

Bilgisayar Sistemi Mühendisliği



Yazılım İsterleri Çözümlemesi



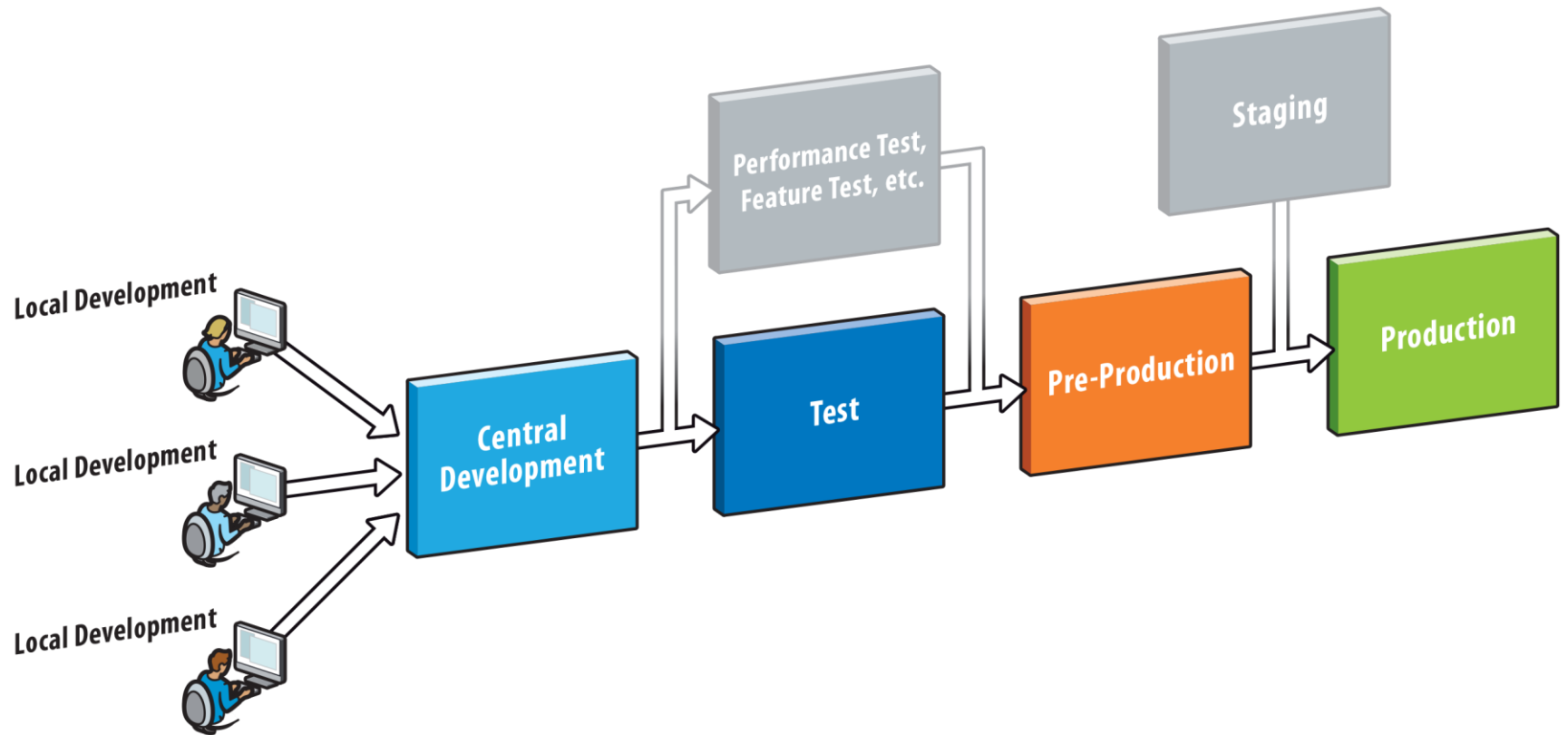
Yazılım Tasarımı

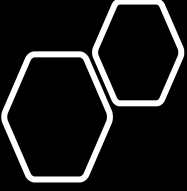


Gerçekleştirim(Kodlama)



Test ve Teslim





GERÇEKLEŞTİRİM

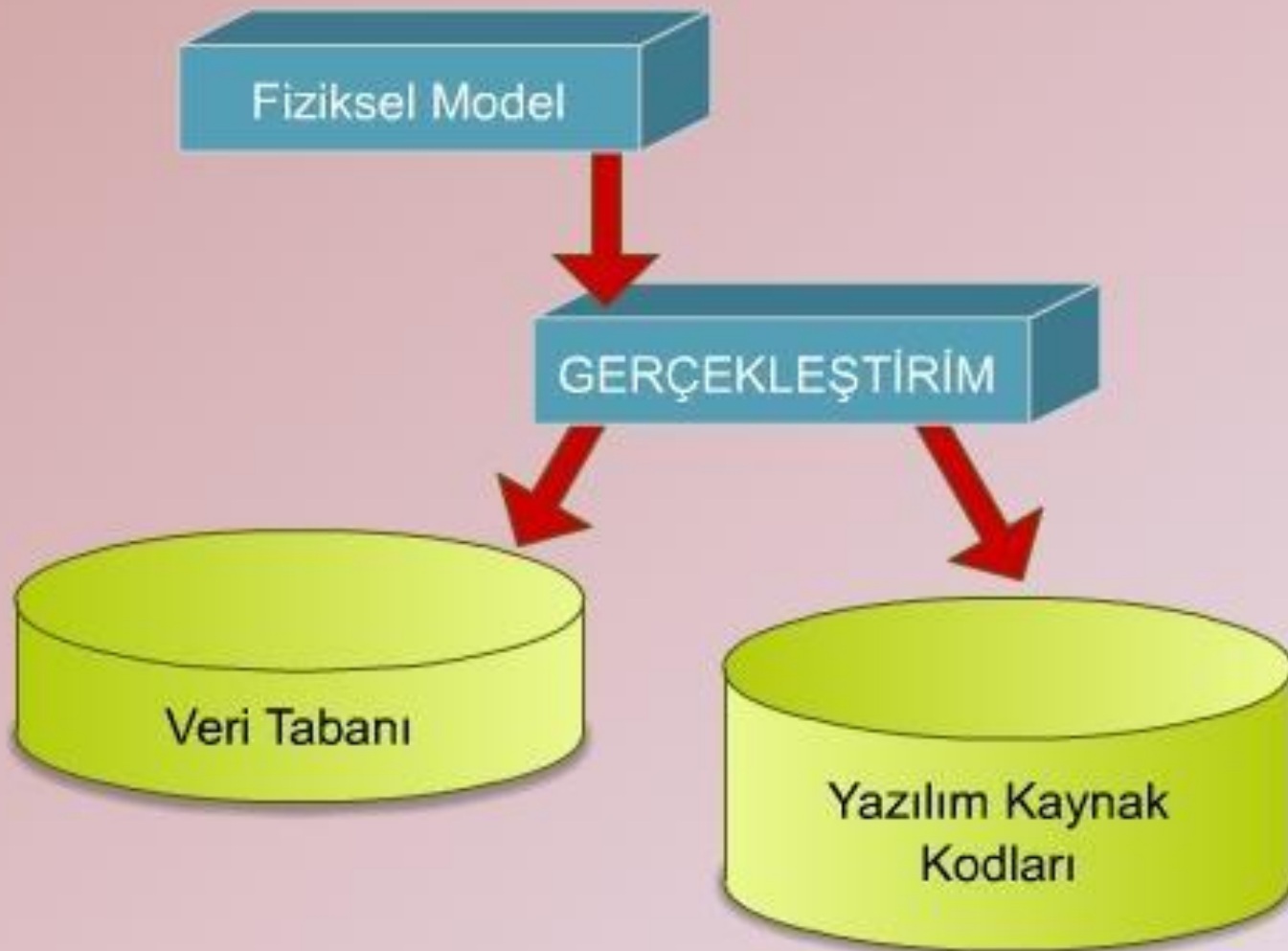
- Gerçekleştirim çalışması, tasarım sonucu üretilen süreç ve veri tabanının fiziksel yapısını içeren **fiziksel modelin bilgisayar ortamında çalışan yazılım biçimine dönüştürülmesi** çalışmalarını içerir.
- Yazılımın geliştirilmesi için her şeyden önce belirli bir **yazılım geliştirme ortamının** seçilmesi gerekmektedir.
- Söz konusu ortam, kullanılacak **programlama dili ve yazılım geliştirme araçlarını** içerir.





GERÇEKLEŞTİRİM

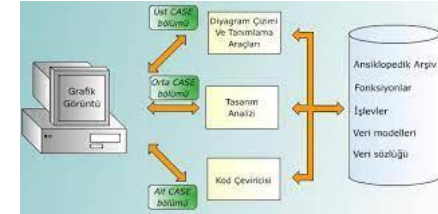
- Söz konusu ortamda belirli bir standartta geliştirilen programlar, **gözden geçirilir, sinanır ve uygulamaya hazır** hale getirilir.
- **Üretilen kaynak kodların belirlenecek bir standartta üretilmesi yazılımın daha sonraki aşamalarda bakım açısından çok önemlidir.**
- Tersî durumda kaynak **kodların okunabilirliği, düzeltilebilirliği zorlaşır** ve yazılımın işletimi süresince ortaya çıkabilecek sorunlar kolayca çözülemez.



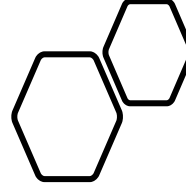
YAZILIM GELİŞTİRME ORTAMLARI

- Yazılım geliştirme ortamı, tasarım sonunda üretilen fiziksel modelin, bilgisayar ortamında çalıştırılabilmesi için gerekli olan:

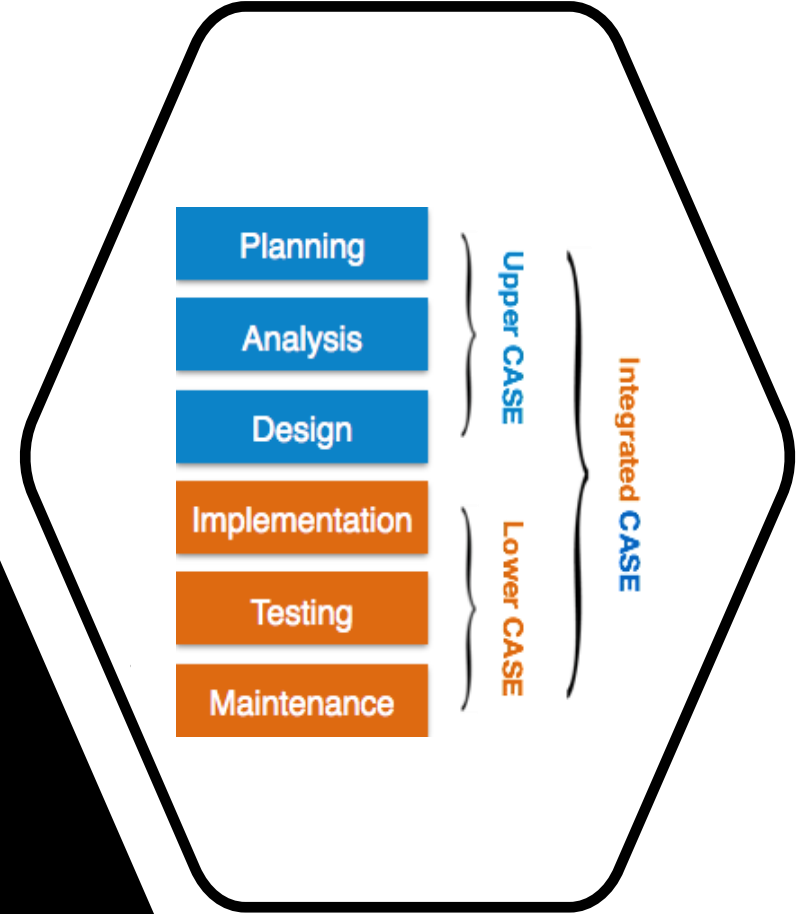
- Programlama dili
- Veri tabanı yönetim sistemi
- CASE araçları (Computer aided software engineering)



