



下载APP



25 | 节点四：架构规划之如何进行需求确认？

2022-03-29 郭东白

《郭东白的架构课》

课程介绍 >

**讲述：郭东白**

时长 23:08 大小 21.19M



你好，我是郭东白。上节课我们讲了架构规划这个环节的第一个部分，也就是统一语义。那么这节课我们就来讲第二个部分——需求确认。

需求确认与统一语义的过程是密不可分的。需求确认是在统一语义赋能之下进行的，所以两者并不是先后顺序的关系。

领资料

通过上节课的学习我们知道了，统一语义的主要场景和目的，就是对目标进行无损地分解和传递，以及理解、表达和传递参与者的诉求。那么需求确认，则是在统一的语义之下将这种诉求继续无损地分解成研发人员的执行任务的过程。



所以在学习这节课的过程中，你仍然需要深度理解并随时应用我们前两节课讲的统一语义的内容。

需求确认前的准备工作

在开始需求确认之前，你需要梳理如下内容。

首先要从产品定位的角度来梳理。一般来说，你应该从产品经理那里拿到三个信息。

第一是**客户**，也就是为你整个架构活动买单的人。在公司内部，站在客户身后的受益人一般就是这个架构活动的赞助者。不过也有的赞助者是站在用户身后的。

第二是**用户**，他们是最终产出的软件产品的使用者，也是你创造的用户价值的受益者。所以用户是指整个架构活动设计需要考虑到的所有内外部的用户角色。既包括常规的用户角色，比如我们上节课提到的商家、发行商和平台运营。还包括代理角色，比如某个 SaaS 服务，也可能是某个银行金融授信的自动化审批的角色。

第三是**核心场景**，比如客户 / 用户在这个架构活动中的核心诉求是什么？他们因为什么价值而买单？这些需求将会在哪些场景中出现？

其次要从执行者定位的角度来梳理。一般来说，你需要从技术团队的管理者那里拿到以下信息。

第一是**执行团队**，也就是架构活动的相关执行团队。你需要确认他们将会为哪些用户角色服务，可以为这些角色创造什么价值。

第二是**执行域划分**，也就是每个执行团队所负责的领域。你特别需要关注的是执行领域之间互相重叠的部分，以及没有被任何执行领域覆盖到的部分。

第三是**需求承接方**。除了架构活动中已知的执行者外，整个公司内部，谁应该、谁愿意、谁有能力或者谁有带宽承接某个需求呢？

最后是取舍规则，你应该从决策者和赞助者那里来确认如下信息。

第一是**需求优先级的决策信条**，比如不同需求方的优先级，是按什么规则来排序的？

第二是**必保需求**。哪些需求是必须完成的？需要注意的是，这些并不是单个角色视角上的必保需求，而是由参与方和决策者达成共识的必保需求。

第三是隐含的技术需求。架构活动肯定会为公司带来新的外部适应性，那么外部适应性将会在哪里创新，怎么度量，谁来承接，等等。这些都是你这个架构师要必须完成的工作。最终，这个隐含需求也要被显现地表达出来，也需要跟其他需求一样，有执行者，接受取舍。

