

系统分析与设计能力模型

文档修订历史

版本号	作者	备注	修订日期
0.1	奕梵	初稿	2016-01-13
1.0	奕梵	补充修正	2016-07-14

目 录

1	概述.....	7
1.1	关键术语.....	7
1.2	能力模型.....	8
2	业务分析能力.....	8
2.1	业务知识积累.....	9
2.1.1	概述.....	9
2.1.2	维度包含的知识点.....	9
2.1.3	维度应用场景.....	9
2.1.4	如何掌握这个维度.....	9
2.1.5	如何进行能力值评估.....	9
2.2	业务需求分析.....	10
2.2.1	概述.....	10
2.2.2	维度包含的知识点.....	10
2.2.3	维度应用场景.....	10
2.2.4	如何掌握这个维度.....	10
2.2.5	如何进行能力值评估.....	11
2.3	业务流程分析.....	11
2.3.1	业务流程梳理概述.....	11
2.3.2	维度包含的知识点.....	11
2.3.3	维度应用场景.....	11
2.3.4	如何掌握这个维度.....	11
2.3.5	如何进行能力值评估.....	11
2.3.6	如何考察学习结果.....	12
3	设计能力.....	12
3.1	掌握 UML 和相关工具.....	12
3.1.1	概述.....	12
3.1.2	维度包含的知识点.....	12
3.1.3	维度应用场景.....	13
3.1.4	如何掌握这个维度.....	13
3.1.5	如何进行能力值评估.....	13
3.1.6	如何考察学习结果.....	13
3.2	理解目标应用架构.....	13
3.2.1	概述.....	13
3.2.2	维度包含的知识点.....	13
3.2.3	维度应用场景.....	14
3.2.4	如何掌握这个维度.....	14
3.2.5	如何进行能力值评估.....	14
3.2.6	如何考察学习结果.....	14
3.3	领域驱动设计.....	14
3.3.1	概述.....	14
3.3.2	维度包含的知识点.....	14
3.3.3	维度应用场景.....	15

3.3.4	如何掌握这个维度.....	15
3.3.5	如何进行能力值评估.....	15
3.3.6	如何考察学习结果.....	15
3.4	常用设计模式.....	15
3.4.1	概述.....	15
3.4.2	维度包含的知识点.....	16
3.4.3	维度应用场景.....	16
3.4.4	如何掌握这个维度.....	16
3.4.5	如何进行能力值评估.....	16
3.4.6	如何考察学习结果.....	16
3.5	非功能性设计.....	17
3.5.1	概述.....	17
3.5.2	维度包含的知识点.....	17
3.5.3	维度应用场景.....	17
3.5.4	如何掌握这个维度.....	17
3.5.5	如何进行能力值评估.....	18
3.5.6	如何考察学习结果.....	18
3.6	数据建模.....	18
3.6.1	概述.....	18
3.6.2	维度包含的知识点.....	18
3.6.3	维度应用场景.....	18
3.6.4	如何掌握这个维度.....	18
3.6.5	如何进行能力值评估.....	19
3.6.6	如何考察学习结果.....	19
4	编码能力.....	19
4.1	CodeReview.....	20
4.1.1	概述.....	20
4.1.2	维度包含的知识点.....	20
4.1.3	维度应用场景.....	20
4.1.4	如何掌握这个维度.....	20
4.1.5	如何进行能力值评估.....	21
4.1.6	如何考察学习结果.....	21
4.2	OO 原则.....	21
4.2.1	概述.....	21
4.2.2	维度包含的知识点.....	21
4.2.3	维度应用场景.....	22
4.2.4	如何掌握这个维度.....	22
4.2.5	如何进行能力值评估.....	22
4.2.6	如何考察学习结果.....	22
4.3	开发人员的测试.....	22
4.3.1	概述.....	22
4.3.2	维度包含的知识点.....	23
4.3.3	维度应用场景.....	24
4.3.4	如何掌握这个维度.....	24
4.3.5	如何进行能力值评估.....	24