CASE Araç ve Ortamları



- Yazılım süreci etkinlikleri için otomatik destek sağlamak üzere geliştirilmiş yazılım araçlarıdır
- Günümüzde bilgisayar destekli yazılım geliştirme ortamları (CASE) oldukça gelişmiş durumdadır.
- CASE araçları, yazılım üretiminin hemen her aşamasında (planlama-çözümleme-tasarım-gerçekleştirim-sınama) üretilen bilgi ya da belgelerin bilgisayar ortamında saklanmasını, bu yolla kolay erişilebilir ve yönetilebilir olmasını olanaklı kılar.
- *Bu yolla yapılan üretimin yüksek kalitede olması sağlanır.*



CASE Araç ve Ortamları

CASE sistemleri genelde yöntemleri desteklemek için kullanılır.

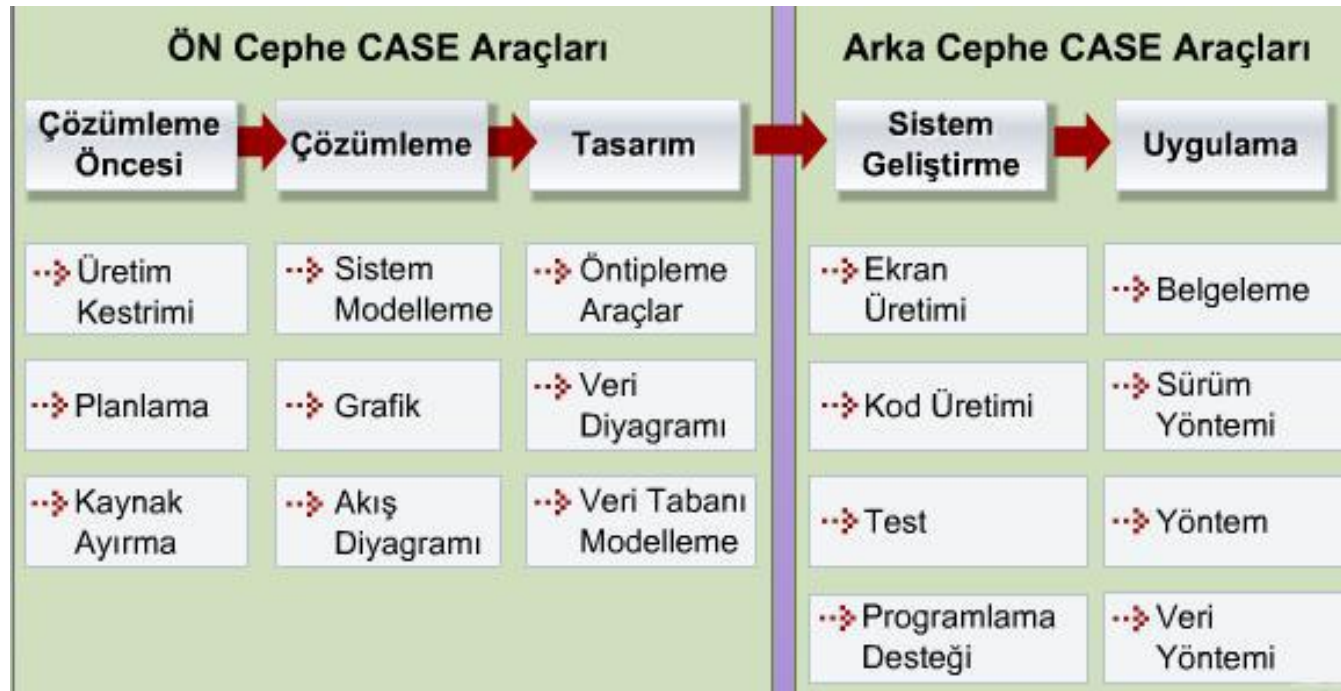
“Upper-CASE”

Analiz ve tasarım gibi erken geliştirme etkinliklerini destekleyen araçlardır.

“Lower-CASE”

Programlama, hata ayıklama ve test gibi geliştirme etkinliklerini destekleyen araçlardır.

CASE Araçları ve Kullanım Evreleri



“CASE” Ne Yapar, Ne Yapamaz?

CASE teknolojisi, yazılım sürecini desteklemek üzere gelişme göstermiştir.

- Sistem modelini geliştirme için grafiksel editörler

- Kullanıcı arayüzü oluşturmak için grafik kullanıcı arayüzü yazılımları

- Programdaki hataları bulmak için hata ayıklayıcılar

- Detay tasarımdan kod oluşturan dönüştürücüler

Ancak bu gelişmelerden hiçbiri, yazılım sürecinin insana bağımlı unsurlarını adresleyemez.

- Yazılım geliştirme bilişsel algılama ve ifade gerektirir.

- Yazılım geliştirme ekip işidir ve özellikle büyük kapsamlı projelerde, proje zamanının önemli kısmı iletişimle geçer.

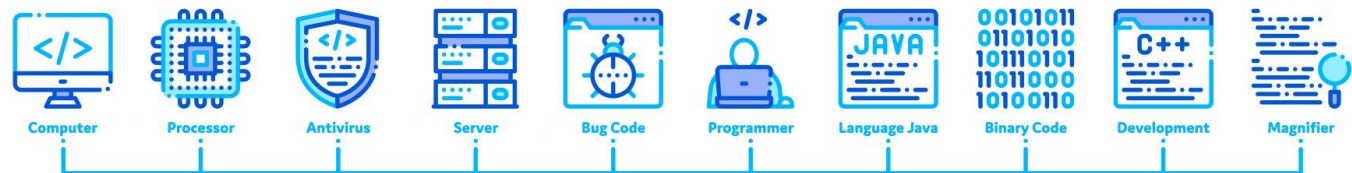
CASE ARAÇ GRUPLARI

CASE araçları değişik açılardan sınıflandırılabilir. Bu sınıfların isimlendirilmesi de değişebilir. Bu bölümde işlevler açısından araç sınıflandırması sunulmaktadır.

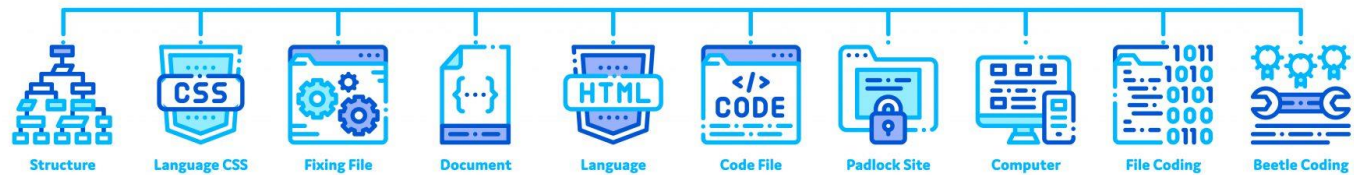
- Bilgi Mühendisliği Araçları
- Süreç Modelleme Araçları
- Proje Planlama Araçları
- Risk Çözümleme Araçları
- Proje Yönetimi Araçları
- Gereksinim Tanımlama Araçları
- Ölçüm ve Ölçme Yönetimi Araçları
- Belgeleme Araçları
- Sistem Yazılımı Araçları
- Kalite Araçları
- Veritabanı Yönetim Araçları
- Yazılım Konfigürasyon Yönetimi Araçları
- Çözümleme ve Tasarım Araçları
- Benzetim Araçları
- Kullanıcı Arayüzü Araçları
- Öntipleme Araçları
- Programlama Araçları
- Bütünleştirme ve Sınama Araçları
- Sınama Yönetim Araçları
- İstemci/sunucu Sınama Araçları

İşlevsel CASE Araçları

Tool type	Examples
• Planning tools	PERT tools, estimation tools, spreadsheets
• Editing tools	Text editors, diagram editors, word processors
• Change management tools	Requirements traceability tools, change control systems
• Configuration management tools	Version management systems, system building tools
• Prototyping tools	Very high-level languages, user interface generators
• Method-support tools	Design editors, data dictionaries, code generators
• Language-processing tools	Compilers, interpreters
• Program analysis tools	Cross reference generators, static analysers, dynamic analysers
• Testing tools	Test data generators, file comparators
• Debugging tools	Interactive debugging systems
• Documentation tools	Page layout programs, image editors



CODING



Visual Paradigm for UML - UML tool for software application development

Visual Paradigm for UML (VP-UML) is a UML design tool and UML CASE tool designed to aid software development. VP-UML supports key industry standards such as Unified Modeling Language (UML), SysML, BPMN, XMI, etc. It offers complete toolset software development teams need for requirements capturing, software planning, test planning, class modeling, data modeling, and etc.

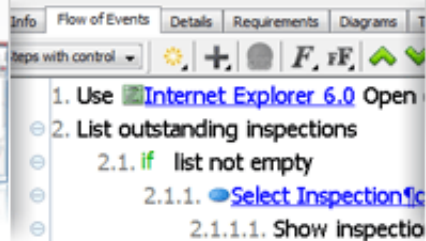
UML Modeling

- Software design with UML
- UML profile support
- Design pattern support



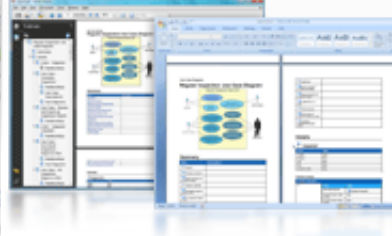
Use Case Modeling

- Draw use case diagrams
- Write flow of events
- Generate sequence diagram



Report Generation

- PDF, HTML and MS Word
- Fully customizable
- Publish to Web



Code Engineering

- Java to sequence diagram
- Java round-trip
- C++ round-trip



- <http://www.visual-paradigm.com/>
- <http://case-tools.org/>

Tasarımdan sonra iskelet program için yazılım mühendisliği araçları (CASE tools) kullanılabilmektedir. Araçlar, ara yüzlerin tanımlanması ve uygulanması için kodları içermekte, geliştiriciler ise bazı detayları ekleyebilmektedir (Sommerville, 2000).