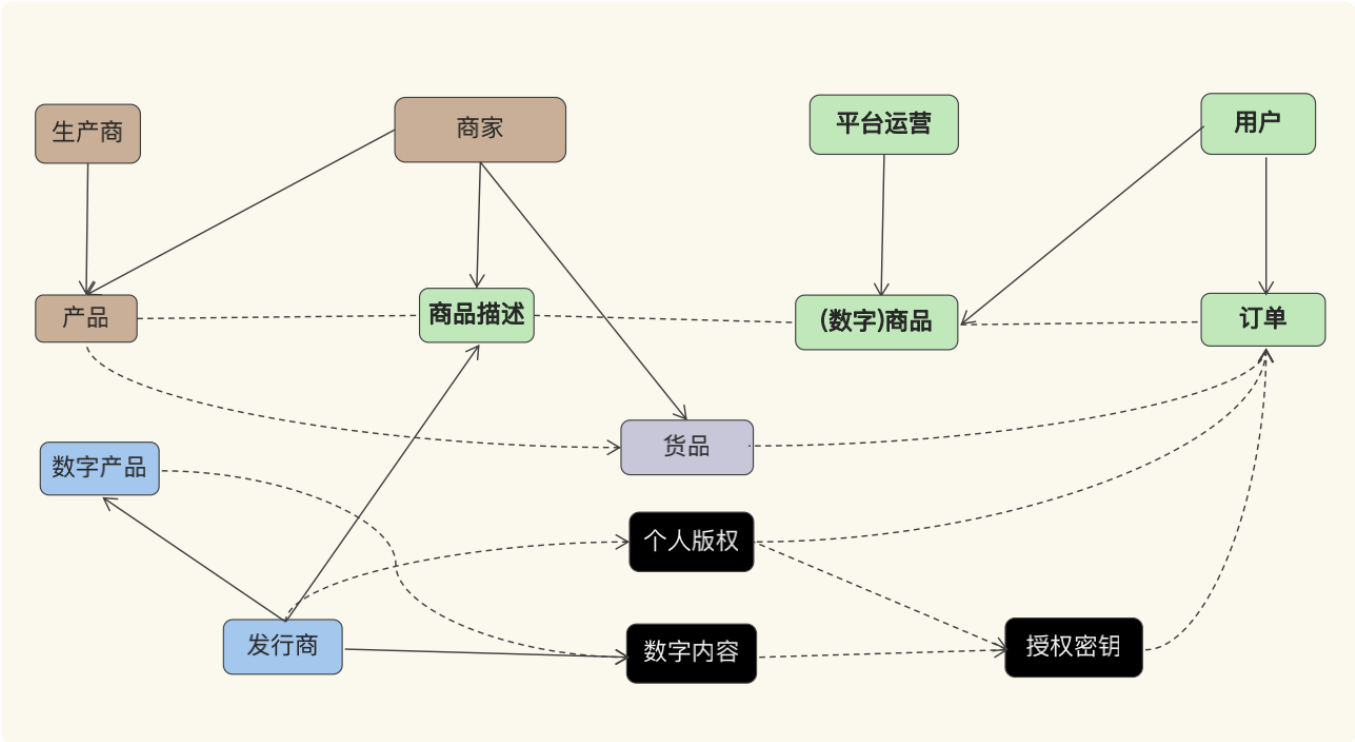
问题域划分

需求确认的过程，也是从问题域到执行领域的映射过程。一个大的架构活动会有几千条需求，有好几百人参与开发。如果你把需求一条一条都映射到执行领域中，那将是非常繁琐和低效的过程。

应该怎么做呢？在最开始的时候，我们就要**把问题域定义出来**。多数需求可以由产品经理或者 PMO 整理到不同的问题域中。而你这个架构师需要做的，就是把问题域映射到执行域上。

这样的话，你并不是把上千条需求分配给几百个研发，而是把几十个问题域映射到几十个执行域中。在这之后，就是逐层分解了。只有在边界模糊的场景下，才需要你这个架构师的介入。

其中比较核心的是问题域的定义。我们还延用上节课的案例。如下图所示，是一种问题域的划分办法：



不同的颜色代表不同问题域的划分；

绿色，代表数字和实物商品整合之后的商品域；

古铜色，代表实物商家域；

蓝色，代表发行商域；

紫色，代表货品和履约域；

而黑色的部分，则属于数字商品域。

多数时候，问题域和执行域都是同构的。也就是说，如果你把这张图复制一份，就可以在
每个问题域上标注出相应执行团队的名字。当然也有粒度不同的情况存在：

在一个合并后的公司里，你会看到一对多的情况；

在一个收缩的公司里面，你会看到多对一的情况；

在一个架构和管理混乱的公司里，你也能看到多对多的情况。

如果沿用我们上节课的例子，假设你整合了两个不同的部门。那么整合之后可能会发现某
些问题域有多个执行域，也就是踩脚的情况。当然，你也可能发现某些问题域没有任何已
知的执行域，也就是新建或者烂尾的情况。

