

Seminar Software Qualität im WS 2017/18

Vorbesprechung am 13. Juli 2017

Prof. Dr. Dr. h.c. Manfred Broy

Dr. Maximilian Junker, Dr. Elmar Jürgens,

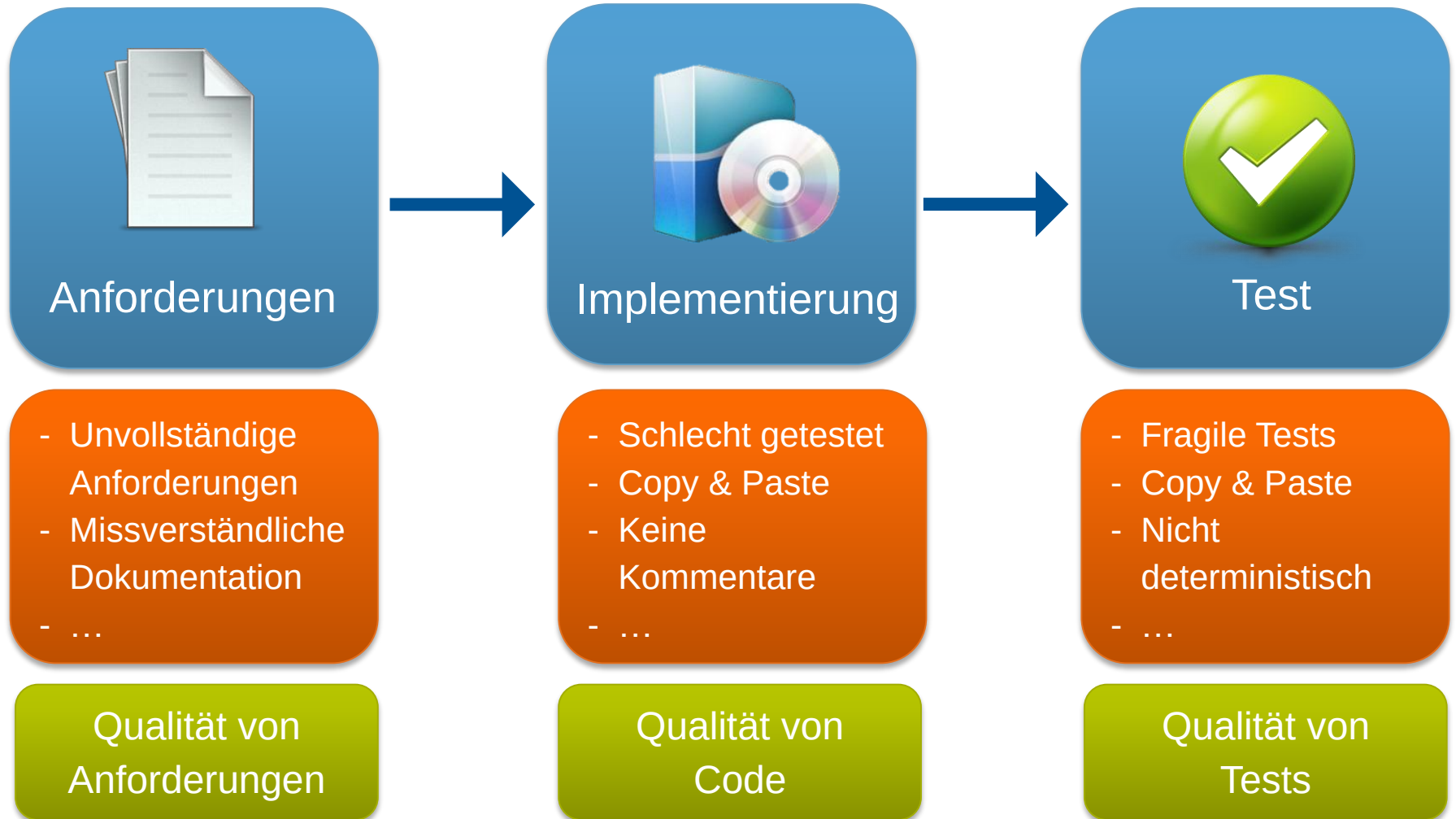
Dr. Daniel Mendez, Dr. Benedikt Hauptmann,

Henning Femmer, Dr. Sebastian Eder,

Roman Haas, Daniel Veihelmann



Seminar Software Qualität



Teilnahme am Seminar

Mail an **junkerm@in.tum.de**

bis **Mittwoch, 19. Juli**

Inhalt

- Motivation für Seminarteilnahme u. Themen (<1 Seite)
- Stand im Studium
- Erfahrungen (Arbeit, Praktikum, Open-Source-Projekt, ...)
- **Themenwünsche** (sortiert nach Priorität)

→ **Rückmeldung Ende Juli über das Matching System**

Ablauf

Anfang des Semesters: Auftaktveranstaltung

- Überblick Software Qualität
- Literaturrecherche
- Effektiv Präsentieren

vstl. Januar 2018: Seminar als Blockveranstaltung

- Block: 3 Tage Vorträge mit Diskussion
- genaue Termine klären wir zum Semesterstart

Seminararbeit und Vortrag

Inhalt

- **Theorie**
- **Anwendung des Seminarthemas in der Praxis**
 - Ergebnisse
 - Erfahrungen
 - Probleme & Grenzen

Abgaben

- Seminararbeit: 2 Wochen vor Vortrag
- Probevortrag (verpflichtend): 2 Wochen vor Vortrag

Umfang

- Arbeit: max. 15 Seiten (wir stellen Vorlage bereit)
- Vortrag: 25min + 20min Diskussion

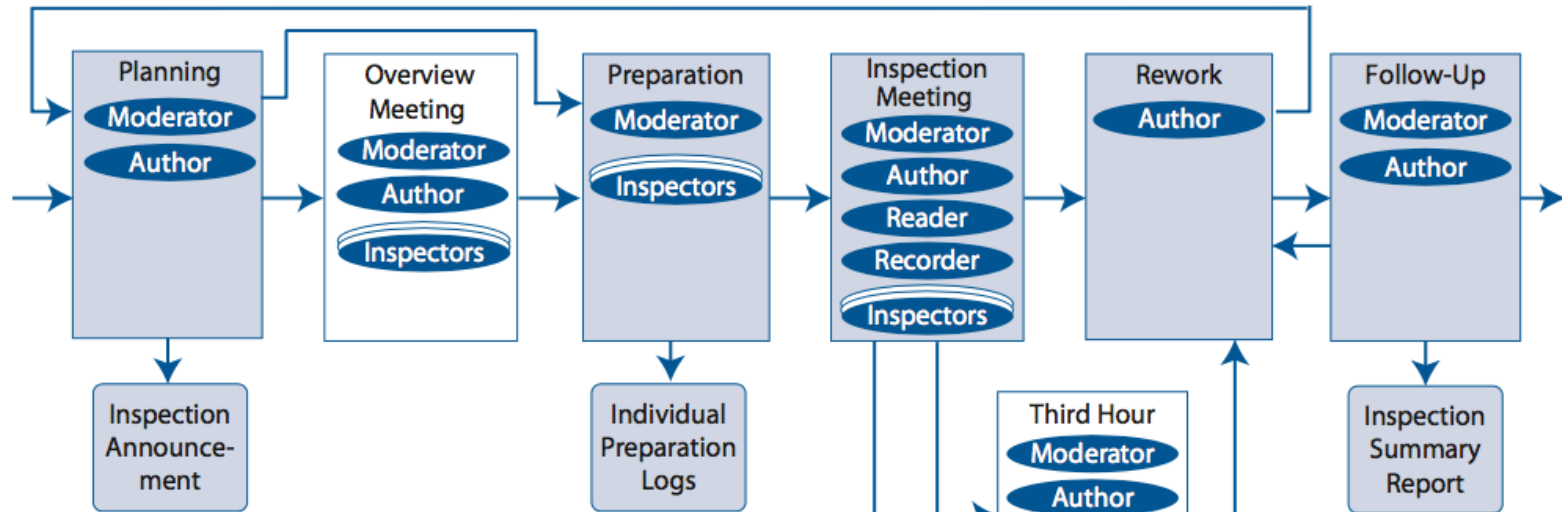
Themen (I)

Qualität von Anforderungen

1. **Walkthroughs, Inspektionen, Prototypen:** Techniken für manuelle Requirements Reviews
(Henning)
2. **Wie gut sind meine Anforderungen?** Defekte in Anforderungen automatisch detektieren.
(Henning)
3. **Model-based Requirements Engineering:** Bessere Anforderungsqualität durch Modellierung?
(Maximilian)

Walkthroughs, Inspektionen, Prototypen

- Techniken für manuelle Requirements Reviews



- Bei der NASA sind Fehler in Anforderungen manchmal nicht mehr zu fixen.
- Deshalb machen sie strenge Reviews, so wie in diesem Prozess.
- Wie funktioniert das, was kann man (anders) machen?

Wie gut meine Anforderungen? Defekte in
Anforderungen automatisch detektieren

The shop system
sends an E-Mail to
registered customers,
occasionally with
attachment.

