软件测试基础与实践

实验报告

实验名称：黑盒测试实验一

实验地点：计算机楼 268

实验日期：2018 年 11 月 15 日星期四

学生姓名：白丰硕

学生学号：71116233

东南大学 软件学院 制

一、实验目的

- 能熟练应用黑盒测试中的等价类划分方法设计测试用例；
- 能熟练应用黑盒测试中的边界值分析方法设计测试用例；
- 能数量综合使用等价类划分和边界值分析解决黑盒测试需求；
- 能够在黑盒测试用例设计中同时考虑正面测试和负面测试；
- 学习测试用例的书写

二、实验环境

- 硬件环境： PC 机一台
- 软件环境： Java 编程环境： Java SDK + Intellij
- 待测程序： NextDate（可执行文件）
- 实验指导书和待测程序可从 FTP 服务器中下载

三、实验内容

实验 1：NextDate 问题的黑盒测试

实验背景：

日期是软件中被频繁处理的信息之一，软件开发人员有必要了解的一些公历历法的相关知识。

公历的前身是古罗马凯撒修订的儒略历。 根据儒略历的规定，每 4 年有 1 个闰年，闰年为 366 日，其余 3 年(称为平年)各有 365 日。公元年数能被 4 除得尽的是闰年。儒略历 1 年平均长 365.25 日，比实际公转周期的 365.2422 日长 11 分 14 秒，即每 400 年约长 3 日。这样到公元 16 世纪时已经积累了有 10 天误差。可以明显感觉到两至两分提 前了。在此情况下，教皇格列高里十三世于 1582 年宣布改历。先是一步到位把儒略历 1582 年 10 月 4 日的下一天定为格列历 10 月 15 日，中间跳过 10 天。同时修改了儒略历置闰法则。除了保留儒略历年数被 4 除尽的是闰年外。增加了被 100 除得尽而被 400 除不尽的则不是闰年的规定。这样的做法可在 400 年中减少 3 个闰年。在格列高里历历法里， 400 年中有 97 个闰年(每年 366 日)及 303 个平年(每年 365 日)，所以每年平均长 365.2425 日，与 公转周期的 365.2422 日十分接近。可基本保证到公元 5000 年前误差不超过 1 天。在软件开发和测试中，我们需要注意以下的一些有用信息：

- 1582 年 10 月 5 日至 10 月 14 日排除在公历外

- 2038 年 1 月 19 日是 BIOS 提供的记时基准时间 1970 年 1 月 1 日的最大值(下一个千年虫问题的根源)
- 英国 1752 年才采用阳历，他们扣除 9/3/1752 到 9/13/1752 年同步以月亮为参照的立法

备注：

- 以上信息中，后两条并不影响我们所进行的测试活动,可不用考虑。
- NextDate 程序中有 3 个输入，分别对应一个日期的年、月、日，程序能输出给定日期的下一天。
- 程序能接收的日期输入范围为 1582 年 1 月 1 日到 3000 年 12 月 31 日。

要求：

- 综合使用等价类划分和边界值分析方法对该程序进行黑盒测试；
- 设计的测试用例都要有充分的设计理由。

等价类划分：

序号	等价类
1	1582.1.1-1582.10.4 的有效等价类
2	1582.10.15-3000.12.31 的有效等价类
3	1582.10.5-1582.10.14 的无效等价类
4	日向上超出的无效等价类
5	日小于 1 的无效等价类
6	月大于 12 的无效等价类
7	月小于 1 的无效等价类
8	年大于 3000 的无效等价类
9	年小于 1582 的无效等价类
10	年月日输入不全的无效等价类
11	年月日中至少一个包含字符的无效等价类
12	年月日中至少一个包含中文的无效等价类
13	年月日中至少一个包含浮点数的无效等价类

边界条件确定：

- 固定日、年的月边界条件

边界条件	月	日	年
1	1	1-28	1582-3000
2	12	1-28	1582-3000

- 固定月、年的日边界条件

边界条件	月	日	年
3	30 天的月	1	1582-3000
4	30 天的月	30	1582-3000
5	31 天的月	1	1582-3000
6	31 天的月	31	1582-3000

- 固定日、月的年边界条件

边界条件	月	日	年
7	1-12	1-28	1582
8	1-12	1-28	3000
9	2	1	闰年
10	2	29	闰年
11	2	1	平年
12	2	28	平年

