**Q1: Schedule the design reviewmeeting (1)**

1 - We can take ourbest guess at an estimate on our own

2 - We should work fora couple of days to gauge our progress, and then complete our estimate fromthere

4 - We should consulta coworker who has more relevant experience on this type of task

3 - We should conductour own investigation utilizing online research materials and internaldocumentation

5 - Let's ask ourmanager how we should go about developing an estimate

**Q2: Schedule thedesign review meeting (2)**

3 - Ask all parties toidentify a back-up person who could meet during a designated time

3 - See if there is abackup person on the Localization Team that can meet

5 - Set-up videoconferencing to include all POC's regardless of their physical location

1 - Agree to postponethe design review for two weeks when all parties have more availability

2 - Discuss the designreview over email

4 - Agree to schedulethe meeting at Xavier's location an hour away

**Q3: Response toRavi (1)**

3 - We should miss theconference and increase the timeline to four weeks because we have four weeksof work

4 - Take a day toinvestigate whether adding additional resources would allow us to meet theoriginal timeline, and re-evaluate afterwards

1 - Tell theLocalization team if can't be done in the timeline, so we should go ahead withthe US launch and delay the global launch even though it means adding anadditional week of effort to the four week estimate

5 - Take two weeks tocreate a prototype of the feature to demo at the conference, then take theadditional two weeks needed to fully complete the feature

2 - We can still hitthe two week deadline without any changes by working harder and putting inovertime

**Q4: Response toRavi (2)**

Begin yourinvestigation using the old error logs, but tell Ravi he will need to run thenew logs if the old logs aren't useful

**Q5: Response toAaron and Jacob (1)**

Can you tell me moreabout what you're talking about?

**Q6: Response toAaron and Jacob (2)**

You said we have aninternal database of both digital and physical books. How did we get thephysical book data if the Book Database API doesn't give it to us?

-baidu 1point3acres

**Q7: Response toAaron and Jacob (3)**

I recommend you gowith Jacob's solution. We should miss the deadline to build our own service andmeet all the requirements.

**Q8: Roadmap**

Since you know moreabout the programming language than anyone else, you revise the estimate forporting to Java.

**Q9: Response toNadia**

What were the internaltest case results?

**Q10: Most likelycause of German language issue**

Site is using proxyserver location to determine displayed language

. From 1point 3acres bbs

**Q11: Most likelycause of invalid recommendation issue**

Database field storingusername is too short

**Q12: Log traceinvestivation success**

5 - Increase timealotted for testing in overall lifecycle

5 - Update automatedend-to-end tests to include broader data coverage

3 - Write more unittests to include edge cases

3 - Have team membersperform more manual testing before checking code in

1 - Increase the sizeof QA team

1 - Have more usertesting in beta phase

**Q13: Response formeeting the deadline**

2 - Work on theproject on your own, putting in extra effort to finish on time

3 - Work on theproject on your own until Priya is available, then continue to work on ittogether

4 - Work on theproject with Ben, being sure to watch his work closely because of his lack ofexperience

5 - Tell your manageryou will not be able to complete the project in the time avaliable

1 - Cut features fromthe product so you will be able to meet the two week deadline

3 - Start working onthe project right away with Ben. Then ask Priya to contribute what she can whenshe is avaliable

**Q14: Response forcompleting this work on time**

4 - Work with theCustomer Incentives Team to identify the critical features that they need bythe deadline, and focus on those

2 - Push the timelineback another week to ensure there is enough time for all work to be completedaccurately

3 - Ask your wholeteam for help, explaining the urgency that another team is blocked

5 - Ask your managerfor help in determining the best approach to meet the new deadline

1 - Put in extra hoursyourself to make sure everything gets done on time

**Q15: Upgrade**

4 - We should notperform this upgrade at this point in time. We promised the Retail Website Teamwe would have their new features complete by the proposed deadline. Let'spostpone the upgrade to another time

2 - We should notperform the upgrade because it will not have a significant impact on the RetailWebsite Team's experience. We should focus on the Retail Website Team'srequests

