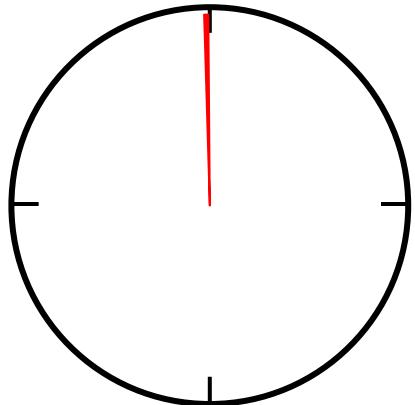




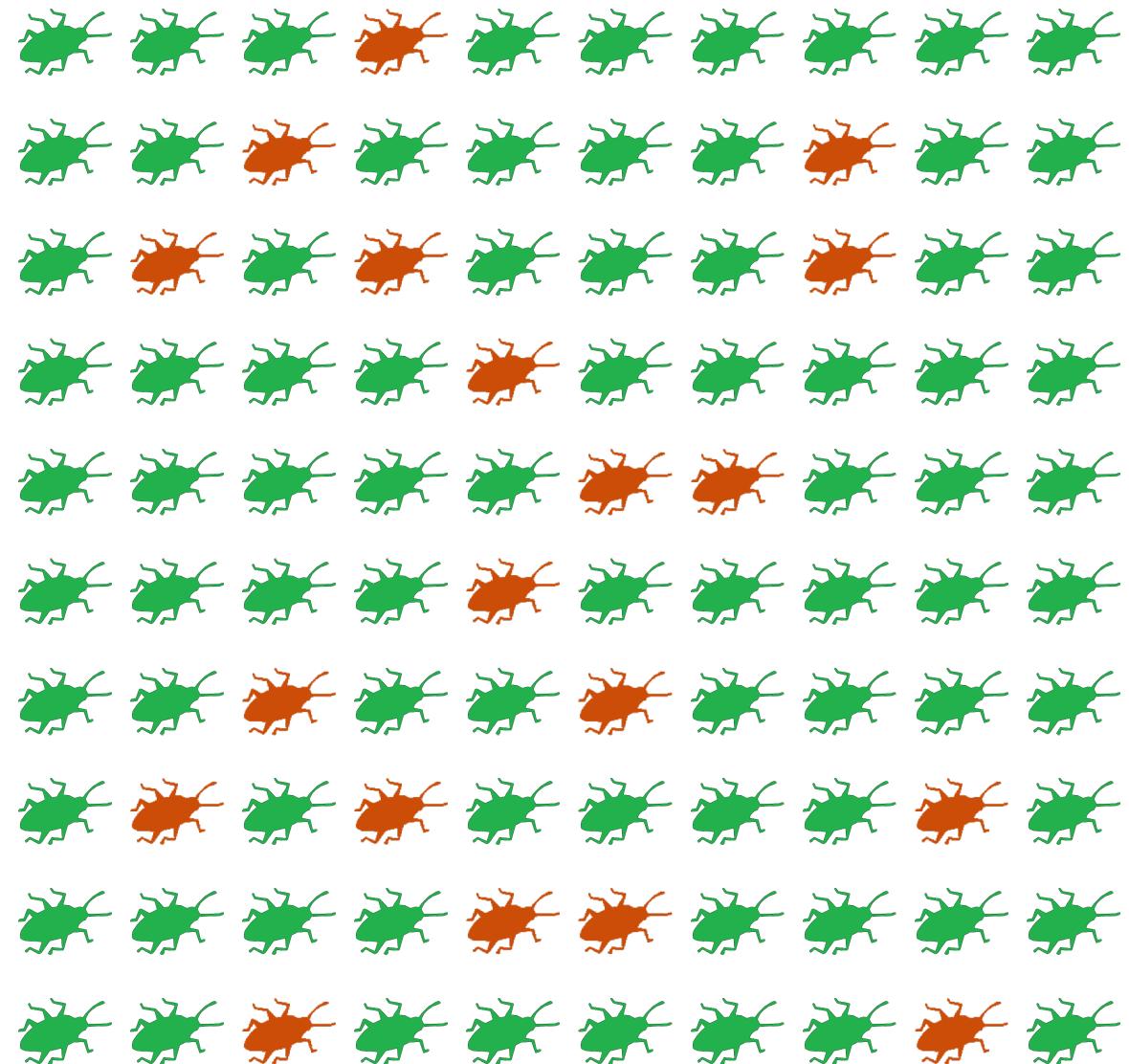
1

2

1%



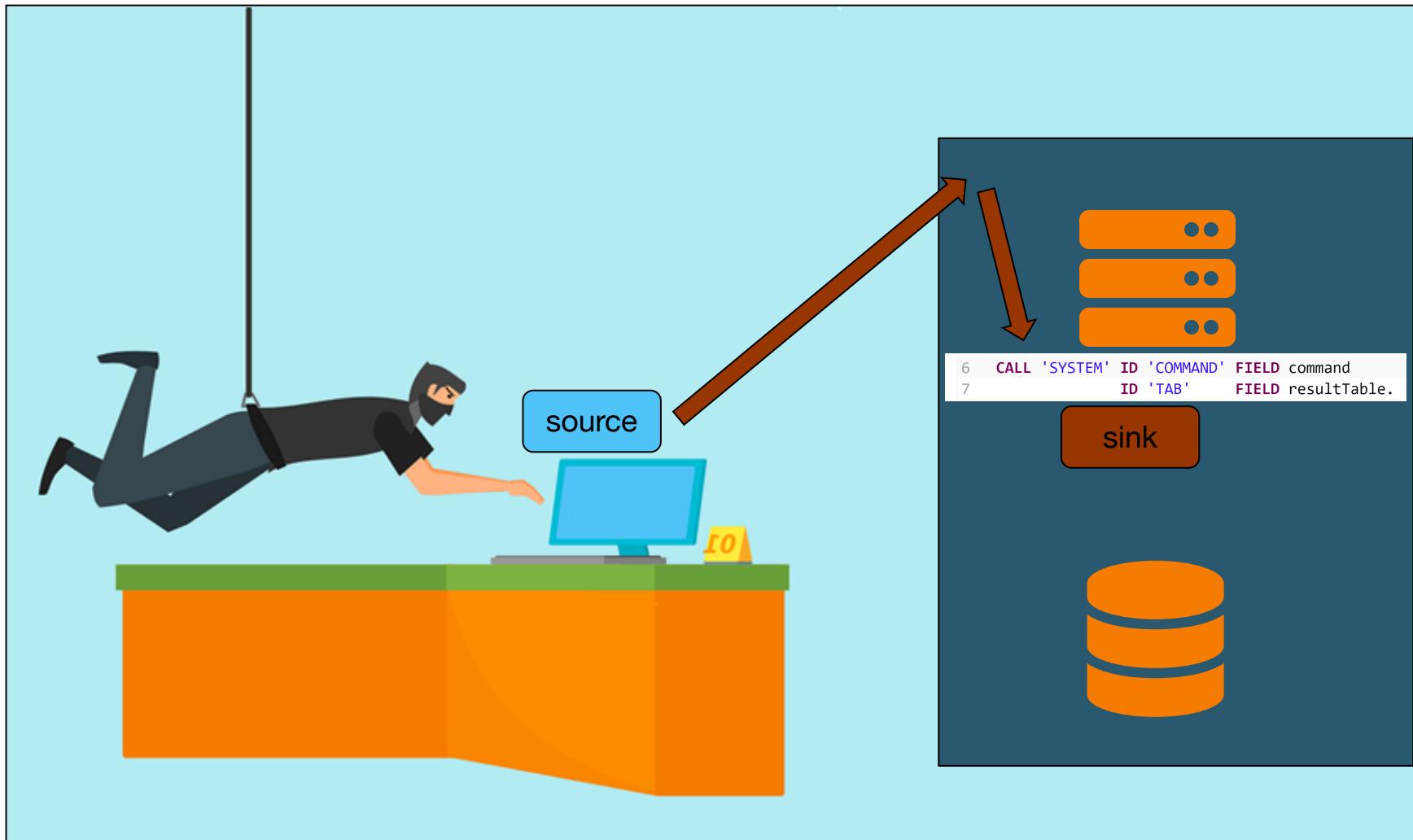
80%



Presenting Findings of Complex Analyses

Open Topic

Security Threat: System Command Injection



Interpretation of Findings

Clone with 2 instances of length 11

Code Duplication > Cloning > Redundancy/Clones

Clones are source code duplicates that are very similar and needlessly increase the amount of program code and maintenance activities. Changes, such as bug fixes, that affect one clone will likely affect the other.