Yüksek seviyede yazılmış kodu makine koduna çeviren ve çalıştıran derleyiciye (compiler), çalışma zamanında birçok modülü birbirine bağlayan bağlayıcıya (linker), hafızaya ürünün çalışan versiyonunu yüklemek için ise yükleyiciye (loader) ihtiyaç vardır.

Programlama Dilleri

Programlama dillerini çeşitli açılardan sınıflandırabiliriz. En sık kullanılan sınıflandırmalar;

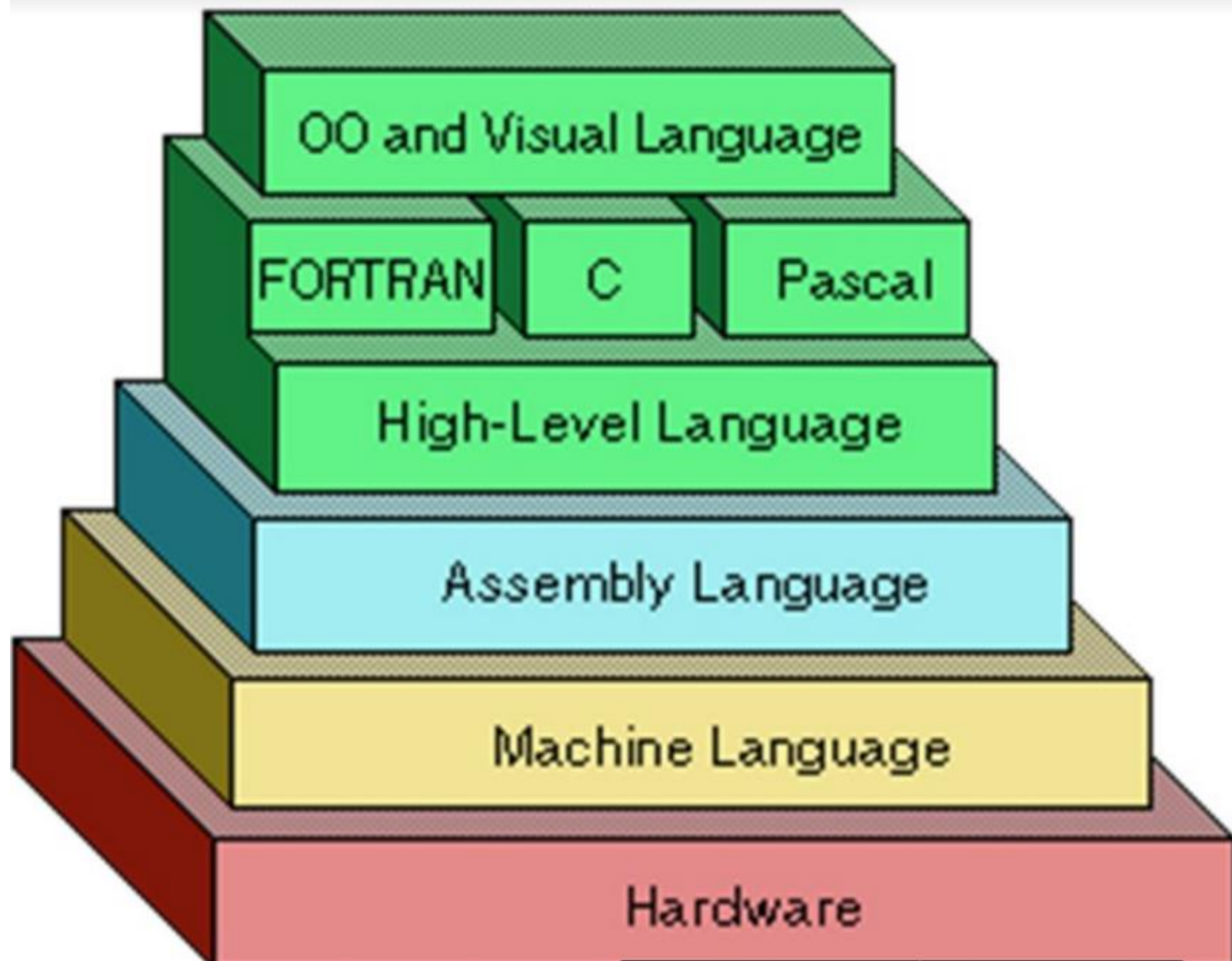
- 1) seviyelerine göre sınıflandırma
- 2) uygulama alanlarına göre sınıflandırma