- 补充确定的关联边界条件

边界条件	月	日	年
13	1	1	1582
14	12	31	3000
15	10	5	1582
16	10	14	1582

测试用例设计：

编号	测试用例	期望输出	实际输出	设计理由
1	~~1582.01.01~~	~~1582-1-2~~	~~1582-1-2~~	~~1582.1.1-1582.10.4 的有效等价类~~
2	~~1582.10.04~~	~~1582-10-15~~	~~1582-10-15~~	~~1582.1.1-1582.10.4 的有效等价类~~
3	~~1582.10.15~~	~~1582-10-16~~	~~1582-10-16~~	~~1582.10.15-3000.12.31 的有效等价类~~
4	~~3000.12.31~~	~~30001-1-1~~	~~30001-1-1~~	~~1582.10.15-3000.12.31 的有效等价类~~
5	~~1582.10.05~~	~~Error~~	~~Error~~	~~1582.10.5-1582.10.14 的无效等价类~~
6	~~1582.10.14~~	~~Error~~	~~Error~~	~~1582.10.5-1582.10.14 的无效等价类~~
7	~~1582.03.32~~	~~Error~~	~~Error~~	~~日向上超出的无效等价类~~
8	~~1999.02.29~~	~~Error~~	~~Error~~	~~日向上超出的无效等价类~~
9	~~2000.02.30~~	~~Error~~	~~Error~~	~~日向上超出的无效等价类~~
10	~~1582.06.31~~	~~Error~~	~~Error~~	~~日向上超出的无效等价类~~
11	~~1582.06.00~~	~~Error~~	~~Error~~	~~日小于 1 的无效等价类~~
12	~~1582.13.01~~	~~Error~~	~~无响应~~	~~月大于 12 的无效等价类~~
13	~~1582.00.28~~	~~Error~~	~~Error~~	~~月小于 1 的无效等价类~~

14	~~3001.01.01~~	~~Error~~	~~Error~~	~~年大于 3000 的无效等价类~~
15	~~1581.12.31~~	~~Error~~	~~Error~~	~~年小于 1582 的无效等价类~~
16				年月日输入不全的无效等价类
17	无法输入			年月日中至少一个包含字符的无效等价类
18	无法输入			年月日中至少一个包含中文的无效等价类
19	无法输入			年月日中至少一个包含浮点数的无效等价类
20	1582.00.28	Error		边界条件 1 的检测
21	1582.01.28	1582-1-29	1582-1-29	边界条件 1 的检测
22	1582.02.28	1582-2-29	1582-2-29	边界条件 1 的检测
23	1582.11.01	1582-11-2	1582-11-2	边界条件 2 的检测
24	1582.12.01	1582-12-2	1582-12-2	边界条件 2 的检测
25	1582.13.01	Error	无响应	边界条件 2 的检测
26	1582.06.00	Error	Error	边界条件 3 的检测
27	1582.06.01	1582-6-2	1582-6-2	边界条件 3 的检测
28	1582.06.02	1582-6-3	1582-6-3	边界条件 3 的检测
29	1582.06.29	1582-6-30	1582-6-30	边界条件 4 的检测
30	1582.06.30	1582-7-1	1582-7-1	边界条件 4 的检测
31	1582.06.31	Error	Error	边界条件 4 的检测
32	1582.03.00	Error	Error	边界条件 5 的检测
33	1582.03.01	1582-3-2	1582-3-2	边界条件 5 的检测
34	1582.03.02	1582-3-3	1582-3-3	边界条件 5 的检测
35	1582.03.30	1582-3-31	1582-3-31	边界条件 6 的检测
36	1582.03.31	1582-4-1	1582-4-1	边界条件 6 的检测
37	1582.03.32	Error	Error	边界条件 6 的检测
38	1581.06.06	Error	Error	边界条件 7 的检测
39	1582.06.06	1582-6-7	1582-6-7	边界条件 7 的检测
40	1583.06.06	1583-6-7	1583-6-7	边界条件 7 的检测
41	2999.06.06	2999-6-7	2999-6-7	边界条件 8 的检测
42	3000.06.06	3000-6-7	3000-6-7	边界条件 8 的检测
43	3001.06.06	Error	Error	边界条件 8 的检测
44	2000.02.00	Error	Error	边界条件 9 的检测
45	2000.02.01	2000-2-2	2000-2-2	边界条件 9 的检测
46	2000.02.02	2000-2-3	2000-2-3	边界条件 9 的检测
47	2000.02.28	2000-2-29	2000-2-29	边界条件 10 的检测
48	2000.02.29	2000-3-1	2000-3-1	边界条件 10 的检测
49	2000.02.30	Error	Error	边界条件 10 的检测
50	1999.02.00	Error	Error	边界条件 11 的检测
51	1999.02.01	1999-2-2	1999-2-2	边界条件 11 的检测
52	1999.02.02	1999-2-3	1999-2-3	边界条件 11 的检测
53	1999.02.27	1999-2-28	1999-2-28	边界条件 12 的检测
54	1999.02.28	1999-3-1	1999-3-1	边界条件 12 的检测

