

软件工程实践
Rational 用户 组



uml.org.cn
UML 软件工程组织

软件需求管理及其在项目中的应用



UML 软件工程组织

樊晓明



Evolve by case

技术

管理

需求

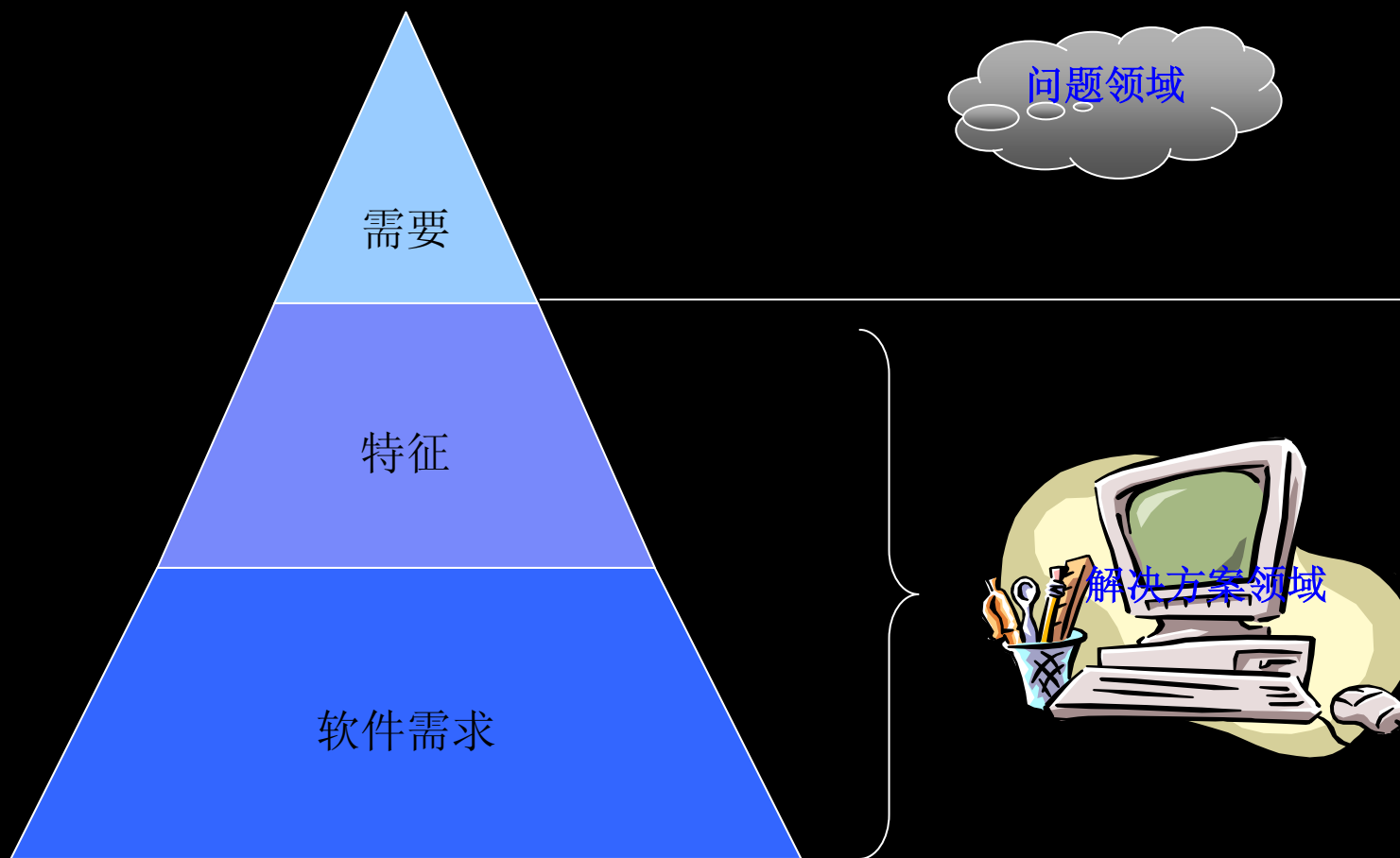


Roadmap

- 需求的层次
- 需求管理对团队技能的要求
- 需求管理的工具
- 软件需求管理的相关文档
- 交流



需求的层次



RM对团队技能的要求

- 技能一：分析问题
- 技能二：理解用户的需要
- 技能三：定义系统
- 技能四：管理广度
- 技能五：细化系统定义
