

# 软件测试方法之优化组合法

周立 原创

## 1. 适用对象

单元测试完成后，准备做多个测试点/功能点的集成测试的时候，如果每个测试点包含多个测试因子，此方法可以协助你设计比较有效的测试案例。

## 2. 目标

使用 20% 的测试案例去发现 80% 的缺陷（80-20 原则）。

## 3. 试图解决的问题

某一个功能/模块牵涉的测试点及其测试因子比较多，导致：

- ◇ 针对该模块测试者不知道如何设计出比较有效的测试案例，因为如果想要发现所有的缺陷，也就是达到 100% 的缺陷发现率，就要遍历所有测试因子的完全组合，test case 的数量有可能会非常惊人，无法实现；
- ◇ 虽然花了很多时间测试这个功能/模块，仍担心有遗漏的地方；
- ◇ 经过比较长的时间或者多个版本测试后这个功能/模块已经很稳定了，但是某一处的代码变更又导致了较多问题的出现。测试人员不得不又要花上相当多的时间去测试该功能/模块；

## 4. 优点

- 把很多零散、无序的测试点和测试因子尽量全面的集中整理到一张表格中，使得测试点和测试因子比较完整，不容易遗漏；
- 排列测试因子的优先级，突出重点，使用尽量少的 case 发现尽可能多的缺陷，从而达到最有效的测试目的；
- 方法中使用的表格清晰明了，易于评审；
- 通过设计 case 的优先级，减少同一模块需要重复测试时需要投入的工作量。

## 5. 使用时间

- ✓ 软件测试的设计阶段，输出记录在软件测试大纲中；
- ✓ 虽然产品已经发布，但是部分功能/模块仍然存在比较多的缺陷，且该模块属于本方法的测试对象范畴，也可以尝试本方法。

## 6. 使用方法

步骤一：列出所有的测试点和每个测试点涵盖的所有测试因子。横坐标为测试点，纵坐标为测试因子。

Table 1 （缩写：TP – Test Point, V – Value ）

		A	B	C	D	E	F	G	H
1.	TP1	V1A	V1B	V1C					
2.	TP2	V2A	V2B	V2C	V2D	V2E	V2F	V2G	V2H
3.	TP3	V3A	V3B						

4.	TP4	V4A	V4B	V4C	V4D				
5.	TP5	V5A	V5B	V5C					
6.	TP6	V6A	V6B						
7.	TP7	V7A	V7B	V7C	V7D	V7E			
8.	TP8	V8A	V8B						
9.	TP9	V9A	V9B						

步骤二：定义所有测试因子的 Severity, 并作出标识, 如 HIGH 标识为红色, MEDIUM 表示为黄色, LOW 表示为绿色。定义 Severity 的依据包括客户的使用频率和测试因子参与集成的时候产生缺陷的概率。在定义 Severity 的时候要考虑到 HIGH 级别的数量, 参与完全组合的测试因子建议不超过测试因子总数的 30%。(如果某个测试点只有一个测试因子, 则该因子不会影响完全组合的数量, 所以这样的测试因子建议不参与完全组合)

Table 2

		A	B	C	D	E	F	G	H
1.	TP1	V1A	V1B	V1C					
2.	TP2	V2A	V2B	V2C	V2D	V2E	V2F	V2G	V2H
3.	TP3	V3A	V3B						
4.	TP4	V4A	V4B	V4C	V4D				
5.	TP5	V5A	V5B	V5C					
6.	TP6	V6A	V6B						
7.	TP7	V7A	V7B	V7C	V7D	V7E			
8.	TP8	V8A	V8B						
9.	TP9	V9A	V9B						

步骤三：调整表格, 纵向按照每个测试点包含的 HIGH 级别的测试因子的数量降序排列, 如 TP7 包含 3 个 HIGH 的测试因子, 应将其放在第一行。TP2 次之包含 2 个测试因子, 放在第二行, 以此类推。横向按照测试因子的级别降序, HIGH 的放在左边, MEDIUM 放在中间, LOW 的放在最后。

Table 3

		A	B	C	D	E	F	G	H
1.	TP7	V7A	V7C	V7D	V7B	V7E			
2.	TP2	V2A	V2C	V2B	V2D	V2E	V2F	V2G	V2H
3.	TP4	V4B	V4C	V4D	V4A				
4.	TP9	V9B	V9A						
5.	TP1	V1A	V1B	V1C					
6.	TP3	V3B	V3A						
7.	TP5	V5A	V5B	V5C					
8.	TP6	V6A	V6B						
9.	TP8	V8A	V8B						