这都是整个架构活动风险多发的领域，你必须及早发现，及早处置。

这里我也要强调一些，问题域和执行域的划分，不能靠你这个架构师的想象，而必须在多
方参与下对事实形成一个准确的描述。

事实上，你会发现问题域和执行域划分的过程，并没有我们描述得那么简单。从形成统一
的问题域模型，再到问题域划分，最后到执行域的划分，一次迭代肯定没法完成。

因为大家的语境本来就不一样，不可能给你足够自治的答案。而且，一个实体往往不会正
好映射到一个执行域，而是会因为各种历史原因形成错综复杂的映射关系。

那么接下来，我就来讲讲从项目目标到需求的映射过程。

从项目目标到需求的映射过程

评估需求

在这个环节，你需要从架构活动的整体目标出发，确认需求存在的必要性。很多时候，尤其是大的项目，需求方经常会夹带私货。虽然他们并没有什么恶意，但是这些附加的需求，不仅会消耗研发资源，还会增加项目复杂度和规划难度。而最坏的情况，就是附加需求会引入风险，导致整个项目的失败。

所以这个时候，作为架构师，你需要准确区分**最小必要的需求**和**无关紧要的需求**。只有那些与项目目标形成因果关系的强依赖需求，才是属于架构活动的需求。

这个梳理过程，之所以要放在问题域和执行域划分之后，是因为需求属于问题域的范畴，而需求的执行则属于执行域的范畴。如果没有领域划分，那么你还需要砍掉附加需求。估计你这个架构师很快就混不下去了。

要知道，砍需求是个非常得罪人的事情。你做这件事情必须要有个同盟，也就是与需求对应的、问题域映射到执行域的负责人。因为在砍掉不必要的需求上，你们两个人的利益是一致的。你为了整个架构活动的成功，他则是为了控制自己领域的风险。你可以站出来表达比较客观的立场，而他则可以帮助你证实这个立场的合理性。

这个过程与确认目标的过程类似，你的关注点应该放在如下五个方面。

第一是**需求的必要性**。这个需求绝对有必要吗？与整体目标是因果关系吗？无论这个需求的价值有多大，只要它不是架构活动的必要条件，那就应该分开考虑，最好完全隔离在架构活动之外。架构规划最忌好大喜功。

第二是**需求的正确性**。需求是否和架构目标相匹配？想要得到赞助者所期望的商业价值，那你这个需求目标的数量级应该是多少？如果一个需求的预期目标，比赞助者所需要的值还要小一个数量级，那这个目标就是不正确的。

事实上，需求不正确的情况在架构活动中频繁发生。表面上看，可能是执行的队友不给力。但更真实的原因是你这个架构师不给力，没有及早发现规划的软肋。

第三是**需求合理性**。在当前交付时间的约束下，需求的交付时间和质量要求是否合理？是否会出现研发团队动作和设计完全变形的情况？

第四是**需求的可达性**。当前的时间要求和资源投入，能产出质量上可以接受的实施方案吗？实施方案是否高于质量底线？如果某个需求需要多个团队协同，那么我们能留出足够的时间，让团队处理集成中出现的各种问题吗？