- 技能六：构建正确的系统



RM对团队技能的要求

- 技能一：分析问题
 - ▶ 分析问题的步骤
 - ▶ 业务建模



RM对团队技能的要求

■ 技能二：理解用户的需要

▶ 三个综合症：

- “是的，但是”综合症
- “尚未发现的遗址”综合症
- “用户和开发人员”综合症

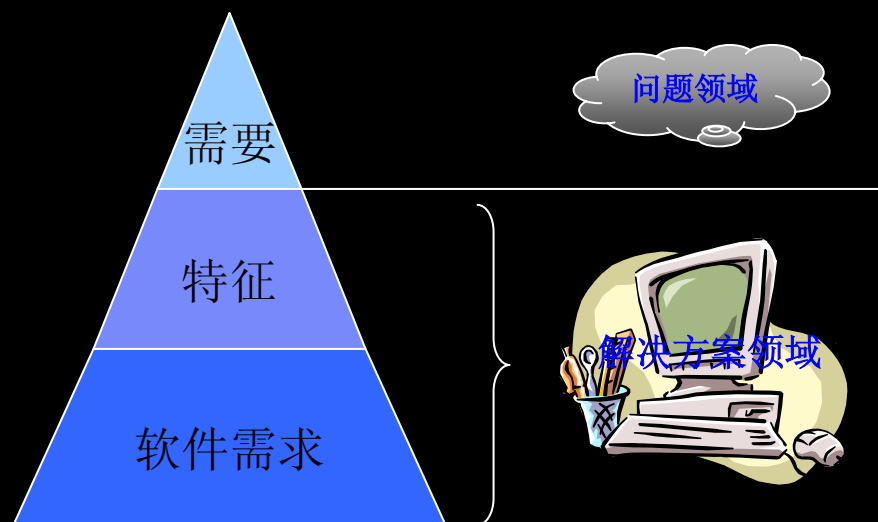
▶ 需求启发的技术：

- | | |
|-------------|--------|
| • 面谈 | • 用例 |
| • 需求专题讨论会 | • 角色扮演 |
| • 自由讨论和意见精简 | • 原型开发 |
| • 情节串联板制作 | |