3 - We should notperform this upgrade at this point in time. Our top focus is meeting our agreedupon commitment with the Retail Website Team, so we should finish that first.We can focus on the upgrade afterwards by pushing our deadlines for some of ourother projects

1 - I think we shouldperform the upgrade. The right thing to do is push back on the Retail WebsiteTeam because it will keep our team from having to do the same work twice

5 - I think we shouldperform the upgrade. As a compromise, we can include the gift recommendationfeature the Retail Website Team wants by the deadline and then complete theupgrade. We can finish the seasonal-based gift recommendations feature afterthe deadline

2 - I think we shouldperform the upgrade. The right thing to do is push back on the Retail WebsiteTeam because it will allow us to more efficiently serve the customer and thecustomer will be helped in the long run.

**Q16: New productdesign**

2 - A, C, D, G

1 - A, C, D, G, H

4 - A, B, D

3 - A, C, F

5 - A, D, F

3 - F, G

**Q17: ？**

?

**Q18: Problem withProduct.wasPurchasedByUser()**

It has performanceissue

**Q19: Most effectiveway of improving ShoppingCart()**

Change the design ofShoppingCart by removing ShoppingCart user and making shopping cart a propertyof User instead

**Q20: Five testswithin ShoppingCartTest()**

Fail - Test1

Pass - Test2

Fail - Test3

Pass - Test4

Fail - Test5

**Q21: Ask Jacob aquestion**

3 - Do any otherprojects depend on fixing this problem?

5 - How many customersis this affecting?

5 - How does thisaffect customers

4 - Are we receivingcomplaints from customers?

2 - How long will ittake to solve this problem?

1 - If I help you withthis problem, will you help me finish my work today?

Work simulation(原则有先后顺序)  
目前两大做题中最重要原则：  
1.requirement排在第一，deadline第二。  
2.有manager出现的选项无脑选manager，manager就是一个组的地头蛇。  
  
[Amazon](http://www.amazon.com/b?_encoding=UTF8&tag=1p3a-guanlian-20&linkCode=ur2&linkId=89c11e2c5b86155c5422f19cca1e9880&camp=1789&creative=9325&node=5)9条主要原则  
原则1：客户是上帝，requirement优先，任何影响上帝的事情都不能干，  
        如某个requirement影响了上帝的体验，   
        你就是死键盘上也不能砍了，宁愿miss deadline  
原则2：为长远考虑，即客户几年之后可能会出现的需求也要考虑到，  
        不会为了交付短期的deadline，-baidu 1point3acres  
        而牺牲长期的价值。（比如 global api  和 local api）  
原则3：最高标准，“最高”对应上面的“长远”。  
原则4：一般情况，能请示manager就请示manager，manager一般不会出错  
原则5：速度很重要，决策和行动都可以改变，因此不需要进行过于广泛的推敲  
        ，但提倡在深思熟虑下进行冒险。  
原则6：不需要一定要坚持“非我发明”，需求帮助也是可以的，四处寻找创意  
        ，并且接受长期被误导的可能  
原则7：敢于承担责任，任劳任怨，比如领导说谁会java，你会你就跳出来说我会  
原则8：对问题刨根问底，探究细节  
原则9：服从大局（团队比个人重要）  
  
打分不是关键，排序才是关键。  
大部分情况下其实并没有deadline 和 requirement谁更好，更多还是在  
这个组合中你对ddl 和 requirement整体的权衡。  
  
每个选项可以评1~5分，most effective 是5，然后1是least effective  
刚开始让你看一些介绍amazon工作环境的视频  
1.上来给一段video，场景是项目的晨会，就是把team正在推进的项目描述一下，  
期间会有多个项目和你有关系，后面会遇到  
2.进入工作界面，可以看到接受到邮件，接收到的instant message  
3.进入工作状态。会有同事给你发邮件，发信息。需要你对他们提出的问题做一些判断，也就是给解决问题的选项评分  
4.一个21题，有log分析bug，有给报告出问题结论，有判断项目 走向的  
  
