

Le bon déroulement de la maintenance nécessite une bonne instrumentation du code, et surtout un très bon jeu de tests, avec un moniteur de tests pour automatiser l'exécution et la vérification des résultats.

Bonus

Question N°4 : Testabilité avec le « design pattern » d'architecture MVC (2 points)

Le nombre de combinaisons d'écrans est le produit des possibilités, soit : $5 \times 3 \times 2 = 30$; il est clair que certaines de ces combinaisons doivent donner lieu à des incohérences, et donc à des messages d'erreurs et des aides en lignes appropriés.

Chacun des écrans comporte en moyenne 10 champs, soit au total 100 champs valeurs à vérifier. Si l'on veut un message d'erreur par champ, cela fait donc 100 messages d'erreurs à prévoir dans le « Controller ».

Question N° 5 : Assurance qualité et effort de tests (2 points)

On peut prévoir un scénario par écran. Ce scénario doit permettre de vérifier :

- Que les valeurs autorisées ne déclenchent aucun message d'erreur.
- Que toute valeur erronée déclenche un message d'erreur. Pour une valeur numérique, il faut donner des valeurs hors des plages de validité ; pour des types énumérés il faut donner des valeurs qui ne correspondent à rien ; pour des libellés alpha-numérique, il faut donner des chaînes de caractères comportant des erreurs de syntaxe.

Dans ce type de travail, le testeur dispose d'un « cahier » décrivant ce qu'il doit saisir, et la réaction qu'il doit observer.

Si l'on compte 10 lignes par donnée, cela fait un document de 1000 lignes, soit 20 pages, qui doivent être elles-mêmes soigneusement validées. Soit au minimum, 1 à 2h de travail ; pour s'assurer des réponses, le testeur doit utiliser les programmes, ou aller discuter avec les programmeurs pour connaître les réponses du système.

Pour les combinaisons d'écrans, il faut au minimum 30 scénarios. Certain de ces scénarios devront vérifier que les actions effectuées sont cohérentes entre elles, donc s'assurer d'une nouvelle catégorie de messages d'erreurs (NB : la combinatoire des incohérences n'est certainement pas linéaire ; la détection des erreurs associées est beaucoup plus délicate, et requiert un plus grand volume de programmation). Pour rendre les écrans le plus indépendant possible les uns des autres il faut faire une analyse très fine des données associées à ces écrans, ce qui est très difficile à faire la première fois ; d'où l'importance du prototype qu'il faut clairement séparer de la maquette. Rappelons que le rôle d'un prototype est d'abord de valider l'architecture.

Ces scénarios, qui s'appuient sur les précédents, sont évidemment beaucoup plus difficiles à élaborer, et donc beaucoup plus coûteux à spécifier. Ils ne peuvent être faits que par des personnes connaissant bien la partie métier du projet PTC, et l'architecture de l'application. C'est la raison pour laquelle l'intégration est faite (dans le modèle en V), en regard de la conception.

198

```

70 END LOOP: -- ZONLU
71 END LICARTE:
72
73 BEGIN -- pgm ppal
74 LICARTE:
75
76
77 END quatre_couleurs:
78
jan 29 2003 15:22      sdcokl.ada.8
Page 2

```

```

1 Module pour colorier 1 carte plane en utilisant 4 couleurs.
2
3
4
5
6
7
8
9
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
```


messages au module TDE de télémaintenance pour intervention.

L'état des équipements surveillé est visualisé sur un « mur » d'écrans des différents postes de travail.

- TDE : 50 KLS - le temps maximum d'intervention sur le matériel, en cas d'incident, est de une heure. Ce module reçoit des messages du module IHMS et permet aux opérateurs de maintenance d'envoyer des commandes vers les équipements et les capteurs. Ce module est simple dans sa conception, mais chaque équipement surveillé est un cas particulier qui nécessite des valeurs de données très précises.
- TSD : 90 KLS - ce module reçoit les données synthétisées au même rythme que IHMS et stocke ces données dans la base. Les traitements statistiques sont effectués à la demande des utilisateurs sur différents pas de temps : horaire, journée, semaine, mois, année. Il n'y a donc pas de temps réel ; c'est un système d'information, avec un jeu de requêtes paramétrables en fonction des configurations des équipements et des capteurs. Les traitements statistiques peuvent être assez complexes mais les algorithmes sont connus. Dans une étape ultérieure, il est prévu de permettre aux usagers du système d'enrichir les bibliothèques standards d'algorithmes de traitements, par des bibliothèques qui leur sont propres.

Ce qu'il faut savoir sur l'organisation du projet :

- La société Starindus développe les modules TSD, MSD et IHMS. Les modules MA et TDE sont sous-traités à une SSII, la société Fabulog, spécialisée dans les matériels utilisés et le temps réel.
- Le responsable du projet chez Starindus aurait bien voulu avoir la responsabilité du développement de l'ensemble ou, au moins, celui du module TDE ; en conséquence la documentation de cette API n'est pas de grande qualité et manque de rigueur. Pour cette raison, il donne le minimum d'information sur l'API du module de synthèse (MSD) à la société Fabulog. La communication n'est donc pas la meilleure entre les deux sociétés.

Les équipes :

- TSD : équipe très compétente
- MSD : à l'origine elle est prévue de niveau nominal
- IHMS : équipe compétente dans le domaine métier considéré mais la technologie est nouvelle pour cette équipe (IHM en Java avec JBuilder). L'équipe vient d'être formée à Java mais n'a pas eu de formation JBuilder.
- MA : équipe compétente, sans être extraordinaire. Connaît bien les familles de capteurs utilisés
- TDE : équipe de niveau nominal. Fabulog considère que ce module est simple et il n'y a pas d'innovation technologique. La société a déjà développé des modules semblables.