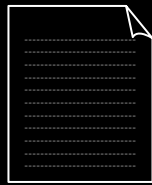
RM对团队技能的要求

- 技能三：定义系统
 - ▶ 前景文档
 - ▶ δ 前景文档



RM对团队技能的要求

- 技能三：定义系统
 - ▶ 前景文档



前景文档1.0版本

- 概括
- 用户和市场
- 1.0特征
- 其他需求
- 未来的特征



RM对团队技能的要求

■ 技能三：定义系统

▶ δ 前景文档



概况
+1.0特征
+将来特征

全面起点

1.0版本



+新特征
-删除特征
+将来特征

+

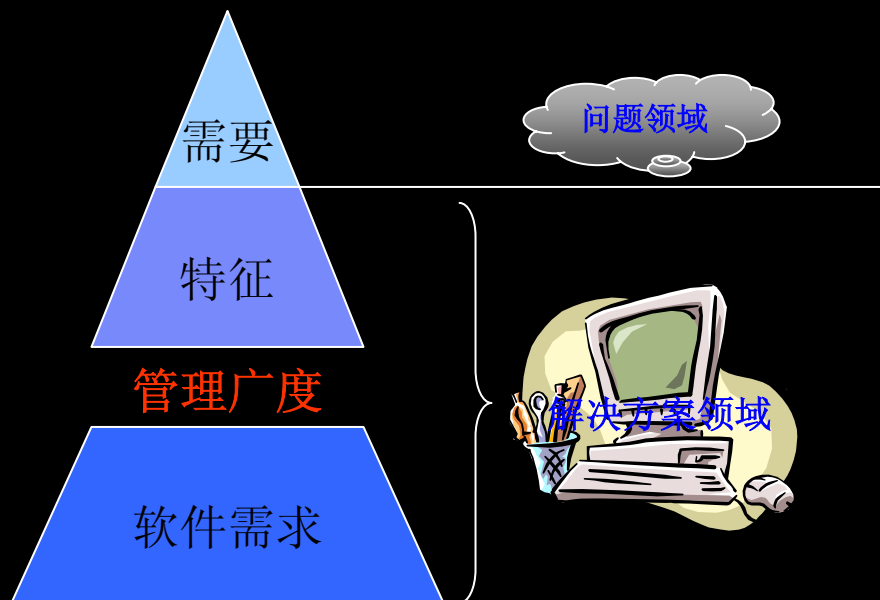
δ 2.0版本

=完整产品定义



RM对团队技能的要求

- 技能四：管理广度



RM对团队技能的要求

- 技能四：管理广度
 - ▶ 广度

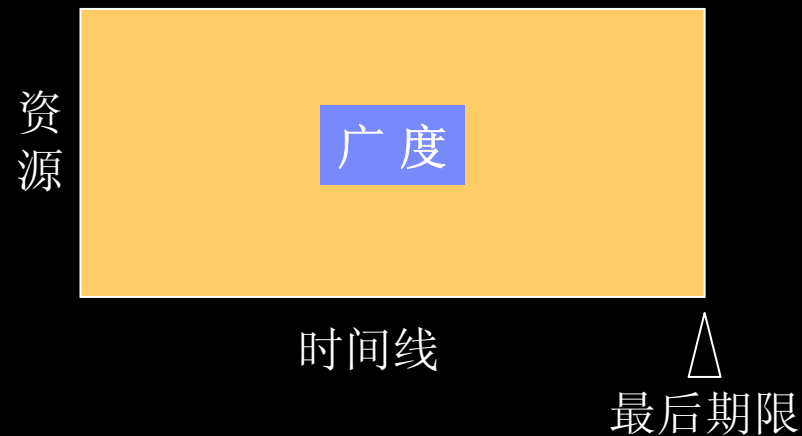


