

回顾产品的快速迭代开发过程



扫码试看/订阅

《 Django 快速开发实战 》视频课程

快速迭代开发过程

- 快速迭代的价值与挑战
- 使用 OOPD 方法识别产品核心功能
- 如何做好技术方案设计与工作拆解
- 如何保证交付质量和持续迭代

快速迭代的价值与挑战

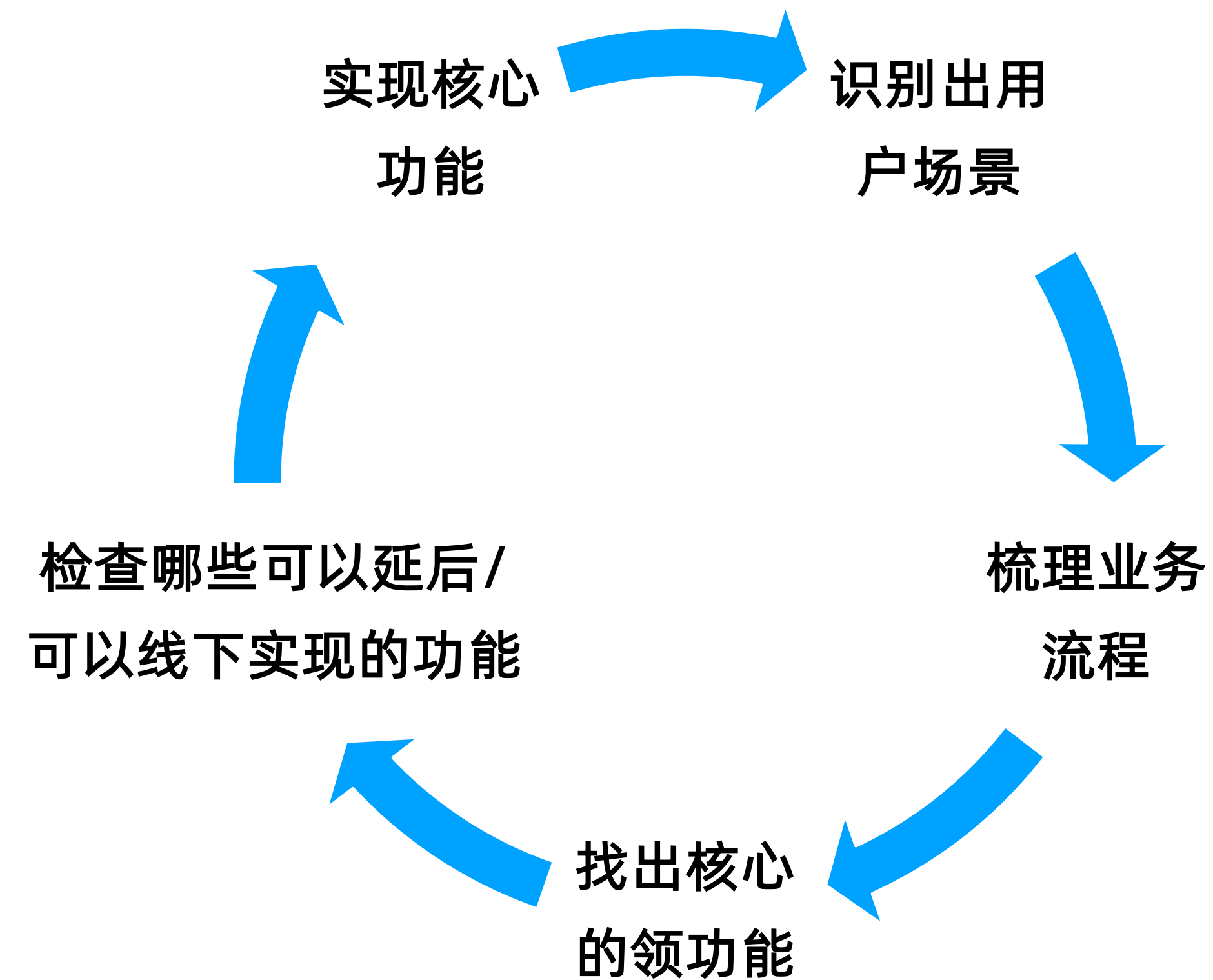
- 快速迭代：以天，甚至小时为单位，持续完善产品，交付到用户的循环过程
- 快速迭代的价值
 - 时间是最大的成本：机会瞬息即逝，赢得市场先机
 - 快速验证需求，减少不对用户产生价值的投入（Fail fast, fail better）
 - 快速验证方案，提高研发效率
 - 加速反馈回路，给到团队和自己即时的激励