55	1999.02.29	Error	Error	边界条件 12 的检测
56	1581.12.31	Error	Error	边界条件 13 的检测
57	1582.01.01	1582-1-2	1582-1-2	边界条件 13 的检测
58	1582.01.02	1582-1-3	1582-1-3	边界条件 13 的检测
59	3000.12.30	3000-12-31	3000-12-31	边界条件 14 的检测
60	3000.12.31	3001-1-1	3001-1-1	边界条件 14 的检测
61	3001.01.01	Error	Error	边界条件 14 的检测
62	1582.10.04	1582-10-15	1582-10-15	边界条件 15 的检测
63	1582.10.05	Error	Error	边界条件 15 的检测
64	1582.10.06	Error	Error	边界条件 15 的检测
65	1582.10.13	Error	Error	边界条件 16 的检测
66	1582.10.14	Error	Error	边界条件 16 的检测
67	1582.10.15	1582-10-16	1582-10-16	边界条件 16 的检测

备注：其中被划去的测试用例是两种方法得到的相同测试用例被合并的的结果。

实验 2： 四边形覆盖问题的黑盒测试

四边形覆盖问题描述：

- 程序输入： 2 个四边形:（ X1Coord, Y1Coord, Width1, Height1）和 （ X2Coord, Y2Coord, Width2, Height2），其中前 2 个参数是四边形左上角坐标，后 2 个参数指四边形的宽和高；
- 程序输出： 两个四边形的覆盖关系。
- 四边形覆盖：判断 2 个四边形在平面上的覆盖关系。

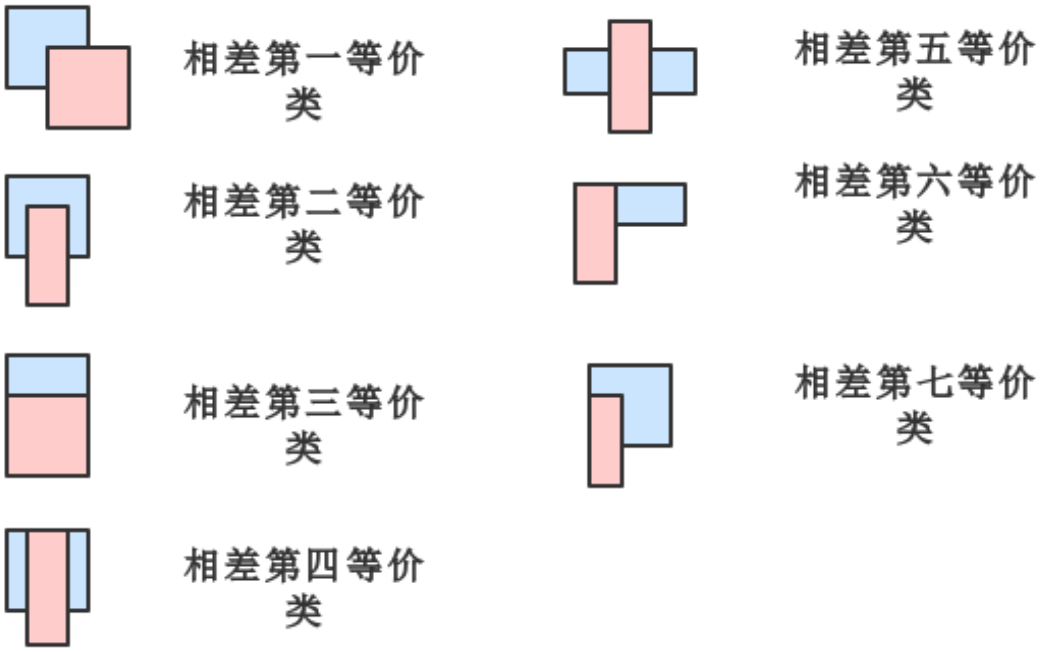
要求：

- 利用等价类划分和边界值分析方法，设计四边形覆盖问题的测试用例。请给出测试用例的具体设计思路。
- Github 上有一个少有人关注的项目 https://github.com/cuthullu/box-black-box，（可从 FTP 上下载到该项目的源码 box-black-box-gh-pages.zip，解压后可运行 index.html）。这个项目中，给出了四边形问题的可视化测试界面， 其中还包含 5 种判断四边形关系的函数。