RM对团队技能的要求

■ 技能四：管理广度

▶ 难题：

- 要把广度削减下来；
- 同时还要使客户满意！



RM对团队技能的要求

■ 技能四：管理广度

▶ 建立项目广度：

- 第一步：建立需求基线；
- 第二步：建立基线内每个特征需要的工作量等级；
- 第三步：评估基线内每个特征的风险；
- 第四步：用获得的数据建立项目广度。



RM对团队技能的要求

系统1.0版本基线

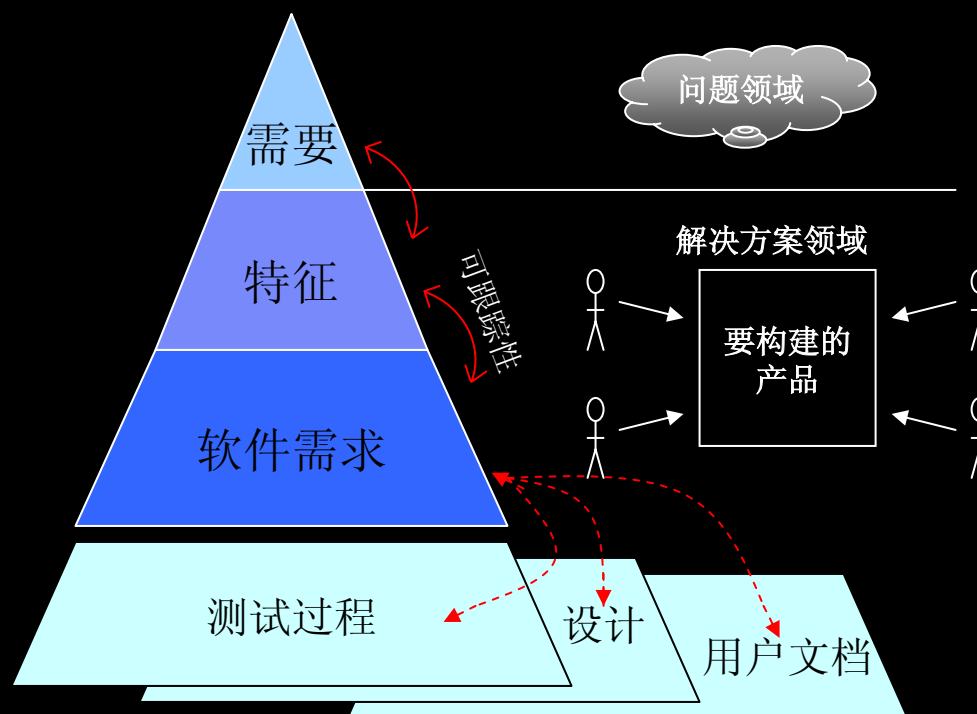
- 技能四：管理广度
 - ▶ 结果

特 征	备 注
特征4	
特征8	
特征12	
...	
1.0版本强制性基线：以上各行必须实现，否则将推迟发布	
特征2	
特征3	
...	
1.0版本可选特征：尽可能去做	
未来特征（以下各行，现在不开发）	
特征5	
特征9	
...	