步骤四：将 HIGH 级别的所有因子做完全组合(如果某个测试点只有一个 HIGH 的测试因子,

则不参与完全组合，因为它们不会影响到完全组合的数量)，并生成测试案例表格。此时所有测试点放在第一行，纵坐标为测试案例的编号。

此处参与完全组合的为 TP7 和 TP2，共生成  $3 \times 2 = 6$  个测试案例。

Table 4

	TP7	TP2	TP4	TP9	TP1	TP3	TP5	TP6	TP8
1.	V7A	V2A							
2.	V7A	V2C							
3.	V7C	V2A							
4.	V7C	V2C							
5.	V7D	V2A							
6.	V7D	V2C							
7.									
8.									
9.									
10.									

步骤五：删除你认为不需要测试的 case，如第三行是 V7C 和 V2A 组合在一起，如果你觉得没有意义，可以将其删除或者做上特别标识。

	TP7	TP2	TP4	TP9	TP1	TP3	TP5	TP6	TP8
1.	V7A	V2A							
2.	V7A	V2C							
3.	V7C	V2A							
4.	V7C	V2C							
5.	V7D	V2A							
6.	V7D	V2C							
7.									
8.									
9.									
10.									

步骤六：补充表格，保证拥有最多测试因子的测试点的测试因子都被涵盖到。如此处 TP2 拥有 8 个测试因子，为最多，而 HIGH 级别的测试因子经过完全组合后只有 V2A 和 V2C 被 cover 到了，所以至少还要增加六行来保证 TP2 的所有测试因子都被 cover 到。

Table 5

	TP7	TP2	TP4	TP9	TP1	TP3	TP5	TP6	TP8
1.	V7A	V2A							
2.	V7A	V2C							
3. *	V7B	V2A							
4.	V7B	V2C							
5.	V7C	V2A							

6.	V7C	V2C							
7.		V2B							
8.		V2D							
9.		V2E							
10.		V2F							
11.		V2G							
12.		V2H							

步骤七：根据经验填充表格中剩下的测试点的测试因子，原则是级别越高测试因子的出现的频率越高，级别越低的测试因子出现的频率越低，但是至少要保证每个测试因子都被 cover 到一次或一次以上。

Table 6

	TP7	TP2	TP4	TP9	TP1	TP3	TP5	TP6	TP8
1.	V7A	V2A	V4B	V9B	V1A	V3B	V5A	V6A	V8A
2.	V7A	V2C	V4C	V9A	V1B	V3A	V5B	V6B	V8B
3. *	V7C	V2A	V4B	V9B	V1C	V3B	V5C	V6A	V8A
4.	V7C	V2C	V4D	V9B	V1A	V3B	V5A	V6B	V8A
5.	V7D	V2A	V4B	V9B	V1B	V3B	V5B	V6A	V8A
6.	V7D	V2C	V4E	V9B	V1C	V3B	V5C	V6B	V8A
7.	V7B	V2B	V4B	V9B	V1A	V3A	V5A	V6A	V8A
8.	V7E	V2D	V4C	V9B	V1B	V3A	V5B	V6B	V8A
9.	V7B	V2E	V4B	V9B	V1C	V3A	V5C	V6A	V8A
10.	V7E	V2F	V4D	V9B	V1A	V3A	V5A	V6B	V8A
11.	V7A	V2G	V4B	V9A	V1B	V3B	V5B	V6A	V8A
12.	V7C	V2H	V4B	V9A	V1C	V3B	V5C	V6B	V8A

步骤八：增加你认为不足的 test case。

步骤九：对这些 case 定义优先级，优先级高的可用于 Smoke 测试。

步骤十：根据 Table 6 生成测试案例。

## 7. 总结

针对上面操作步骤中的那个表格，数据汇总如下：

测试点总数：9 个

测试因子总数：3+8+2+4+3+2+5+2+2=31 个 (HIGH 7 + MEDIUM 15 + LOW 9)

参与完全组合的 HIGH 级别测试因子的比例：5/31=16%

如果在保证每个测试点及其测试因子都被 cover 到，设计最少的 case 数量是：8

如果使用最完整的组合法，case 的数量为：3X8X2X4X3X2X5X2X2=23040

优化组合后的 case 数量为：12

## 8. 建议

使用者可以根据实际情况、测试的最终目的，灵活变更使用此方法。以达到最佳使用的目的。

如果在设计表格的时候觉得 word 中不太好用，可以考虑在 excel 中设计使用。