Question 1 (4 points)

A partir des informations fournies, utiliser le modèle intermédiaire d'estimation COCOMO 81 pour chacun des modules et déterminer :

- Le type de projet pour chacun des modules
- L'effort en h.m pour chacun des modules. On ajoutera 8% d'effort pour les modules qui doivent spécifier leur interface et on comptera 4% pour chacun des modules MA et TDE qui se partagent une interface commune.
- Le temps de développement, en mois, pour chacun des modules.
- En déduire le nombre de personnes, en moyenne sur chacun des modules.

Remarque : en annexe figure le tableau des facteurs de coût du modèle intermédiaire de COCOMO 81

Réponse à la question 1

QUESTION 1								
Module	Type	Nombre de lignes (en KLS)	Effort nominal (en H.M)	Plus Effort interfaces	Facteurs de coûts appliqués	Effort résultant (en H.M)	TDEV (en mois)	Nbre de personnes (FSP)
MA	E	30	165,85	172,48	RELY = 1,15 ACAP=0,86 AEXP=0,82 VEXP=0,9	125,89	11,75	11
MSD	P	120	639,45	690,60	Neant	690,60	24,64	28
IHMS	S	35	133,79	144,49	AEXP=0,91 LEXP=1,07 MODP=0,91	128,03	15,80	8
TDE	S	50	194,57	233,48	Neant	233,48	19,86	12
TSD	P	90	463,31	500,38	ACAP=0,71 AEXP=0,82 PCAP=0,7 VEXP=0,9 LEXP=0,95	174,35	15,22	11
Total		325	1596,96	1741,43		1352,36		

Explications**- Module MA**

Le module MA est naturellement de type E car il est critique et fait de l'acquisition en temps réel.

Son niveau de fiabilité doit être élevé (RELY=1,15). Comme l'équipe est compétente et connaît bien les familles de capteurs utilisés, on considère que ACAP est à un niveau élevé, AEXP à un niveau très élevé et VEXP à un niveau élevé.

L'effort nominal est calculé avec la formule $2,8(30)^{1,2}$. On ajoute 4% pour l'API vers le réseau de capteurs. TDEV est calculé avec la formule $2,5(\text{Effort})^{0,32}$.

- Module MSD

Le module MSD est considéré de type P car le code est nominal mais il reçoit les données acquises en temps réel. L'API n'est pas très complexe. L'équipe étant prévue de type nominale, aucun facteur de coût n'est à appliquer.

L'effort nominal est calculé avec la formule $3(120)^{1,12}$. On ajoute 8% pour l'interface dont il a la charge. TDEV est calculé avec la formule $2,5(\text{Effort})^{0,35}$.

- Module IHMS

Le module IHMS est de type S car on dit qu'il est assez simple et l'interface ne semble pas sophistiquée. L'équipe est compétente dans le domaine métier (AEXP=0,91). Par contre, le niveau en Java est bas (l'équipe a juste suivi une formation), donc LEXP=1,07. Bien que l'équipe n'est pas eu de formation à l'environnement Websphère, on considère que cet outil va, au total, faciliter le travail (MODP=0,91), mais ceci peut être contesté...

L'effort nominal est calculé avec la formule $3,2(35)^{1,05}$. On ajoute 8% pour l'API dont le module a la charge. TDEV est calculé avec la formule $2,5(\text{Effort})^{0,38}$.

- Module TDE

Le module TDE est de type simple, il faut simplement être minutieux car certaines données doivent être très précises. L'équipe étant de niveau nominal aucun facteur de coût n'est appliqué.

L'effort nominal est calculé avec la formule $3,2(35)^{1,05}$. On ajoute 20% pour les interfaces (2 interfaces à 8% plus 4% pour celle qui est partagée avec le module MA). TDEV est calculé avec la formule $2,5(\text{Effort})^{0,38}$.

- Module TSD

Le module TSD est de type P car les traitements statistiques peuvent être complexes même si les algorithmes sont connus. De plus le jeu de requêtes doit être paramétrable. Sa conception

n'est pas toute simple car il y a également une base de données et, à terme, les utilisateurs doivent pouvoir enrichir les bibliothèques standards de traitements.

L'effort nominal est calculé avec la formule $3(90)^{1,12}$. On ajoute 8% pour l'API. TDEV est calculé avec la formule $2,5(\text{Effort})^{0,35}$.

L'équipe étant très compétente, tous les facteurs de coût concernant le personnel ont été pris au plus haut.

Question 2 (3 points)

En vous aidant du tableau de ventilation des efforts et des délais (TABLE 6-8), par phases, calculer, en présentant les résultats sous forme de tableaux, l'effort et le délai par phase pour chacun des modules. On considérera la phase de programmation comme un tout (on ne détaillera pas entre la conception détaillée et le codage et tests unitaires). Il faut prendre en compte la phase "plans and requirements".

Appliquer la formule de pondération pour donner le coût total du projet, tous modules confondus.

Réponse à la question 2

QUESTION 2									
Module	Type	Plans and requirements		Conception générale		Réalisation		tests et intégration	
		Effort	Délai	Effort	Délai	Effort	Délai	Effort	Délai
MA	E	10,07	3,76	22,66	3,99	67,98	4,70	35,25	3,05
MSD	P	48,34	5,42	117,40	6,65	379,83	10,84	193,37	7,15
IHMS	S	7,68	1,90	20,48	3,00	79,38	8,69	28,17	4,11
TDE	S	14,01	2,38	37,36	3,77	144,76	10,92	65,37	5,16
TSD	P	12,20	3,35	29,64	4,11	95,89	6,70	48,82	4,41
Effort moyen pondéré									
Calcul des coefficients p, q et r :									
p	q	r							
0,26	0,65	0,09							
Calcul des efforts (EFFs, EFFp, EFFe) :									
EFFs =	1368,76								
EFFp =	1951,78								
EFFe =	2893,46								
Effort moyen pondéré =	1891,45								

On constate que l'effort moyen pondéré est bien supérieur à la somme des efforts (1352,36)

Explications

C'est le tableau suivant qui a été appliqué en respectant le type de chaque module et en considérant (par rapport au tableau) que les modules MA, IHMS et TDE étaient de 32 KLS et les modules MSD et TSD de 128 KLS.