快速迭代的价值与挑战

- 快速迭代的挑战
 - 产品设计者：能梳理清楚业务流程，抓住客户的重点需求，能把客户需求转化为系统需求
 - 开发者：充分理解用户需求;有足够的的能力，能用简洁的方案来设计出易维护的系统
- 根本挑战
 - 市场、用户、技术、环境变化太快，产品开发赶不上节奏
 - 几乎不能从一开始就设计出一个完美的，能够适应未来长时间变化的方案
 - 几乎没有人愿意承认，自己没有足够的的能力（或条件）设计出一个完美的产品（系统）

使用 OOPD 方法识别产品核心功能

- OOPD (Online and Offline integrated Product development) 产品开发流程



使用 OOPD 方法识别产品核心功能

OOPD 快速迭代的原则

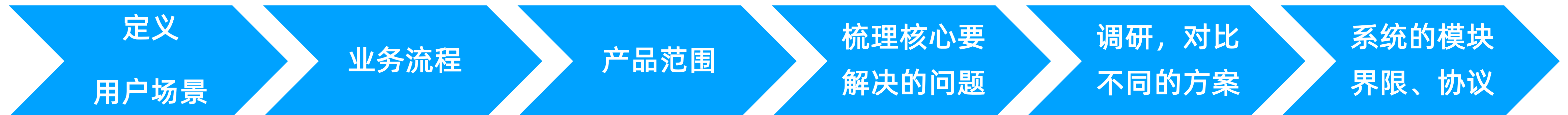
- 自助原则：做自己用的产品，自己用自己的产品，吃自己的狗食
- 0day 原则：找到明确的核心问题 拆解目标，抓住核心的问题，忽略掉一切细节，0day 发布
- 时限原则：设定时限，挑战自我 不给自己写 Bug 的时间
- 不完美原则：不做完美的产品（没有完美的产品，不去为了完美而浪费宝贵的资源）
- 谦卑原则：能够看到自己的局限性，获取用户反馈，持续迭代，听取用户声音