4.3.6	如何考察学习结果.....	24
4.4	代码可读性.....	24
4.4.1	概述.....	24
4.4.2	维度包含的知识点.....	25
4.4.3	维度应用场景.....	25
4.4.1	如何掌握这个维度.....	26
4.4.2	如何进行能力值评估.....	26
4.4.3	如何考察学习结果.....	26
4.5	代码严谨可扩展性.....	26
4.5.1	概述.....	26
4.5.2	维度包含的知识点.....	26
4.5.3	维度应用场景.....	27
4.5.1	如何掌握这个维度.....	27
4.5.2	如何进行能力值评估.....	27
4.6	代码规范性.....	27
4.6.1	概述.....	27
4.6.2	维度包含的知识点.....	28
4.6.3	维度应用场景.....	28
4.6.1	如何掌握这个维度.....	28
4.6.2	如何进行能力值评估.....	28
4.6.3	如何考察学习结果.....	28
4.7	代码设计理念和方法.....	28
4.7.1	概述.....	28
4.7.2	维度包含的知识点.....	28
4.7.3	维度应用场景.....	29
4.7.1	如何掌握这个维度.....	29
4.7.2	如何进行能力值评估.....	29
4.7.3	如何考察学习结果.....	29
5	基础技术水平.....	29
5.1	JAVA 基础.....	30
5.1.1	概述.....	30
5.1.2	维度包含的知识点.....	30
5.1.3	维度应用场景.....	31
5.1.4	如何掌握这个维度.....	31
5.1.5	如何进行能力值评估.....	31
5.1.6	如何考察学习结果.....	31
5.2	J2EE 基础.....	31
5.2.1	概述.....	31
5.2.2	维度包含的知识点.....	32
5.2.3	维度应用场景.....	32
5.2.4	如何掌握这个维度.....	33
5.2.5	如何进行能力值评估.....	33
5.2.6	如何考察学习结果.....	33
5.3	通用框架.....	33
5.3.1	通用框架技术认知能力概述.....	33

5.3.2	维度包含的知识点.....	33
5.3.3	维度应用场景.....	34
5.3.4	如何掌握这个维度.....	34
5.3.5	如何进行能力值评估.....	34
5.3.6	如何考察学习结果.....	34
5.4	阿里云技术.....	34
5.4.1	概述.....	34
5.4.2	维度包含的知识点.....	34
5.4.3	维度应用场景.....	34
5.4.4	如何掌握这个维度.....	35
5.4.5	如何进行能力值评估.....	35
5.4.6	如何考察学习结果.....	35
6	软性能力.....	35
6.1	沟通协作能力.....	35
6.1.1	概述.....	35
6.1.2	维度包含的知识点.....	36
6.1.3	维度应用场景.....	36
6.1.4	如何掌握这个维度.....	36
6.1.5	如何进行能力值评估.....	36
6.1.6	如何考察学习结果.....	37
6.2	技术写作表达能力.....	37
6.2.1	概述.....	37
6.2.2	维度包含的知识点.....	37
6.2.3	维度应用场景.....	37
6.2.4	如何掌握这个维度.....	37
6.2.5	如何进行能力值评估.....	37
6.2.6	如何考察学习结果.....	37
6.3	项目管理能力.....	38
6.3.1	概述.....	38
6.3.2	维度包含的知识点.....	38
6.3.3	维度应用场景.....	38
6.3.4	如何掌握这个维度.....	38
6.3.5	如何进行能力值评估.....	38
6.3.6	如何考察学习结果.....	39
6.4	个人时间管理能力.....	39
6.4.1	概述.....	39
6.4.2	维度包含的知识点.....	39
6.4.3	维度应用场景.....	39
6.4.4	如何掌握这个维度.....	39
6.4.5	如何进行能力值评估.....	39
6.4.6	如何考察学习结果.....	39
6.5	学习能力.....	40
6.5.1	概述.....	40
6.5.2	维度包含的知识点.....	40
6.5.3	维度应用场景.....	40

6.5.4	如何掌握这个维度.....	40
6.5.5	如何进行能力值评估.....	40
6.5.6	如何考察学习结果.....	40
6.6	执行能力.....	41
6.6.1	概述.....	41
6.6.2	维度包含的知识点.....	41
6.6.3	维度应用场景.....	41
6.6.4	如何掌握这个维度.....	41
6.6.5	如何进行能力值评估.....	41
6.6.6	如何考察学习结果.....	41
6.7	自我调节能力.....	41
6.7.1	概述.....	41
6.7.2	维度包含的知识点.....	42
6.7.3	维度应用场景.....	42
6.7.4	如何掌握这个维度.....	42
6.7.5	如何进行能力值评估.....	42
6.7.6	如何考察学习结果.....	42