CORRIGE DE L'EXAMEN DE GENIE LOGICIEL B5 SESSION I

Effort distribution		Medium	Large	Very Large
Mode	Phase	32 KISL	128 KISL	512 KISL
Organic (S)	Plans and requirements (%)	6	6	
	Product design	16	16	
	Programming	62	59	
	Detailed design	24	23	
	Code and unit test	38	36	
	Integration and test	22	25	
Semidetached (P)	Plans and requirements (%)	7	7	7
	Product design	17	17	17
	Programming	58	55	52
	Detailed design	25	24	23
	Code and unit test	33	31	29
	Integration and test	25	28	31
Embedded (E)	Plans and requirements (%)	8	8	8
	Product design	18	18	18
	Programming	54	51	48
	Detailed design	26	25	24
	Code and unit test	28	26	24
	Integration and test	28	31	34
Schedule distribution		32 KISL	128 KISL	512 KISL
Organic (S)	Plans and requirements (%)	12	13	
	Product design	19	19	
	Programming	55	51	
	Integration and test	26	30	
Semidetached (P)	Plans and requirements (%)	20	22	24
	Product design	26	27	28
	Programming	48	44	40
	Integration and test	26	29	32
Embedded (E)	Plans and requirements (%)	32	36	40
	Product design	34	36	38
	Programming	40	36	32
	Integration and test	26	28	30

En vertu de ce tableau, les pourcentages appliqués sont les suivants :

CORRIGE DE L'EXAMEN DE GENIE LOGICIEL B5 SESSION I

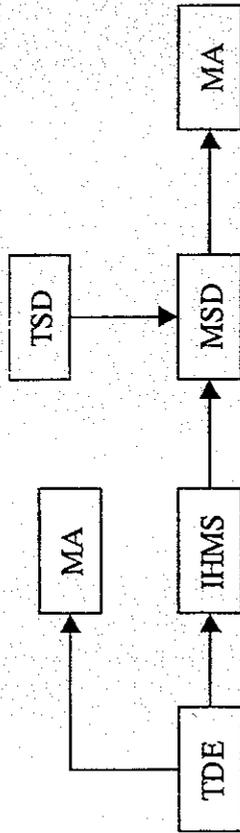
Module	Type	Plans and requirements		Conception générale		Réalisation		tests et intégration	
		Effort	Délai	Effort	Délai	Effort	Délai	Effort	Délai
MA	E	8%	32%	18%	34%	54%	40%	28%	26%
MSD	P	7%	22%	17%	27%	55%	44%	28%	29%
IHMS	S	6%	12%	16%	19%	62%	55%	22%	26%
TDE	S	6%	12%	16%	19%	62%	55%	22%	26%
TSD	P	7%	22%	17%	27%	55%	44%	28%	29%

Question 3 (3 points)

En fonction des dépendances entre les modules et en considérant une fourchette d'erreur des estimations de $\pm 10\%$, tracer le PERT du projet en indiquant date de début au plus tôt, au plus tard et date de fin au plus tôt, au plus tard pour chacune des tâches de chacun des modules. On considérera les tâches : conception globale (product design), programmation (programming) et "integration et tests" (on ne fera pas figurer "plans and requirements". La date de début de projet est appelée T0. On donnera toutes les dates par rapport à T0.

Réponse

Les dépendances entre les modules sont les suivantes :