The Evolution Of Computer Programming Languages

```

00 01 02 03 04 05 06 07
08 09 10 11 12 13 14 15
16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31
32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47
48 49 50 51 52 53 54 55
56 57 58 59 60 61 62 63
64 65 66 67 68 69 70 71
72 73 74 75 76 77 78 79
80 81 82 83 84 85 86 87
88 89 90 91 92 93 94 95
96 97 98 99 100 101 102 103
104 105 106 107 108 109 110 111
112 113 114 115 116 117 118 119
120 121 122 123 124 125 126 127
128 129 130 131 132 133 134 135
136 137 138 139 140 141 142 143
144 145 146 147 148 149 150 151
152 153 154 155 156 157 158 159
160 161 162 163 164 165 166 167
168 169 170 171 172 173 174 175
176 177 178 179 180 181 182 183
184 185 186 187 188 189 190 191
192 193 194 195 196 197 198 199
200 201 202 203 204 205 206 207
208 209 210 211 212 213 214 215
216 217 218 219 220 221 222 223
224 225 226 227 228 229 230 231
232 233 234 235 236 237 238 239
240 241 242 243 244 245 246 247
248 249 250 251 252 253 254 255
256 257 258 259 260 261 262 263
264 265 266 267 268 269 270 271
272 273 274 275 276 277 278 279
280 281 282 283 284 285 286 287
288 289 290 291 292 293 294 295
296 297 298 299 300 301 302 303
304 305 306 307 308 309 310 311
312 313 314 315 316 317 318 319
320 321 322 323 324 325 326 327
328 329 330 331 332 333 334 335
336 337 338 339 340 341 342 343
344 345 346 347 348 349 350 351
352 353 354 355 356 357 358 359
360 361 362 363 364 365 366 367
368 369 370 371 372 373 374 375
376 377 378 379 380 381 382 383
384 385 386 387 388 389 390 391
392 393 394 395 396 397 398 399
400 401 402 403 404 405 406 407
408 409 410 411 412 413 414 415
416 417 418 419 420 421 422 423
424 425 426 427 428 429 430 431
432 433 434 435 436 437 438 439
440 441 442 443 444 445 446 447
448 449 450 451 452 453 454 455
456 457 458 459 460 461 462 463
464 465 466 467 468 469 470 471
472 473 474 475 476 477 478 479
480 481 482 483 484 485 486 487
488 489 490 491 492 493 494 495
496 497 498 499 500 501 502 503
504 505 506 507 508 509 510 511
512 513 514 515 516 517 518 519
520 521 522 523 524 525 526 527
528 529 530 531 532 533 534 535
536 537 538 539 540 541 542 543
544 545 546 547 548 549 550 551
552 553 554 555 556 557 558 559
560 561 562 563 564 565 566 567
568 569 570 571 572 573 574 575
576 577 578 579 580 581 582 583
584 585 586 587 588 589 590 591
592 593 594 595 596 597 598 599
600 601 602 603 604 605 606 607
608 609 610 611 612 613 614 615
616 617 618 619 620 621 622 623
624 625 626 627 628 629 630 631
632 633 634 635 636 637 638 639
640 641 642 643 644 645 646 647
648 649 650 651 652 653 654 655
656 657 658 659 660 661 662 663
664 665 666 667 668 669 670 671
672 673 674 675 676 677 678 679
680 681 682 683 684 685 686 687
688 689 690 691 692 693 694 695
696 697 698 699 700 701 702 703
704 705 706 707 708 709 710 711
712 713 714 715 716 717 718 719
720 721 722 723 724 725 726 727
728 729 730 731 732 733 734 735
736 737 738 739 740 741 742 743
744 745 746 747 748 749 750 751
752 753 754 755 756 757 758 759
760 761 762 763 764 765 766 767
768 769 770 771 772 773 774 775
776 777 778 779 780 781 782 783
784 785 786 787 788 789 790 791
792 793 794 795 796 797 798 799
800 801 802 803 804 805 806 807
808 809 810 811 812 813 814 815
816 817 818 819 820 821 822 823
824 825 826 827 828 829 830 831
832 833 834 835 836 837 838 839
840 841 842 843 844 845 846 847
848 849 850 851 852 853 854 855
856 857 858 859 860 861 862 863
864 865 866 867 868 869 870 871
872 873 874 875 876 877 878 879
880 881 882 883 884 885 886 887
888 889 890 891 892 893 894 895
896 897 898 899 900 901 902 903
904 905 906 907 908 909 910 911
912 913 914 915 916 917 918 919
920 921 922 923 924 925 926 927
928 929 930 931 932 933 934 935
936 937 938 939 940 941 942 943
944 945 946 947 948 949 950 951
952 953 954 955 956 957 958 959
960 961 962 963 964 965 966 967
968 969 970 971 972 973 974 975
976 977 978 979 980 981 982 983
984 985 986 987 988 989 990 991
992 993 994 995 996 997 998 999
1000 1001 1002 1003 1004 1005 1006 1007
1008 1009 1010 1011 1012 1013 1014 1015
1016 1017 1018 1019 1020 1021 1022 1023
1024 1025 1026 1027 1028 1029 1030 1031
1032 1033 1034 1035 1036 1037 1038 1039
1040 1041 1042 1043 1044 1045 1046 1047
1048 1049 1050 1051 1052 1053 1054 1055
1056 1057 1058 1059 1060 1061 1062 1063
1064 1065 1066 1067 1068 1069 1070 1071
1072 1073 1074 1075 1076 1077 1078 1079
1080 1081 1082 1083 1084 1085 1086 1087
1088 1089 1090 1091 1092 1093 1094 1095
1096 1097 1098 1099 1100 1101 1102 1103
1104 1105 1106 1107 1108 1109 1110 1111
1112 1113 1114 1115 1116 1117 1118 1119
1120 1121 1122 1123 1124 1125 1126 1127
1128 1129 1130 1131 1132 1133 1134 1135
1136 1137 1138 1139 1140 1141 1142 1143
1144 1145 1146 1147 1148 1149 1150 1151
1152 1153 1154 1155 1156 1157 1158 1159
1160 1161 1162 1163 1164 1165 1166 1167
1168 1169 1170 1171 1172 1173 1174 1175
1176 1177 1178 1179 1180 1181 1182 1183
1184 1185 1186 1187 1188 1189 1190 1191
1192 1193 1194 1195 1196 1197 1198 1199
1200 1201 1202 1203 1204 1205 1206 1207
1208 1209 1210 1211 1212 1213 1214 1215
1216 1217 1218 1219 1220 1221 1222 1223
1224 1225 1226 1227 1228 1229 1230 1231
1232 1233 1234 1235 1236 1237 1238 1239
1240 1241 1242 1243 1244 1245 1246 1247
1248 1249 1250 1251 1252 1253 1254 1255
1256 1257 1258 1259 1260 1261 1262 1263
1264 1265 1266 1267 1268 1269 1270 1271
1272 1273 1274 1275 1276 1277 1278 1279
1280 1281 1282 1283 1284 1285 1286 1287
1288 1289 1290 1291 1292 1293 1294 1295
1296 1297 1298 1299 1300 1301 1302 1303
1304 1305 1306 1307 1308 1309 1310 1311
1312 1313 1314 1315 1316 1317 1318 1319
1320 1321 1322 1323 1324 1325 1326 1327
1328 1329 1330 1331 1332 1333 1334 1335
1336 1337 1338 1339 1340 1341 1342 1343
1344 1345 1346 1347 1348 1349 1350 1351
1352 1353 1354 1355 1356 1357 1358 1359
1360 1361 1362 1363 1364 1365 1366 1367
1368 1369 1370 1371 1372 1373 1374 1375
1376 1377 1378 1379 1380 1381 1382 1383
1384 1385 1386 1387 1388 1389 1390 1391
1392 1393 1394 1395 1396 1397 1398 1399
1400 1401 1402 1403 1404 1405 1406 1407
1408 1409 1410 1411 1412 1413 1414 1415
1416 1417 1418 1419 1420 1421 1422 1423
1424 1425 1426 1427 1428 1429 1430 1431
1432 1433 1434 1435 1436 1437 1438 1439
1440 1441 1442 1443 1444 1445 1446 1447
1448 1449 1450 1451 1452 1453 1454 1455
1456 1457 1458 1459 1460 1461 1462 1463
1464 1465 1466 1467 1468 1469 1470 1471
1472 1473 1474 1475 1476 1477 1478 1479
1480 1481 1482 1483 1484 1485 1486 1487
1488 1489 1490 1491 1492 1493 1494 1495
1496 1497 1498 1499 1500 1501 1502 1503
1504 1505 1506 1507 1508 1509 1510 1511
1512 1513 1514 1515 1516 1517 1518 1519
1520 1521 1522 1523 1524 1525 1526 1527
1528 1529 1530 1531 1532 1533 1534 1535
1536 1537 1538 1539 1540 1541 1542 1543
1544 1545 1546 1547 1548 1549 1550 1551
1552 1553 1554 1555 1556 1557 1558 1559
1560 1561 1562 1563 1564 1565 1566 1567
1568 1569 1570 1571 1572 1573 1574 1575
1576 1577 1578 1579 1580 1581 1582 1583
1584 1585 1586 1587 1588 1589 1590 1591
1592 1593 1594 1595 1596 1597 1598 1599
1600 1601 1602 1603 1604 1605 1606 1607
1608 1609 1610 1611 1612 1613 1614 1615
1616 1617 1618 1619 1620 1621 1622 1623
1624 1625 1626 1627 1628 1629 1630 1631
1632 1633 1634 1635 1636 1637 1638 1639
1640 1641 1642 1643 1644 1645 1646 1647
1648 1649 1650 1651 1652 1653 1654 1655
1656 1657 1658 1659 1660 1661 1662 1663
1664 1665 1666 1667 1668 1669 1670 1671
1672 1673 1674 1675 1676 1677 1678 1679
1680 1681 1682 1683 1684 1685 1686 1687
1688 1689 1690 1691 1692 1693 1694 1695
1696 1697 1698 1699 1700 1701 1702 1703
1704 1705 1706 1707 1708 1709 1710 1711
1712 1713 1714 1715 1716 1717 1718 1719
1720 1721 1722 1723 1724 1725 1726 1727
1728 1729 1730 1731 1732 1733 1734 1735
1736 1737 1738 1739 1740 1741 1742 1743
1744 1745 1746 1747 1748 1749 1750 1751
1752 1753 1754 1755 1756 1757 1758 1759
1760 1761 1762 1763 1764 1765 1766 1767
1768 1769 1770 1771 1772 1773 1774 1775
1776 1777 1778 1779 1780 1781 1782 1783
1784 1785 1786 1787 1788 1789 1790 1791
1792 1793 1794 1795 1796 1797 1798 1799
1800 1801 1802 1803 1804 1805 1806 1807
1808 1809 1810 1811 1812 1813 1814 1815
1816 1817 1818 1819 1820 1821 1822 1823
1824 1825 1826 1827 1828 1829 1830 1831
1832 1833 1834 1835 1836 1837 1838 1839
1840 1841 1842 1843 1844 1845 1846 1847
1848 1849 1850 1851 1852 1853 1854 1855
1856 1857 1858 1859 1860 1861 1862 1863
1864 1865 1866 1867 1868 1869 1870 1871
1872 1873 1874 1875 1876 1877 1878 1879
1880 1881 1882 1883 1884 1885 1886 1887
1888 1889 1890 1891 1892 1893 1894 1895
1896 1897 1898 1899 1900 1901 1902 1903
1904 1905 1906 1907 1908 1909 1910 1911
1912 1913 1914 1915 1916 1917 1918 1919
1920 1921 1922 1923 1924 1925 1926 1927
1928 1929 1930 1931 1932 1933 1934 1935
1936 1937 1938 1939 1940 1941 1942 1943
1944 1945 1946 1947 1948 1949 1950 1951
1952 1953 1954 1955 1956 1957 1958 1959
1960 1961 1962 1963 1964 1965 1966 1967
1968 1969 1970 1971 1972 1973 1974 1975
1976 1977 1978 1979 1980 1981 1982 1983
1984 1985 1986 1987 1988 1989 1990 1991
1992 1993 1994 1995 1996 1997 1998 1999
2000 2001 2002 2003 2004 2005 2006 2007
2008 2009 2010 2011 2012 2013 2014 2015
2016 2017 2018 2019 2020 2021 2022 2023
2024 2025 2026 2027 2028 2029 2030 2031
2032 2033 2034 2035 2036 2037 2038 2039
2040 2041 2042 2043 2044 2045 2046 2047
2048 2049 2050 2051 2052 2053 2054 2055
2056 2057 2058 2059 2060 2061 2062 2063
2064 2065 2066 2067 2068 2069 2070 2071
2072 2073 2074 2075 2076 2077 2078 2079
2080 2081 2082 2083 2084 2085 2086 2087
2088 2089 2090 2091 2092 2093 2094 2095
2096 2097 2098 2099 2100 2101 2102 2103
2104 2105 2106 2107 2108 2109 2110 2111
2112 2113 2114 2115 2116 2117 2118 2119
2120 2121 2122 2123 2124 2125 2126 2127
2128 2129 2130 2131 2132 2133 2134 2135
2136 2137 2138 2139 2140 2141 2142 2143
2144 2145 2146 2147 2148 2149 2150 2151
2152 2153 2154 2155 2156 2157 2158 2159
2160 2161 2162 2163 2164 2165 2166 2167
2168 2169 2170 2171 2172 2173 2174 2175
2176 2177 2178 2179 2180 2181 2182 2183
2184 2185 2186 2187 2188 2189 2190 2191
2192 2193 2194 2195 2196 2197 2198 2199
2200 2201 2202 2203 2204 2205 2206 2207
2208 2209 2210 2211 2212 2213 2214 2215
2216 2217 2218 2219 2220 2221 2222 2223
2224 2225 2226 2227 2228 2229 2230 2231
2232 2233 2234 2235 2236 2237 2238 2239
2240 2241 2242 2243 2244 2245 2246 2247
2248 2249 2250 2251 2252 2253 2254 2255
2256 2257 2258 2259 2260 2261 2262 2263
2264 2265 2266 2267 2268 2269 2270 2271
2272 2273 2274 2275 2276 2277 2278 2279
2280 2281 2282 2283 2284 2285 2286 2287
2288 2289 2290 2291 2292 2293 2294 2295
2296 2297 2298 2299 2300 2301 2302 2303
2304 2305 2306 2307 2308 2309 2310 2311
2312 2313 2314 2315 2316 2317 2318 2319
2320 2321 2322 2323 2324 2325 2326 2327
2328 2329 2330 2331 2332 2333 2334 2335
2336 2337 2338 2339 2340 2341 2342 2343
2344 2345 2346 2347 2348 2349 2350 2351
2352 2353 2354 2355 2356 2357 2358 2359
2360 2361 2362 2363 2364 2365 2366 2367
2368 2369 2370 2371 2372 2373 2374 2375
2376 2377 2378 2379 2380 2381 2382 2383
2384 2385 2386 2387 2388 2389 2390 2391
2392 2393 2394 2395 2396 2397 2398 2399
2400 2401 2402 2403 2404 2405 2406 2407
2408 2409 2410 2411 2412 2413 2414 2415
2416 2417 2418 2419 2420 2421 2422 2423
2424 2425 2426 2427 2428 2429 2430 2431
2432 2433 2434 2435 2436 2437 2438 2439
2440 2441 2442 2443 2444 2445 2446 2447
2448 2449 2450 2451 2452 2453 2454 2455
2456 2457 2458 2459 2460 2461 2462 2463
2464 2465 2466 2467 2468 2469 2470 2471
2472 2473 2474 2475 2476 2477 2478 2479
2480 2481 2482 2483 2484 2485 2486 2487
2488 2489 2490 2491 2492 2493 2494 2495
2496 2497 2498 2499 2500 2501 2502 2503
2504 2505 2506 2507 2508 2509 2510 2511
2512 2513 2514 2515 2516 2517 2518 2519
2520 2521 2522 2523 2524 2525 2526 2527
2528 2529 2530 2531 2532 2533 2534 2535
2536 2537 2538 2539 2540 2541 2542 2543
2544 2545 2546 2547 2548 2549 2550 2551
2552 2553 2554 2555 2556 2557 2558 2559
2560 2561 2562 2563 2564 2565 2566 2567
2568 2569 2570 2571 2572 2573 2574 2575
2576 2577 2578 2579 2580 2581 2582 2583
2584 2585 2586 2587 2588 2589 2590 2591
2592 2593 2594 2595 2596 2597 2598 2599
2600 2601 2602 2603 2604 2605 2606 2607
2608 2609 2610 2611 2612 2613 2614 2615
2616 2617 2618 2619 2620 2621 2622 2623
2624 2625 2626 2627 2628 2629 2630 2631
2632 2633 2634 2635 2636 2637 2638 2639
2640 2641 2642 2643 2644 2645 2646 2647
2648 2649 2650 2651 2652 2653 2654 2655
2656 2657 2658 2659 2660 2661 2662 2663
2664 2665 2666 2667 2668 2669 2670 2671
2672 2673 2674 2675 2676 2677 2678 2679
2680 2681
```



Bilgisayar Dillerinin Seviyelerine Göre Sınıflandırılması

1. Kuşak diller: Doğal dilidir ve bilgisayarın donanımsal tasarımına bağlıdır. Bilgisayarların geliştirilmesiyle birlikte onlara iş yaptırmak için kullanılan ilk diller de makine dilleri olmuştur. Bu yüzden makine dillerine 1. kuşak diller denilebilir.