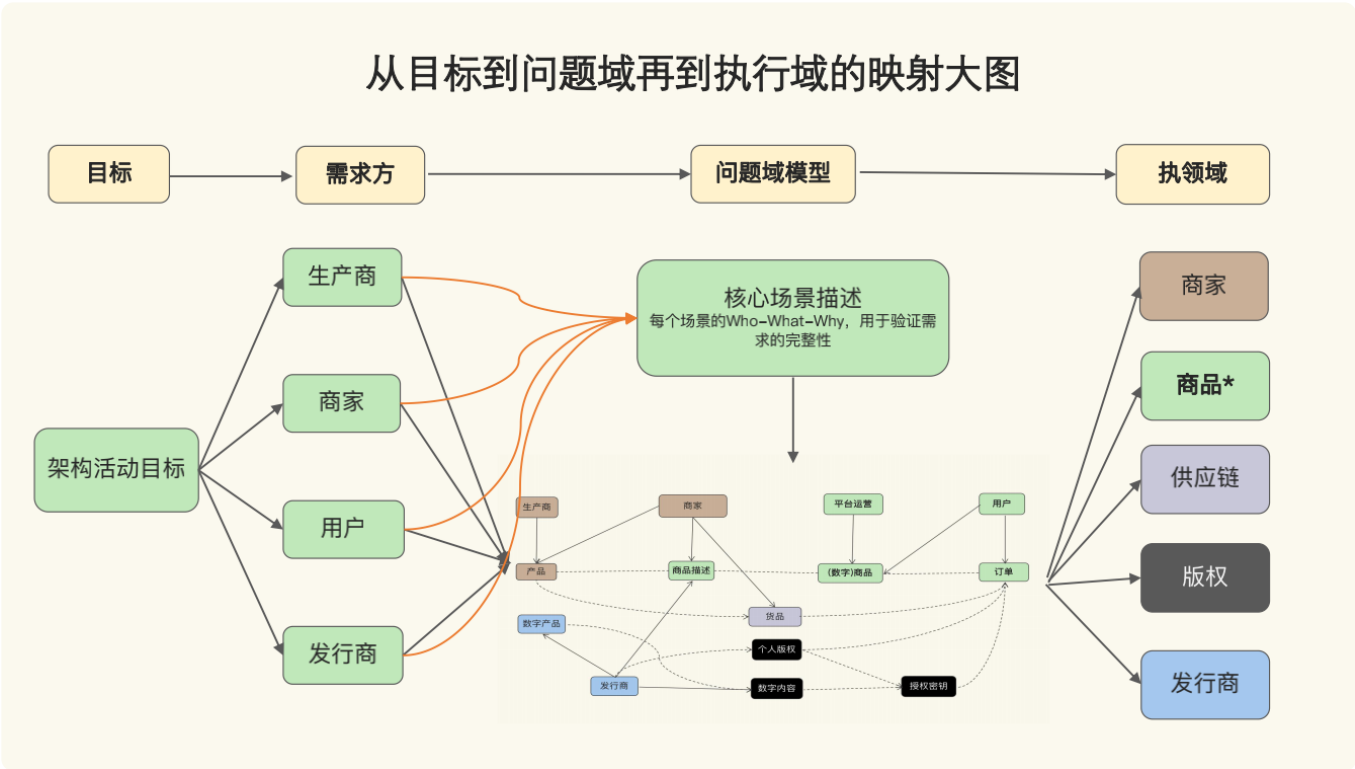
第五是**需求的承接方**。这个需求有且仅有一个团队承接吗？是他们应该承接的吗？他们愿意吗？能力够吗？有研发带宽吗？

一个架构活动中最终要承接的需求，在这五个问题的答案上必须是肯定的。而这个梳理需求的过程，其实也是执行风险的梳理过程。

完成这个梳理过程后，你应该对需求与架构目标之间的因果联系有了信心。那么接下来，我们就需要形成一张完整的映射关系表。

需求到承接的映射

当梳理完所有需求后，我们还需要将那些满足最小必要条件的需求，整理成一张从需求到承接的关系映射表。如下图所示：



你需要跟每个执行团队，逐个确认他们是否是某个需求的承接方，这是你这个架构师在架构规划前必须确认的大图。因为你需要有足够的信心，来保证从目标到需求、从需求到核心场景、从核心场景到问题域、从问题域到执行域的映射，可以被无损地表达且没有冗余。

这个映射是对事实的反映。你会发现有些需求有好几个承接方，而有些需求却没有一个承接方。有些高优先级需求的对应执行者，没有任何研发带宽；而有些低优先级需求的对应执行者，却有充足的带宽。

这些冲突都是我们要解决的问题。而对于那些高优先级需求，也就是从需求到承接都满足一对一映射。这个时候，它们领域内的规划设计就可以启动啦！

问题域和执行域中的冲突

在统一语义的过程中，你会发现不同角色在不同的语境中隐藏了很多冲突。日常工作时这些冲突可能并不明显，因为大家都在自己的隔离语境中与几个团队进行了小范围的合作。直到我们把不同语境中的概念，拿到一个统一的语境中来抢夺有限资源的时候，这些冲突才会全面爆发。