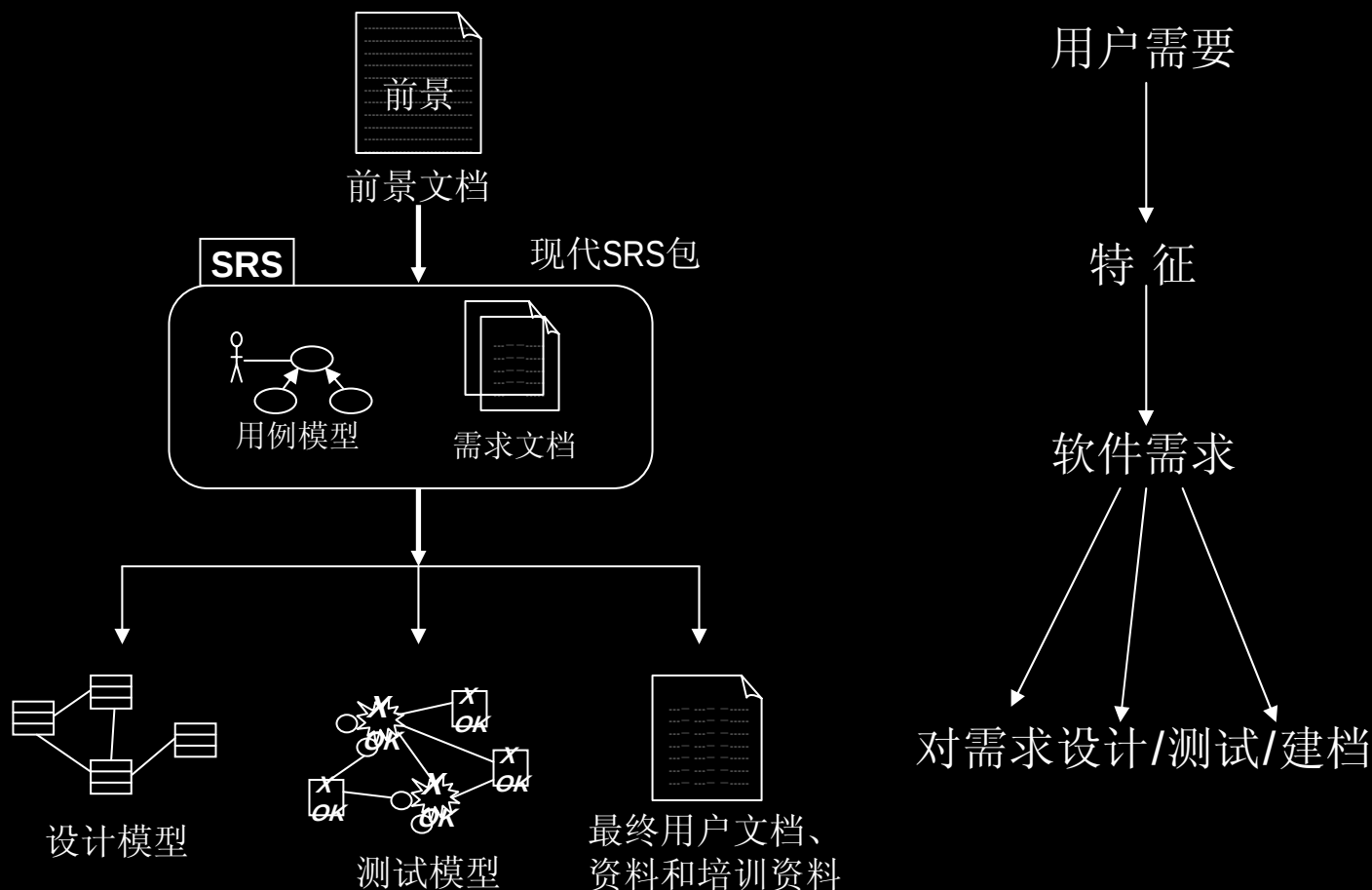
RM对团队技能的要求

- 技能五：细化系统定义



RM对团队技能的要求

■ 技能五：细化系统定义



RM对团队技能的要求

- 技能六：构建正确的系统
 - ▶ 团队理解需求但不能构建符合需求的系统！
 - ▶ 确定正在正确地构建正确系统
 - 不断证实开发是沿着正确的方向；
 - 证实开发的结果是正确的；
 - 处理变更。