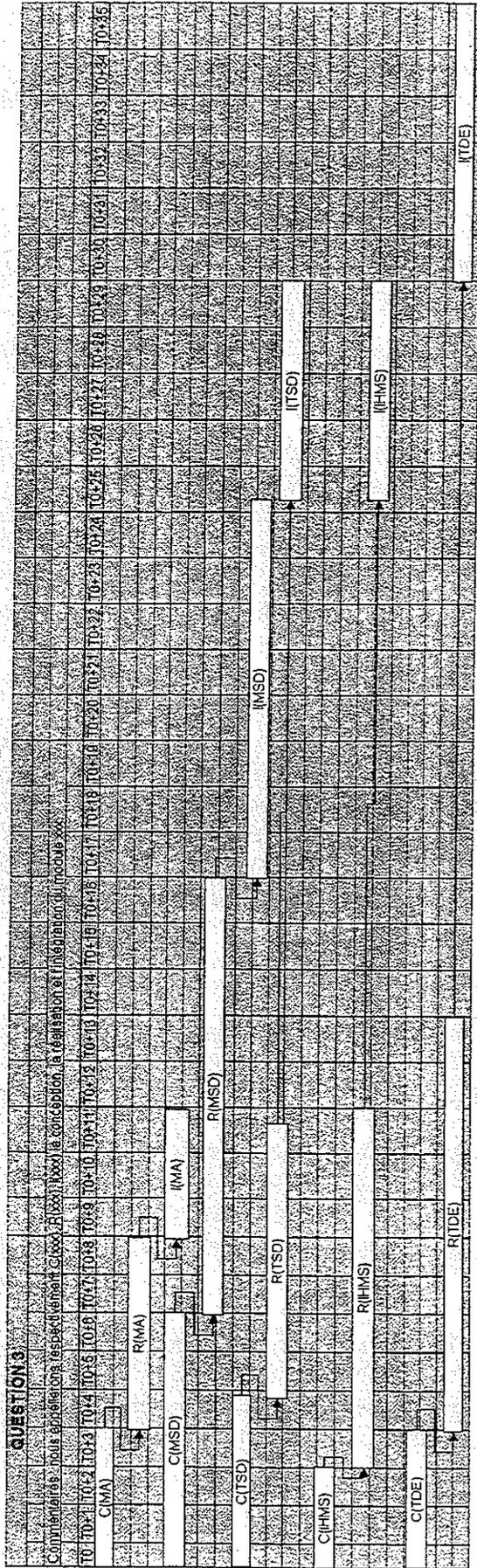


Il y a 2 fois MA car TDE dépend directement et indirectement de ce module.

Ces dépendances sont importantes pour l'ordre d'intégration car si les API sont bien définies, la réalisation des différents modules peut être parallélisée.

Le PERT (page suivante) montre que c'est le module MSD qui est sur le chemin critique car 3 modules dépendent de celui-ci pour l'intégration (TSD, TDE et IHMS).

CORRIGE DE L'EXAMEN DE GENIE LOGICIEL B5 SESSION I



Remarques

On voit que même si le dialogue entre Starindus et Fabulog ne se passe pas très bien pour l'API du module MSD, il y a de la marge pour le développement du module MA (environ 5 mois). Ce délais comble aisément le handicap, estimé à 10% d'effort supplémentaire, concernant le manque d'information sur cette interface.

Afin de réduire la durée du projet (environ 35 mois), il y a intérêt à simuler ce qui arrive du module IHMS en phase d'intégration car ceci pourrait ramener la durée du projet à 30 mois.

Dans l'absolu, les phases de conceptions des 5 modules ne vont pas démarrer en même temps. C'est surtout la conception de MSD qui est cruciale et qui doit débiter en premier. Les modules TSD, IHMS et TDE n'ont pas de problème de délai en phase de conception et de réalisation.

206

Question 4 (2 points)

Finalement, pour le module MSD, le responsable du projet ne dispose pas d'une équipe de niveau nominal. En fait, l'équipe de programmeurs est de niveau très bas, mais le responsable ne s'en aperçoit pas tout de suite. Les difficultés deviennent manifestes qu'en milieu de la phase de programmation. Quelles en sont les conséquences sur le projet global ?

Réponse

Le fait que les programmeurs soient de niveau très bas se traduit par un coefficient LEXP de 1,42 au lieu de 1 (soit 42% de coût supplémentaire pour la phase de réalisation !).

On va considérer que ce facteur n'agit que sur la phase de réalisation qui nécessitait un effort de 380 h.m et un délai de 10,84 mois.

L'effort devient : $380 \text{ h.m} * 1,42 = 540 \text{ h.m}$. L'effort total passe donc de 690 h.m à $690 + (540 - 380) = 850 \text{ h.m}$. TDEV devient donc $2,5 (850)^{0,35}$, soit 26,5 mois. Le pourcentage de réalisation étant de 44% en délai, le nouveau délai de réalisation est donc de $26,5 * 0,44 = 11,66$ mois. Le délai de réalisation est donc allongé d'environ 1 mois.

Comme le module MSD est sur le chemin critique, le projet subira donc un retard d'environ 1 mois.

Question 5 (bonus de 2 points)

Que feriez-vous, à la place de la société Fabulog, pour essayer de régler le problème de communication avec le responsable du projet chez Starindus. Le responsable du projet chez Fabulog doit sans cesse aller "à la pêche aux informations" concernant la fameuse interface du module MSD dont Starindus est responsable. On peut estimer à environ 10% le retard pris, en phase de conception globale, pour le module MA. De plus, l'équipe en charge du module MSD étant maintenant de niveau très bas, ce problème rejaillit également sur la phase de codage du module MA.

Réponse

Afin de réduire le risque de retard pour le module MA, il est nécessaire d'encapsuler au maximum les fonctions qui interfacent avec l'API du module MSD. Ceci nécessite un travail supplémentaire de conception et de réalisation qui a été estimé à environ 10% dans l'énoncé.

Nous avons vu cependant que ceci n'est pas très grave car le PERT montre que le développement du module MA n'est pas sur le chemin critique et qu'il y a même une marge d'environ 5 mois (en fait il y a même plus puisque le délai de réalisation du module MSD sera allongé d'un mois par rapport aux prévisions compte tenu du niveau très bas de l'équipe de réalisation de celui-ci !).

Le responsable du projet chez Fabulog peut aussi faire valoir qu'un trop grand dérapage du développement du module MA (en l'absence de spécifications claires de l'API du module MSD) pourrait compromettre la date d'intégration du module MSD, ce qui n'est pas de l'intérêt du responsable de projet chez Starindus...

Cours Génie logiciel B5 – Année universitaire 2004-2005

1^{re} session

I - Etude de cas (12 points+bonus 6 points)

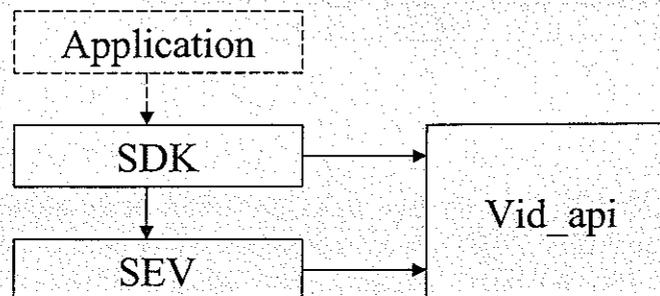
La Société VZ S.A. a mis en place un système de visioconférence sophistiqué permettant la commande à distance de plusieurs caméras et le traitement des informations transmises par celles-ci.