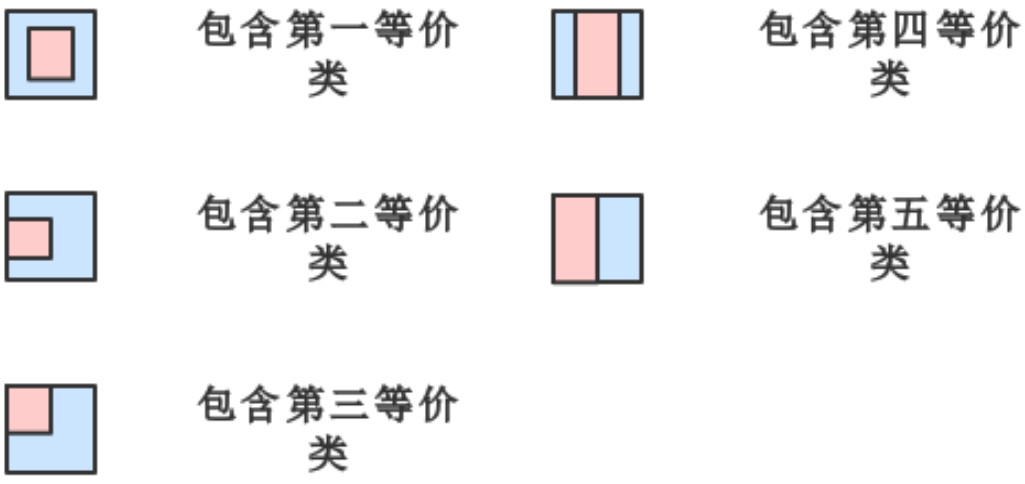
- 请利用（1）中设计的测试用例来对 box-black-box 项目进行黑盒测试， 通过黑盒测试，分析该项目给出的 6 种函数中是否存在 BUG。

等价类划分：

- 非法输入无效等价类
- 覆盖（差集不为空）



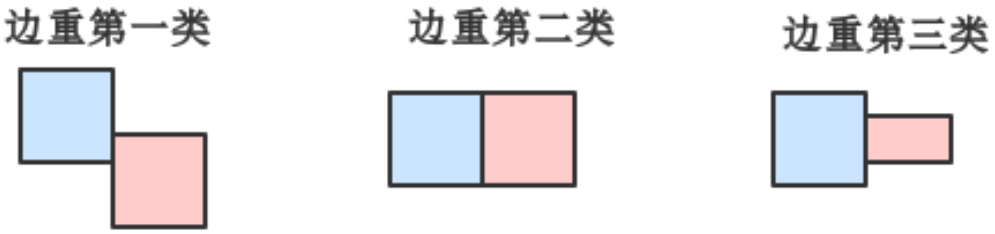
- 覆盖（包含）



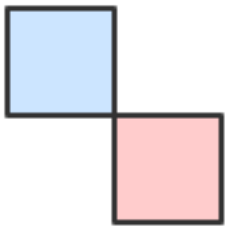
- 覆盖（重合）有效等价类
- 不覆盖的有效等价类

边界条件确定：

- 面重合（完全重合）
- 边重合



- 点重合



测试用例设计：

编号	测试用例		预期输出	实际输出						设计理由
	Rect1	Rect2		a	b	c	d	e	f	
1	-1,1,1,1	-1,1,1,1	Error							非法输入无效等价类
2	1.2,1,1,1	1.2,1,1,1	Error							非法输入无效等价类
3	a,1,1,1	a,1,1,1	Error							非法输入无效等价类