2. kuşak diller: 1950'li yılların hemen başlarında makine dili kullanımının getirdiği problemleri ortadan kaldırmaya yönelik çalışmalar yoğunlaşmıştır. Sembolik makine dilleri (Assembly languages) yalnızca 1 ve 0 dan oluşan makine dilleri yerine İngilizce bazı kısaltma sözcüklerden oluşuyordu.

"Compiler" derleyici buluşu bu dönemde yapılmıştır. Assembly diller 2. kuşak diller olarak tarihte yerini almıştır.

3. kuşak diller: Tarihsel süreç içinde Assembly dillerinden daha sonra geliştirilmiş ve daha yüksek seviyeli diller 3. kuşak diller sayılmaktadır. Bu dillerin hepsi algoritmik dillerdir. Fortran, Pascal, Basic, Cobol,...

4. kuşak diller: Çok yüksek seviyeli ve genellikle algoritmik yapı içermeyen programların görsel bir ortamda yazıldığı diller ise 4. kuşak diller olarak isimlendirilirler. Access, Foxpro, Paradox, Xbase, Visual Basic, Oracle Forms,...

Ortalama bir satırındaki kodlama 3. kuşak dildeki 10 satır kodlamaya eşittir.

Bilgisayar Dillerinin Seviyelerine Göre Sınıflandırılması

3. kuşak diller prosedürlüdür. Programcı her işin bilgisayar tarafından “nasıl” yerine getirileceğini detaylı bir şekilde tanımlamak zorundadır.

4.kuşak dillerde buna gerek yoktur. Sadece “ne” yapılması gerektiği tanımlanır. Farklı bir deyişle;

4. kuşak diller öncekilere göre daha problem amaçlıdır.

4. kuşak dille geliştirme daha hızlıdır, daha az programcı gerektirdiğinden daha az maliyetlidir.

Dilin öğrenilmesi ve o dilde program yazılması daha kolaydır (Britton ve Doake, 1993).

Uygulama Alanlarına Göre Sınıflandırma

a) Bilimsel ve mühendislik uygulama dilleri: Pascal, C, FORTRAN

b) Veri tabanı dilleri : XBASE, Oracle, Clipper, Visual Foxpro....

c) Genel amaçlı programlama dilleri: Pascal, C, Basic.

d) Yapay zekâ dilleri: Prolog, Lisp

e) Sistem programlama dilleri: C, Assembler, sembolik makine dilleri.

Programlama Dili Seçimi

Hangi dilin seçileceği belirlenirken tüm kriterlerin göz önünde bulundurulması gerekir. En önemli kriterler:

Uygunluk: Bazı diller genel amaçlıdır ve çeşitli programlarda kullanılabilir, bazıları ise özel amaçlıdır ve bazı sınırlı işlerde kullanılabilirler. Dil seçimi kullanıcının ve kullanımın ayırt edilmesini gerektirir.

Karmaşıklık: Yüksek seviyeli diller karmaşık kontrol yapılarını ve veri yapılarını içermelidir. Kontrol yapıları, okuması ve oluşturması kolay, açık mantıksal yapıdaki programların dış yapısını oluştururlar. Diller birçok veri yapısını destekleyecek şekilde seçilmelidir.

Programlama Dili Seçimi

Organizasyonel Düşünce: Bir dilin etkili olması için kullanıcı için çabuk öğrenilebilir olması gerekmektedir. Oluşturulması ve değerlendirilmesi kolay ve organizasyonla birlikte büyüyebilecek kadar esnek olmalıdır.

Destek: Bir yazılım satın alınırken başka organizasyonlar tarafından geniş bir kullanıma sahip, destek veren firmalar ve servisler tarafından kontrol edilmiş olması önemlidir.

Etkinlik: Bir yazılım satın alınırken o dilin derlendiğinde ve çalıştığında kalan performansının etkinliği önemlidir. Birim hafıza başına düşen makine fiyatı düşerken, makine zamanında etkin olmayan fakat programlama zamanında etkin olan dillerin önemi giderek artmaktadır (Atan, 2004).

Hata tipleri

Kullanılan Ölçümler Hata sayısı tahmin etmek için basit ölçüm kod satırı sayısıdır. Hata sayısı önemlidir, çünkü bulunan hata sayısı önceden belirlenen maksimumu geçerse modül tekrar tasarım ve kod aşamalarına girmelidir. Hata tipleri de önemlidir.

Tipik hata tipleri; tasarımı anlayamamayı, birbirini tutmayan değişkenlerin kullanımını içerir. Tutulacak hata dataları, daha sonraki gelecek ürünlerde hata kontrolü sırasında kullanılan kontrol listesine eklenebilir (Schach, 1993).

<https://polen.itu.edu.tr/bitstream/11527/2015/1/3978.pdf>

Programlama dilleri





Hangi platformda geliştirilirse geliştirilsin, **yazılımın belirli bir düzende kodlanması** yazılım yaşam döngüsünün uygulama boyutu açısından oldukça önem taşımaktadır.

KODLAMA STİLİ

Yazılım ya da bilgi sistemleri, doğaları gereği durağan değildir.

Uygulamanın gerektirdiği güncel değişikliklerin ilgili yazılıma da aktarılması gerekir.

Bu nedenle yazılımın kodlarına zaman zaman başvurmak, yeni kod parçaları eklemek ya da var olan kodlarda değişiklikler yapmak yazılım yaşam döngüsünün işletimsel boyutunun en önemli işlevlerinden biridir. "**Bakım Programcısı**" kavramı bu tür gereksinimlerden doğmuştur.

Bakım programcısının temel görevleri, varolan yazılıma ilişkin kodlar dahil üretilmiş tüm bilgi ve belgeleri incelemek ve yazılım üzerinde değişiklikler yapmak biçiminde özetlenebilir.

Kolay okunabilir, anlaşılabilir kodları olmayan yazılımın bakımı oldukça zorlaşır ve büyük maliyetlere ulaşır.

• Etkin kod yazılım stili için kullanılan yöntemler:

- Açıklama Satırları
- Kod Yazım Düzeni
- Anlamlı İsimlendirme
- Yapısal Programlama Yapıları



Açıklama Satırları

- Bir program parçasını anlaşılabilir kılan en önemli unsurlardan biri bu program kesiminde içerilen açıklama satırlarıdır. Her bir program modülü içerisine
 - **Modül başlangıç açıklamaları**
 - **Modül Kod Açıklamaları**
 - **Boşluk satırları** eklenmelidir.
- Her bir modülün temel işlevleri, yazar kişi vb bilgiler ilgili modülün en başına modül başlangıç açıklama satırları olarak eklenmelidir.

```
1 <?php
2 /*Aşağıdaki programda
3 iki değişken toplanıp
4 sonuç ekrana yazdırılmıştır
5 */
6 $a=20;
7 $b=10;
8 $topla=$a+$b;
9 echo $topla;
10 ?>
```

Açıklama Satırları

1

Bir programın karmaşıklığını arttıran en önemli bileşenler, program içerisinde kullanılan denetim yapılarıdır (**Koşullu deyimler, döngü deyimleri**).

2

Bu tür deyimlerin hemen öncesinde bu denetim işleminin açıklamasını ve bu deyimde olabilecek **olağan dışı durumları içeren kod açıklama satırları** koyulmalıdır

Kod Biçimlemesi

Programın **okunabilirliğini artırmak** ve anlaşılabilirliğini kolaylaştırmak amacıyla açıklama satırlarının kullanımının yanı sıra, belirli bir kod yazım düzeninin de kullanılması gerekmektedir.

```
for i:=1 to 50 do begin i:=i+1; end;    // Kötü kodlanmış
```

```
for i:=1 to 50 do                        // İyi kodlanmış
begin
    i:=i+1;
end;
```


Anlamalı İsimlendirme

Kodların okunabilirliğini ve anlaşılabilirliğini sağlayan önemli unsurlardan biri de kullanılan ve kullanıcı tarafından belirlenen belirteçlerin (**Değişken adları, kütük adları, Veri tabanı tablo adları, işlev adları, yordam adları vb**) anlamlı olarak isimlendirilmesidir.

İsimlendirme yöntemi uygulamayı geliştirenler tarafından çözümleme-tasarım aşamalarında belirlenmeli ve gerçekleştirim aşamasında uygulanmalıdır.



Anlamlı İsimlendirme

Bütün standartlarda ortak olması gereken noktaları ise şöyle sıralayabiliriz :

Tanımlayıcının(değişkenin,sınıfın,metodun vb...) amacı doğrultusunda isimler verilmesi gerekir.

Mesela okuldaki öğrenci sayısını tutan bir değişkene “tamsayı” şeklinde isim vermek yerine “**OğrenciSayisi**” şeklinde isim vermek daha mantıklı olacaktır.

Anlamlı İsimlendirme

Tanımlayıcının ismi **büyük ve küçük harfleriyle** okunabilir ve anlaşılır uzunlukta olmalıdır.

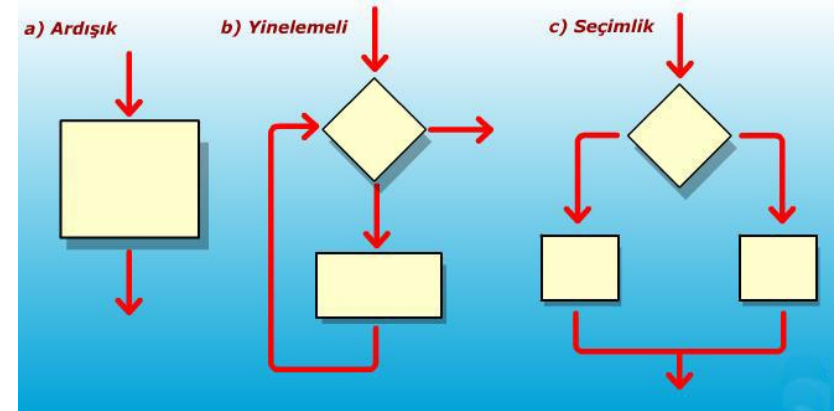
Mümkün olduğunca kısaltmaları azaltmalıdır. Çünkü kısaltmalar çoğu zaman tehlikeli olabilmektedir. Örneğin “Ctr” “Control” olarak anlaşılabilceği gibi “counter” olarak da anlaşılabilir.

- <http://www.csharpnedir.com/articles/read/?id=99&title=.NET%20%C4%B0%C3%A7in%20Tavsiye%20Edilen%20%C4%B0simlendirme%20Konvansiyonlar%C4%B1%20-%201> (Macar Notasyonu, String olarak strFirstName, integer olarak iNumberOfDays bu notasyona uygun isimlendirmelerdir.)

Yapısal Programlama Yapıları

Yazılım mantıksal bütünlük gösteren bloklara (bölümlere) bölünebilirler. Bu yolla uzun ve karmaşık programların, **bloklara ayırarak** daha kolay biçimde yazılabilmesi mümkün olmaktadır.

Bloklar halinde alt programlardan oluşturulan lojik birimler; ardışık, yinelemeli ve seçimlik olarak üç ayrı bileşim şeklinde düzenlenmektedir.