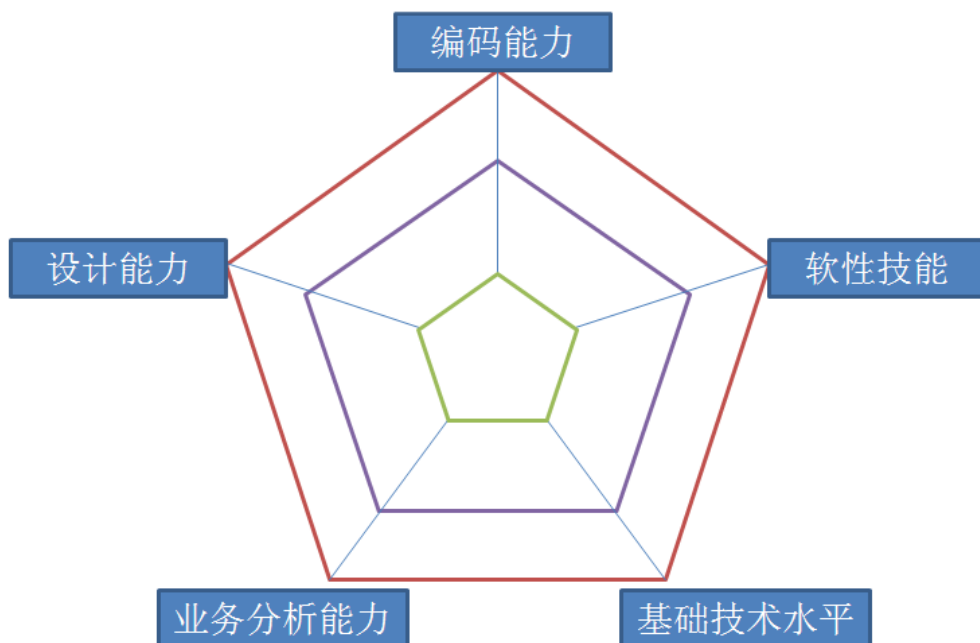
1 概述

系统分析师(以下简称系分)作为技术团队的腰部力量,承担着业务与系统结合的重要职责,在整个项目周期中起着至关重要的作用。开发人员如何能正常为一个优秀的系分?一个优秀的系分应该具备哪些能力?这些能力有哪些知识需要掌握?这些能力如何衡量和提升?本文旨在建立一个合理的系分能力模型,并针对该模型进行阐述。

1.1 关键术语

术语	说明
业务分析能力	业务分析能力是一个合格系统分析师重要的专业技能之一,理解业务和需求才能更好的完成系分工作。
设计能力	作为一个合格的系统分析师,设计能力是必不可少的能力之一。通过优良的设计能力,保证项目的稳定实施,为后续的开发可延续性和伸展性提供保障。
基础技术水平	基础技术水平主要是指系分人员应该掌握的基础技术能力,并能在系分中(技术选型、评估等等)熟练应用
编码能力	编码能力是系分能力的基础,是在系统开发的过程中高质量解决一系列技术需求和业务需求的保障;高质量的编码能力在开发阶段起着非常重要的作用。
软性技能	软性能力属于非专业技能之一,良好的软能力是系分不可或缺的,包含沟通技能等。

1.2 能力模型



系分能力模型从整体上分为编码能力、设计能力、业务分析能力、基础技术水平、软性技能五个部分；其中，编码能力、设计能力、业务分析能力、基础技术水平属于专业技能。每个方面的能力都包含很多维度，如设计能力包含：UML 掌握及工具使用、领域驱动设计等。

2 业务分析能力

业务分析能力是一个合格系统分析师重要的专业技能之一。业务分析是一组任务和技能的集合，目的是为了让大家更好的理解业务价值及其运作模式，并针对该业务提出相应的解决方案。包含维度列表

维度编号	维度名称	备注
1	业务知识积累	相关联的业务和行业知识掌握，“广”
2	业务需求分析	业务功能整体性掌握，“全”
3	业务流程分析	业务理解要有深度，“深”

2.1 业务知识积累

2.1.1 概述

作为系分应该了解和掌握各自领域的业务知识及周边关系域的业务知识,主要包含以下两部分:

1. 自身负责的业务域所涉及到的业务知识
2. 周边关系域涉及到的业务知识

2.1.2 维度包含的知识点

1. 普通选座票购票
2. 会员卡
3. 卖品
4. 取票机
5. 营销活动
6. 券
- 7.
8.