Ce système est constitué des composants suivants :

- Une sur-couche du système d'exploitation hôte, appelée SEV (Système d'Exploitation Vidéo), qui est le cœur du système ;
- Un SDK (Software Development Kit) qui permet à toute application qui le désire d'utiliser certaines fonctionnalités du SEV ;
- Une librairie d'APIs (Application Programming Interfaces), appelée Vid_api, utilisée par le SDK et le SEV.

Toute application qui sera bâtie au dessus de ce système de visioconférence interagira avec celui-ci par le biais uniquement du composant SDK.

On peut ainsi représenter ce système selon le schéma suivant, les flèches indiquant une relation d'utilisation entre composants :



Chacun des composants du système comporte plusieurs modules. Le tableau ci-après en donne la liste et, pour chacun, le nombre de lignes de code ainsi que la complexité (S, P, E).

Il faut noter que les modules de Vid_api sont **totalelement indépendants** des uns des autres et de ce fait n'interagissent pas entre eux.

208

Nom	#Lignes	Complexité
Vid_api		
Vid_utils	5000	P
Vid_codec	8000	E
Vid_event	1000	S
Vid_io	17000	P
Vid_snmp	1000	S
Vid_stream	27000	E
Vid_ar	44000	E
SEV		
Vid_Serv	61000	E
Vid_Update	1600	P
Vid_sensor	2000	P
Vid_detector	200	S
Vid_mib	4200	S
Driver modèle-1	47000	P
Driver modèle-2	23000	P
SDK		
Audio+video AccessManager	19000	P
Vid_DController	1300	S
Vid_EController	1000	S
Vid_Monitor	24000	E
Vid_Remote	1200	S
Vid_SAController	4800	P
Vid_SMController	1500	P
Vid_VM	1500	P
XVid_Display	26000	E

Question 1 (4 points)

En utilisant le modèle COCOMO 81 de base, calculer l'effort de développement pour chacun des trois composants Vid_api, SEV, SDK, en les considérant ici comme indépendants des uns des autres. La formule de l'effort pondéré s'applique-t-elle pour le calcul de l'effort de Vid_api ?

Les modules de Vid_api étant totalement indépendants les uns des autres, on peut considérer que le développement de chacun de ses modules se fait de manière totalement indépendante. De ce fait l'effort de développement de la librairie est égal à la somme de l'effort de développement de chacun de ses modules. Le calcul de l'effort de SFK et de SEV se fait en revanche au moyen de la formule de l'effort pondéré car les modules de chacun de ces composants interagissent entre eux.

Note : le calcul de l'effort module par module pour SEV et SDK n'est pas nécessaire (sauf pour Vid_monitor, utilisé à la question 7). C'est pour cela que les lignes correspondantes sont grisées. Enfin, rappelons que la valeur de l'effort pondéré sera toujours supérieur à celle de la somme des efforts des modules pris un à un, constat que l'on peut faire ici.

209

Nom	#Lignes	Complexité	#Lignes S	#Lignes P	#Lignes E	Effort hxmois
Question1						
Librairie Vid_api						
Vid_utils	5 000	P	0	5 000	0	18,2
Vid_codec	8 000	E	0	0	8 000	43,7
Vid_event	1 000	S	1 000	0	0	2,4
Vid_io	17 000	P	0	17 000	0	71,7
Vid_snmp	1 000	S	1 000	0	0	2,4
Vid_stream	27 000	E	0	0	27 000	187,9
Vid_ar	44 000	E	0	0	44 000	337,6
Effort modules séparemment						663,8
Sous-total LS	103 000		2 000	22 000	79 000	
SEV						
Vid_Serv	61 000	E			61 000	499,7
Vid_Update	1 600	P		1 600		5,1
Vid_sensor	2 000	P		2 000		6,5
Vid_detector	200	S	200			0,4
Vid_mib	4 200	S	4 200			10,8
Driver modèle-1	47 000	P		47 000		223,8
Driver modèle-2	23 000	P		23 000		100,5
Effort modules séparemment						846,9
Sous-total LS	139 000		4 400	73 600	61 000	
Pourcentage			3	53	44	
Effort pondéré			14	399	589	1001,9
SDK						
Audio+video AccessManager	19 000	P		19 000		81,2
Vid_DController	1 300	S	1 300			3,2
Vid_EController	1 000	S	1 000			2,4
Vid_Monitor	24 000	E			24 000	163,1
Vid_Remote	1 200	S	1 200			2,9
Vid_SAController	4 800	P		4 800		17,4
Vid_SMController	1 500	P		1 500		4,7
Vid_VM	1 500	P		1 500		4,7
XVid_Display	26 000	E			26 000	179,6
XVid_Time	5 000	P		5 000		18,2
Effort modules séparemment						477,4
Sous-total LS	80 300		3 500	26 800	50 000	
Pourcentage			4	33	62	
Effort pondéré			10	136	433	579,3
TOTAL Question1	322 300					2245,0

Question 2 (3 points)

Tous les modules de Vid_api sont utilisés et fortement sollicités par SEV et SDK. De ce fait il a paru suffisant aux équipes de développement de n'effectuer les tests d'intégration de Vid_api

depuis soit SEV, soit SDK. Selon vous quelle sera la stratégie d'intégration de Vid_api la moins coûteuse ? Justifiez votre réponse de manière qualitative.