4	一，1,1,1	一 1,1,1,	Error	\	\	\	\	\	\	非法输入无效等价类
5	M+1，M+1，M，M，	M+1，M+1，M，M，	Error	\	\	\	\	\	\	非法输入无效等价类
6	0,0,2,2	1,1,2,2	T	T	T	T	T	T	T	相差第一等价类
7	1,1,2,2	0,0,2,2	T	T	F	T	T	T	T	相差第一等价类
8	1,1,2,2	2,0,2,2	T	T	F	T	T	T	T	相差第一等价类
9	2,0,2,2	1,1,2,2	T	T	F	T	T	T	T	相差第一等价类
10	1,1,3,3	2,0,1,2	T	F	F	T	T	T	F	相差第二等价类
11	1,1,3,3	2,3,1,2	T	T	T	T	T	T	T	相差第二等价类
12	1,1,3,3	0,2,2,1	T	T	F	T	T	T	T	相差第二等价类
13	1,1,3,3	3,2,2,1	T	T	T	T	T	T	T	相差第二等价类
14	0,0,2,2	0,1,2,2	T	T	T	T	T	T	T	相差第三等价类
15	0,1,2,2	0,0,2,2	T	T	F	T	T	T	T	相差第三等价类
16	0,0,2,2	1,0,2,2	T	T	T	T	T	T	T	相差第三等价类
17	1,0,2,2	0,0,2,2	T	T	F	T	T	T	T	相差第三等价类
18	0,0,3,3	1,0,1,5	T	T	T	T	T	T	T	相差第四等价类
19	0,0,3,3	0,1,5,1	T	T	T	T	T	T	T	相差第四等价类
20	1,1,3,3	0,2,4,1	T	T	F	T	T	T	T	相差第四等价类
21	1,1,3,3	2,0,1,4	T	F	F	T	T	T	F	相差第四等价类
22	0,1,3,1	1,0,1,3	T	F	F	T	T	T	T	相差第五等价类
23	0,0,2,1	0,0,1,2	T	T	T	T	T	T	T	相差第六等价类
24	0,0,1,2	0,0,2,1	T	T	T	T	T	T	T	相差第六等价类
25	1,0,1,2	0,1,2,1	T	F	T	T	T	T	T	相差第六等价类
26	0,1,2,1	1,0,1,2	T	F	T	T	T	T	T	相差第六等价类
27	1,1,2,2	0,1,2,1	T	T	F	T	T	T	T	相差第七等价类
28	1,1,2,2	1,2,1,2	T	T	T	T	T	T	T	相差第七等价类
29	1,1,2,2	2,0,1,2	T	T	F	T	T	T	T	相差第七等价类
30	1,1,2,2	2,2,2,1	T	T	T	T	T	T	T	相差第七等价类
31	1,1,1,1	0,0,3,3	T	T	F	T	T	T	T	包含第一等价类
32	0,1,1,1	0,0,3,3	T	T	F	T	T	T	T	包含第二等价类
33	1,0,1,1	0,0,3,3	T	T	F	T	T	T	T	包含第二等价类
34	2,1,1,1	0,0,3,3	T	T	F	T	T	T	T	包含第二等价类
35	1,2,1,1	0,0,3,3	T	T	F	T	T	T	T	包含第二等价类
36	0,0,1,1	0,0,2,2,	T	T	T	T	T	T	T	包含第三等价类
37	0,1,1,1	0,0,2,2,	T	T	F	T	T	T	T	包含第三等价类
38	1,0,1,1	0,0,2,2,	T	T	F	T	T	T	T	包含第三等价类
39	1,1,1,1	0,0,2,2,	T	T	F	T	T	T	T	包含第三等价类
40	1,0,1,3	0,0,3,3	T	T	F	T	T	T	T	包含第四等价类
41	0,1,3,1	0,0,3,3	T	T	F	T	T	T	T	包含第四等价类
42	0,0,1,2	0,0,2,2,	T	T	T	T	T	T	T	包含第五等价类
43	0,0,2,1	0,0,2,2,	T	T	T	T	T	T	T	包含第五等价类

44	1,0,1,2	0,0,2,2,	T	T	F	T	T	T	T	包含第五等价类
45	0,1,2,1	0,0,2,2,	T	T	F	T	T	T	T	包含第五等价类
46	3,3,3,3	3,3,3,3	T	T	T	T	F	T	T	重合有效等价类
47	2,2,2,2	3,0,2,2,	T	F	F	T	T	F	F	边重第一类检测
48	2,2,2,2	0,1,2,2	T	F	F	T	T	F	F	边重第一类检测
49	2,2,2,2	1,4,2,2	T	F	F	T	T	F	F	边重第一类检测
50	2,2,2,2	4,3,2,2	T	F	F	T	T	F	F	边重第一类检测
51	2,2,2,2	2,0,2,2,	T	F	F	T	F	F	F	边重第二类检测
52	2,2,2,2	0,2,2,2	T	F	F	T	F	F	F	边重第二类检测
53	2,2,2,2	2,4,2,2	T	F	F	T	F	F	F	边重第二类检测
54	2,2,2,2	4,2,2,2	T	F	F	T	F	F	F	边重第二类检测
55	1,1,3,3	2,0,1,1	T	F	F	T	T	F	F	边重第三类检测
56	1,1,3,3	0,2,1,1	T	F	F	T	T	F	F	边重第三类检测
57	1,1,3,3	4,2,1,1	T	F	F	T	T	F	F	边重第三类检测
58	1,1,3,3	2,4,1,1	T	F	F	T	T	F	F	边重第三类检测
59	1,1,1,1	2,2,1,1	T	F	F	T	F	F	F	点重合类检测
60	2,2,1,1	1,1,1,1	T	F	F	T	F	F	F	点重合类检测
61	1,1,1,1	2,0,1,1	T	F	F	T	F	F	F	点重合类检测
62	2,0,1,1	1,1,1,1	T	F	F	T	F	F	F	点重合类检测
63	1,1,1,1	3,3,1,1	F	F	F	F	F	F	F	不覆盖的有效等价类