2.1.3 维度应用场景

1. 业务需求讨论及评审
2. 业务发展规划参与
3. 业务需求合理性评估

2.1.4 如何掌握这个维度

1. 查看公司内部相关业务架构文档
2. 参加相关业务培训及分享
3. 参加各种业务分享,了解一个相关联业务知识
4. 自学相关业务

2.1.5 如何进行能力值评估

1. 对领域类的业务能整体画出业务流程图

2. 能较为清晰的描述业务流程中关键点及相关背景。

2.2 业务需求分析

2.2.1 概述

业务需求分析的目的是解决"要做什么"的问题，需要全面地理解用户的各项要求，并准确地表达所接受的用户需求

2.2.2 维度包含的知识点

1. 理解业务需求
2. 做需求分析的方法：工具、技巧
3. 需求功能点挖掘
4. 发现 PRD 需求中的问题
5. 业务需求如何与现行系统结合
6. 业务未来发展方向了解

2.2.3 维度应用场景

1. 业务需求学习及讨论
2. 对需求中相关功能，用图表或其他方式初步描述需求各角色见的协作关系。
3. 参与 PRD 评审，看 PRD 中对正常业务的分析、异常业务的分析、业务状态是否都考虑到了。
4. 评估需求合理性，了解业务未来发展方向及规划

2.2.4 如何掌握这个维度

1. 资深的系分分享
2. 参与需求评审
3. 参与需求讨论及分析
4. 让 PD 讲解需求的背景、价值及未来方向

2.2.5 如何进行能力值评估

1. 需求与系分理解是否一致
2. 系分产出与需求功能要求是否一致

2.3 业务流程分析

2.3.1 业务流程梳理概述

业务流程分析是对业务需求分析的进一步细化。业务流程分析的目的是：形成合理、科学的业务流程，产出新的或是更为合理的业务流程来满足需求。

2.3.2 维度包含的知识点

1. 根据 PRD 了解需求中的业务流程在现行系统中的交互关系及涉及到的功能
2. 根据 PRD 了解业务流程在本系统中的职责、定位、涉及功能点
3. 业务流程合理性评估

2.3.3 维度应用场景

1. 编写系分文档中业务流程，业务用例等
2. 与项目组人员讨论业务流程

2.3.4 如何掌握这个维度

1. 与 PD/BD 对业务流程进行讨论
2. 与相关系统 Owner、架构师交流
3. 向项目组人员介绍业务流程

2.3.5 如何进行能力值评估

1. 画出一个主要业务流程，涉及功能
2. 描述业务流程中的关键点

2.3.6 如何考察学习结果

1. 项目组反馈
2. 给出对应需求的业务流程并讲解

3 设计能力

作为一个合格的系统分析师，设计能力是必不可少的能力之一。通过优良的设计能力，保证项目的稳定实施，为后续的开发可延续性和伸展性提供保障。

设计能力包含维度列表

维度编号	维度名称	备注
1	掌握 UML 和相关工具	
2	理解目标应用架构	
3	领域驱动设计	
4	常用设计模式	
5	非功能性设计	
6	数据建模	

3.1 掌握 UML 和相关工具

3.1.1 概述

通过 UML 描述了一个系统的静态结构和动态行为，为系统设计提供依据，为开发、测试人员沟通提供便利，为后续开发人员学习系统提供资料

3.1.2 维度包含的知识点

1. 掌握常用 UML 图(包括类图、时序图、活动图、用例图、状态机图等)图表语法。后续圈定常用 UML 元素范围，作为系分需要掌握的基础
2. 熟练掌握 UML 工具使用(推荐 VP)
3. 了解各种 UML 图的应用场景，每个 UML 图在编写文档或在作为沟通工具时，都有自己的作用范围，比如要描述业务的流程，用活动图会优于使用时序图。

