一，能力 沟通 敏捷方法

学习和思考

学习技术， 学习要解决的问题需求，社区的巧妙技术

二，新手到专家

新手，没经验 ，重复经验不算 不会分析情境 需要给予指示，照着做，不会处理意外

高级新手 尝试解决问题 缺少全局思维，查找解决方法但是没重温体系

对公司，团队整体状况不关心

胜任者 解决遇到的问题 探索新发现的问题 寻找专家意见

精通 全局看待问题 深入研究原理，主动纠正和优化 自我改进，学习他人经验

专家，从依赖规则到依赖直觉 问题从整体关注

自我认知，大部分人高估自己

2,必须实践 模仿 吸收 创新 这三个过程可以一起的 先对原程序分析，写中文注释，或者伪代码 然后写出自己的程序 改成其他形式

在此过程中使用代码规范，重构的知识

3，工具陷阱 uml 模型不是万能，不一定能构建所有问题

三 认识大脑

双cpu

l逻辑和语言 线性 2搜索引擎 匹配记忆

r后台记忆 是异步的

记忆必须刷新，不用就一定会忘

没关注的事情虽然记下来了，但是没有索引，所以想不起来

随时记录想法，用笔记 语音等 不断记录这些想法 需要处理想法

L 语言 分析 符号 抽象 时间 推理 数字 逻辑 线性

2，设计优于功能 吸引力大于作用

四，利用右脑

加强感观体验 界面鉴赏

画图笔记 多感官反馈

用右脑绘画

画画只关注形和线，不关注画的意思

结对编程，另一个人看

五 调试大脑

思维定势

归因错误

自私和偏见

不一定非要确定答案

用惯了就不换更好的技术方法

很少不等于没有，低概率在样本足够大的时候会大量发生

记忆靠不住，多做记录

七 积累经验

必须实践， 必须有收获

真正的学习不是死记硬背，要找到乐趣 用好玩的方式学习，界面化，游戏化

2，怎样解题 利用现有知识 未知知识 条件 计划 画图，过去相关知识，缩小范围 找相似

3，失败 需要管理失败，你做错了什么，你没做什么

需要记录失败并整理分析

探索实验，不同方法，回退

4，内在诀窍 用探索来学习

允许失败

脑内模拟程序过程，先从整体开始，再设想细节，先中文表述，再想程序写法

八，注意力

提高注意力

人注意力容易被分散 需要关注一件事

写中文伪代码

单任务，减少干扰

九，超越专家

改变很难

制定短期计划，保证实施，失败的计划直接划掉 ，计划动态调整，计划越小越容易完成越短越好，比如看几页笔记，做五个俯卧撑

长计划拆成小计划，每次只关注一步

做最近的事

不怕做错，就怕不做

永远关注新技术