备注：

M 表示八个输入数值最大值（M = 99999999999999934463）

标注为红色表示与期望输出有差异的实际输出

四、实验体会

1. 通过测试，是否发现程序中存在的缺陷？

- 实验一

  实验一中，当设置月大于 12 时，点击 nextDate 按钮后没有响应，不提示任何错误提示信息。

- 实验二

  a 函数在识别两个矩形有覆盖区域（差集不为空）情形下，部分情况无法正确识别；在边重合的三种情况以及点重合的情况都不可以正确识别。

  b 函数在识别两个矩形有覆盖区域（差集不为空）情形下，部分情况无法正确识别；在识别一个矩形在另一个矩形内部情况（包含）中，有很多都无法正确识别；在边重合的三种情况以及点重合的情况都不可以正确识别。

d 函数完全重合的情况下无法正确识别;在很多边重合的情况中也无法正确识别。

e 函数在边重合的三种情况以及点重合的情况都不可以正确识别。

f 函数在识别两个矩形有覆盖区域（差集不为空）情形下，部分情况无法正确识别；在边重合的三种情况以及点重合的情况都不可以正确识别。

2. 在黑盒测试中，测试用例的设计实际上是一件非常具有挑战性的工作。谈谈你在进行黑盒测试过程中所碰到的难题。

在做黑盒测试实验中，实验一和二都是使用等价类划分和边界值分析综合来设计测试用例。实验一情况比较简单，比较好分析，主要是因为它的主要输入是三个整数（并且还带有范围），这对于寻找不同的等价类和分析边界值都是可以比较顺利的分析。但是实验二中，要进行这一过程，由于其输入是两个元组，使用这两个元组来表示两个矩形的平面位置关系。一共是八个输入值，如果直接进行值的分析，就会陷入很大麻烦。但是要是按照几何关系分析，虽然很直观清晰但是很容易把一些情形（等价类）漏掉，而造成覆盖不完全。并且在设计测试用例也要花费很多的时间来保证测试用例的正确性。

3. 思考为什么现在企业内大量的项目主要采用黑盒测试，而比较少而且有限的使用白盒测试技术？谈谈你对企业这样做的原因的理解和这样做的危害。

结合我对白盒测试和黑盒测试的理解与一些社会认知来说一下我的看法。

- **白盒测试成本太高。**白盒测试的重要部分就是要理解整个程序的框架和其中的逻辑，那么就一定要阅读源码，但是企业项目的代码不是往往是不同团队的协作，所以代码量大，不同模块的代码风格可能不同，以及代码规范以及可读性都是不合格的，在这种情况下做白盒测试，这对于商业化的企业要付出的代价成本太高了，很可能会导致项目时间延长，投入团队人数增多。
- **黑盒测试更贴近实际。**黑盒测试相比白盒测试，着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。更重要的是，黑盒测试包含了用户的视角，使用黑盒测试更容易发现用户使用时可能会出现的程序缺陷。

对于企业重黑盒，轻白盒的这种思路，可能的危害就是，对于一些程序可能已经通过了黑盒测试。其功能可能是在表面是正确的，但是其内部的一些控制流或者数据流并没有正确的发生变化，只是没有明显的暴露出来。但是这个可能就会导致软件有漏洞，漏洞要是被一些人利用可能会让开发公司或者软件使用者蒙受损失。所以一些公司会不断的更新自己的程序，不定期的修补软件中的漏洞。他们依赖用户的使用情况（相当于黑盒测试）以及其反馈情况来发现其中的漏洞，这样比较被动，但是巨大的项目情况过于复杂，要想在开发时期测试完全覆盖几乎是不可能的。