3.1.3 维度应用场景

1. 系分文档的编写
2. 和开发、测试、业务方交流的过程
3. 企业架构文档的编写

3.1.4 如何掌握这个维度

1. 提供专门的 UML 课程
2. 通过对优秀 UML 图的分析，达到标杆的效果

3.1.5 如何进行能力值评估

1. 已有系分文档中 UML 图的评估
具体评估参考如下几点：
 - (a) 图表的语法是否符合规范
 - (b) 是否使用不适合场景的图表进行表达
 - (c) 是否有过多文字描述，缺少 UML 图表达的地方

3.1.6 如何考察学习结果

1. 围绕特定的课题，画出相关 UML 图，描述思路

3.2 理解目标应用架构

3.2.1 概述

理解当前开发系统的目标架构，从架构层面分析当前的重点，使整体设计能够和目标架构的期望保持统一。

3.2.2 维度包含的知识点

1. 理解当前项目涉及到的系统目标架构。系分人员需要通过查看相关文档、和应用架构师沟通的方式，理解现有系统目标架构。
2. 根据一定时间段内的目标架构，定义设计准则。比如目标架构是年度维稳，那么设计准则可能是需要以稳定性为重点，作为设计的准则。

3.2.3 维度应用场景

1. 系分评审、和架构师沟通
2. 在系分设计的初期，作为指导性原则
3. 企业架构文档的编写

3.2.4 如何掌握这个维度

1. 让应用架构师给系分进行目标架构分享
2. 完善各系统的目标架构文档，让开发人员有据可依

3.2.5 如何进行能力值评估

1. 评估系分文档和目标架构的一致程度
具体可以从如下几点评估：
 - (1) 系分文档中对系统架构的描述是否和当前的目标架构一致
 - (2) 系分文档中是否以目标架构作为指导性思路

3.2.6 如何考察学习结果

1. 让架构师参与到考察中来，通过面对面的沟通的形式，考察学员对目标架构的理解以及对设计的影响

3.3 领域驱动设计

3.3.1 概述

领域模型使开发人员可以表达丰富的软件功能需求，由此实现的软件可以满足用户真正的需要，因此被公认为是软件设计的关键所在，其重要性显而易见。

3.3.2 维度包含的知识点

领域驱动意识的建立，如何在设计之前深层次理解领域知识

1. 如何建立统一的语言
2. 模型和实现的绑定关系
3. 领域驱动设计中的分层架构

4. 常用模型：entity(实体)、vo(值对象)、service(服务)
5. 领域模型的生命周期管理：aggregate、factory、repository
6. 领域驱动的重构技巧：隐式概念到显式概念的转化、柔性设计等
7. 了解领域驱动设计中常用的模式的思想
8. 了解领域驱动设计在业务系统中的应用

3.3.3 维度应用场景

1. 系统分析与设计
2. 系统重构
3. 团队沟通
4. 系统框架搭建

3.3.4 如何掌握这个维度

1. 系统全面有深度的学习和思考
2. 组织相关主题学习培训
3. 组织领域驱动设计分享

3.3.5 如何进行能力值评估

1. 系分文档、代码、系统设计及相关沟通阶段、codereview、重构中领域驱动知识和原则的使用

3.3.6 如何考察学习结果

1. 安排每人 20 分钟，对领域驱动设计进行讲述，考察是否做到深入浅出、条理清晰
2. 在本系统中是有能够应用领域驱动设计的思想、方法和技巧。

3.4 常用设计模式

3.4.1 概述

设计模式（Design pattern）是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结。使用设计模式是为了可重用代码、让代码更容易被