RM对团队技能的要求

- 技能六：构建正确的系统
 - ▶ V&V
 - ▶ 深度&覆盖



RM对团队技能的要求

- 技能六：构建正确的系统
 - ▶ 需求变更
 - 为什么需求会变更？



RM对团队技能的要求

■ 技能六：构建正确的系统

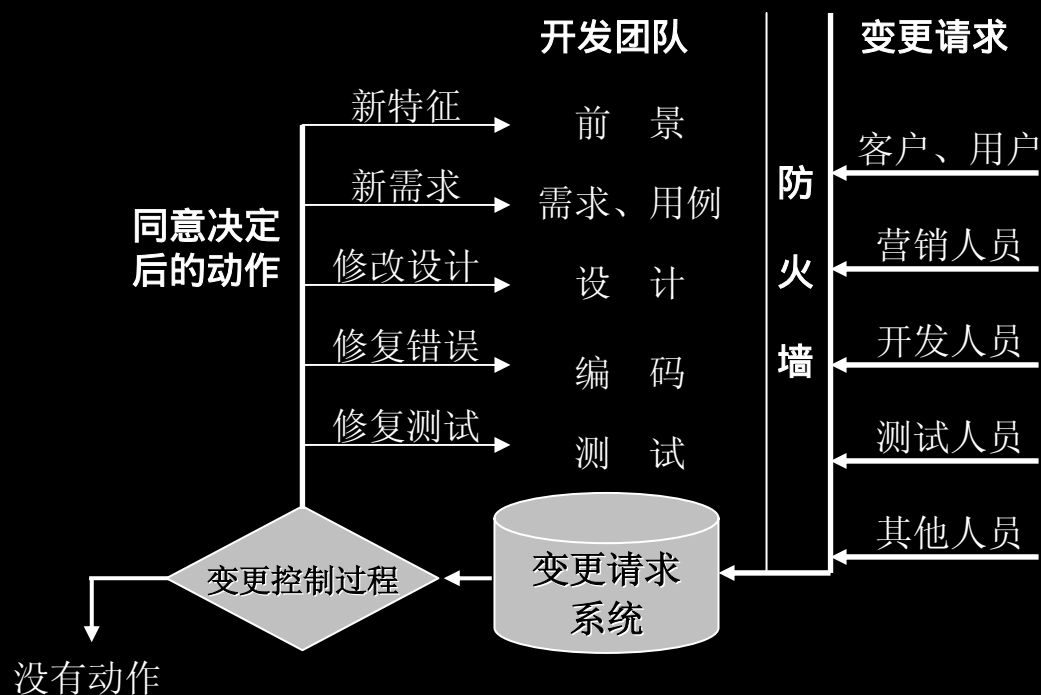
▶ 需求变更

- 需求变更管理过程：
 - 认识到变更是不可避免的并为之制定计划；
 - 确认需求的基线；
 - 建立控制变更的惟一渠道；
 - 使用变更控制系统来捕获变更；
 - 分层次管理变更。



RM对团队技能的要求

- 技能六：构建正确的系统
 - ▶ 需求变更
 - 变更控制系统



需求管理的工具

- Rational RequisitePro
- Rational Rose



需求管理相关文档

- 合同
- 需求规格说明书
- 需求管理计划
- 需求变更管理计划
- 情境文档
- 前景文档
- 需求原型系统/文档



交流

别投机！



交流

别镀金！



交流

别泄漏！



交流

安排设计人员参与!



交流

组织提供过程指导!



交流

需求工作人员受控!



交流

激励对需求工作的影响!



交流

使用有经验人员!



交流

需求小组配置

建议三人：

沟通

技术

新手



交流

从简单中获利!



交流

注意冲突管理!



交流

注意管理客户！



交流

使用沟通技巧!



交流

稳! 稳! 稳!



交流

需求的所有者！



交流

没有“捷径”！



交流

失败是控制客户的唯一结果！



交流

尽量使用界面原型，
尽量少的在界面原型上投入！



交流

引导用户参与需求变更!



交流

管理项目后期人员流动对需求工作的风险。



交流

技术既不是项目失败的主要原因，
也不是项目成功的主要原因！



交流

不要开发太多的特征！



交流

每个复杂问题都有一个简单的解决方案!