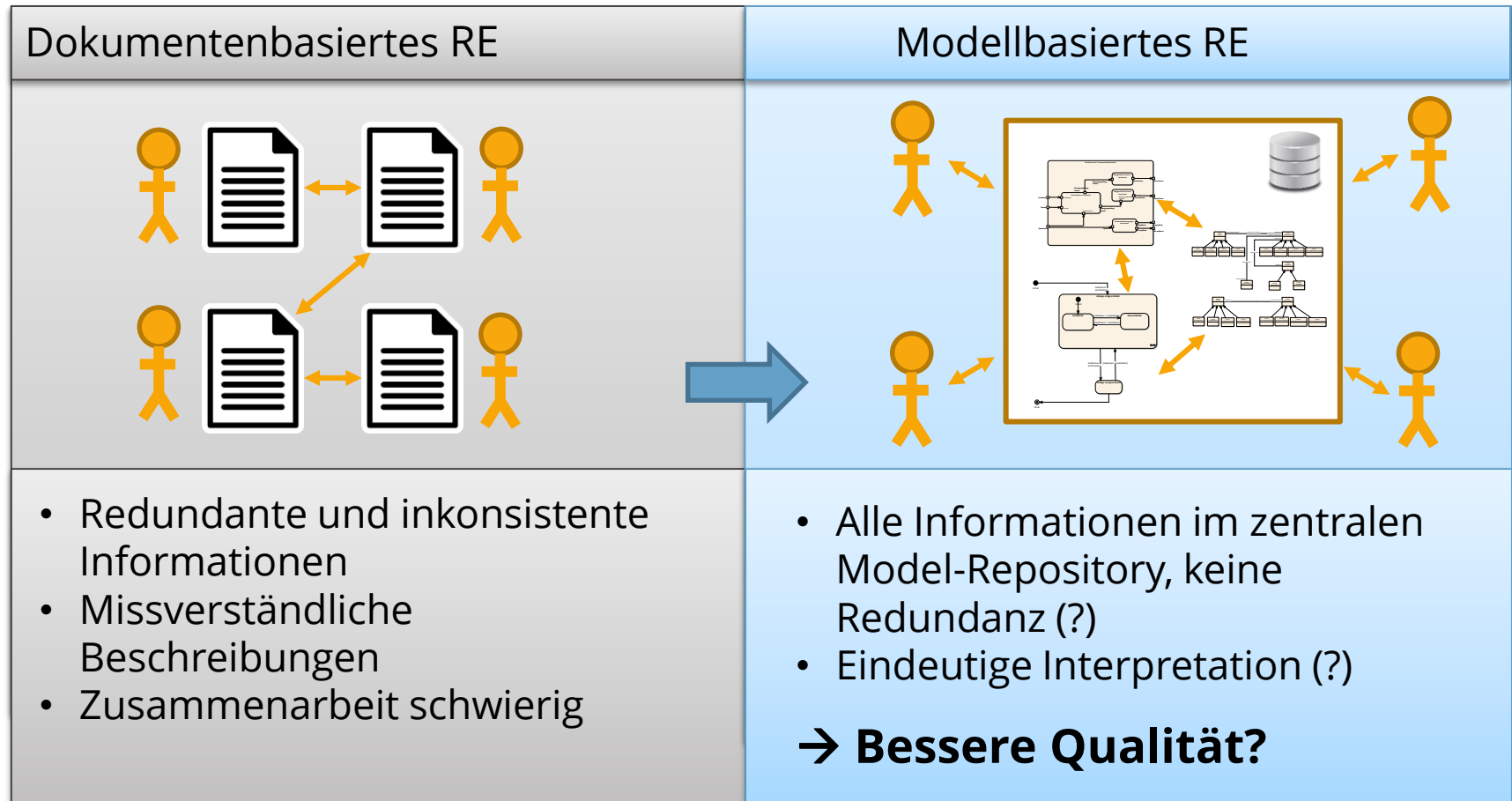
- Schlechte Anforderungen sind riskant
 - Manuelle Qualitätssicherung ist teuer und langsam.
 - Deshalb: Automatisierung!
1. Was lässt sich automatisch erkennen?
 2. Was nicht?
 3. Wie funktioniert das?

Themen (I)

Qualität von Anforderungen

1. **Walkthroughs, Inspektionen, Prototypen:** Techniken für manuelle Requirements Reviews
(Henning)
2. **Wie gut sind meine Anforderungen?** Defekte in Anforderungen automatisch detektieren.
(Henning)
3. **Model-based Requirements Engineering:** Bessere Anforderungsqualität durch Modellierung?
(Maximilian)

Model-based Requirements Engineering



Themen (II)

Qualität von Code

4. **Statische Analyse:** Automatisches Auffinden von Bugs in Quellcode.
(Sebastian)
5. **Clone Detection:** Wo ist kopierter Quelltext im System?
(Elmar)
6. **Clone Management:** Wie wurde kopierter Quelltext gewartet?
(Elmar)
7. **Sinn und Unsinn von Metriken zur Komplexität von Quellcode**
(Daniel Veihelmann)
8. **Code-Nutzungsanalysen:** Welcher Code wird (nicht) genutzt?
(Daniel Veihelmann)

```
// 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
```

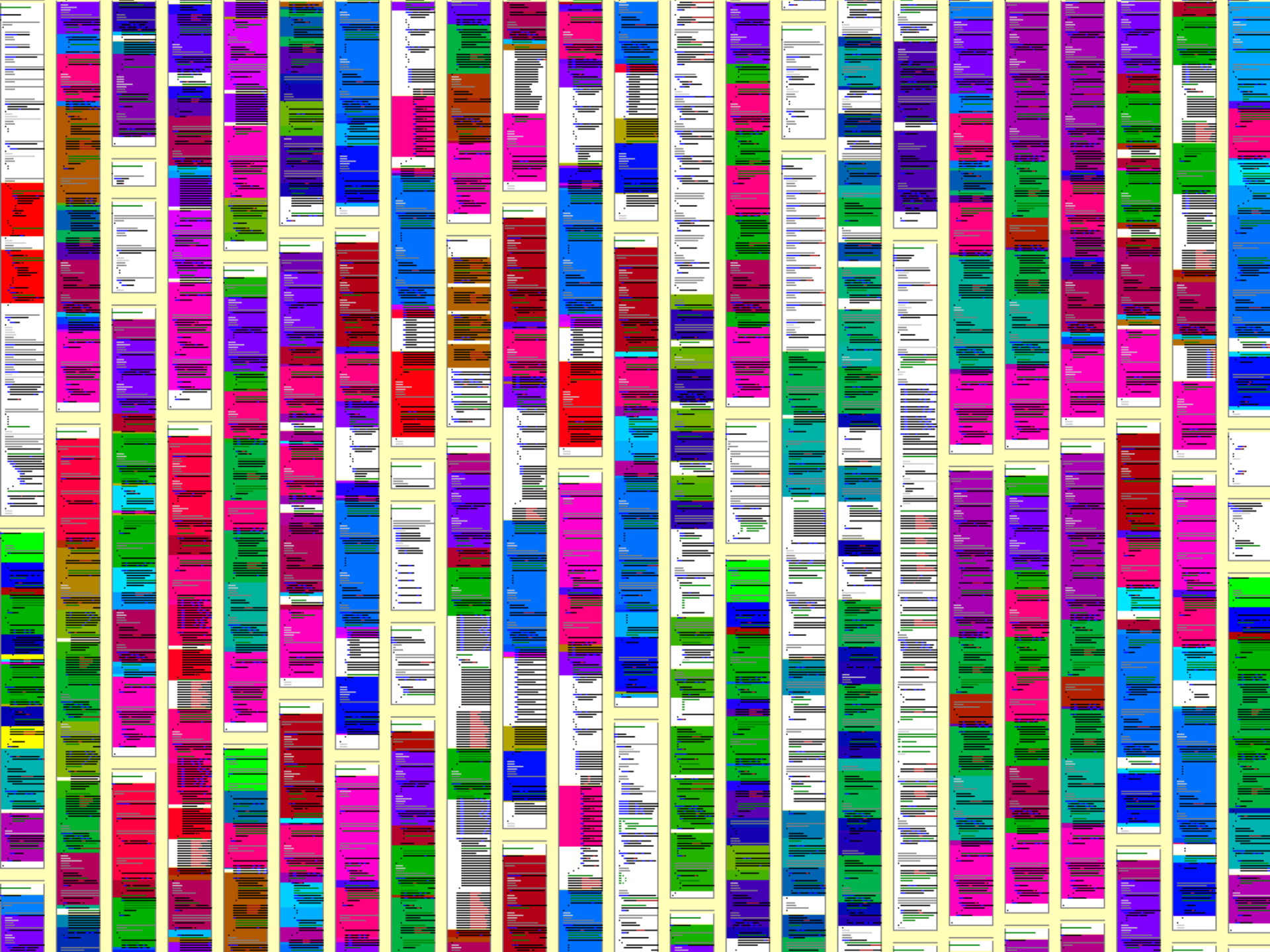
```
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; 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();
}
```









Themen (II)

Qualität von Code

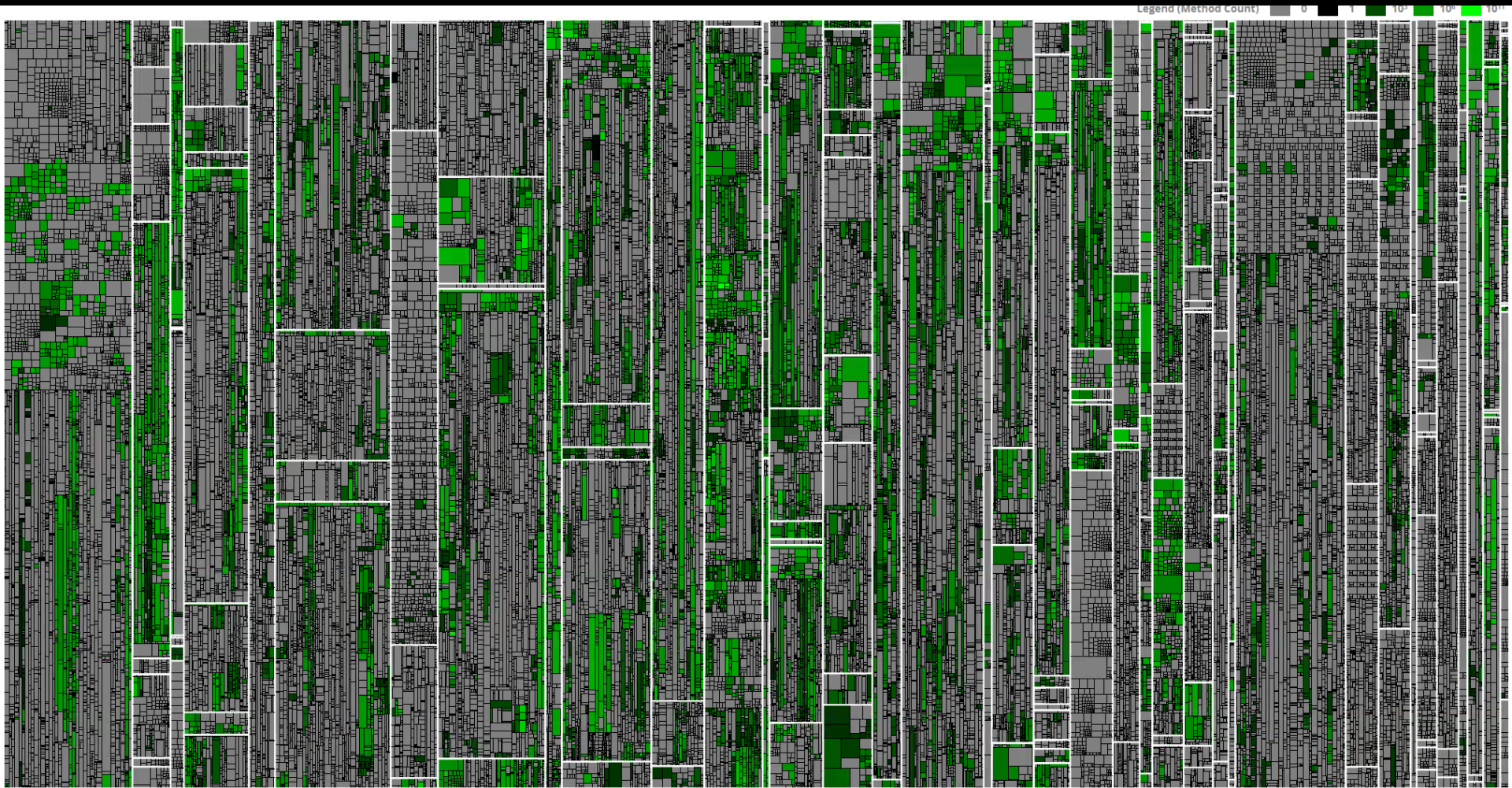
- 4. **Statische Analyse:** Automatisches Auffinden von Bugs in Quellcode.
(Sebastian)
- 5. **Clone Detection:** Wo ist kopierter Quelltext im System?
(Elmar)
- 6. **Clone Management:** Wie wurde kopierter Quelltext gewartet?
(Elmar)
- 7. **Sinn und Unsinn von Metriken zur Komplexität von Quellcode**
(Daniel Veihelmann)
- 8. **Code-Nutzungsanalysen:** Welcher Code wird (nicht) genutzt?
(Daniel Veihelmann)