La stratégie d'intégration la moins coûteuse sera avec le composant de taille la plus petite, c'est à dire SDK. En effet l'effort d'intégration est égal à la différence de l'effort de développement des 2 composants à intégrer et des efforts de développement de chacun des 2 composants pris séparément, calculés à la question 1. Donc bien entendu l'intégration de Vid_api avec le composant dont l'effort de développement est le plus petit. D'autre part les hypothèses de l'énoncé s'avèrent justes : en effet plus généralement l'utilisation d'API dans tout développement d'application ne demande pas en principe d'effectuer des tests d'intégration de celles-ci avec l'application que l'on développe.

Déterminer ensuite le coût d'intégration de SEV avec Vid_api et celui de SDK avec Vid_api. Pour cela on utilisera la formule de l'effort pondéré sur les composants à intégrer 2 à 2 à laquelle on déduira l'effort unitaire de développement de chacun des composants, calculé à la question 1. Comparer les résultats obtenus. Cela confirme-t-il votre réponse qualitative ?

L'effort d'intégration (constitué en fait de la somme des coûts de conception générale et de l'intégration proprement dite) le moins élevé est effectivement celui de SDK avec Vid_api (voir ci-dessous). Cela confirme bien notre réponse.

Nom	#Lignes	Complexité	#Lignes S	#Lignes P	#Lignes E	Effort hxmois
Intégration SEV Vid_api						
TOTAL des lignes de code	242 000		6 400	95 600	140 000	
POURCENTAGE			3	40	58	
EFFORT pondéré			20	554	1 511	2085,1
EFFORT d'intégration						419,4
Intégration SDK Vid_api						
TOTAL des lignes de code	183 300		5 500	48 800	129 000	
POURCENTAGE			3	27	70	
EFFORT pondéré			17	274	1 317	1607,5
EFFORT d'intégration						364,4

En réalité, pour être plus fin, le coût d'intégration de tout composant avec Vid_api est égal à la somme des coûts d'intégration de ce composant avec chacun des modules de Vid_api. Il se calcule en appliquant la formule de l'effort pondéré à chacun des modules de Vid_api avec le composant à intégrer, puis en faisant la somme des efforts d'intégration. La valeur obtenue sera alors de 307 hommes.mois au lieu des 364 hommes.mois obtenus.

Nom	#Lignes	Complexité	#Lignes S	#Lignes P	#Lignes E	Effort hxmois
<i>Intégration SDK Vid_utils</i>						
TOTAL des lignes de code	85 300		3 500	31 800	50 000	
POURCENTAGE			4	37	59	
EFFORT total			10	163	438	611,1
EFFORT d'intégration						13,7
<i>Intégration SDK Vid_codec</i>						
TOTAL des lignes de code	88 300		3 500	26 800	58 000	
POURCENTAGE			4	30	66	
EFFORT total			11	138	512	659,7
EFFORT d'intégration						36,8
<i>Intégration SDK Vid_event</i>						
TOTAL des lignes de code	81 300		4 500	26 800	50 000	
POURCENTAGE			6	33	62	
EFFORT total			13	136	434	583,5
EFFORT d'intégration						1,9
<i>Intégration SDK Vid_lo</i>						
TOTAL des lignes de code	97 300		3 500	43 800	50 000	
POURCENTAGE			4	45	51	
EFFORT total			11	228	450	687,8
EFFORT d'intégration						36,9
<i>Intégration SDK Vid_snmp</i>						
TOTAL des lignes de code	81 300		4 500	26 800	50 000	
POURCENTAGE			6	33	62	
EFFORT total			13	136	434	583,5
EFFORT d'intégration						1,9
<i>Intégration SDK Vid_stream</i>						
TOTAL des lignes de code	107 300		3 500	26 800	77 000	
POURCENTAGE			3	25	72	
EFFORT total			11	141	706	857,7
EFFORT d'intégration						90,5
<i>Intégration SDK Vid_ar</i>						
TOTAL des lignes de code	124 300		3 500	26 800	94 000	
POURCENTAGE			3	22	76	
EFFORT total			11	143	828	1041,9
EFFORT d'intégration						125,0
EFFORT d'intégration affiné						306,6

Question 3 (2 points)

Déterminer le coût d'intégration SDK-SEV et en déduire l'effort de développement total du système. Faire l'hypothèse que l'ensemble du système peut être considéré comme une application de type P est-il une approximation légitime ? Justifiez votre réponse.

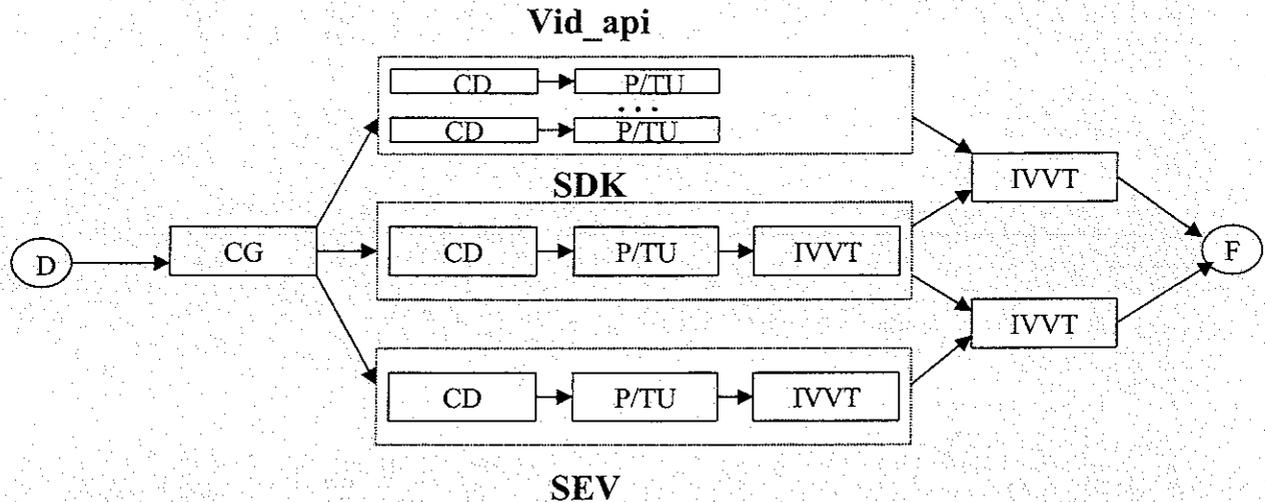
L'effort de développement total du système est égal à la somme de l'effort de développement de chacun des composants pris un à un, soit 2245 hommes.mois, auquel s'ajoute l'effort d'intégration SDK-SEV et celui de SDK avec Vid_api. Le fait de considérer que l'ensemble du système comme de type P n'est pas vraiment légitime, la valeur obtenue se situant médiane entre les résultats P et E.