交流

管理项目广度，
如果可能，削减之！



交流

管理客户，适度引发冲突！



交流

让客户明白，对功能要求越来越多将损害项目的质量和整体生存能力：

“更多”会成为“足够”的敌人；

“更好”会成为“足够好”的敌人。



交流

尽早触发“是的，但是”。



交流

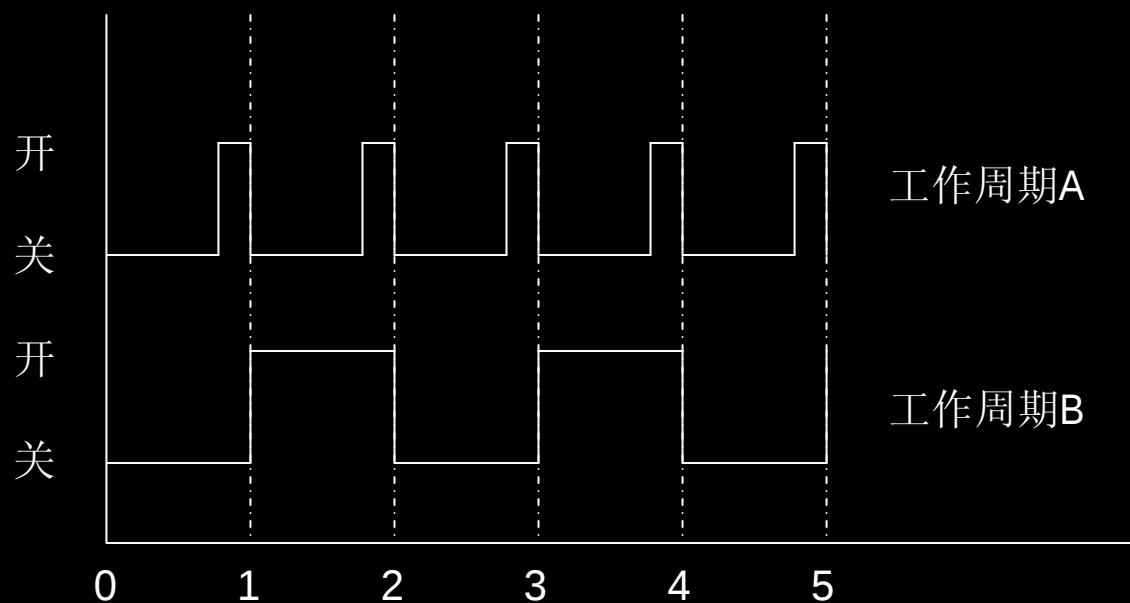
“如果你的惟一工具是锤子，那么你的所有问题看起来都像是钉子。”



交流

需求的歧义性

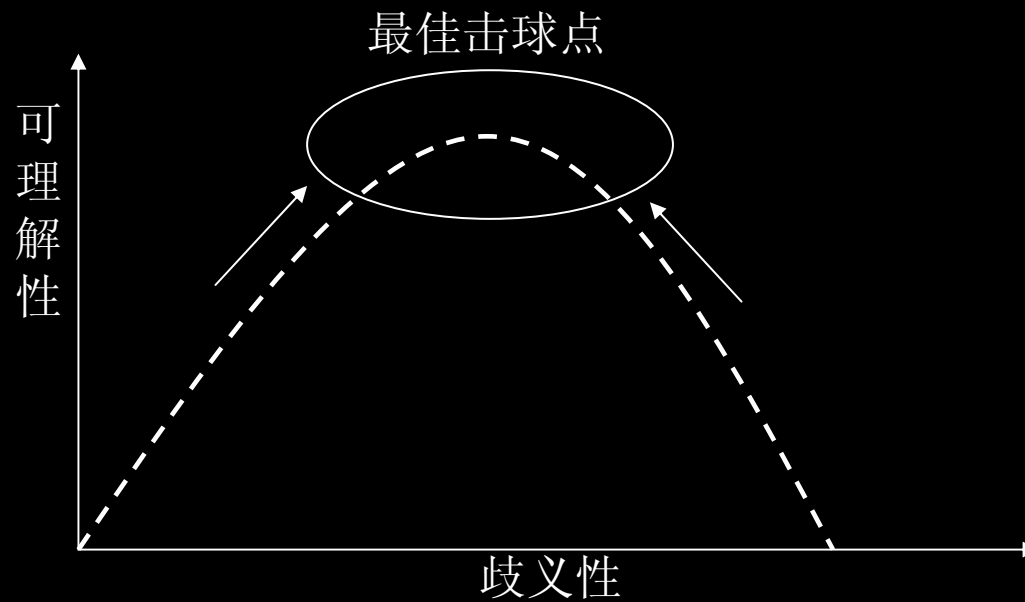
- ▶ “每隔1秒灯泡闪一次。”



交流

需求的歧义性

- ▶ “最佳击球点。”



交流

是谁打败了我们？



GOOD LUCK !