他人理解、保证代码可靠性。毫无疑问，设计模式于己于他人于系统都是多赢的，设计模式使代码编制真正工程化，设计模式是软件工程的基石，如同大厦的砖石一样。

3.4.2 维度包含的知识点

1. 了解常用设计模式在业务系统中的应用
2. 掌握常用设计模式 UML 图及其思想
3. 与各种面向对象设计原则的结合
4. 各种模式适用的场景

3.4.3 维度应用场景

1. 系统架构设计
2. 代码重构
3. 系分文档编写
4. 高标准代码开发

3.4.4 如何掌握这个维度

1. 系统全面有深度的学习和思考
2. 组织相关课程进行分享

3.4.5 如何进行能力值评估

1. 系分文档或代码中查看设计模式的使用情况
2. 介绍几个常用的设计模型及实际应用情况

3.4.6 如何考察学习结果

1. 安排每人 20 分钟，对设计模式进行讲述，考察是否做到深入浅出、条理清晰
2. 在本系统中能够应用设计模式及面向对象的思想、方法和技巧。

3.5 非功能性设计

3.5.1 概述

非功能性需求是指软件产品为满足用户业务需求而必须具有且除功能需求以外的特性，包括稳定性、性能、安全性、可用性等，非功能性设计是评估一个软件产品做到了多好或者说评估一个软件产品的品质的非常重要考核点。非功能性设计部分关键是让大家多了解非功能性设计的一些方法，然后在设计中去关注和运用这些方法

3.5.2 维度包含的知识点

1. 容量与性能：使用缓存，线程池等
2. 安全性：防篡改，验签，敏感信息处理等
3. 稳定性：异步设计，恢复机制等
4. 高可用：无单点设计,短事务与柔性事务,异步设计,可重试，可禁用等
5. 可伸缩：异步设计，可水平拆分，可缓存，容量与资源使用可监控
6. 可维护：

以上各知识点之间有重叠，其中 1,2,3 属于关键非功能性要求，4,5,6 属于重点了解内容

3.5.3 维度应用场景

1. 项目系分设计都应该考虑到非功能性设计,一般会涉及到非功能性设计中的某些点，但不一定全都涉及

3.5.4 如何掌握这个维度

1. 自学现有系统的优秀方案和成功经验
2. 提供专题培训（此部分可培训内容较多）
3. 日常互相分享见到的一些优秀设计方案资料，分享日常工作中遇到的一些非功能性问题及解决的办法。

通过如上方式掌握一些非功能性设计的一些方法论和原则，逐渐的让非功能性设计成为各系分人员设计思想中根生地固的一部分

3.5.5 如何进行能力值评估

1. 系分文档非功能性设计 checkList，系分文档是否体现了关键的非功能性设计的思想
2. 介绍一个自己完成的非功能性设计

3.5.6 如何考察学习结果

1. 日常系分文档，有没有体现出非功能性设计的思想，方案是否优秀
2. 思考现有系统在非功能性方面有哪些问题，可以用哪些方式优化
3. 采用交流的方式学习非功能性设计规范

3.6 数据建模

3.6.1 概述

数据建模指的是对现实世界各类数据的抽象组织，确定数据库需管辖的范围、数据的组织形式等直至转化成现实的数据库，DDD 模式主要是如何从领域模型映射到数据模型。

3.6.2 维度包含的知识点

1. 数据建模工具的使用(如 powerdesigner,ERWin)
2. 从领域模型到数据模型的映射
3. 数据建模的思路和方法(扩展字段)
4. 了解数据建模常见的特点和潜规则，如：无外键

3.6.3 维度应用场景

1. 新系统的建立，大方案的调整时涉及数据库变更的项目
2. 系分文档编写
3. 数据库结构变更

3.6.4 如何掌握这个维度

1. 形成体系要求后，提供一些方法论的资料和培训

2. 形成数据库规范，并做宣讲

3.6.5 如何进行能力值评估

1. 给出某一个系统的领域模型,推导出数据模型，看是否合理
2. 被评估者描述曾经做过的数据建模，设计方案和思路

3.6.6 如何考察学习结果

1. 涉及到较大数据库变更的项目，系分文档中是否有数据模型的内容，设计是否合理
2. 专题性的项目，进行数据建模，说明设计原因，评估建模的效果

4 编码能力

编码能力是系分设计能力的基础，是在日常系统开发中锻炼出来的硬功，在编码能力基础上进行抽象可以作为提升设计能力的基础。优秀的编码能力更能得到项目组认可。

编码能力维度列表：

维度编号	维度名称	备注
1	CodeReview	
2	OO 原则	
3	开发人员的测试	
4	代码可读性	
5	代码严谨和可扩展性	
6	代码规范性	
7	代码设计理念和方法	