Intégration SDK SEV

TOTAL des lignes de code	219 300	7 900	100 400	111 000	
POURCENTAGE		4	46	51	
EFFORT pondéré		25	575	1 174	1774,4
EFFORT d'intégration					193,3
Effort total					2802,7
Hypothèse P	322 300		322 300		1933,6
Hypothèse E				322 300	3683,1

Question 4 (3 points)

Quelle organisation de projet mettriez-vous en place ? Pour cela construire un diagramme de PERT grossier sans y faire figurer les dates, en y indiquant les tâches clés de conception, de développement et d'intégration et leur liens de dépendances. Affiner notamment l'organisation de développement de Vid_api, sachant que l'équipe impliquée dans cette réalisation possède une forte expérience et compétence du domaine vidéo et qu'elle a décidé de réaliser cette API en utilisant un processus de développement de type agile, c'est à dire en développant les modules un par un, indépendamment des autres.



Question 5 (bonus 2 points)

On évalue le coût d'apprentissage de chaque module de Vid_api à une demi journée homme par personne impliquée dans le projet. Déterminer le coût d'apprentissage total par les développeurs impliqués respectivement dans la réalisation de SEV et de SDK.

Sur la base d'un projet de type P (cf. question 3), la durée de développement est, selon le calcul $T = 2,5.Eff^{0,35}$ (formule de calcul de la durée pour un projet de type P, appliquée à l'effort réel de 2802,7 hommes.mois), de 40 mois. On en déduit le nombre de moyen de développeurs impliqués, en divisant l'effort par la durée, égal à 71. L'effort d'apprentissage total est donc de 248 jours.hommes, soit 12,5 hommes.mois sur la base d'un homme.mois = 20 jours.hommes, c'est à dire 0,44% de l'effort total.

Note : en toute rigueur, il ne faudrait pas compter les développeurs qui travaillent sur Vid_api. En l'absence de données précises sur ces modules, il est difficile de dire quel développeur connaît quels modules (au pire, chacun ne connaît que celui sur lequel il travaille). Les 248 jours.hommes ci-dessus sont donc un majorant de l'effort d'apprentissage.

Question 6 (bonus 2 points)

L'équipe impliquée (architectes, développeurs) dans le développement de ce système possède des fortes expérience et compétence du domaine vidéo. Elle maîtrise parfaitement la structure et les APIs du langage C++ dans lequel est codé ce système.

Nous allons considérer les facteurs de coût ACAP, LEXP, PCAP, AEXP du modèle COCOMO 81 intermédiaire. A l'aide de ce modèle, déterminer le coefficient de réduction de l'effort de développement du système. Pour cela, on évaluera les valeurs de ces facteurs, en considérant que la valeur des autres est égale à 1. Quel effort obtenez-vous alors, en repartant de l'effort total déterminé dans la question 3 et en lui appliquant l'équation de l'effort du modèle intermédiaire ?

On prend les valeurs les plus basses de ces facteurs, et il suffit de faire le produit de la valeur de leur produit avec l'effort obtenu, 2842 hommes.mois. On obtient alors une valeur ajustée de l'effort égale à 1101 hommes.mois.

Que se passerait-il maintenant en considérant la situation inverse, si l'équipe ne possède pas du tout l'expérience et la compétence requises ?

La valeur obtenue de l'effort sera alors de 8667 hommes.mois. D'où un rapport de 1 à 8 entre les deux valeurs extrêmes de l'effort !

Question 7 (bonus 2 points)

Nous allons nous intéresser au module Vid_monitor de SDK. Ce module si l'on le regarde de plus près, peut être vu comme une implémentation du pattern MVC, selon par exemple le framework J2EE STRUTS. Selon cette hypothèse, il sera constitué d'un automate complexe (20% du code de type E) gérant événements en entrée et sortie provenant de l'extérieur, appelant un grand nombre d'actions élémentaires (80% du code de type S). Calculer l'effort de développement de ce module, compte tenu de ces hypothèses. Quelle conclusion faites-vous relativement à la conception/réalisation de ce module ?

Vid_Monitor	24 000	S/E	19 200	0	4 800	
Pourcentage			80	0	20	
Effort pondéré			54	0	33	86,6

La valeur de l'effort obtenue est de 86,6 hommes.mois au lieu de 163,1 hommes.mois, soit un rapport 2. Cela signifie que le choix d'une architecture adaptée est fondamental pour la maîtrise des coûts d'un projet logiciel.

CORRIGÉ DE L' EXERCICE DE TEST

Soit à tester, par la méthode des couvertures, la procédure **TROUVER_SYMB** ci-après. Cette procédure (lignes 60 à 114) représente la *brique de base* (très simplifiée) d'un analyseur lexical pour un programme écrit en un certain langage (style Pascal). Elle permet d'obtenir le *symbole lexical suivant* dans le flot d'entrée enregistré dans un fichier de texte (le fichier prédéfini `Standard_Input` dans le module proposé).