Yapısal Programlama Yapıları

Yapısal program, **sadece bir giriş ve bir çıkışı bulunan** soyutlanmış bloklardan oluştuğu için, ayrı kişiler tarafından bağımsız olarak tasarlanıp düzenlenebilmekte, sınanması-değiştirilmesi ve işletilmesi kolaylaştırılmaktadır.



OLAĞANDIŞI DURUM ÇÖZÜMLEME

1

Olağan dışı durum, bir programın çalışmasının, geçersiz ya da yanlış veri oluşumu ya da başka nedenlerle istenmeyen bir biçimde sonlanmasına neden olan durum olarak tanımlanmaktadır.

2

Genelde kabul edilen kural, bir programın işletiminin sonlandırılması işleminin elektrik kesintisi vb. donanım hataları dışında bütünüyle program denetiminde olmasıdır.

OLAĞANDIŐI DURUM ÇÖZÜMLEME

Bu kural olađan dıőı sonlandırma iőlemini de kapsamaktadır. Bu nedenle program kodları oluőturulurken, olađandıőı durumların da dikkate alınması ve bu durumlardaki program davranıőına iliőkin yöntemler geliőtirilmesi gerekmektedir.



OLAĞANDIŞI DURUM ÇÖZÜMLEME

Örneğin, normalde sıfır değeri almaması gereken bir değişken sıfır değerini aldığı anda program, bu değişkene ilişkin bir bölme işleminde "**sıfıra bölme hatası**" nedeniyle işletim sistemi tarafından kesilmemeli, bu tür bir yanlış uyarısı vererek durmalıdır.

Günümüzdeki programlama dilleri, bu tür olağan dışı durumlarda programın yapması gereken işlevi kapsayacak "**olağan dışı durum çözümleyicileri**" ya da yordamları" tanımlarını içermektedir.

Olağandışı Durum Tanımları

Olağan dışı durumlar, **programlama dili tarafından tanımlı durumlar olduğu gibi kullanıcı tarafından da tanımlanabilmektedir.**

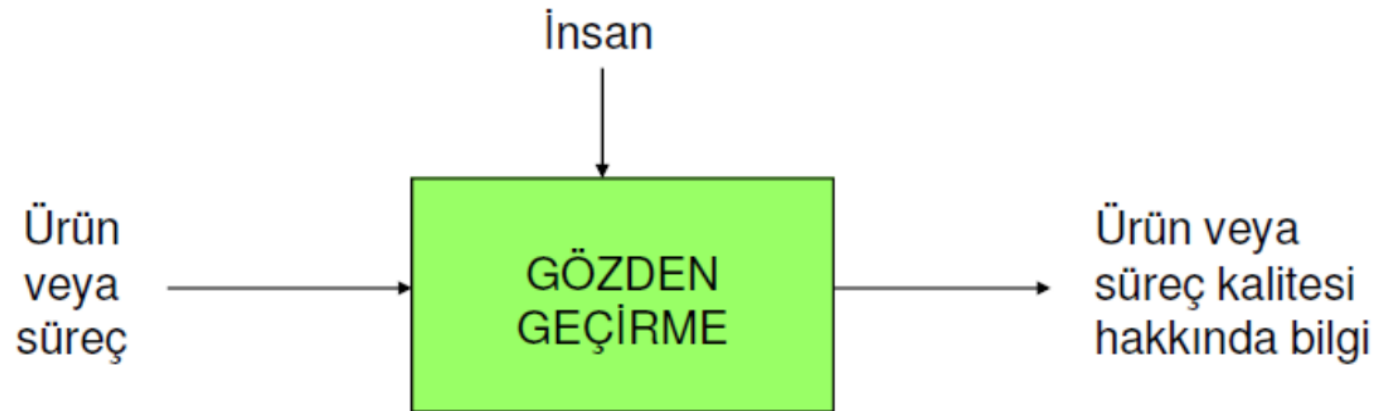
Bu tür olağan dışı durumlar, olağan dışı durumu tanımlayan ve sonucunda "doğru" ya da "yanlış" değeri üreten biri mantıksal yordam ya da fonksiyon tanımından oluşmaktadır

Farklı Olağandışı Durum Çözümleme Yaklaşımları

Herhangi bir olağandışı durumda, program tarafından değişik işlemler yapılabilir, değişik önlemler alınabilir. Bu konudaki yaklaşımlar, aşağıda belirtilmiştir.

- Anında durdurma,
- Hata kodu verme,
- Tanımlı olağandışı yordam çalıştırma,
- Hata yordamı çalıştırma.

Yazılım gözden geçirme



Yazılım gözden geçirme

- **IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits** “A process or meeting during which a software product, set of software products, or a software process is presented to project personnel, managers, users, customers, user representatives, auditors or other interested parties for examination, comment or approval.”

- Proje personeline, yöneticilere, kullanıcılara, müşterilere, kullanıcı temsilcilerine, denetçilere veya diğer ilgili taraflara inceleme, yorum veya onay için bir yazılım ürününün, yazılım ürünleri setinin veya bir yazılım sürecinin sunulduğu bir süreç veya toplantı.



Gözden geçirme

Çözümleme, tasarım ve kodlama bir yada birkaç kişi tarafından ne kadar iyi yapılmış olursa olsun, **bir başka kişi ya da kişilerin hata bulması her zaman olasıdır.** Geliştirme aşamasında bulunan her hata, ürünü mükemmelliğe daha çok yaklaştırır. Bu nedenle de herkesin her yaptığı işe gözden geçirme (review) uygulanmalıdır.



Gözden geçirme

Gözden geçirmeler yazılım mühendisliği için bir tür süzgeç gibidir. Geliştirme sürecinin çeşitli evrelerinde yazılım ürününün yazarından başka kişiler tarafından incelenmesi şeklinde uygulanarak ;

- Kusurların ortaya çıkarılmasını,
- Uygun şekilde düzeltilmesini,
- Ürünün daha da iyileştirmesini sağlarlar.

Kusur her zaman bir hata olmayabilir, fakat yanlış anlama yada deneyim eksikliği nedeniyle asıl isterlerden ve standartlardan sapma olarak da değerlendirilebilir.



Neden Gözden Geçirme?

- Ürününün veya sürecin farklı bakış açılarıyla sistematik olarak değerlendirilmesini sağlar.
- Proje zamanlamasını ve maliyetini iyileştirir.
- Test etkinliğini destekler ve maliyetini düşürür.
- Yatırım geri dönüşü yüksektir.
- Bir çeşit eğitim yöntemidir.



Yazılım Kalite Güvence: Yöntemler

- **Statik yöntemler: Kodu çalıştırmadan yapılır. (GÖZDEN GEÇİRME)**

Örnek: Gereksinim ,tasarım belgeleri, kodun denetimi veya gözden geçirilmesi

- İnceleme (“inspection”)
- Gözden geçirme (“review”)
- Denetleme (“audit”)

- **Dinamik yöntemler: Kodu çalıştırarak yapılır. (TEST)**

Örnek: Ürünün veya bileşenlerinin gerçeğe yakın testi

- Birim test (“unit test”)
- Türleştirme test (“integration test”)
- Sistem test (“system test” / “functional test” / “qualification test”)
- Kabul test (“acceptance test”)



Kalite Nedir?

- “Ability to satisfy requirements”
- **İhtiyaçları karşılama yeteneğidir.**
- “Conformance to user requirements”
- **Kullanıcı gereksinimlerine uygunluktur.**
- “Doing right it the first time”
- **İlk seferde doğrusunu yapmaktır.**
- “Fitness for intended use”
- **Amaçlanan kullanıma uygunluktur.**



Kalite Nasıl Sağlanır?

- **Geleneksel anlayış: Hata ayıklama**
 - “Kalite kontrol”
 - Bir ürün veya hizmetin tanımlanmış gereksinimleri karşılayıp karşılamadığının tespitinde kullanılan teknikler ve uygulanan faaliyetler
 - Ürün/hizmet üzerinden kaliteyi sağlama anlayışı



HOW?

Gözden geçirme türleri

- **Eş düzey gözden geçirme:** Proje çalışanlarının genellikle aynı düzeyde bulunan personel ile birlikte yürüttükleri gözden geçirmelere eş düzey gözden geçirme denir. Tasarımcılar tasarımla ilgili, kodlayıcılar da kodla ilgili gözden geçirmelere katılırlar.
- **Birleşik gözden geçirme:** Geliştirici ile müşterinin, sözleşmede yer alan yönetsel ve teknik işleri, iş adımlarını, aşamaları gözden geçirmek için beraber yaptıkları toplantıya birleşik gözden geçirme denir.
- **Resmi teknik gözden geçirme:** 3 aşaması vardır. İnceleme, denetleme ve kod geçişleri. Bu etkinliklerin her biri toplantı şeklinde gerçekleştirilir, dikkatli bir planlama yapılır, eksiksiz katılım sağlanması, denetim altında yürütülmesi ve sonuçların kayıt altına alınıp açıklanması gerekir.

Resmi teknik gözden geçirme türleri



Denetleme(Audit): Bir yazılım ürününün, bir yazılım sürecinin veya bir dizi yazılım süreç faaliyetlerinin belirtiler , standartlar, sözleşme veya diğer unsurlar bakımından uyumunun değerlendirilmesi için yapılan sistematik değerlendirmedir.



İnceleme(Inspection): Bir yazılım ürünündeki hatalar ve standartlardan sapmalara neden olan anormalliklerin belirlenip tanımlanması için inceleme teknikleri konusunda eğitilmiş, tarafsız kişilerin rehberliğinde denek kişilerin katılımıyla gerçekleştirilen sorgulamadır.



Kod geçişleri(Walk-through): Bir yazılım geliştiricisi tarafından diğer geliştirme ekip üyelerine anlatılarak, yazılım ürününün iyileştirilmesine yönelik görüşlerin alınması ve standartların ihlali veya olası hataların belirlenmesidir.

Resmi teknik gözden geçirme süreci

- **Planlama:** Gözden geçirme sürecinin hazırlanması ve organize edilmesidir. Bu kapsamda gözden geçirme materyalleri, yordamları, toplantı takvimi, gözden geçirmeye katılacak olan kişiler ve rolleri hakkında hazırlıklar gerçekleştirilir.
- **Bilgilendirme:** Bu aşamanın amacı gözden geçirmeye katılacak olanların gözden geçirme hakkında ve gözden geçirilecek olan ürün hakkında eğitilmesidir. Bu aşamada amaç, gözden geçirme ve ürün hakkında tüm ekibin temel bilgi düzeyine ulaşmalarını sağlamaktır.
- **Bireysel Hazırlık:** Ürün hakkında gerekli bilgileri öğrenen ekip elemanları daha sonra kendilerine ait roller ile gözden geçirilecek olan ürünü inceler ve ilgili gözden geçirme kayıtlarını doldurur. Bununla gözden geçirme toplantısından önce ürün üzerindeki hata, kusur ve eksikliklerin keşfedilmesi amaçlanır

Resmi teknik gözden geçirme süreci

- **Grup Toplantısı:** Bu toplantı ile bireysel olarak tespit edilen hata, kusur ve aksaklıklar bir araya getirilir. Gözden geçirme toplantısı, genelde, gözden geçirilecek ürün sorumlusunun ürünü kısaca tanıtımı ile başlar. Daha sonra bireysel gözden geçirmelerde tespit edilen hata, kusur ve eksiklikler teker teker gündeme getirilir. Gerekli düzeltici faaliyetler planlanarak ilgili kişilere göre ataması yapılır.
- **Tekrar Çalışma:** Bu süreç «hata düzeltme süreci» olarak da adlandırılır. Grup toplantısında karar verilen düzeltici faaliyetlerin ilgili kişilerce ilgili kişilerce gerçekleştirildiği süreçtir.
-
- **İzleme:** Bu aşamada, belirlenen tüm eylem maddelerinin yerine getirilip getirilmediği gözden geçirme sorumlusu tarafından izlenir ve kontrol altında tutulur. Gereken durumlarda ürün için yeni bir toplantı daha yapılabilir.