如何做好技术方案设计与工作拆解

- 做技术方案设计的前提条件
 - 有明确的用户场景，用户如何跟产品交互，期望拿到什么样的预期结果
 - 有清晰定义的业务流程

如何做好技术方案设计与工作拆解

- 技术方案设计流程



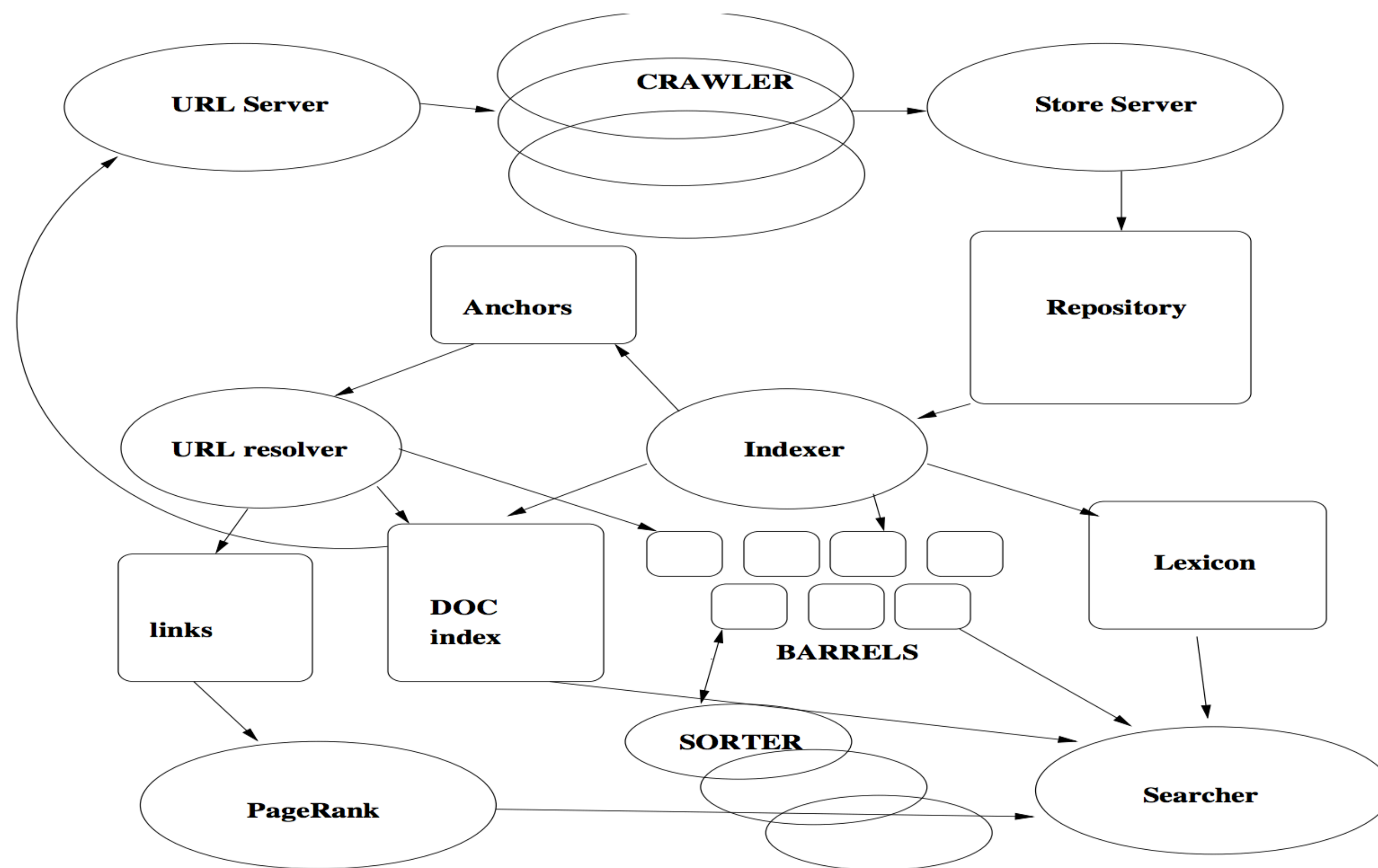
如何做好技术方案设计与工作拆解

- 用什么工具?
 - Visual Paradigm
 - Lucid Chart
 - Visio
 - Gliffy
 - Draw.io
 - Astash
 - StarUML
 - ...
- 推荐的工具：白板/白纸，不用工具是最好的工具

如何做好技术方案设计与工作拆解

- 产出的技术方案设计文档要素
 - 产品背景（用户场景，产品目标，引用到的业务流程，产品需求文档）
 - 要解决的问题列表，系统不解决的问题列表，系统的限制
 - 对于问题的不同解决方案的对比，阐述各个主要的问题如何被解决
 - 所选整体的流程图（序列图），模块关系图，重要的接口、实体的概念定义
 - 除了功能之外的其它方面的设计，包括安全，性能，可维护性，稳定性，监控，扩展性，易用性等

如何做好技术方案设计与工作拆解



如何做好技术方案设计与工作拆解

- 工作拆解
 - 任何事情，只要把它拆解到足够细，你都能够完成它
- 工作拆解的原则
 - 优先级：主流程上，不确定的工作优先完成（建议提前一个迭代做调研）
 - 核心流程优先：核心工作优先，先把主流程跑通
 - 依赖：减少不同人之间的工作依赖；并且保持团队工作拆解的透明，预留20% Buffer
 - 拆解粒度：拆解到每项子任务 0.5-1天的粒度，最长不超过2天

如何保证交付质量和持续迭代

- 定义好清晰产品需求，产品需求从根本上决定了软件的质量
- 系统有整体上的架构方案的设计，评估，评审；系统设计决定了软件实现的质量
- 工程的角度持续交付的最佳实践推荐
 - Code Review：每一次提交都有 CR，每次 commit 代码量 < 200 行，频繁commit
 - 单元测试：项目开始建立好单元测试的机制，在持续集成中自动运行
 - 自动化回归：对预发/线上系统做 API/页面自动化测试 (Postman/Robot Framework)
 - 使用 CICD 机制对系统进行自动化的打包、测试、部署、线上验证