4.1 CodeReview

4.1.1 概述

CodeReview 是对开发的产出代码的评审活动，目前代码评审有了更全面的描述，例如结对 review，交叉 review 等。而高效、丰富的 review 活动会给团队长期发展带来支持，不仅仅局限于当前升级包的质量。CodeReview 能力是一项综合能力，是对已有知识的掌握应用的、更是将知识传播到团队的能力。系分掌握了良好的 CodeReview 方法知识后，了解该活动的开销、成本。在系分文档的把控上，可以准确的标记出哪些内容是需要重点 review 的，需要组内提供什么样的资源来做 review，review 后对于团队、项目的收益。

4.1.2 维度包含的知识点

1. 掌握编码规范
 - a) 系统代码简版规范。
2. 掌握代码评审工具
 - a) Tala(适用于异地开发)，该插件目前还仅限于阿里巴巴集团内使用，目前还在洽谈中
 - b) Jupiter(第三方工具)
3. 掌握敏捷方法中的结对概念
4. 掌握会议的组织、把控能力
5. 掌握良好的沟通能力

4.1.3 维度应用场景

1. 系分需要了解 CodeReview 的开销、收益，才能够通过系分文档将开发资源做到最极致的分配，准确投放开发资源获得团队、项目的收益最大化。结对不仅仅保障版本质量，更重要的是团队的成长。
2. 项目过程中参与 REVIEW 活动

4.1.4 如何掌握这个维度

1. 积极参与到 CodeReview 过程中

2. 提出代码中的问题，并给出好的建议和方法
3. 学习敏捷中结对的知识

4.1.5 如何进行能力值评估

1. 介绍自己的 REVIEW 经历

4.1.6 如何考察学习结果

1. 主管、PM、开发人员反馈

4.2 OO 原则

4.2.1 概述

OOP 作为当前最广泛应用的编程语言方法论之一，其强大业务抽象的能力被大家所公认。铭记 OO 原则是所有优秀 JAVA 开发人员所必需的，著名的设计模式中大部分也建立在 OO 原则基础上的高级应用。只有熟悉 OO 的开发人员，其设计才能将 JAVA 抽象复杂业务的能力发挥到极致。

4.2.2 维度包含的知识点

1. 类原则
 - a) SRP，单一职责原则
 - b) OCP，开放封闭原则
 - c) LSP，Liskov 替换原则
 - d) DIP，依赖倒置原则
 - e) ISP，接口隔离原则
2. 包内聚原则
 - a) REP，重用发布等价原则
 - b) CCP，共同封闭原则
 - c) CRP，共同重用原则
3. 包耦合原则
 - a) ADP，无环依赖原则
 - b) SDP，稳定依赖原则

c) SAP, 稳定抽象原则

4. 其他原则

a) BBP(Black Box Principle)黑盒原则

b) DAP(Default Abstraction Principle)缺省抽象原则

c) IDP(Interface Design Principle)接口设计原则

d) DCSP(Don't Concrete Supperclass Principle)

e) 迪米特法则

4.2.3 维度应用场景

Java 代码设计无处不 OO, 大到系统间的交互界面设计, 小到一个类里的属性设计。

4.2.4 如何掌握这个维度

1. 学习 OO 相关理论知识
2. 从工作中去总结与理解
3. 在工作中应用
4. 积极参与各种设计相关的活动, 相互交流经验

4.2.5 如何进行能力值评估

1. 评审系分中设计部分是否都符合 OO 原则
2. 编写的代码是否符合 OO 原则
3. 面谈的方式介绍自己度 OO 的理解

4.2.6 如何考察学习结果

1. 架构 review

4.3 开发人员的测试

4.3.1 概述

在软件开发方法中, 测试与编码的分工界限不再清晰。优秀系分总能从测试角度分析并影响设计, 测试提升设计的思想也被业界认可。系分应该拥有足够的

测试相关知识、方法论。

4.3.2 维度包含的知识点

1. 常用测试框架
 - a) Junit
 - b) TestNG
2. 常用测试工具
 - a) JMOCK
 - b) Mock4j
3. 测试
 - a) 持续集成环境
 - b) 持续集成的指标
4. 开发测试方法
 - a) 测试驱动开发
 - b) 单元测试
 - c) 集成测试
 -
5. 软件测试方法
 - a) 白盒测试
 - b) 黑盒测试
 - c) 压力测试
 - d) 负载测试
 - e) 自动化测试
 -
6. 测试原则
 - a) 保持测试独立
 - b) 隔离 SUT
 - c) 最小化测试重叠
 - d) 保持测试逻辑与产品代码分离
 -