[read more](#)

Sibling findings

- !CQSE!ABAP_D...N_EXPORT/CLAS/!CQSE!CL_SRCOBJ_BW_TRANSFORM.abap

Details

introduced Mar 28 2018 12:01

```
abap-cq!/CQSE!ABAP_DEV//CQSEITMSCALE_CONNECTOR/CQSEITSCON_EXPORT/CLAS/!CQSE!CL_SRCOBJ_BW_TRANSFORM.abap (revision 6e007ba6...) abap-cq!/CQSE!ABAP_DEV//CQSEITMSCALE_CONNECTOR/CQSEITSCON_EXPORT/CLAS/!CQSE!CL_SRCOBJ_BW_QUERY.abap (revision 6e007ba6...)
```

Interpretation of Findings

Execution of OS command: filename read Z_CALLER.abap line 31.

taint analysis > Taint Propagation > Dataflow/Taint Propagation

A data value is read from user input and passed to a statement that executes an OS command. For further information see SAP help page on system command injection.

- Source: PROG/Z_CALLER.abap:3
- Sink: PROG/Z_CALLER.abap:31

```
- in PROG/Z_CALLER.abap

1 REPORT ztestsrc_df_class.
2
3 PARAMETERS: filename TYPE String DEFAULT 'default_filename'.
4   ① This source taints the variable `filename`
5   *
6   * CLASS zlcl_main DEFINITION
7   *-----*
8   *
9   *-----*
10  CLASS zlcl_main DEFINITION.
11    PUBLIC SECTION.
12      CLASS-METHODS main.
13      CLASS-METHODS foo
14          IMPORTING info TYPE String
15          RETURNING VALUE(returnValue) TYPE String.
16      ENDMETHODS.           "lcl_main DEFINITION
17
18   *-----*
19   *     CLASS zlcl_main IMPLEMENTATION
20   *-----*
21   *
22   *-----*
23  CLASS zlcl_main IMPLEMENTATION.
24
25    METHOD main.
26      ② Field `filename` may contain tainted data
27      DATA: secret TYPE string.
28
29      secret = foo( filename ).           ③ Possibly tainted parameter at index 0 can be modified in method call
30      ④ Possibly tainted return value in assignee `MCR_foo^2_ASSIGNEE_`           ⑤ Possibly tainted variable at index 0 can be modified in method call
31      CALL 'SYSTEM' ID 'ls' FIELD secret.           ⑥ Possibly tainted return value in assignee `MCR_foo^2_ASSIGNEE_`           ⑦ OS command injection with tainted variable `secret`
32      ENDMETHOD.           "main
33      METHOD foo.
34          returnValue = 'CLEAN'.
35      ENDMETHOD.           "main
36  ENDMETHODS.           "lcl_main IMPLEMENTATION
37
38  START-OF-SELECTION.
39      zlcl_main=>main( ).           ⑧ Possibly tainted field `PROG/Z_CALLER.abap##filename` can be modified in method call
```

Agenda

- 11:40 Introduction
- 11:45 Static Analysis for Quality Improvement
- 12:05 Continuous Quality Improvement
- 12:50 Innovation & Improvement
- 13:10 Q&A**
- 13:20 End of Lecture

Additional Literature

- **»Containing the Findings Flood« (Blog)**
<https://cqse.eu/en/blog/containing-the-findings-flood/>
- **»Does Quality Control Really Make Your Code Better?« (Blog)**
<https://cqse.eu/en/blog/effect-of-quality-control/>
- **»McCabe's Cyclomatic Complexity and Why We Don't Use It« (Blog)**
<https://cqse.eu/en/blog/mccabe-cyclomatic-complexity/>
- **»Haben wir das Richtige getestet?« und »Immer kürzere Testphasen?« (OBJEKTspektrum 2018)**
<https://cqse.eu/tga-ticket-coverage-objektspektrum>
- **»Fehler früh erkennen trotz großer, langlaufender Test-Suites« (OBJEKTspektrum 2018)**
<https://cqse.eu/test-impact-analyse-objektspektrum>
- **»Teamscale – The Official User Reference«**
<https://cqse.eu/userguide>

Contact



Dr. Sven Amann · amann@cqse.eu · +49 172 1860063

CQSE GmbH
Hilpertstraße 31 (HUB31)
64289 Darmstadt

CQSE
Continuous Quality in Software Engineering