331	301:	353:	869:	749:	831:	998:	1078:	1181:	1243:	1329:	1407:	1490:	1574:	1694:	1738:	1819:	1901:	1954:	* z. B. der Parameter a_display richtig gesetzt
332:	302:	354:	868:	750:	832:	997:	1079:	1182:	1244:	1328:	1408:	1491:	1575:	1695:	1739:	1820:	1902:	1955:	PREPARE lt_item_300.
333:	303:	355:	867:	751:	833:	996:	1080:	1183:	1245:	1327:	1409:	1492:	1576:	1696:	1740:	1821:	1903:	1956:	ASSIGN ar_data_changed=>mp_mod_rowa->* TO
334:	304:	356:	866:	752:	834:	995:	1081:	1184:	1246:	1326:	1410:	1493:	1577:	1697:	1741:	1822:	1904:	1957:	LOOP AT <lt_itema> INTO ga_itema_300.
335:	305:	357:	865:	753:	835:	994:	1082:	1185:	1247:	1325:	1411:	1494:	1578:	1698:	1742:	1823:	1905:	1958:	READ TABLE gt_itema_300 INTO ls_item_300
336:	306:	358:	864:	754:	836:	993:	1083:	1186:	1248:	1324:	1412:	1495:	1579:	1699:	1743:	1824:	1906:	1959:	WITH KEY itm_number = ga_itema_300->itm
337:	307:	359:	863:	755:	837:	992:	1084:	1187:	1249:	1323:	1413:	1496:	1580:	1700:	1744:	1825:	1907:	1960:	APPEND ls_item_300 TO lt_item_300.
338:	308:	360:	862:	756:	838:	991:	1085:	1188:	1250:	1322:	1414:	1497:	1581:	1701:	1745:	1826:	1908:	1961:	ENDLOOP.
339:	309:	361:	861:	757:	839:	990:	1086:	1189:	1251:	1321:	1415:	1498:	1582:	1702:	1746:	1827:	1909:	1962:	IF ar_subrc EQ 0 AND <lt_itema> IS ASSIGN
340:	310:	362:	860:	758:	840:	989:	1087:	1190:	1252:	1320:	1416:	1499:	1583:	1703:	1747:	1828:	1910:	1963:	PREPARE <lt_itema>.
341:	311:	363:	859:	759:	841:	988:	1088:	1191:	1253:	1319:	1417:	1500:	1584:	1704:	1748:	1829:	1911:	1964:	<lt_itema> = lt_item_300[].
342:	312:	364:	858:	760:	842:	987:	1089:	1192:	1254:	1318:	1418:	1501:	1585:	1705:	1749:	1830:	1912:	1965:	ENDIF.
343:	313:	365:	857:	761:	843:	986:	1090:	1193:	1255:	1317:	1419:	1502:	1586:	1706:	1750:	1831:	1913:	1966:	*
344:	314:	366:	856:	762:	844:	985:	1091:	1194:	1256:	1316:	1420:	1503:	1587:	1707:	1751:	1832:	1914:	1967:	* Fallstr. protokolliert ausgeben
345:	315:	367:	855:	763:	845:	984:	1092:	1195:	1257:	1315:	1421:	1504:	1588:	1708:	1752:	1833:	1915:	1968:	CALL METHOD ar_data_changed->display_prot
346:	316:	368:	854:	764:	846:	983:	1093:	1196:	1258:	1314:	1422:	1505:	1589:	1709:	1753:	1834:	1916:	2000:	----
347:	317:	369:	853:	765:	847:	982:	1094:	1197:	1259:	1313:	1423:	1506:	1590:	1710:	1754:	1835:	1917:	2001:	ENDMETHOD. "handle_data_ch
348:	318:	370:	852:	766:	848:	981:	1095:	1198:	1260:	1312:	1424:	1507:	1591:	1711:	1755:	1836:	1918:	----	
349:	319:	371:	851:	767:	849:	980:	1096:	1199:	1261:	1311:	1425:	1508:	1592:	1712:	1756:	1837:	1919:	----	
350:	320:	372:	850:	768:	850:	979:	1097:	1200:	1262:	1310:	1426:	1509:	1593:	1713:	1757:	1838:	1920:	----	
351:	321:	373:	849:	769:	851:	978:	1098:	1201:	1263:	1309:	1427:	1510:	1594:	1714:	1758:	1839:	1921:	----	
352:	322:	374:	848:	770:	852:	977:	1099:	1202:	1264:	1308:	1428:	1511:	1595:	1715:	1759:	1840:	1922:	----	
353:	323:	375:	847:	771:	853:	976:	1100:	1203:	1265:	1307:	1429:	1512:	1596:	1716:	1760:	1841:	1923:	----	
354:	324:	376:	846:	772:	854:	975:	1101:	1204:	1266:	1306:	1430:	1513:	1597:	1717:	1761:	1842:	1924:	----	
355:	325:	377:	845:	773:	855:	974:	1102:	1205:	1267:	1305:	1431:	1514:	1598:	1718:	1762:	1843:	1925:	----	
356:	326:	378:	844:	774:	856:	973:	1103:	1206:	1268:	1304:	1432:	1515:	1599:	1719:	1763:	1844:	1926:	----	
357:	327:	379:	843:	775:	857:	972:	1104:	1207:	1269:	1303:	1433:	1516:	1600:	1720:	1764:	1845:	1927:	----	
358:	328:	380:	842:	776:	858:	971:	1105:	1208:	1270:	1302:	1434:	1517:	1601:	1721:	1765:	1846:	1928:	----	
359:	329:	381:	841:	777:	859:	970:	1106:	1209:	1271:	1301:	1435:	1518:	1602:	1722:	1766:	1847:	1929:	----	
360:	330:	382:	840:	778:	860:	969:	1107:	1210:	1272:	1300:	1436:	1519:	1603:	1723:	1767:	1848:	1930:	----	
361:	331:	383:	839:	779:	861:	968:	1108:	1211:	1273:	1299:	1437:	1520:	1604:	1724:	1768:	1849:	1931:	----	
362:	332:	384:	838:	780:	862:	967:	1109:	1212:	1274:	1298:	1438:	1521:	1605:	1725:	1769:	1850:	1932:	----	
363:	333:	385:	837:	781:	863:	966:	1110:	1213:	1275:	1297:	1439:	1522:	1606:	1726:	1770:	1851:	1933:	----	
364:	334:	386:	836:	782:	864:	965:	1111:	1214:	1276:	1296:	1440:	1523:	1607:	1727:	1771:	1852:	1934:	----	
365:	335:	387:	835:	783:	865:	964:	1112:	1215:	1277:	1295:	1441:	1524:	1608:	1728:	1772:	1853:	1935:	----	
366:	336:	388:	834:	784:	866:	963:	1113:	1216:	1278:	1294:	1442:	1525:	1609:	1729:	1773:	1854:	1936:	----	
367:	337:	389:	833:	785:	867:	962:	1114:	1217:	1279:	1293:	1443:	1526:	1610:	1730:	1774:	1855:	1937:	----	
368:	338:	390:	832:	786:	868:	961:	1115:	1218:	1280:	1292:	1444:	1527:	1611:	1731:	1775:	1856:	1938:	----	
369:	339:	391:	831:	787:	869:	960:	1116:	1219:	1281:	1291:	1445:	1528:	1612:	1732:	1776:	1857:	1939:	----	
370:	340:	392:	830:	788:	870:	959:	1117:	1220:	1282:	1290:	1446:	1529:	1613:	1733:	1777:	1858:	1940:	----	
371:	341:	393:	829:	789:	871:	958:	1118:	1221:	1283:	1289:	1447:	1530:	1614:	1734:	1778:	1859:	1941:	----	
372:	342:	394:	828:	790:	872:	957:	1119:	1222:	1284:	1288:	1448:	1531:	1615:	1735:	1779:	1860:	1942:	----	
373:	343:	395:	827:	791:	873:	956:	1120:	1223:	1285:	1287:	1449:	1532:	1616:	1736:	1780:	1861:	1943:	----	
374:	344:	396:	826:	792:	874:	955:	1121:	1224:	1286:	1286:	1450:	1533:	1617:	1737:	1781:	1862:	1944:	----	
375:	345:	397:	825:	793:	875:	954:	1122:	1225:	1287:	1285:	1451:	1534:	1618:	1738:	1782:	1863:	1945:	----	
376:	346:	398:	824:	794:	876:	953:	1123:	1226:	1288:	1284:	1452:	1535:	1619:	1739:	1783:	1864:	1946:	----	
377:	347:	399:	823:	795:	877:	952:	1124:	1227:	1289:	1283:	1453:	1536:	1620:	1740:	1784:	1865:	1947:	----	
378:	348:	400:	822:	796:	878:	951:	1125:	1228:	1290:	1282:	1454:	1537:	1621:	1741:	1785:	1866:	1948:	----	
379:	349:	401:	821:	797:	879:	950:	1126:	1229:	1291:	1281:	1455:	1538:	1622:	1742:	1786:	1867:	1949:	----	
380:	350:	402:	820:	798:	880:	949:	1127:	1230:	1292:	1280:	1456:	1539:	1623:	1743:	1787:	1868:	1950:	----	
381:	351:	403:	819:	799:	881:	948:	1128:	1231:	1293:	1279:	1457:	1540:	1624:	1744:	1788:	1869:	1951:	----	
382:	352:	404:	818:	800:	882:	947:	1129:	1232:	1294:	1278:	1458:	1541:	1625:	1745:	1789:	1870:	1952:	----	
383:	353:	405:	817:	801:	883:	946:	1130:	1233:	1295:	1277:	1459:	1542:	1626:	1746:	1790:	1871:	1953:	----	
384:	354:	406:	816:	802:	884:	945:	1131:	1234:	1296:	1276:	1460:	1543:	1627:	1747:	1791:	1872:	1954:	----	
385:	355:	407:	815:	803:	885:	944:	1132:	1235:	1297:	1275:	1461:	1544:	1628:	1748:	1792:	1873:	1955:	----	
386:	356:	408:	814:	804:	886:	943:	1133:	1236:	1298:	1274:	1462:	1545:	1629:	1749:	1793:	1874:	1956:	----	
387:	357:	409:	813:	805:	887:	942:	1134:	1237:	1299:	1273:	1463:	1546:	1630:	1750:	1794:	1875:	1957:	----	
388:	358:	410:	812:	806:	888:	941:	1135:	1238:	1300:	1272:	1464:	1547:	1631:	1751:	1795:	1876:	1958:	----	
389:	359:	411:	811:	807:	889:	940:	1136:	1239:	1301:	1271:	1465:	1548:	1632:	1752:	1796:	1877:	1959:	----	
390:	360:	412:	810:	808:	890:	939:	1137:	1240:	1302:	1270:	1466:	1549:	1633:	1753:	1797:	1878:	1960:	----	
391:	361:	413:	809:	809:	891:	938:	1138:	1241:	1303:	1269:	1467:	1550:	1634:	1754:	1798:	1879:	1961:	----	
392:	362:	414:	808:	810:	892:	937:	1139:	1242:	1304:	1268:	1468:	1551:	1635:	1755:	1799:	1880:	1962:	----	
393:	363:	415:	807:	811:	893:	936:	1140:	1243:	1305:	1267:	1469:	1552:	1636:	1756:	1800:	1881:	1963:	----	
394:	364:	416:	806:	812:	894:	935:	1141:	1244:	1306:	1266:	1470:	1553:	1637:	1757:	1801:	1882:	1964:	----	
395:	365:	417:	805:	813:	895:	934:	1142:	1245:	1307:	1265:	1471:	1554:	1638:	1758:	1802:	1883:	1965:	----	
396:	366:	418:	804:	814:	896:	933:	1143:	1246:	1308:	1264:	1472:	1555:	1639:	1759:	1803:	1884:	1966:	----	
397:	367:	419:	803:	815:	897:	932:	1144:	1247:	1309:	1263:	1473:	1556:	1640:	1760:	1804:	1885:	1967:	----	
398:	368:	420:	802:	816:	898:	931:	1145:	1248:	1310:	1262:	1474:	1557:	1641:	1761:	1805:	1886:	1968:	----	
399:	369:	421:	801:	817:	899:	930:	1146:	1249:	1311:	1261:	1475:	1558:	1642:	1762:	1806:	1887:	1969:	----	
400:	370:	422:	800:	818:	900:	929:	1147:	1250:	1312:	1260:	1476:	1559:	1643:	1763:	1807:	1888:	1970:	----	
401:	371:	423:	799:	819:	901:	928:	1148:	1251:	1313:	1259:	1477:	1560:	1644:	1764:	1808:	1889:	1971:	----	
402:	372:	424:	798:	820:	902:	927:	1149:	1252:	1314:	1258:	1478:</								