情境1：给图书馆写图书推荐系统，关于book api  
两个人，在表达不同的观点  
选择：tell me more. From 1point 3acres bbs  
一开始其实每个人都在强调自己是对的，即使有一个人更对一些，  
也应该选tell me more（原则8），选了之后会得到更多信息  
  
情境2：选图书馆的服务器有没有开放关于实体书的api  
两个小哥讨论图书推荐的api应该是自己做还是用现成的。  
自己做api覆盖面广，但是due赶不上，别人做的能赶上due。   
requirement优先（原则2），tell me more层层递进  
  
情境3：经理说咱们最近服务器老挂，什么情况？  
先选看internal bug的记录  
选 I think service 3 is the problem,   
but I would like to see another report to confirm  
烙印，义正言辞说自己做了20年服务器，不可能有错误，  
刚刚调试过服务器，不可能是内部错误。  
选自己去查，问题的关键在于不要麻烦别人  
增加开发过程中测试的时间/测试覆盖更多case，放5  
写Manuel test，放3  
还有个是unit test，也放3  
增加QA的人手，放1  
让客户来当小白鼠发现问题，放1  
. check 1point3acres for more.  
  
情境4：Amazon recommendation system item，  
给你推荐一些你感兴趣的item，第一个issue总是失败，  
第二个issue总是显示germany  
第一个问题是因为username 太长所以一直报错。  
第二个问题是因为他用proxy的name来决定是不是语言了。  
  
情境5：德国amazon除了什么问题，让你看log回答问题。问你大概哪里除了问题  
亚马逊推荐广告，给英国人推了德文广告，给你log文件，  
问你可能在哪，找bug in error log  
  
情境6：员工们讨论case media network服务器最近好多compliants  
有德国的，有invalid recommendation，有返回404，  
找出错原因的相同点  
德语因为服务器， 一个因为用户名太长，一个是有些用户的语言变成德语  
  
情境7：具体客户ddl 只有两周，两个方案，延到四周，做完整。  
另一个说先实现一部分功能做个demo，再慢慢做。  
先做demo放5，按部就班四周放3， 通知其他组说两走做不完接着做美国放1  
  
情境8：估计项目开发时间  
Manager放5，找有经验的人请教4，上网查资料或是先做一段时间再估计都放3，  
还有其他裸上的就1。  
  
情境9：一个项目时间表设计  
说你是这里最会用什么语言的，比如java  
  
情境10：安排会议  
视频会议 5   三个老二开会和老二去找老大开会 3   推迟会议和邮件开会 1  
  
情境11：搞个数据库  
两周时间可以搞个数据，\*\*ben可以帮忙，大腿priya可以帮，但是要等一周半  
报告manager放5，和\*\*合作等大腿放4，合作/等大腿是3  
自己单干，cut feaure都是1  
  
情境12：系统是否升级  
做两个feature，一个让100%用户爽，一个让20%用户爽，  
但要升级系统，升级系统自己组会爽，但是升级会推迟做的feature，. check 1point3acres for more.  
不升级吧，升级之后还得做一遍  
这题的中心是不升级，先做feature，先让用户爽。  
先做100的feature再升级，再做20的feature，放5  
不升级，因为我们承诺要做feature，放4。. From 1point 3acres bbs  
不升级，要搞定feature，可以以后推了其他ddl再升级，放3  
不升级，因为对其他组没影响，我们应该focus在request上面，放2  
升级，推迟这两个feature的ddl，因为升级造福子孙后代，放2  
升级，不然要做两次，放1. 1point3acres  
这题的关键在于升不升级，要坚定的站在一边  
  
情境13：新产品设计  
给8周时间，选择题，让你pick up 一个features的组合要求利益最大化，  
每个feature都有相应的价值，H >> M >> L 都代表远大于  
首先ddl是前提，中位数不能超过8太多，那样的话就算feature再多也没意义，  
同价值，按照ddl排序，同ddl按照价值排序。  
  
情境17：代码分析  
三段一长选最长