Gözden geçirmede dikkat edilmesi gerekkenler

Gözden geçirmeler planlanmalıdır.

Katılımcı sayısı ne az ne de fazla olmalıdır.

Bir gündem belirlenmeli ve uyulmalıdır.

İnatçı tartışmalar sınırlandırılmamalıdır.

Problemler hakkındaki düşünceler açıkça ortaya konulmalıdır.

Tüm problem alanlarına değinilmelidir.

Yazılı notlar alınmalıdır.

Yazılım ürün deęerlendirmesi

- Her yazılım ařamasının sonunda mutlaka bir çıktı bulunur. Bu çıktı bir belge ya da bir yazılım kodu olabilir. Uygulanan nitelik güvence sistemine göre bu kapsamdaki her yazılım ürününün ilgili kişilerce gözden geçirilmesi ve Yazılım Nitelik Raporu düzenlenmesi gerekir. Bu raporda, ürünü gözden geçirecek kişilerin sorumluluk ve ilgi alanlarına göre dağıtım yapılmalıdır.



İnceleme kontrol listeleri

Resmi teknik gözden geçirmelerin her zaman mükemmel bir şekilde yapılması beklenemez. Geliştiricilerin ve yöneticilerin deneyim eksikliği, sistemin karmaşıklığı, personel sayısının azlığı bu toplantıların etkin bir şekilde yapılmasını güçleştirir. Gözden geçirme toplantılarında dikkat edilmesi gereken çeşitli kontrol listeleri vardır. Bu listeler aşağıdaki evreler için oluşturulmuştur:

- Sistem mühendisliği
- Yazılım proje planlaması
- Yazılım tasarımı
- Kodlama
- Yazılım tümleştirme ve test
- Bakım

Örnek bir inceleme kontrol listesi

Kontrol Sorusu	Karar	Notlar
Her öge için tasarım kısıtlamaları belirli midir?		
Başarım ölçütleri ve nasıl ölçüleceği belirlenmiş midir?		
Sistemin öğeleri arasında herhangi bir tutarsızlık var mıdır?		
Sistemin gerçekleştirim çözümünü teknik olarak uygulamak mümkün müdür?		
Önemli işlevler belirgin, anlaşılır ve sınırlandırılmış şekilde tanımlanmış mıdır?		
Sistemin doğrulama ve geçerlemesinin yapılabilmesi için yöntemler belirlenmiş midir?		
Sistemi oluşturan alt sistemlerin ve ana öğelerin işlevleri, bunların birbiriyle olan ara yüzleri tanımlanmış mıdır?		
Sistemi gerçekleştirmek üzere seçilen yol ve geliştirme yöntemi tanımlanmış mıdır, bu konuda diğer alternatif seçenekler göz önüne alınmış mıdır?		

Kalite Nasıl Sağlanır?

- **Gelişmiş anlayış: Hata önleme**

“Kalite güvence”

- Bir ürün veya hizmetin tanımlanmış gereksinimleri karşılanmasını yeterli derecede güvence altına almak için gerekli olan, tüm planlanmış ve sistematik faaliyetler
- Ürünü/hizmeti oluşturan sistem üzerinden kaliteyi sağlama anlayışı



Kaliteli Yazılım Ne Demektir?

- Gereksinimleri karşılayan,
- Amaçlanan kullanıma uygun,
- Zamanında tamamlanmış,
- Belirlenen bütçe sınırları içinde gerçekleştirilmiş,
- Standartlara uyumlu,
- Bakımı sağlanabilen yazılımdır.



Yazılımda Kalite

Geliştirdiğimiz yazılımın kalitesi, yazılımı nasıl geliştirdiğimize büyük ölçüde bağlıdır.

- Yazılım geliştirme süreci, yazılımı nasıl geliştirdiğimizi tanımlar.
- Kaliteyi, yazılım geliştirme aşamaları boyunca yazılım ürününe yerleştirmek zorundayız.
- **Kaliteyi en sonunda sağlamaya çalışmak hem zordur, hem de maliyet çok yükselir.**



Yazılımda Kalite için: Gözden geçirme ve test

Gözden geçirme ve test birbirini tamamlar

- **Her ikisi Doğrulama ve Geçerleme sürecinde kullanılır**
- **Doğrulama (Verification):** "Biz ürünü doğru mu geliştiriyoruz"
 - Yazılım belirteçlere uymalıdır
- **Geçerlilik (Validation):** "Biz doğru ürün mü geliştiriyoruz"
 - Yazılım gerçek kullanıcı isteklerini yerine getirmelidir.
- **Verification:** "Are we building the product right"
 - The software should conform to its specification
- **Validation:** "Are we building the right product"
 - The software should do what the user really requires



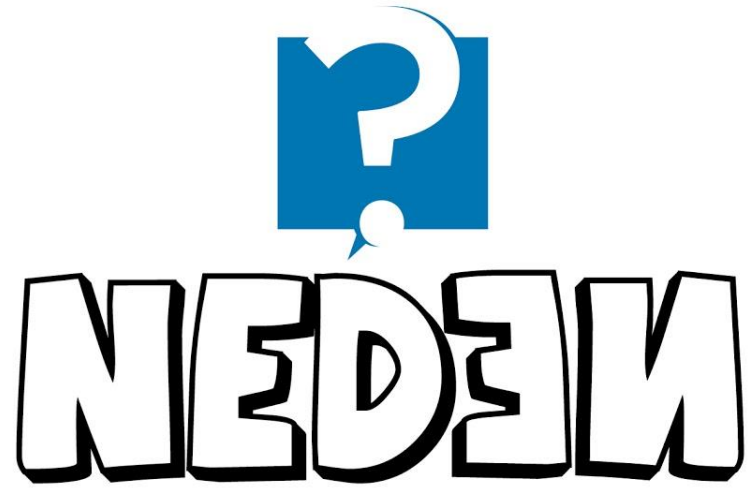
Yazılımda Kalitesizliğin Sorunları

- Müşteri tarafındaki sorunlar:
 - Gereksinimlerin sağlanamayışı
 - Kolay anlaşılabilir ve kullanılabilir olmaması
 - İstenilen zamanda bakım yapılabilir olmaması
 - Eğitim desteğinin yetersiz olması
- Yazılım firması tarafındaki sorunlar:
 - Geciken ya da bitmeyen projeler
 - Yüksek maliyet
 - Çalışanların tatminsizliği
 - Firmaya olan güven kaybı



Yazılımda Kalite Neden Gerekli?

- Tecrübeli bir mühendis her 7-10 satırda bir hata üretmektedir.
- Orta ölçekli bir projede binlerce hataya karşılık gelir.
- Hataların çoğunun test aşamasında düzeltilmesi gerekmektedir.
- Testler uzadıkça maliyet artmakta, teslimat gecikmektedir.



Yazılım Gözden Geçirme Süreci

Çıktılar:

- Projenin ihtiyaçlarına göre yönetim ve teknik gözden geçirmeler yapılır.
- Bir sürecin etkinliğine ait durum ve ürünler, gözden geçirme etkinlikleri ile değerlendirilir.
- Gözden geçirme sonuçları, etkilenen tüm birimlere duyurulur.
- Gözden geçirmeler sonucunda oluşan düzeltici faaliyetler, kapanana kadar izlenir.
- Riskler ve problemler belirlenir ve kaydedilir.





Gözden Geçirme Türleri

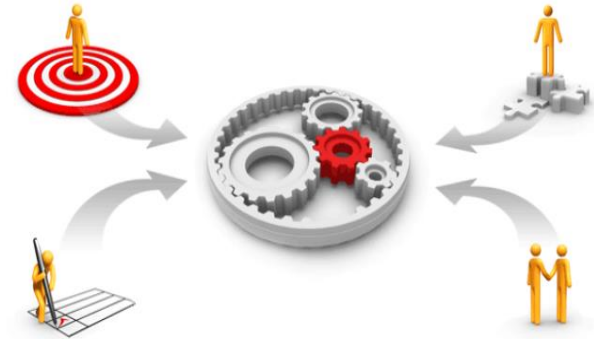
- Management reviews (“yönetim gözden geçirme”)
- Technical reviews (“teknik gözden geçirme”)
- Inspections (“detaylı inceleme”)
- Walkthroughs (“genel gözden geçirme”)

Yazılım Gözden Geçirme Süreci – Etkinlikler ve Görevler

• Teknik gözden geçirmeler

Yazılım ürünlerini veya hizmetlerini değerlendirmek için teknik gözden geçirmeler yapılacak ve aşağıdakiler hakkında kanıtlar sağlayacaktır (Örnek: **Kod gözden geçirme**)

- Ürün veya hizmet tamdır.
- Ürün veya hizmet standartlara ve tanımlara uygundur.
- Ürün veya hizmet tanımlı takvime uymaktadır.
- Ürün veya hizmet, planlanmış bir sonraki etkinlik için hazırdır.
- **Ürün veya hizmetin geliştirilmesi, işletilmesi ve bakımı; projenin tanımlı planlarına, takvimine, standartlarına ve talimatlarına göre yapılmaktadır.**



Yazılım Gözden Geçirme Süreci – Etkinlikler ve Görevler

Proje yönetim gözden geçirmeleri

Proje gelişiminin durumu; proje planlarına, takvime, standartlara ve yönergelere kıyasla değerlendirilecektir.