Themen (II)

Qualität von Code

- 4. **Statische Analyse:** Automatisches Auffinden von Bugs in Quellcode.
(Sebastian)
- 5. **Clone Detection:** Wo ist kopierter Quelltext im System?
(Elmar)
- 6. **Clone Management:** Wie wurde kopierter Quelltext gewartet?
(Elmar)
- 7. **Sinn und Unsinn von Metriken zur Komplexität von Quellcode**
(Daniel Veihelmann)
- 8. **Code-Nutzungsanalysen:** Welcher Code wird (nicht) genutzt?
(Daniel Veihelmann)



Themen (III)

Qualität von Code

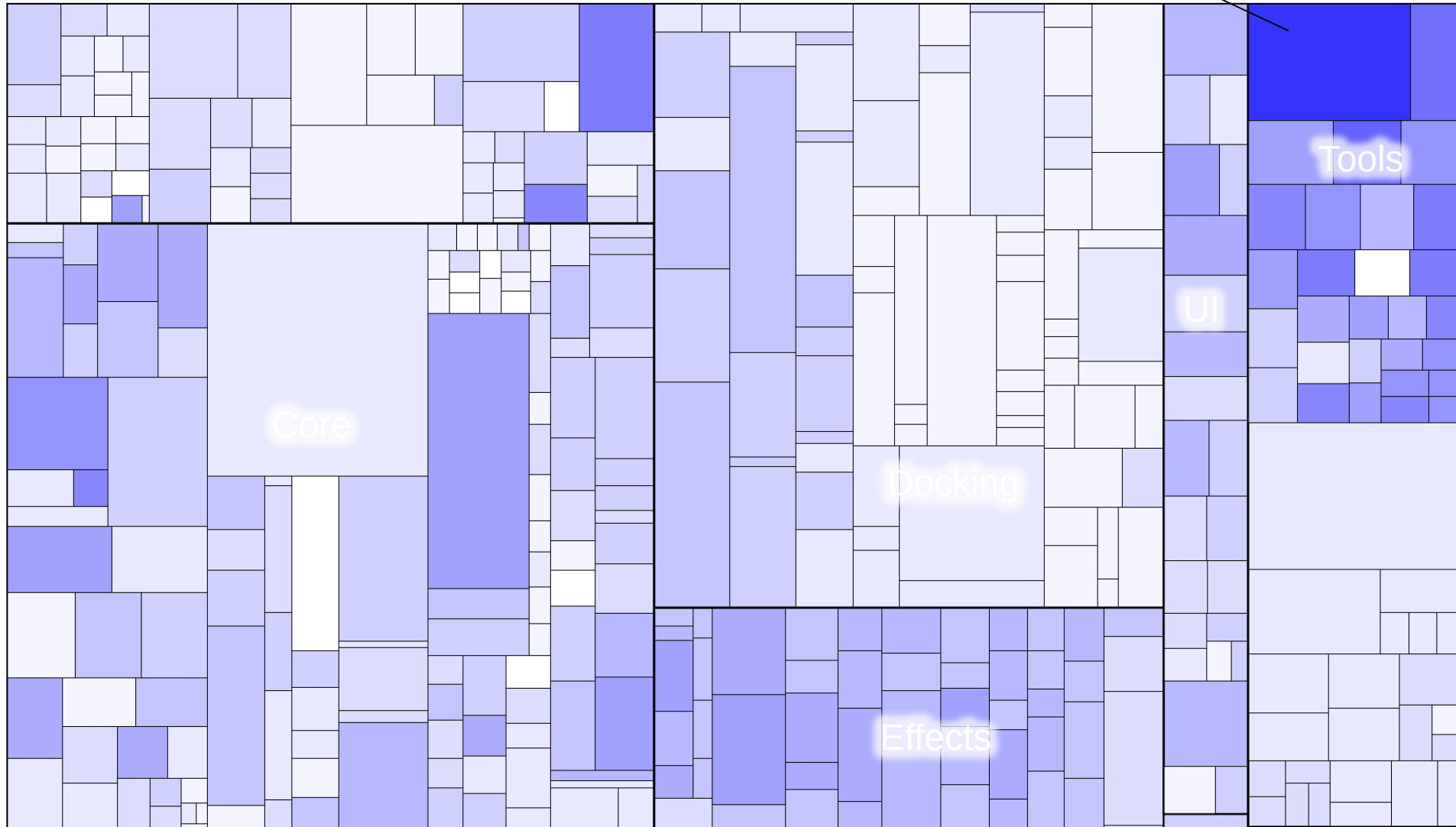
- 9. **Historienanalyse:** Wie entwickeln sich Teams und wer kennt welchen Code?
(Roman Haas)
- 10. **Refactoring Empfehlungen:** Automatische Generierung von Verbesserungsvorschlägen für Code.
(Roman Haas)
- 11. **Code-Zentralität:** Was sind in diesem Softwaresystem die wichtigsten Klassen?
(Roman Haas)





TextTool.cs (17 Entwickler)

Ownership Distribution for [pinta](#)



Themen (III)

