1、什么时候做技术调研？

一般来说是在做完了需求评审之后，我们已经评审完了需求，知道了要做一个什么样的系统，大概也知道了这个系统的复杂度，功能有多少，对性能的要求是什么样子的，数据量会有多大，并发量会有多大。接着我们就会做需求分析，在需求分析之后，我们从技术角度去定义清楚了这个系统要做成什么样子，包括功能长什么样子，系统运转的流程是什么样子，有哪些特殊的需求（并发性、可用性、数据量、安全性）。

接着就是要做技术了，但是这里有个问题，可能在做技术设计之前，我们会发现说，作为架构师，你可能想要用一些新的技术，来做这个项目。比如说用Spring Boot来作为整个系统的脚手架框架， 让系统的框架整合和启动效率更加高一些；我们还希望使用一些设计模式，来更好的设计我们的系统中的各个代码组件，保证我们的系统的代码的可扩展性是非常高的；因为团队越来越大，需求越来越复杂，多团队并行开发需求，特别频繁，这个时候可能需要微服务的架构来解决这个问题，可能又要去学习Spring Cloud、DevOps等知识来做微服务的架构。

所以严格意义上来说，在需求评审过后、需求分析完成后、技术设计开始前，需要进行一些技术调研，为后续的系统设计，储备需要的技术

2、如何去做这个技术调研？

一般来说，我的风格，一般做一个新的项目，我们都会尽可能去增加一些最新的技术，最流行的一些东西。互联网行业，好就好在这里，互联网行业是一个比较开放，快速变化，会承担很多的技术挑战的一个行业，传统IT行业是不太一样的。互联网行业，你一半要做的是说，开发一个系统出来，支撑你的这个互联网公司的整个业务的发展，此时你要承载的就是更多的用户，更大的访问量，高并发，高性能，高可用，安全性，快速迭代，你要负责去解决这些技术挑战。

而行业里，不断的每年都涌现出一些新的技术出来，都是为了解决各种各样的问题而产生的，此时对于互联网行业来说，一般就是快速的去吸收这些全新的技术和理念，思想，架构，基于这些最新的东西，来重构自己的架构设计，让自己的架构更好，可以解决更多的技术挑战和问题，更好的服务公司的业务发展。

传统的IT行业，我开发出来一个IT产品，IT系统，去服务于我的客户，为我的客户提供软件系统的功能，去更好的让我的客户的业务实现发展。IT系统，承担的用户都是一个公司内的一些员工。一般技术挑战很少，更加重要的是说，我每次接到一个单子，要做一个系统，如何能够以更低的成本，直接复用之前的代码，稍微改一改，就可以服务于新的客户，快速交付，用更少的人力去交付产品，获取利润。

如果要不断的使用新技术，就意味着传统的IT行业要不断的投入大量的人力物力去研究技术，不断的去重构自己之前已经做好的系统。这个对于传统IT行业来说，是没有意义的。如果10年前开发好的一套系统，10年后还能继续给新的用户来使用。成本很低，但是可以不断获取利润。我是传统IT企业的老板，我也不会干这事儿。对于传统IT企业来说，如果要用新技术，重新做这个系统，一定是有原因的。

之前大数据火了，其实很多传统的数据项目，用oracle搞定，但是人家客户不认，我要做大数据，流行，跟风啊，数据量很少，还得去部署那种几台机器的小集群，与运行hadoop，hbase，spark，一整套技术。oracle集群可能就搞定了。

互联网行业的公司，一般技术更好，用的技术更新，承担的技术挑战更大，出来的人技术能力更强，工资更高。互联网行业工作5年的人，薪资就在30~35k之间了。

传统IT行业的公司，一般技术较为陈旧，技术更新很慢，系统没什么技术挑战，出来的人技术能力较为平庸，工资较为平平。20~25k，少了10k，这都是有可能的。

每次做新的项目，必定会有技术调研的环节，因为在互联网行业中，拥抱新技术，快速使用新技术，尽快的去实践新技术，保证自己的系统架构是最好的。

spark课程，2015年出过spark课程，2013年~2014年某个时间点，spark 0.2.x版本，刚刚出来，很原始的时候，我当时还在BAT某公司里，我应该就是最早去实践和使用spark的人之一，在国内。spark的技术大量的实践，踩过大量的坑，积累很多的经验。所以在2015年推出了spark课程。