如何保证交付质量和持续迭代

- 发布过程做到可监控，可回滚
- 对于大量用户使用的产品，使用灰度机制
- 架构上对于意外的并发访问进行限流，降级
- 架构上使用配置开关，对系统功能提供实时的开启/关闭的服务；
- 对产品建立 A/B Test 机制，通过数据来快速对比不同版本，不同方案的效果
- 自动化所有事情，代码化所有过程：代码化配置，代码化部署流程，代码化基础设施
 - 声明式 API, CI/CD Pipeline, K8S, Helm, Terraform

通往 Hacker 之路

通往 Hacker 之路

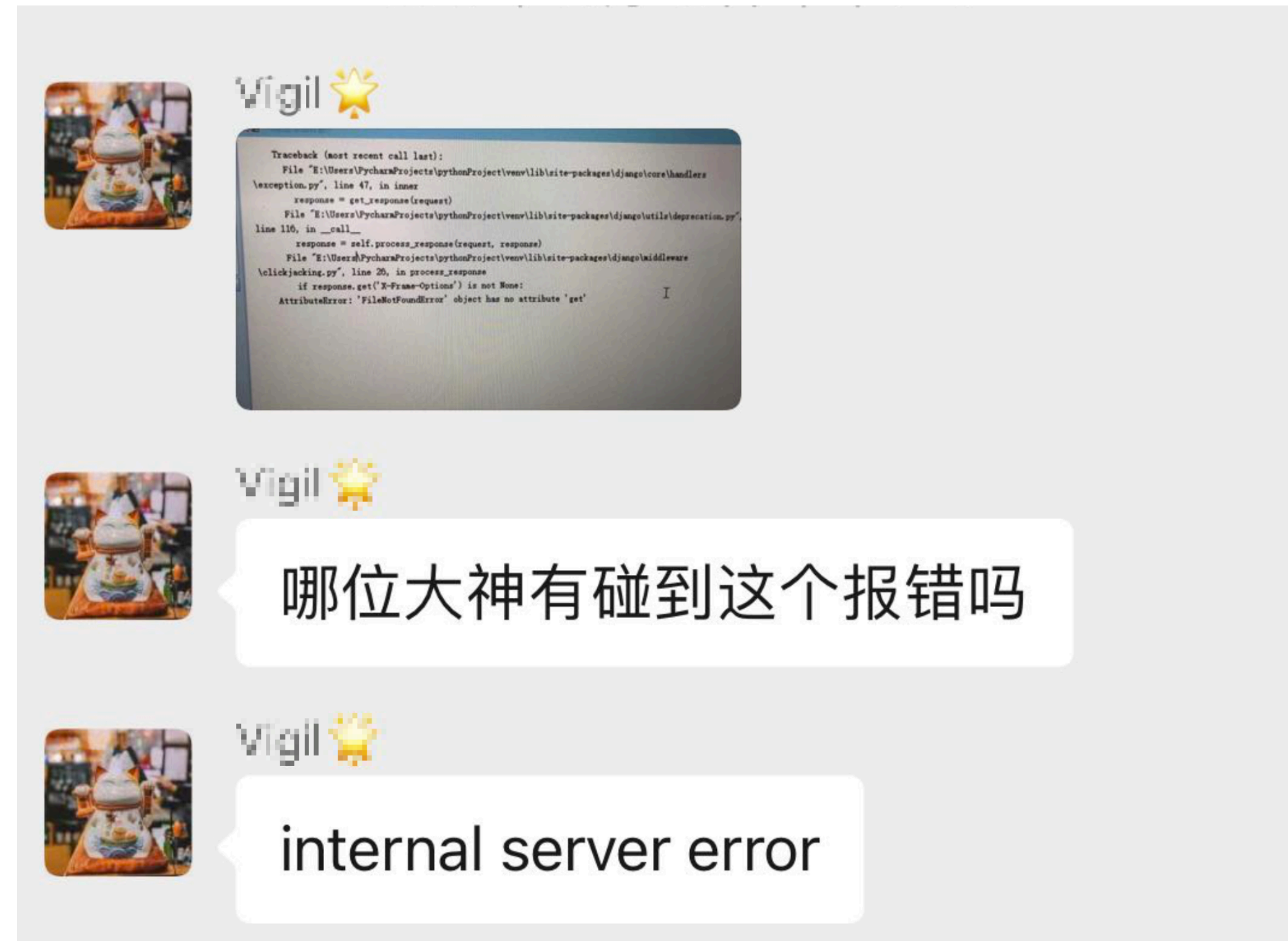
- 如何更高效地寻求帮助？
- Hacker 的成长之路
- 结束语

如何更高效地寻求帮助？

如何更高效地寻求帮助？

举几个栗子

- 怎么样创建 Django 项目？
- 为什么我的 k8s 应用部署不成功？
- Python 里面怎么样读取环境变量？



如何更高效地寻求帮助？

1. 判断问题的类别，对于这个问题，只需要快速解决，还需要系统化了解背后的原理
2. 如果是对自己非常重要的领域，先系统化了解相关知识（帮助文档，官方文档，书籍等）
3. 如果不需要系统化了解，只需要快速解决，使用 Google 搜索问题
4. 如果 Google 找不到答案，查阅帮助文档，看能否找到（特别是帮助文档中的FAQ）
5. 仍然找不到答案，社区、群里提问；开源社区、论坛；身边同事朋友，行业内的专家
6. 清晰描述你的问题：什么环境下出现的，贴出来代码，错误信息，期望的结果
7. 问题解决，或者没有解决，都对回复者做一个回复

如何更高效地寻求帮助？

- 自助解决问题的方案
 - 查阅软件/程序自带的帮助文档；
 - man 命令
 - tldr 工具
 - Google & Baidu
 - 软件调试
- 习惯 Google 带来的问题：不知道使用什么关键词进行搜索（建议减少 Google 的使用）

Hacker 的成长之路

Hacker 的成长之路

- 什么是 Hacker? 对技术有热情, 不断创造和建设的人们

https://translations.readthedocs.io/en/latest/hacker_howto.html

By Eric Steven Raymond, 2001

Hacker 的成长之路

Hacker 应有的态度

1. 这个世界充满了令人着迷的问题等着我们解决
2. 一个问题不应该被解决两次
3. 无聊和乏味的工作是罪恶
4. 崇尚自由
5. 态度不能替代能力

Hacker 的成长之路 - 如何成为 Hacker



To follow the path: (沿着这样一条道路)

- look to the master, (寻找大师,)
- follow the master, (跟随大师,)
- walk with the master, (与大师同行,)
- see through the master, (洞察大师,)
- become the master. (成为大师。)