Qualität von Code

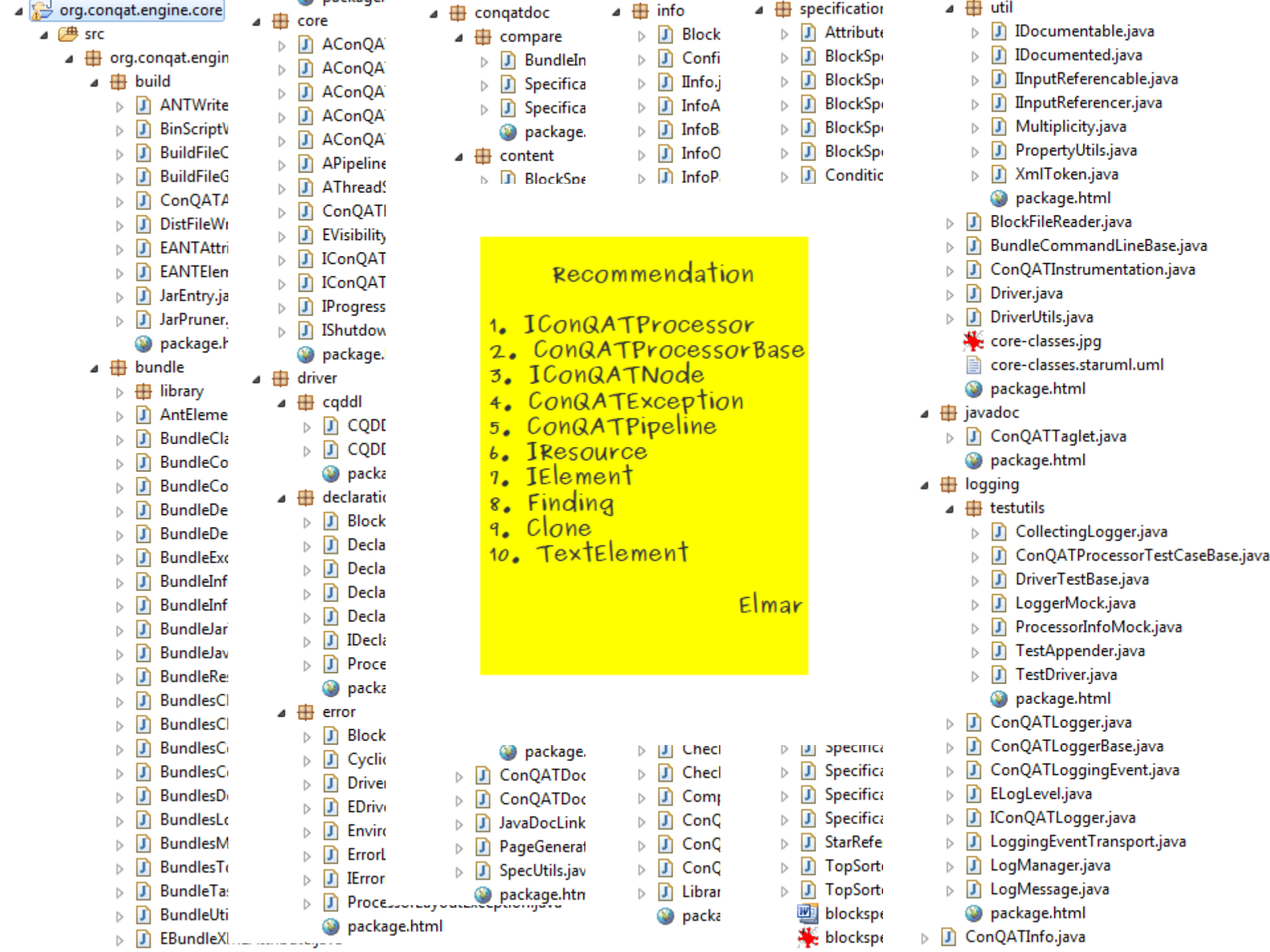
- 9. **Historienanalyse:** Wie entwickeln sich Teams und wer kennt welchen Code?
(Roman Haas)
- 10. **Refactoring Empfehlungen:** Automatische Generierung von Verbesserungsvorschlägen für Code.
(Roman Haas)
- 11. **Code-Zentralität:** Was sind in diesem Softwaresystem die wichtigsten Klassen?
(Roman Haas)

331	301:	353:	869:	749:	831:	998:	1078:	1181:	1243:	1329:	1407:	1490:	1574:	1694:	1738:	1819:	1901:	1954:	* z. B. der Parameter a_display richtig gesetzt
332:	302:	354:	868:	750:	832:	997:	1079:	1182:	1244:	1328:	1408:	1491:	1575:	1695:	1739:	1820:	1902:	1955:	REFRESH lt_item_300.
333:	303:	355:	867:	751:	833:	996:	1080:	1183:	1245:	1327:	1409:	1492:	1576:	1696:	1740:	1821:	1903:	1956:	ASSIGN ar_data_changed->mp_mod_rowa->* TO
334:	304:	356:	866:	752:	834:	995:	1081:	1184:	1246:	1326:	1410:	1493:	1577:	1697:	1741:	1822:	1904:	1957:	LOOP AT <lt_itema> INTO gs_itema_300.
335:	305:	357:	865:	753:	835:	994:	1082:	1185:	1247:	1325:	1411:	1494:	1578:	1698:	1742:	1823:	1905:	1958:	READ TABLE gt_itema_300 INTO ls_item_30
336:	306:	358:	864:	754:	836:	993:	1083:	1186:	1248:	1324:	1412:	1495:	1579:	1699:	1743:	1824:	1906:	1959:	WITH KEY itm_number = gs_itema_300->it
337:	307:	359:	863:	755:	837:	992:	1084:	1187:	1249:	1323:	1413:	1496:	1580:	1700:	1744:	1825:	1907:	1960:	APPEND ls_item_300 TO lt_item_300.
338:	308:	360:	862:	756:	838:	991:	1085:	1188:	1250:	1322:	1414:	1497:	1581:	1701:	1745:	1826:	1908:	1961:	ENDLOOP.
339:	309:	361:	861:	757:	839:	990:	1086:	1189:	1251:	1321:	1415:	1498:	1582:	1702:	1746:	1827:	1909:	1962:	IF sy-subrc EQ 0 AND <lt_itema> IS ASSIGN
340:	310:	362:	860:	758:	840:	989:	1087:	1190:	1252:	1320:	1416:	1499:	1583:	1703:	1747:	1828:	1910:	1963:	REFRESH <lt_itema>.
341:	311:	363:	859:	759:	841:	988:	1088:	1191:	1253:	1319:	1417:	1500:	1584:	1704:	1748:	1829:	1911:	1964:	<lt_itema> = lt_item_300[].
342:	312:	364:	858:	760:	842:	987:	1089:	1192:	1254:	1318:	1418:	1501:	1585:	1705:	1749:	1830:	1912:	1965:	ENDIF.
343:	313:	365:	857:	761:	843:	986:	1090:	1193:	1255:	1317:	1419:	1502:	1586:	1706:	1750:	1831:	1913:	1966:	*
344:	314:	366:	856:	762:	844:	985:	1091:	1194:	1256:	1316:	1420:	1503:	1587:	1707:	1751:	1832:	1914:	1967:	* Fallstr. protokolliert ausgeben
345:	315:	367:	855:	763:	845:	984:	1092:	1195:	1257:	1315:	1421:	1504:	1588:	1708:	1752:	1833:	1915:	1968:	CALL METHOD ar_data_changed->display_prot
346:	316:	368:	854:	764:	846:	983:	1093:	1196:	1258:	1314:	1422:	1505:	1589:	1709:	1753:	1834:	1916:	2000:	----
347:	317:	369:	853:	765:	847:	982:	1094:	1197:	1259:	1313:	1423:	1506:	1590:	1710:	1754:	1835:	1917:	2001:	ENDMETHOD.
348:	318:	370:	852:	766:	848:	981:	1095:	1198:	1260:	1312:	1424:	1507:	1591:	1711:	1755:	1836:	1918:	----	"handle_data_ch
349:	319:	371:	851:	767:	849:	980:	1096:	1199:	1261:	1311:	1425:	1508:	1592:	1712:	1756:	1837:	1919:	----	
350:	320:	372:	850:	768:	850:	979:	1097:	1200:	1262:	1310:	1426:	1509:	1593:	1713:	1757:	1838:	1920:	----	
351:	321:	373:	849:	769:	851:	978:	1098:	1201:	1263:	1309:	1427:	1510:	1594:	1714:	1758:	1839:	1921:	----	
352:	322:	374:	848:	770:	852:	977:	1099:	1202:	1264:	1308:	1428:	1511:	1595:	1715:	1759:	1840:	1922:	----	
353:	323:	375:	847:	771:	853:	976:	1100:	1203:	1265:	1307:	1429:	1512:	1596:	1716:	1760:	1841:	1923:	----	
354:	324:	376:	846:	772:	854:	975:	1101:	1204:	1266:	1306:	1430:	1513:	1597:	1717:	1761:	1842:	1924:	----	
355:	325:	377:	845:	773:	855:	974:	1102:	1205:	1267:	1305:	1431:	1514:	1598:	1718:	1762:	1843:	1925:	----	
356:	326:	378:	844:	774:	856:	973:	1103:	1206:	1268:	1304:	1432:	1515:	1599:	1719:	1763:	1844:	1926:	----	
357:	327:	379:	843:	775:	857:	972:	1104:	1207:	1269:	1303:	1433:	1516:	1600:	1720:	1764:	1845:	1927:	----	
358:	328:	380:	842:	776:	858:	971:	1105:	1208:	1270:	1302:	1434:	1517:	1601:	1721:	1765:	1846:	1928:	----	
359:	329:	381:	841:	777:	859:	970:	1106:	1209:	1271:	1301:	1435:	1518:	1602:	1722:	1766:	1847:	1929:	----	
360:	330:	382:	840:	778:	860:	969:	1107:	1210:	1272:	1300:	1436:	1519:	1603:	1723:	1767:	1848:	1930:	----	
361:	331:	383:	839:	779:	861:	968:	1108:	1211:	1273:	1299:	1437:	1520:	1604:	1724:	1768:	1849:	1931:	----	
362:	332:	384:	838:	780:	862:	967:	1109:	1212:	1274:	1298:	1438:	1521:	1605:	1725:	1769:	1850:	1932:	----	
363:	333:	385:	837:	781:	863:	966:	1110:	1213:	1275:	1297:	1439:	1522:	1606:	1726:	1770:	1851:	1933:	----	
364:	334:	386:	836:	782:	864:	965:	1111:	1214:	1276:	1296:	1440:	1523:	1607:	1727:	1771:	1852:	1934:	----	
365:	335:	387:	835:	783:	865:	964:	1112:	1215:	1277:	1295:	1441:	1524:	1608:	1728:	1772:	1853:	1935:	----	
366:	336:	388:	834:	784:	866:	963:	1113:	1216:	1278:	1294:	1442:	1525:	1609:	1729:	1773:	1854:	1936:	----	
367:	337:	389:	833:	785:	867:	962:	1114:	1217:	1279:	1293:	1443:	1526:	1610:	1730:	1774:	1855:	1937:	----	
368:	338:	390:	832:	786:	868:	961:	1115:	1218:	1280:	1292:	1444:	1527:	1611:	1731:	1775:	1856:	1938:	----	
369:	339:	391:	831:	787:	869:	960:	1116:	1219:	1281:	1291:	1445:	1528:	1612:	1732:	1776:	1857:	1939:	----	
370:	340:	392:	830:	788:	870:	959:	1117:	1220:	1282:	1290:	1446:	1529:	1613:	1733:	1777:	1858:	1940:	----	
371:	341:	393:	829:	789:	871:	958:	1118:	1221:	1283:	1289:	1447:	1530:	1614:	1734:	1778:	1859:	1941:	----	
372:	342:	394:	828:	790:	872:	957:	1119:	1222:	1284:	1288:	1448:	1531:	1615:	1735:	1779:	1860:	1942:	----	
373:	343:	395:	827:	791:	873:	956:	1120:	1223:	1285:	1287:	1449:	1532:	1616:	1736:	1780:	1861:	1943:	----	
374:	344:	396:	826:	792:	874:	955:	1121:	1224:	1286:	1286:	1450:	1533:	1617:	1737:	1781:	1862:	1944:	----	
375:	345:	397:	825:	793:	875:	954:	1122:	1225:	1287:	1285:	1451:	1534:	1618:	1738:	1782:	1863:	1945:	----	
376:	346:	398:	824:	794:	876:	953:	1123:	1226:	1288:	1284:	1452:	1535:	1619:	1739:	1783:	1864:	1946:	----	
377:	347:	399:	823:	795:	877:	952:	1124:	1227:	1289:	1283:	1453:	1536:	1620:	1740:	1784:	1865:	1947:	----	
378:	348:	400:	822:	796:	878:	951:	1125:	1228:	1290:	1282:	1454:	1537:	1621:	1741:	1785:	1866:	1948:	----	
379:	349:	401:	821:	797:	879:	950:	1126:	1229:	1291:	1281:	1455:	1538:	1622:	1742:	1786:	1867:	1949:	----	
380:	350:	402:	820:	798:	880:	949:	1127:	1230:	1292:	1280:	1456:	1539:	1623:	1743:	1787:	1868:	1950:	----	
381:	351:	403:	819:	799:	881:	948:	1128:	1231:	1293:	1279:	1457:	1540:	1624:	1744:	1788:	1869:	1951:	----	
382:	352:	404:	818:	800:	882:	947:	1129:	1232:	1294:	1278:	1458:	1541:	1625:	1745:	1789:	1870:	1952:	----	
383:	353:	405:	817:	801:	883:	946:	1130:	1233:	1295:	1277:	1459:	1542:	1626:	1746:	1790:	1871:	1953:	----	
384:	354:	406:	816:	802:	884:	945:	1131:	1234:	1296:	1276:	1460:	1543:	1627:	1747:	1791:	1872:	1954:	----	
385:	355:	407:	815:	803:	885:	944:	1132:	1235:	1297:	1275:	1461:	1544:	1628:	1748:	1792:	1873:	1955:	----	
386:	356:	408:	814:	804:	886:	943:	1133:	1236:	1298:	1274:	1462:	1545:	1629:	1749:	1793:	1874:	1956:	----	
387:	357:	409:	813:	805:	887:	942:	1134:	1237:	1299:	1273:	1463:	1546:	1630:	1750:	1794:	1875:	1957:	----	
388:	358:	410:	812:	806:	888:	941:	1135:	1238:	1300:	1272:	1464:	1547:	1631:	1751:	1795:	1876:	1958:	----	
389:	359:	411:	811:	807:	889:	940:	1136:	1239:	1301:	1271:	1465:	1548:	1632:	1752:	1796:	1877:	1959:	----	
390:	360:	412:	810:	808:	890:	939:	1137:	1240:	1302:	1270:	1466:	1549:	1633:	1753:	1797:	1878:	1960:	----	
391:	361:	413:	809:	809:	891:	938:	1138:	1241:	1303:	1269:	1467:	1550:	1634:	1754:	1798:	1879:	1961:	----	
392:	362:	414:	808:	810:	892:	937:	1139:	1242:	1304:	1268:	1468:	1551:	1635:	1755:	1799:	1880:	1962:	----	
393:	363:	415:	807:	811:	893:	936:	1140:	1243:	1305:	1267:	1469:	1552:	1636:	1756:	1800:	1881:	1963:	----	
394:	364:	416:	806:	812:	894:	935:	1141:	1244:	1306:	1266:	1470:	1553:	1637:	1757:	1801:	1882:	1964:	----	
395:	365:	417:	805:	813:	895:	934:	1142:	1245:	1307:	1265:	1471:	1554:	1638:	1758:	1802:	1883:	1965:	----	
396:	366:	418:	804:	814:	896:	933:	1143:	1246:	1308:	1264:	1472:	1555:	1639:	1759:	1803:	1884:	1966:	----	
397:	367:	419:	803:	815:	897:	932:	1144:	1247:	1309:	1263:	1473:	1556:	1640:	1760:	1804:	1885:	1967:	----	
398:	368:	420:	802:	816:	898:	931:	1145:	1248:	1310:	1262:	1474:	1557:	1641:	1761:	1805:	1886:	1968:	----	
399:	369:	421:	801:	817:	899:	930:	1146:	1249:	1311:	1261:	1475:	1558:	1642:	1762:	1806:	1887:	1969:	----	
400:	370:	422:	800:	818:	900:	929:	1147:	1250:	1312:	1260:	1476:	1559:	1643:	1763:	1807:	1888:	1970:	----	
401:	371:	423:	799:	819:	901:	928:	1148:	1251:	1313:	1259:	1477:	1560:	1644:	1764:	1808:	1889:	1971:	----	
402:	372:	424:	798:	820:	902:	927:	1149:	1252:	1314:	1258:	1478:								