Les spécifications sont précisées dans les commentaires (*lignes 15 à 23*).

I. PARTIE SYNTAXIQUE

- 3 points** Q 1.1 -Construire le graphe de commande de la procédure **TROUVER_SYMB**.
Cf. document annexe.
- 1 point** Q 1.2 -Calculer son nombre cyclomatique, en précisant la formule utilisée.

Mesure de complexité textuelle du programme:

$M = 1 +$ nombre de décisions binaires simples de la procédure.

Application au problème :

les décisions figurent aux lignes : 74, 78, 83 (x 2 : AND), 87, 93, 96 (x 3 : OR), 98, 99

soient 11 décisions binaires simples. D'où : $M = 1 + 11 = 12$

Mesure de complexité structurelle du graphe de commande :

$M =$ nombre d'ARCS - nombre de NŒUDS + 2 . p

Application au problème :

Sur la présente version du graphe de commande (de très nombreuses variantes sont possibles), on dénombre 24 arcs et 14 nœuds ;

par ailleurs, p = nombre de composantes connexes du graphe = 1 (pour toute procédure)

D'où : $M = 24 - 14 + 2 \times 1 = 12$

Les 2 méthodes de calcul doivent bien sûr fournir le même résultat...

Quel sera le nombre maximum de chemins nécessaires pour effectuer la couverture totale des arcs du graphe ?

Ce nombre maximum est égal au nombre cyclomatique, soit ici 12.

II. PARTIE SEMANTIQUE

- 4 points** Q 2 Définir un ensemble de chemins permettant une couverture exhaustive des arcs (ou liens) du graphe de commande. Préciser avec soin la sémantique de chaque chemin choisi en fonction des données (caractères) du fichier d'entrée.
Cf. document annexe pour la couverture.

Sémantique des chemins :

Soit f le fichier d'entrée ; $f = \{ c_1, c_2, c_3, \dots \}$ où c_i représente 1 caractère.

1. C1 = lettre c2 = caractère autre qu'une lettre ou un chiffre
2. C1 = lettre c2 = chiffre c3 = caractère autre qu'une lettre ou un chiffre
3. C1 = lettre c2 = lettre c3 = caractère autre qu'une lettre ou un chiffre
4. F = suite de 33 lettres, suivies d'un caractère autre qu'une lettre ou un chiffre
5. C1 = chiffre c2 = caractère autre qu'un chiffre

6. C1 = chiffre c2 = chiffre c3 = caractère autre qu'un chiffre
7. C1 = '<' c2 = '='
8. C1 = '<' c2 <> '='
9. C1 = ':' c2 = '='
10. C1 = '>' c2 <> '='
11. C1 = ni lettre, ni chiffre, ni '<', ni ':', ni '>'

Que concluez-vous du nombre de chemins par rapport au nombre cyclomatique?

Il y a 1 chemin de moins que le nombre cyclomatique, car il existe une corrélation entre les prédicats 96 (2) et 99, qui testent la même condition.

Comment procéderiez vous si l'on vous demandait maintenant une couverture totale des nœuds (ou sommets) du graphe?

Il ne reste plus rien à faire, la couverture des arcs impliquant celle des nœuds (laquelle correspond à l'exécution au moins une fois de chaque instruction).

Cours Génie logiciel GLG105 – Année universitaire 2005-2006

1^{re} session

I – Corrigé de l'étude de cas (12 points)

La Société de Génie Civil Rocard', spécialisée dans la conception d'ouvrages d'art, principalement dans le domaine routier, décide de faire un Intranet assez sophistiqué qui va inclure, outre ses référentiels et son savoir faire, des outils de visioconférence (elle a des filiales à l'étranger), workflow, gestion de projet, GED (gestion de documents électroniques) etc. Pour plusieurs de ces outils, il existe des COTS (logiciels sur étagère) mais les spécificités de son métier et le fait que certains documents sont classés « Confidentiel Industrie » l'oblige à des développements spécifiques pour sécuriser certains aspects de cet Intranet. On peut estimer que ce projet contient une assez forte combinatoire lorsque l'on prend l'ensemble des services à fournir.

Le projet est, globalement, de type P et l'expression de besoin et des exigences a coûté 20 h.m.

Question 1 (1 point)

A partir de la valeur de l'effort de la phase expression de besoin et au moyen des tables du modèle COCOMO de répartition de l'effort par phase, calculer le nombre de lignes (en milliers de lignes sources) estimées.

La valeur de l'effort pour la phase EB/EC (phase 1 du modèle COCOMO 81) est égale à 7% de celle de l'effort de la phase de développement (phase 2), ce dernier permettant de déterminer le nombre de lignes de code :

$$\text{Effort}_{\text{développement}} = a \times (KISL)^b$$

Pour un programme de type P, $a = 3$, $b = 1,12$. On obtient alors :

$$KISL = (20 / (0,07 * 3))^{1/1,12} = 58,5 \text{ milliers lignes de code}$$

Question 2 (3 points)

A l'aide du modèle COCOMO 81 basique, calculer l'effort de développement par phase (conception générale, conception détaillé, codage et tests unitaires, intégration), la durée de chaque phase (besoins, conception générale, réalisation, intégration) et le nombre de personnes (besoins, conception générale, réalisation, intégration) pour chacune de ces phases. Tracer la courbe de charge correspondant à l'évolution de l'effectif en fonction du temps.

Nous allons prendre par approximation les tables relatives à 32 KISL pour la répartition de l'effort et de la durée.

Nous obtenons :