其实越早暴露这些冲突，对于架构活动来说越有利。毕竟这些冲突是客观存在的，避免不了。我梳理了架构活动中最常见的几种冲突。

第一，优先级的冲突。这是互联网企业最普遍的冲突，在资源有限的情况下，每个需求方都从自己的视角出发，期望得到最大的支持，导致各方在需求优先级上无法达成一致。

比如营销团队，由行业运营和大促运营两个团队共享，而每个运营团队都认为自己的需求优先级更高。这就是多个依赖方共享一个开发团队导致的。

再比如，弱势角色的高优先级需求很容易被忽视。像合规部门一般没有专职的产品经理，但由于监管形式的变化，合规类需求的优先级其实也很高。不过掌握研发资源的业务团队，却不太愿意让研发人员去开发这类需求。

第二，定位的冲突。不同角色之间天然就是互相制约的，这并不是特例，而是企业中普遍存在的现象。因而一个企业内部必须有某种形式的监督和制约机制，以确保整个企业的决

策有完整且相对平衡的视角，而不仅仅是单一视角中的最优。需要说明的是，这种冲突独立于架构活动，且长期存在。

举个例子，商家运营的定位是商家增长，那么这个角色就要吸引尽可能多的商家到平台上。而网规团队的定位是减少平台风险，那么这个角色就要尽可能地打击作弊和劣质商家。

从算法的视角来看，前者的定位是要最大化召回，后者的定位则是最大化精度。这就是一对矛盾。

第三，**团队和个人的冲突**。团队或者个人之间也会有敌对情绪。有的公司喜欢赛马，针对同一个垂直领域，会让几个部门用不同的商业模式在市场上竞争。或者针对同一个场景，会用多个技术和算法来实现。

这些定位大致重叠的团队，往往会因为内部竞争形成不小的摩擦。久而久之，团队的关系就会变得不和睦。也有的人日常做事缺乏底线，得罪了同事。那么对立双方可能会因为同时参与了一个架构活动，而将个人冲突引入到其中。

第四，**边界冲突**。多个需求方或者执行团队负责的领域边界不够清晰，不确定到底“谁说了算”。这种情况在大公司里尤为常见。语境的差异性也是这种冲突的常见诱因。

边界冲突主要源自以下三个方面：

1. 不同垂直执行域在定位上本身就有重叠。比如商家团队和商品团队之间，交易团队和资金团队之间，流量团队和导购团队之间。
2. 水平分层上的模糊性。比如前端团队和后端团队之间的分层，业务团队和财务团队之间的分层。
3. 因技术进步带来的执行域边界的迁移。比如前端转全栈工程师，导致前端的职能范围拓展到后端去；业务中台化，导致业务线的研发任务迁移到了一个共享的中台里去；前端低代码化，导致之前由前端执行的任务，变成了由产品或者后端工程师来完成。

第五，**问题域到执行域的映射关系的冲突**。需求方到执行者领域的映射关系没有形成共识，导致几个团队互相争夺地盘。

比如上面那张图中的商品，会与数字商品整合成一个商品执行域，从而形成多个业务团队映射到一个执行域的情况。

同样，如果产品团队已经完成整合，但是技术团队的整合还没有完成，那么订单域就可能形成一对多的映射情况。这种一对零、一对多、多对零、多对一、多对多的状态，都会造成设计和执行过程的混乱，必须在这个节点解决掉。

第六，生存空间的冲突。这是映射冲突中一对多和多对多的情形，在大公司里比较常见，所以值得单独列出。

比如一个部门里有多个做网关的团队，平时大家都相安无事。一旦公司要把几个网关合并成一个，那么生存空间的冲突就不可避免了。

第七，决策权的冲突。在规划和执行决策上，有的角色非常强势，导致本来属于架构师的决策权，却被某个领域的领导抢夺了，从而形成架构冲突的热点。

比如某个大厂长期存在这样的现象：对某个领域有控制权的共享技术团队，涉及到他们管辖的领域，需求必须由他们承接，设计也必须由他们决定，谁都不能替代或者更改。这就导致任何由这个团队参与的架构活动，一旦涉及这个团队的需求，设计就全都变形了。