Themen (III)

Qualität von Code

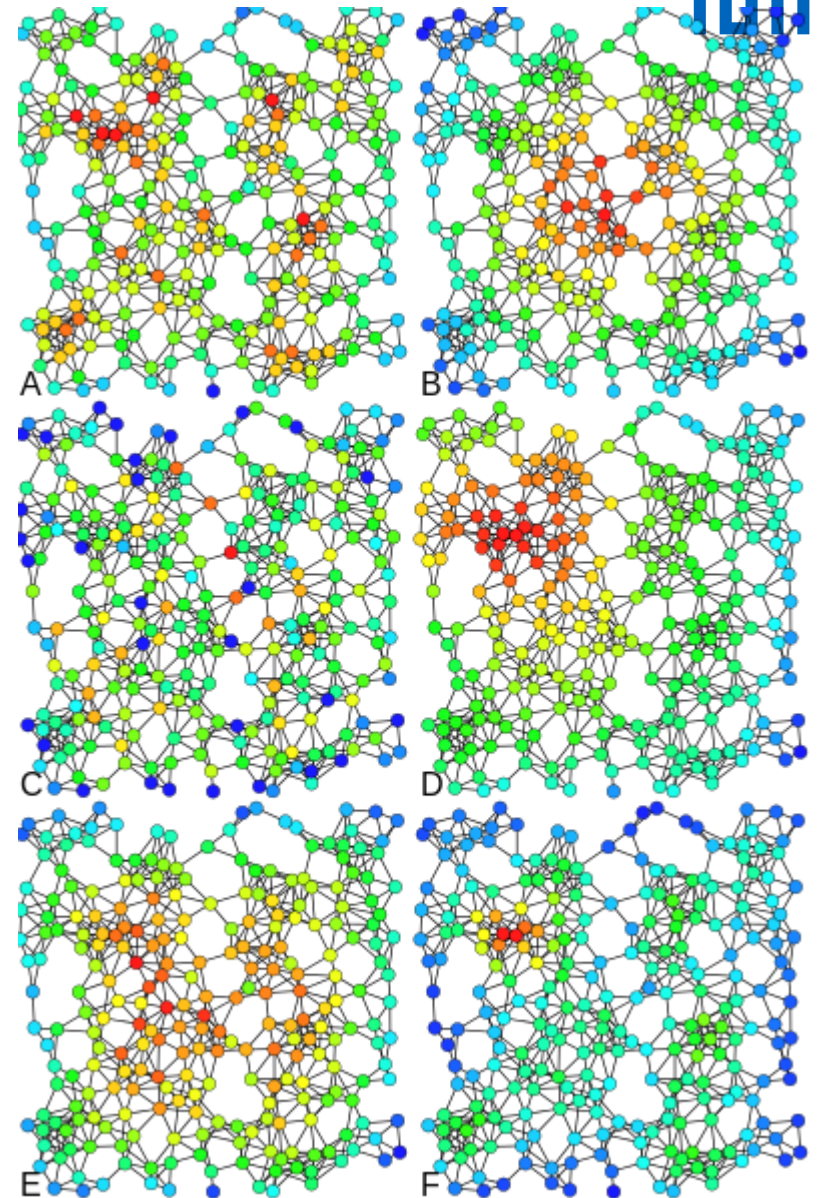
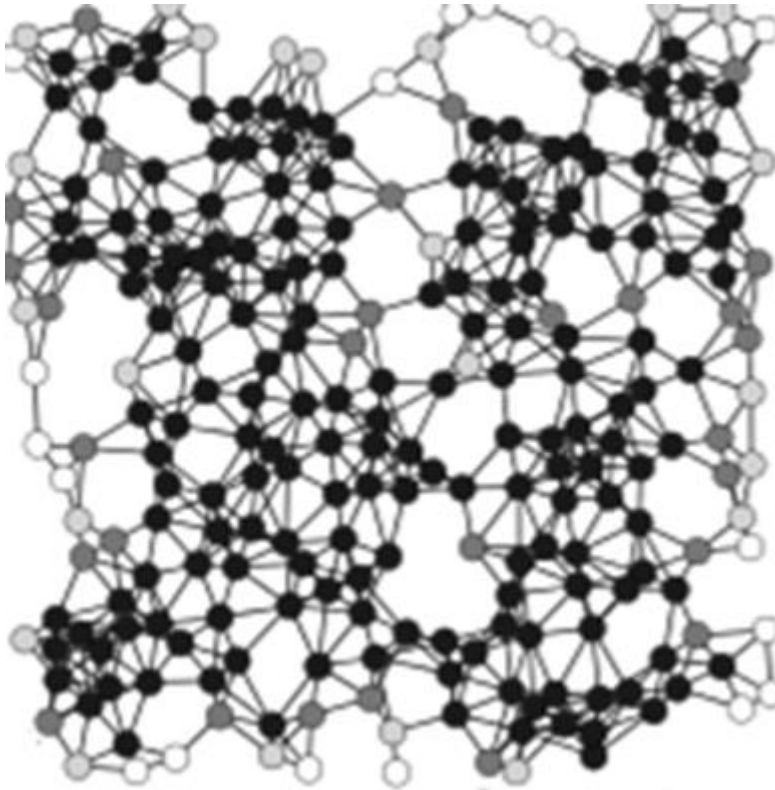
- 9. **Historienanalyse:** Wie entwickeln sich Teams und wer kennt welchen Code?
(Roman Haas)
- 10. **Refactoring Empfehlungen:** Automatische Generierung von Verbesserungsvorschlägen für Code.
(Roman Haas)
- 11. **Code-Zentralität:** Was sind in diesem Softwaresystem die wichtigsten Klassen?
(Roman Haas)