组织团队内的各个同学，每个人负责一个技术，去调研一下这个技术

3、技术调研的流程和规范

3.1 技术有哪些功能

看官网，要求我的同学看官网，不行的话就买一些中文的书，或者技术博客

写一个demo快速入门一下，对那个技术看一下初步的介绍，这个技术是干什么的

仔细看一下这个技术官网相关的官方文档，买一本书看看，里面的重点以及核心的技术都实践一下

3.2 技术的热门和流行程度

（1）开源技术的GitHub的更新频率：如果保持较好的更新频率，那是ok的；但是如果这个技术很少更新，那就不要考虑了

（2）这个技术在国内的运用怎么样：是否足够热门，在国内有哪些公司在使用这个技术

（3）这个技术在国内的中文资料是否足够多：如果中文资料足够多，就说明这个技术较为热门，有更多的资料可以参考

3.3 技术跟公司的整体技术环境是否适合整合

比如说你如果用Spring Cloud，假如说你在阿里，那就肯定不合适了，其他一些大公司。因为那些大公司都是自己研发的微服务的整体架构，都是用自己的东西的。如果你冷不丁搞一个跟公司开发好的基础设施架构重复的东西，那肯定是不合适的。

3.4 看一下这个技术的其他的方方面面

（1）易用性：是否足够简单上手使用

（2）稳定性：是否足够稳定可以使用，bug较少

（3）功能性：功能是否足够完善和强大

（4）与其他技术的整合性：能否良好的跟其他相关的技术进行整合

（5）性能：类似Spring Boot框架，就没什么好说的了；redis，memcached，mongodb，elasticsearch。适当自己在测试环境下，做一些简单的性能测试

（6）并发能力（吞吐量）：每秒可以承载多少请求，做一些简单的压力测试

（7）可用性：这个技术是否提供了高可用部署的机制，比如redis就可以基于redis sentinal哨兵机制实现高可用架构，做一些简单的故障演练

（8）伸缩性：这个技术如果需要扩容的话，是否可以扩容

（9）扩展性：是否可以对这个技术进行一定程度上的扩展，比如说Spring Boot，就可以支持我们自己写spring-boot-starter-x的项目，支持我们自己将Spring Boot跟其他技术进行整合

（10）安全性：这个技术是否足够安全，没有过多的安全漏洞，struts2框架，安全漏洞频频爆出

3.5 技术跟同类技术的横向对比调研

Spring Cloud -> Dubbo -> DubboX

Thymeleaf -> freemarker -> velocity

Redis -> Memcached

Elasticsearch -> Solor

对比哪些方面呢？就是上面说的那些方面，产出一份横向对比的表格

3.6 选择这个技术的理由

调研完了，了解清楚了这个技术了，也跟其他技术对比了优势和劣势了

4、编写技术的培训手册

一般是用ppt的形式来变现的，主要是在决定选择使用一个新技术之后，需要将这个技术编写成使用手册，引入团队中，对团队进行培训

（1）介绍清楚这个技术的基本原理

（2）介绍清楚这个技术的核心功能

（3）介绍清楚这个技术的一些调研情况

（4）给出这个技术使用的demo，不是特别简单的demo，而是将这个技术引入我们的公司环境中，如何跟公司现有的技术栈整合，如何在公司的项目里去使用这个技术

5、对团队里的成员进行培训

每个人都要学习清楚这个技术是怎么回事儿，在项目中来使用的时候，每个人都能得心应手

6、就可以将这个新技术用到我们的项目的技术设计中去了

7、课程里，就会对每个技术的学习，都模拟成是，你现在要为你的公司引入这门技术，然后你学习就是在做调研。完全模拟公司里真实的技术调研和引入的环境和流程以及场景。Spring Boot和设计模式。