Hacker 的成长之路 - 学习之道

- 刻意练习，做自己用的产品
 - 难度合适的目标
 - 解决实际问题 - 有反馈
 - 持续，有意识的练习
- 黑客与画家的共同点：不断反复地练习，犯错

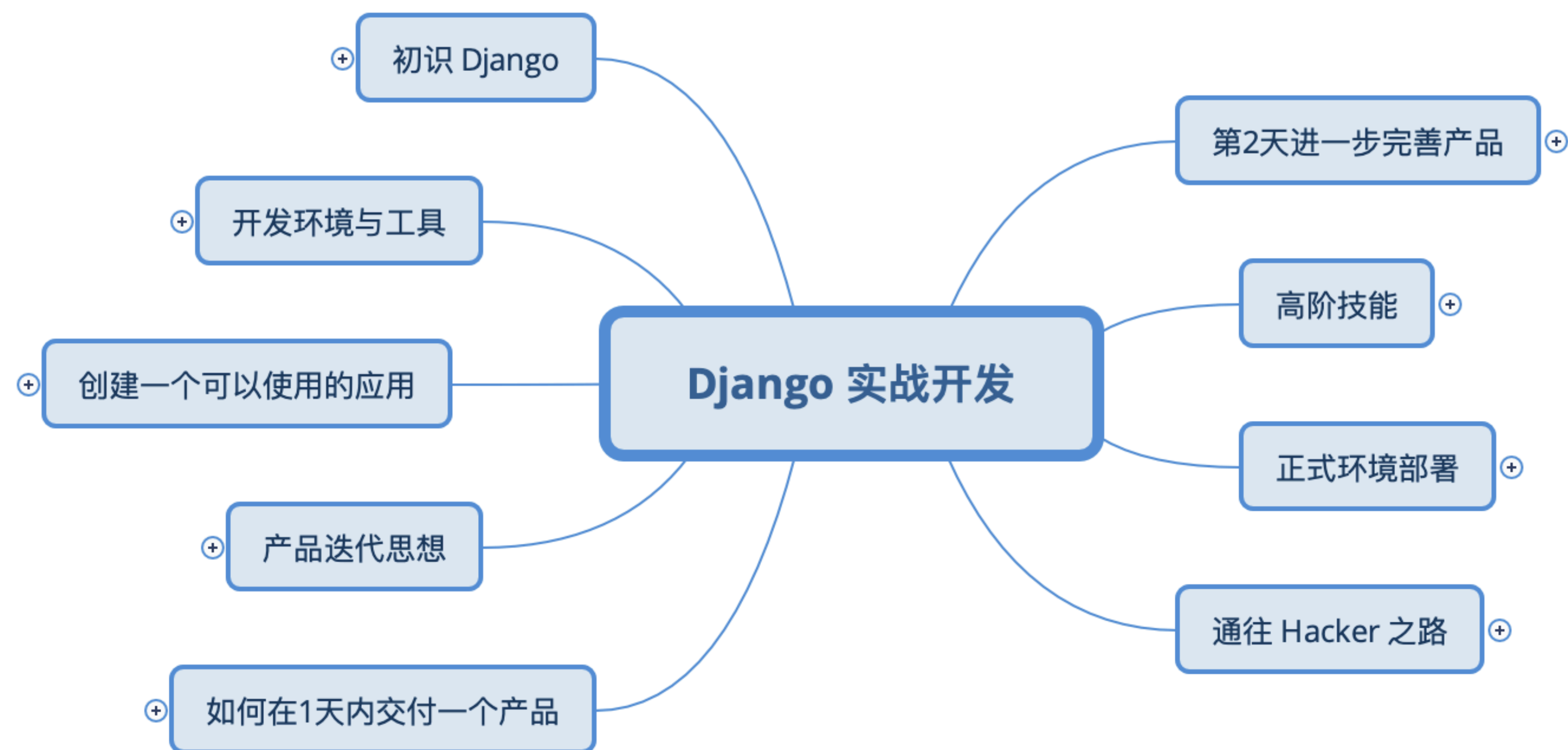
Hacker 的成长之路 - 学习之道

- 快速学习：花20%的时间学习到80%的知识
- 使用 5whys 方法，跟自己不断提问，找到本质
- 在实践中学习，做自己用的产品：
 - 不闻不若闻之，闻之不若见之，见之不若知之，知之不若行之；学至于行之而止矣
 - I hear and I forget. I see and I remember. I do and I understand
- 分享 - 通过教会别人来学习，能够清晰地讲出来，才是真正的掌握
- 学习金字塔 - 分享是最好的学习方法
- 使用不同的分享方式



结束语

课程回顾



推荐资料和学习建议

- 做自己用的产品
 - 使用 Google 搜索资料
 - 养成阅读 英文资料的习惯，阅读领域开创者的论文和书籍（可以通过 Google/Wikipedia 查到）
-
- Django 官方文档 <https://docs.djangoproject.com/en/3.1/>
 - Mozilla 文档 <https://developer.mozilla.org/zh-CN/docs/Learn/Server-side/Django>
 - 如何成为一名黑客 https://translations.readthedocs.io/en/latest/hacker_howto.html
-
- 《The Pragmatic Programmer 程序员修炼之道》
 - 《Kubernetes in Action 中文版》 - 七牛容器云团队反馈
 - 《黑客与画家》 Paul Graham
 - 《启示录- 打造用户喜爱的产品 》 Martin Cagan
 - 《The Lean Startup 精益创业》 Eric Ries



扫码试看/订阅

《 Django 快速开发实战 》视频课程