Recommendation

1. IConQATProcessor
2. ConQATProcessorBase
3. IConQATNode
4. ConQATException
5. ConQATPipeline
6. IResource
7. IElement
8. Finding
9. Clone
10. TextElement

Elmar



Themen (IV)

Qualität von Tests

- 12. Natural Language Test Smells:** Qualitätsdefekte in manuellen Tests automatisch erkennen.
(Benedikt)
- 13. Warum zerbrechen meine automatisierten GUI-Tests?** Statische Analyse von automatisierten Tests.
(Benedikt)
- 14. Anforderungsbasiertes Testen:** Wie stimme ich meine Tests auf die Anforderungen ab?
(Maximilian)

Natural Language Test Smells:

Qualitätsdefekte in manuellen Tests automatisch erkennen.

Step Description		Expected Result
Step ...	...	...
Step 6	Repeat step 4 (Tab Section/Search) and step 5 as long as you want.	...
Step ...	...	...

repeatability?

comprehensibility?

Step Description		Expected Result
Step ...	...	...
Step 12	...	The entries differ depending on chosen View Mode but ...
Step ...	...	...

maintenance?

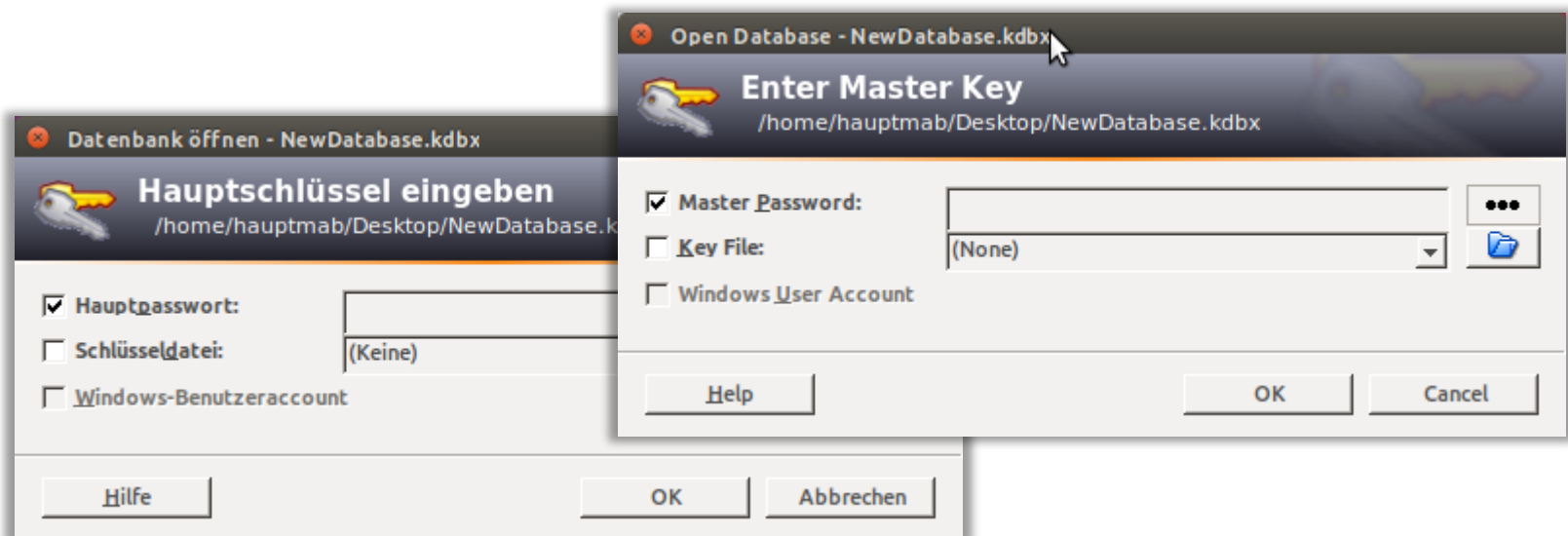
Step 1	Reports -> Execute Report -> Risk Capital Modeling Öffnen des Reports 'NBS Exposures by Group and Scenario'	Keine Reporting wird angezeigt.
Step 2	Eingaben: - Years of History = -14 - Include excluded CV = NO - Default Rating = empty Klick auf Button 'Excel'	Fehlermeldung: "Years of history" darf nicht negativ sein.
Step 3	Eingaben: - Years of History = empty - Include excluded CV = Yes - Default Rating = empty Klick auf Button 'Excel'	Keine Einschränkung, alle Jahre werden angezeigt. Wir beim Pricing werden einzelne Kalenderjahre ausgeschlossen.
Step 4	In Datenblatt 'Summary' überprüfen, ob: - Zeitpunkt der Berichterstattung: - Währung: - Skalierung: generiert sind.	Report wird geöffnet. Die Datenblätter: - 'Summary', - 'TE', - 'SN', - 'Fac', - 'Risk', - 'NL', - 'NL-Struktur', - 'Calendar' wurden generiert.
Step 5	In Datenblatt 'TE' überprüfen, ob die Spalten: - Group ID (Text), - Group ID (Text), - Bufile ID (Text), - Calendar (Text), - CARAT Scenario ID (Text), - Prop Exposure PNL (Gecalcult) generiert sind.	Die Spalten sind vorhanden und mit Daten ausgefüllt.

Warum zerbrechen meine automatisierten GUI-Tests?

Statische Analyse von automatisierten Tests.

Automated Test 01 – Login Dialog

1.	Open application	C:\programs\keepass\2.0\keepass.exe	
2.	Validate	Windows is visible	"Open Database – New Database.kdbx"
3.	Enter value	"Master Password"	"123mysecret\$"
4.	Click	Button	"Cancel"
...	...	...	...

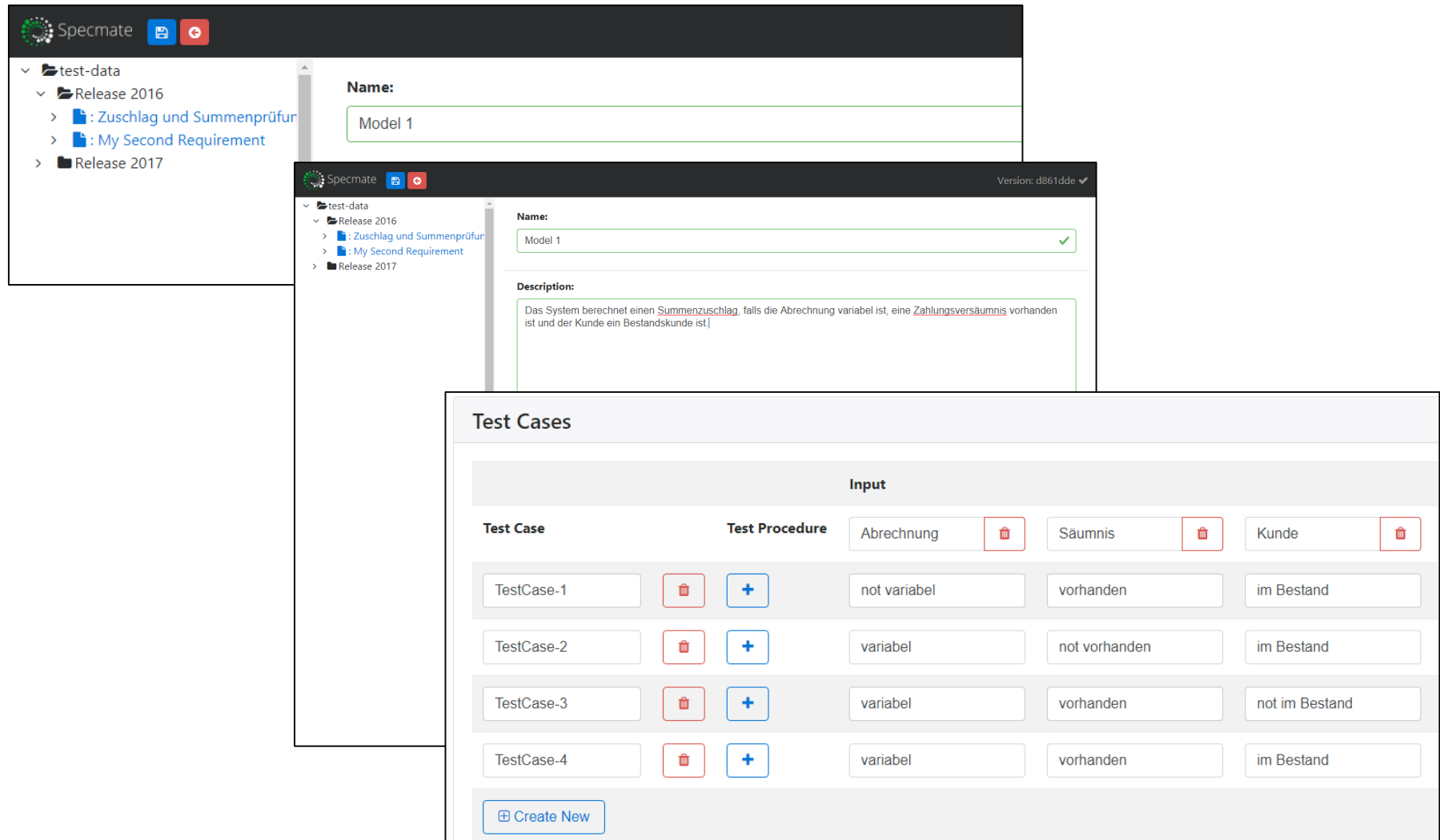


Themen (IV)

Qualität von Tests

- 12. **Natural Language Test Smells:** Qualitätsdefekte in manuellen Tests automatisch erkennen.
(Benedikt)
- 13. **Warum zerbrechen meine automatisierten GUI-Tests?** Statische Analyse von automatisierten Tests.
(Benedikt)
- 14. **Anforderungsbasiertes Testen:** Wie stimme ich meine Tests auf die Anforderungen ab?
(Maximilian)

Anforderungsbasiertes Testen: Wie stimme ich meine Tests auf die Anforderungen ab?