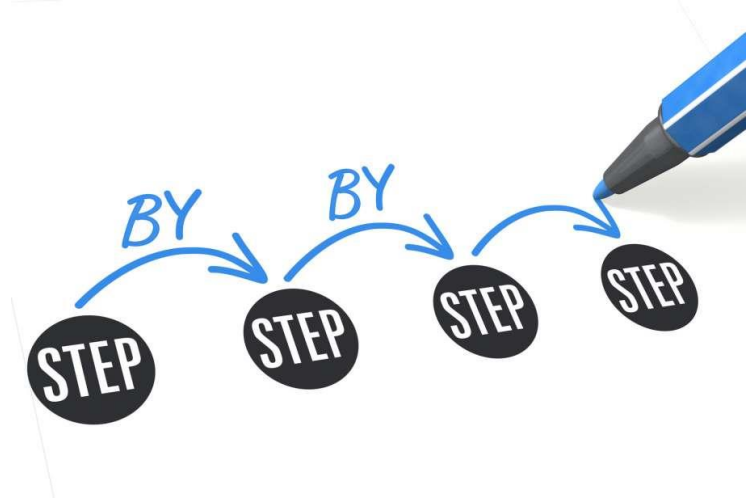
Değerlendirmenin çıktısı, ilişkili yönetim tarafından dikkate alınmalı ve aşağıdakileri sağlamalıdır:

- Etkinliklerin plana göre gelişmesini güvence etmek,
- **Kaynakları uygun şekilde atayarak** projenin genel kontrolünü korumak,
- Projenin yönünü değiştirmek veya **alternatif planlama** için ihtiyacı belirlemek,
- Proje başarısını olumsuz etkileyebilecek **riskleri değerlendirmek** ve yönetmek.



Gözden Geçirme Süreci

- Roller
 - Gözden geçirme lideri (“review leader”)
 - Gözden geçirici (“reviewer”)
 - Kayıt sorumlusu (“recorder”)
 - Yazar (“author”)
- Adımlar
 - Planlama (“planning”)
 - Açılış toplantısı (“kickoff meeting”)
 - Bireysel gözden geçirme (“individual checking”)
 - Toplu gözden geçirme (“logging meeting”)
 - Düzeltme ve izleme (“rework and follow up”)



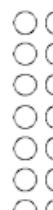
Gözden Geçirme Kontrol Listeleri

- Gözden geçirilecek her tür doküman için gözden geçirme kriterlerini tanımlar.
- Analiz dokümanı, tasarım dokümanı, kod, proje planı, kalite planı, vb.
- Gözden geçirmeyi bir kontrol listesi üzerinden yapmak, gözden geçirmenin etkinliğini artırır.



Yazılım Gereksinimleri Tanımı Belgesi

Örnek Kontrol Listesi



	(E=Evet, H=Hayır, U=Uygulanamaz)	
	(E / H / U)	Açıklamalar
Standartlara Uygunluk		
Doküman için standartlar/kılavuzlar ve isimlendirme kuralları belirlenmiş mi?		
Doküman formatı tanımlı standart/kılavuz ile uyumlu mu?		
Doküman boyunca standartlar ve isimlendirme kuralları takip edilmiş mi?		
Doküman İçeriği		
Üst düzey sistem genel bakışı var mı?		
Üst düzey sistem diyagramları iç ve dış arayüzleri ve veri akışlarını gösteriyor mu?		
Sistemin fonksiyonel akışı açıkça ve tam olarak tanımlanmış mı?		
Yazılım çevresi (ör. donanım, yazılım kaynakları, kullanıcılar) tanımlanmış mı?		
Tüm referans dokümanları listelenmiş mi?		
Tüm tanımlamalar, kısaltmalar ve kısaltmalar konmuş mu?		
Kullanıcı karakteristikleri tanımlanmış mı?		
Genel tasarım ve uygulama kısıtları not edilmiş mi?		
Uygulamayı etkileyen genel varsayımlar belirtilmiş mi?		
Hafıza gereksinimleri sağlanmış mı?		
Zamanlama ve hafıza limitleri donanım kısıtlarıyla uyumlu mu?		
Yazılım performansı üzerindeki tüm limitler ve sınırlamalar tanımlanmış mı?		
Her bir fonksiyon ayrı ayrı tanımlanmış mı?		
Her bir fonksiyon için herhangi yazılım sınırlamaları tartışılmış mı?		
Emniyet-kritik yazılım gereksinimleri tanımlanmış mı?		
Güvenlik gereksinimleri tanımlanmış mı?		
Gizlilik gereksinimleri tanımlanmış mı?		
Yazılım kalite gereksinimleri (güvenilirlik, taşınabilirlik, tekrar kullanılabilirlik, bakılabilirlik) tanımlanmış mı?		
Çevresel gereksinimler ve durumlar tanımlanmış mı?		
Tüm paketleme gereksinimleri tanımlanmış mı?		
Tüm teslim gereksinimleri tanımlanmış mı?		
İşlemsel bilgisayar donanımı için gereksinimler sağlanmış mı?		
Bilgisayar yazılım kaynakları (işletim sistemi, ağ yazılımı, veritabanları, test yazılımı) tanımlanmış mı?		
Tam bütünleştirme, test ve kabul kriterleri belirlenmiş mi?		
Gereksinimler için test metodları (test, gösterim, analiz veya denetleme) tanımlanmış mı?		
Fonksiyonel gereksinimler özgün numaralandırılmış mı?		
Diğer ana fonksiyonlara veya dış varlıklara olan arayüz gereksinimleri açıkça tanımlanmış mı?		
Her bir yazılım gereksinimi bir veya daha fazla üst düzey sistem gereksinimlerine izlenebilir mi?		
Yazılım Gereksinimleri Gözden Geçirmesinden önce doküman dayanaklandırılmış mı?		

KOD GÖZDEN GEÇİRME

- Kod gözden geçirme ile program test(sınama) işlemlerini birbirinden ayırmak gerekir.
- Program testi, programın işletimi sırasında ortaya çıkabilecek yanlış ya da hataları yakalamak amacıyla yapılır.
- **Kod gözden geçirme işlemi ise, programın kaynak kodu üzerinde yapılan bir incelemedir.** Kod gözden geçirmelerinde program hatalarının %3-5 oranındaki kesimi yakalanabilmektedir.
- **Gözden geçirme, müşterinin gerçek gereksinimlerine uyumluluğu değil, belirteçlere uyumluluğu yoklar;**
- **Kodlamadaki Kusurlar bulunur:**
 - Mantıksal hatalar
 - Kodlardaki sapmalar(örn., başlangıç değeri verilmemiş değişken)
 - Standartlarla uyumsuzluk.



```
if (NotificationClient != null)
{
    NotificationClient = new bl.desktop.NotificationClient() { }
    //NotificationClient.Insert();
}
else
{
    NotificationClient.LastRequest = DateTime.Now;
    NotificationClient.RequestCount = NotificationClient.RequestCount + 1;
    //NotificationClient.Update();
}
if (NotificationClient.Deny == false)
{
    NotificationClient.Request NotificationClient.Request = new bl.desktop.NotificationClient() { }
    NotificationClient.Request.ClientId = NotificationClient.Request.ClientId;
    NotificationClient.Request.UserName = NotificationClient.Request.UserName;
```

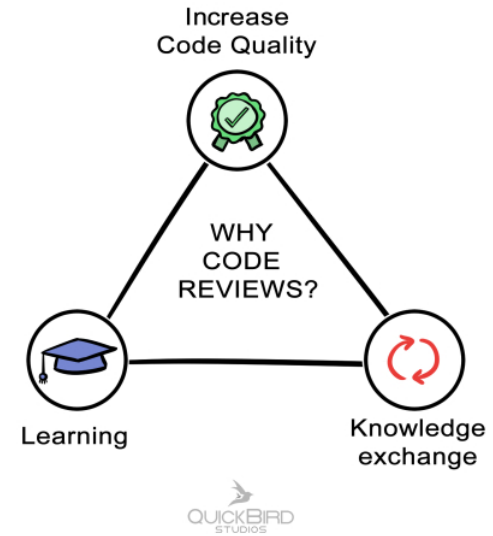

KOD GÖZDEN GEÇİRME

- Kod Gözden geçirme sürecinin temel özellikleri;
 - Hataların bulunması, ancak **düzeltilmemesi** hedeflenir,
 - Olabildiğince küçük bir grup tarafından yapılmalıdır. En iyi durum deneyimli bir inceleyci kullanılmasıdır. Birden fazla kişi gerektiğinde, bu kişilerin, ileride program bakımı yapacak ekipten seçilmesinde yarar vardır.
- Kalite çalışmalarının bir parçası olarak ele alınmalı ve sonuçlar düzenli ve belirlenen bir biçimde saklanmalıdır. biçiminde özetlenebilir. **Burada yanıtı aranan temel soru, programın yazıldığı gibi çalışıp çalışmayacağının belirlenmesidir.**



KOD GÖZDEN GEÇİRME

- **Kod Gözden Geçirme çalışmasının sonucunda:**
 - Programı olduğu gibi kabul etmek,
 - Programı bazı değişikliklerle kabul etmek,
 - Programı, önerilen değişikliklerin yapılmasından sonra tekrar gözden geçirmek üzere geri çevirmek söz konusu olabilir.
- Gözden geçirme çalışması sonucu, önerileri içeren bir rapor biçiminde sunulur. Üst yönetim sonuç ile ilgili olarak bilgilendirilir.



Kod Gözden Geçirme Sırasında Kullanılacak Sorular

Bir program incelenirken, programın her bir **modülü** (yordam ya da işlev) aşağıdaki soruların yanıtları aranır. Bu sorulara ek sorular eklenebilir.

Modül Arayüzü

- Her modül tek bir işlevsel amacı yerine getiriyor mu?
- Modül adı, işlevini açıklayacak biçimde anlamlı olarak verilmiş mi?
- Modül tek giriş ve tek çıkışlı mı?
- Modül eğer bir işlev ise, parametrelerinin değerini değiştiriyor mu?



Kod Gözden Geçirme Sırasında Kullanılacak Sorular

• Giriş Açıklamaları

- Modül, doğru biçimde giriş açıklama satırları içeriyor mu?
- Giriş açıklama satırları, öbeğin amacını açıklıyor mu?
- Giriş açıklama satırları, çıktıları (parametre, kütük vb) ve hata iletilerini tanımlıyor mu?
- Giriş açıklama satırları, öbeğin algoritma tanımını içeriyor mu?
- Giriş açıklama satırları, modülde yapılan değişikliklere ilişkin tanımlamaları içeriyor mu?
- Giriş açıklama satırları, modüldeki olağan dışı durumları tanımlıyor mu?
- Giriş açıklama satırları, modülü yazan kişi ve yazıldığı tarih ile ilgili bilgileri içeriyor mu?
- Her paragrafı açıklayan kısa açıklamalar var mı?



Kod Gözden Geçirme Sırasında Kullanılacak Sorular

Veri Kullanımı

- İşlevsel olarak ilişkili bulunan veri elemanları uygun bir mantıksal veri yapısı içinde gruplanmış mı?
- Değişken adları, işlevlerini yansıtacak biçimde anlamlı mı?
- Değişkenlerin kullanımları arasındaki uzaklık anlamlı mı?
- Her değişken tek bir amaçla mı kullanılıyor?
- Dizin değişkenleri kullanıldıkları dizinin sınırları içerisinde mi tanımlanmış?
- Tanımlanan her gösterge değişkeni için bellek ataması yapılmış mı?



Kaynaklar

- <http://web.firat.edu.tr/mbaykara/8.Hafta -G%C3%B6zdenGecirmeTeknikleri.pdf>
- Yazılım Mühendisliği Temelleri; Dr. M. Erhan Sarıdoğan