所以统一语义的关键作用，就是让你及早看到这些冲突，并及时解决，否则架构活动将会面临很大的不确定性。

从问题域到执行域的领域边界划分

从刚才的分析就可以看到，在架构活动中你可能面临如下四个棘手的问题：

1. 多个团队争夺有限的资源；
2. 多个团队争夺生存空间；
3. 无法调和的个人与团队之间的矛盾；
4. 短期内不合理的组织和决策结构。

我的建议如下：

1. 对于有限资源的争夺，如果没办法缩减交付项，那么必须在这个时间和赞助方、决策方、参与方讲明白。要么调整质量预期，要么调整上线时间。
2. 对于生存空间的争夺，务必请决策者迅速裁决。因为这是没办法调和的冲突。
3. 对于个人与团队之间的矛盾，我认为只有躲开才是上策。如果活动还没开始就没办法相处，可以预见，随着项目压力的增加，矛盾必然会放大。我建议还是请决策者进行处理。如果没办法处理，也要请决策者提前指定一位裁决人。一旦出了问题，你能迅速找到裁决人进行拍板。
4. 对于组织和决策结构的问题，可以通过我们前面讲的搭建架构环境、设立决策信条等方法予以解决。不过我觉得最好还是请相关参与者和决策者一起，来制定和确认一些决策信条。

总的来说，执行域划分是个压力非常大的环节。你这个架构师也必须保持开放心态。在这个过程中，你需要充分表达，大胆建议。不过也要注意，你并没有决策权。你提供的仅仅是技术视角，而不是整个企业的视角。

换句话说，你与决策者所拿到的信息是不对称的。大多数时候，你看不懂，只是因为你的视角有限。换句话说，你与其他参与者一样，各自的视角都是片面的。那么人才培养、团队稳定性、成本控制、商业竞争、监管，甚至是国家间的关系，都可以成为边界划分的理由。你可以不认同决策者的判断，但千万不要下意识地阴谋化别人。这个世界没那么多阴谋，多数时候大家都是见招拆招。

当然，如果可能的话，你应该试图理解每个参与者的视角，并对最终的执行域划分给出一个合理的解释。一个让参与者，如果能主动接受领域划分，那将远远好于自上而下的强制性的边界划分。

| 问题域/执行域 | 0 无执行方 | 1 有且只有一个执行方 | N 多个执行方 |
|---------|---------|----------------------------|------------|
| 0 | — | 团队待重新定位 | 多个团队待重新定位 |
| 1 | 待安排承接团队 | OK | 混乱状态，待领域划分 |
| N | 待安排承接团队 | OK，但如果执行域抽象不合理，可能出现未来的设计问题 | 混乱状态，待领域划分 |

如上图所示，领域映射的最优解是所有问题域到执行域的映射都是无歧义的，也就是图中的绿色部分。否则你会带着巨大的风险进入到下一个环节中，而且你还没有任何决策权和调解手段。

可以说，这个执行域划分的过程，几乎决定了整个架构活动的成败。所以在进入到下一个更细的任务边界划分之前，必须解决图表中展示的那些遗留问题（粉色部分），避免给自己埋下一颗定时核弹。

完成需求确认

如果你将所有问题域和执行域都梳理完了，那么这个需求确认的过程也就初步完成了。

小结

先回顾一下我们上节课讲的统一语义的过程。统一语义可以让所有用户角色都能完整表达自己的需求，并且这些需求与目标保持一致，互相兼容。

这个统一语义的过程，会暴露公司内部很多团队和职能间已经客观存在的冲突。不过这是好事。如果这时候不暴露，等架构活动进行到一半时再暴露，项目就无药可救了。这就是统一语义的王道：把真相呈现给大家。问题的真相再可怕，都有解决的办法。

在这两节课里，我们花了很长时间去分析和寻找语义、语境、问题域定义、执行域定义，以及映射关系的分歧和问题。我们目标没有变，一直是为了在那个真正值得我们每个参与

者去解决的问题上达成共识。只不过今天我们更进了一步，在问题的分解上也达成了共识。

不过你发现没有，我们在这两节课并没有提及任何架构方案的选择，没有微服务，没有分层架构，也没有统一数据模型。

事实上，我在这个过程中一直坚持我们讲过的第二条生存法则：尊重人性，从用户思维出发，先看清楚问题。对应到这节课的上下文中，意味着我们在进入解决方案设计之前，要让所有参与方先坐下来，放弃一切技术执念，先在“我们要解决什么问题”和“怎么分解问题域”上达成共识。这才是彻底的问题域建模。