The image displays the Specmate software interface, which is used for requirement-based testing. It shows a hierarchical view of test data on the left and a detailed view of a test case on the right.

Test Data View (Left):

- test-data
 - Release 2016
 - Zuschlag und Summenprüfung
 - My Second Requirement
 - Release 2017

Test Case View (Right):

Name: Model 1

Description: Das System berechnet einen Summenzuschlag, falls die Abrechnung variabel ist, eine Zahlungsversäumnis vorhanden ist und der Kunde ein Bestandskunde ist.

Test Cases Table:

Test Case	Test Procedure	Abrechnung	Säumnis	Kunde
TestCase-1	not variabel	vorhanden	im Bestand	
TestCase-2	variabel	not vorhanden	im Bestand	
TestCase-3	variabel	vorhanden	not im Bestand	
TestCase-4	variabel	vorhanden	im Bestand	

[Create New](#)

Themen (V)

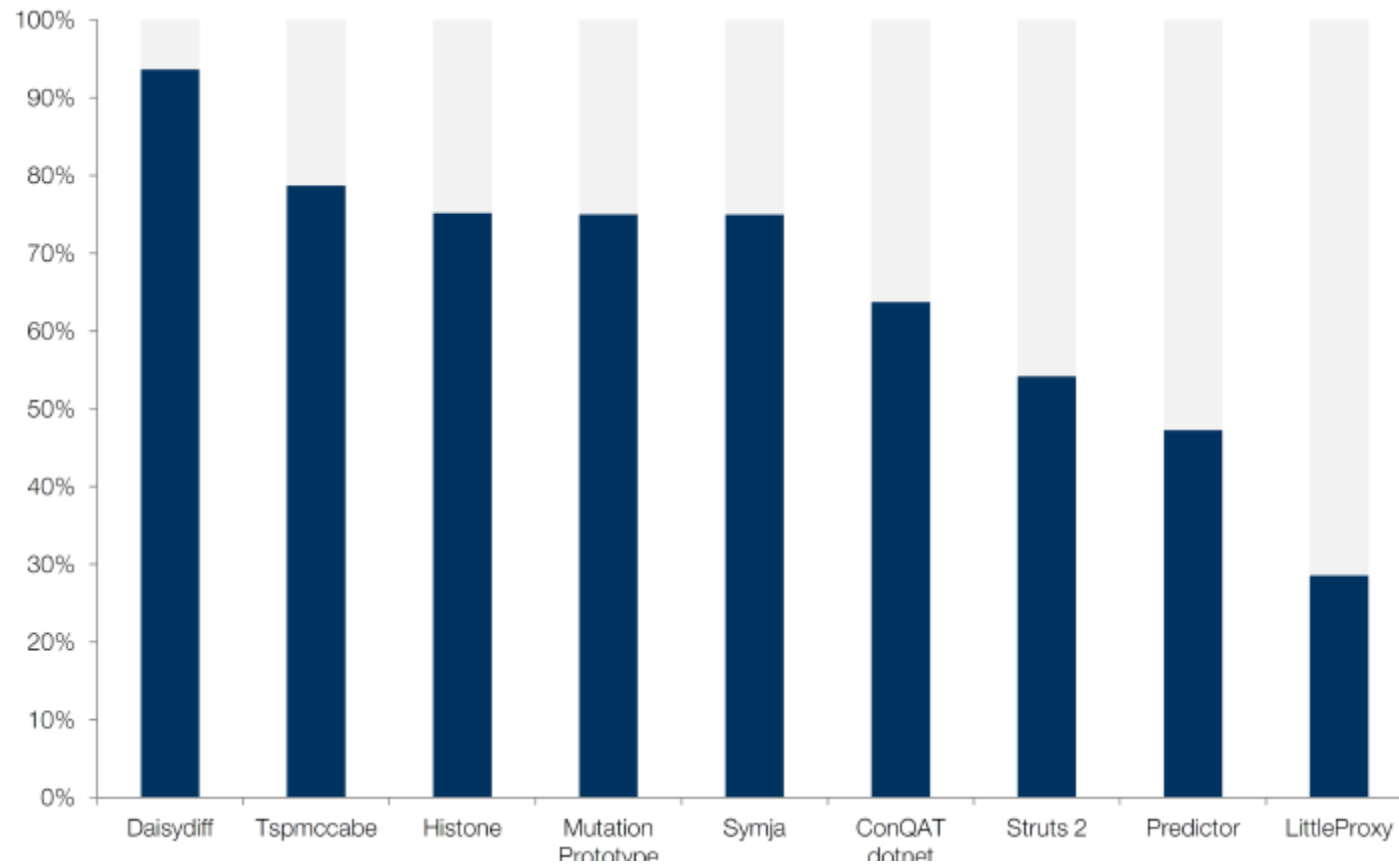
Qualität von Tests

15. Qualitätssicherung von Tests durch Mutation Testing

(Daniel Veihelmann)

16. Test Gap Analyse: Werden Systemänderungen auch wirklich von Tests erfasst?

(Daniel Veihelmann)



Themen (V)

Qualität von Tests

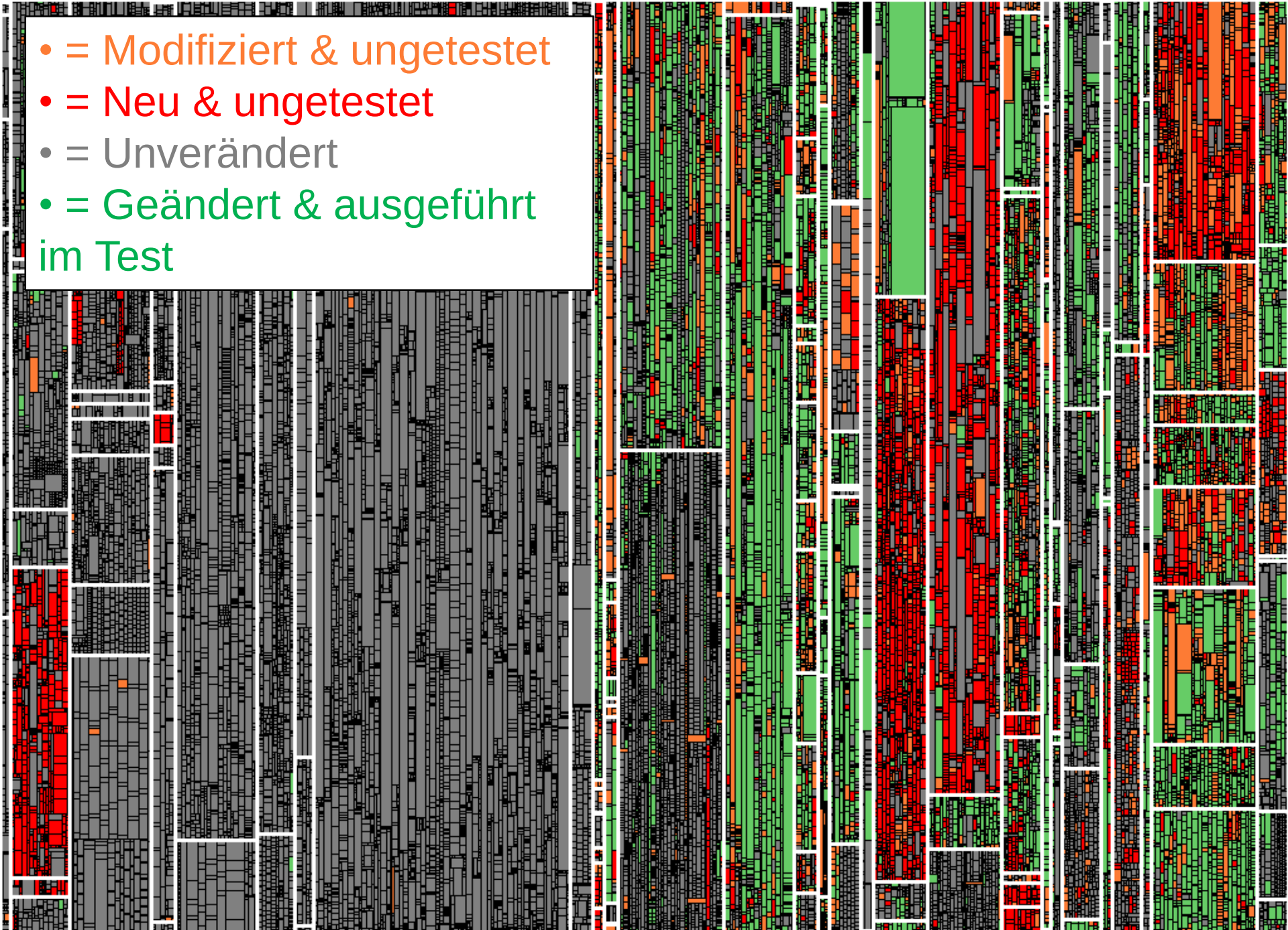
15. Qualitätssicherung von Tests durch Mutation Testing

(Daniel Veihelmann)

16. **Test Gap Analyse:** Werden Systemänderungen auch wirklich von Tests erfasst?

(Daniel Veihelmann)

- = Modifiziert & ungetestet
- = Neu & ungetestet
- = Unverändert
- = Geändert & ausgeführt im Test



Themen (VI)

Prozessqualität / Anforderungsqualität

- 17. Besserer Prozess durch bessere Anforderungen?** Wie hängen die Qualität von Artefakten und die Qualität des Prozesses zusammen?
(Daniel)
- 18. Prozessverbesserung im Requirements Engineering:** Prinzipien und Ansätze
(Daniel)

Fragen? Anmerkungen?