思考题

三个作业，任选一个。

1. 我们这节课强调了要提前暴露问题。但也有人认为暴露组织中本来就存在的问题，其实是没事找事。一个大型的架构活动已经有很多挑战了，没必要暴露更多的问题而人为制造困难。在大规模的软件架构中，你经历过这两种做法吗？结果是什么？你的体会是什么？
2. 问题域和执行域不需要同构，因为执行域可以是多个问题域的抽象。你见到过比较好的抽象是什么？
3. 我们在讲执行域划分时提到四种棘手的情形。你有经历过其中任何一种吗？有没有比较巧妙的解决方法？

“

执行域划分是个压力非常大的环节。你作为架构师，必须要保持开放心态，充分表达，大胆建议。不过也要有充分的心理准备，因为决策权不在你手里，而在企业的决策者。



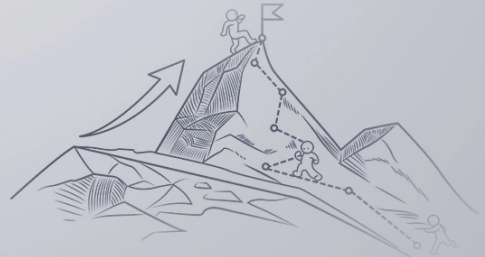
郭东白的架构课

郭东白 / 车好多集团 CTO、前阿里 P10
浙大兼职教授和博导

”



识别二维码
免费试读



如果这节课对你有帮助，欢迎你把课程转发给你的同事或朋友。顺便打个小广告，我刚开了个人抖音号，我会定期发表一些比较新、但是不一定那么成熟的观点。欢迎在抖音上搜索“郭东白”并关注，也欢迎你的批评指正。我们下节课再见！

 生成海报并分享 赞 2  提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 24 | 节点四：如何减少语义上的分歧？

下一篇 26 | 节点四：任务边界划分应该遵循哪些信条？

更多学习推荐

《架构实战营》

跟着阿里 P9
系统提升你的架构能力

立抢课程大额优惠 

李运华
前阿里资深技术专家 (P9)



精选留言 (4)

 写留言

罗均 - Jun

2022-03-29

古人云：“不为良相，愿为良医。”东白老师真是现代科技类企业的“良相良医”，所总结的技术活动中的种种问题，广泛、精确且深入人心！任何企业的任何技术活动，必然存在或多或少的问题，而老师的总结归纳，正如医书一样可以指导学生们对症下药。

听完老师的课程，让学生回忆起Peter Drucker的灵魂拷问：“Who are our customers? W...

展开 ∨

作者回复：惭愧，做不到你说的这个境界。不过能帮助到你很高兴！



👍 5



术子米德

2022-03-30



* 📖：需求确认。

* 😊：每次看到这个词，我就有点想笑，不是因为我能够做好，而是每次都觉得，需求确认是件真戏假做的尴尬演出，至少在我身上经常发生。我个人比较喜欢需求挖掘，尤其喜欢把需求挖掘到，如何还能赚到未来的钱。可是确认需求的时候，经常连最基本的需...
展开 ∨



👍 2



spark

2022-03-29

郭老师，take away~~~做事的思维方式需要完整，缺失的思维维度必然导致局部视角。您在上节课分享，架构师也需要掌握局部视角的细节和痛点，才能保证全局思考和执行的到位~~~
有人分享需求的逻辑，“业务需求(愿景与目标)、客用户需求(问题)、设计方案(方案)、产品需求(功能特性)”，就包括问题域和执行域~~~...
展开 ∨

作者回复：“同样的实力和业务背景，就看谁的洞察力和做减法的本事大~~~” 非常赞同！！！！



👍 1



沈子砚

2022-04-08

这节课强调了要提前暴露问题。但也有人认为暴露组织中本来就存在的问题，其实是没事找事。一个大型的架构活动已经有很多挑战了，没必要暴露更多的问题而人为制造困难。有一次，在大规模的软件架构中，我提前暴露风险，被喷了，我不知道我错在哪里了，罗均的评论给大家指导了一个新的工作方法
展开 ∨



👍