7. 测试套件(test suite)策略

4.3.3 维度应用场景

编码过程中调试、自测占用的时间远超编写代码的时间，优秀的测试技能不仅仅是保证代码的逻辑的正确性，通过优良的单元测试用例实际可以提升代码设计、与代码修改后的验证效率，从而直接提升代码的交付能力。测试能力已经是一个优秀开发人员必要的技能之一。

4.3.4 如何掌握这个维度

1. 和测试同学一起配合，学习测试的思维方式、测试的理论知识
2. 阅读《TDD》《xUnit 测试模式:测试码重构》等大作
3. 了解各种知识后积极实践，积累经验
4. 参与测试技巧分享等

4.3.5 如何进行能力值评估

1. 代码的可测试性
2. 测试代码的设计是否优秀
3. 测试代码的维护成本是否低廉
4. 测试是否提升整体开发效率

4.3.6 如何考察学习结果

1. 开发过程中的测试思想交流
2. 测试收益率

4.4 代码可读性

4.4.1 概述

代码可读性是指编写的代码复杂模块容易理解、注释准确通顺、代码复杂度低，方法长度合理、代码重复率低。

4.4.2 维度包含的知识点

1. 复杂模块拆分重用，代码复杂度低
2. 注释准确通顺，描述清晰
3. 包结构清晰
4. 对外接口清晰

4.4.3 维度应用场景

在实现业务的过程中对代码实现的精雕细琢，设计实现上养成好的编码习惯，做到低耦合，低风险，开发实践中这个维度的能力是随时在锻炼的，随时需要提升的。

例如向秀对事务的常规模式分享：

```
// 将 bizNo 的记录从 I 状态变为 S 状态 （前置条件是：记录存在，并且状态为 I 状态；后置条件是：记录为 S 状态）
1doInTransaction(bizNo) {
2    //启动事务；
3    Record xxLockedRecord = loadAndLock(bizNo); // select * from xxRecord where
bizNo=$bizNo$ for update;
4
5    // 判空
6    if(null == xxLockedRecord) {
7        return errorResult;
8    }
9
10   // 检查状态是否是 I 状态
11   if(Status.I != xxLockedRecord.getStatus()) {
12       return errorResult;
13   }
14
15   // 更新状态为 S
```

```
16.    update(xxLockedRecord, Status.S);  
17  
18 // 其他操作，如发送消费记录事件之类  
19  
20 // 返回结果  
21 return successResult;  
22。 //事务提交  
23}
```

4.4.1 如何掌握这个维度

1. 在项目中进行定向实践
2. 优秀代码阅读与分享
3. 学习他人写的优秀代码

4.4.2 如何进行能力值评估

1. 阅读编写的代码
2. 架构师或项目组成员反馈

4.4.3 如何考察学习结果

1. 定向命题，重构特定的代码。
2. 阶段性 Review 代码。

4.5 代码严谨可扩展性

4.5.1 概述

代码严谨性和可扩展性是在编码过程中对异常情况处理全面严谨，编写代码具备良好的可扩展性业务变化时便于扩展。

4.5.2 维度包含的知识点

1. 异常情况处理

2. 并发处理
3. 幂等性控制
4. 重试机制

4.5.3 维度应用场景

系统架构整体上是分布式的结构，所以编码过程中时刻需要严谨的考虑各种并发处理，需要考虑幂等性控制和重试机制，例如：同时几个付款请求过来的时候，如果不做幂等性控制就可能同一笔单据打款数次导致资损，又例如消息接收，通常消息会丢失或者发送重复，也需要对各种情况考虑周全。

另外业务也是经常会发生变化的，所以在编码设计上需要考虑具备可扩展性，一定程度可以前瞻业务的发展，业务变化后可以以最小的代价进行扩展。例如：水电煤缴费接入机构数量很多，在编码上对机构接入进行分类梳理，将特殊的一些机构处理通过规则的方式进行，新接入一家机构后如果是特殊的只需要配置一些规则就可以满足。

4.5.1 如何掌握这个维度

1. 典型案例分析
2. 结对 Review
3. 实践中琢磨
4. 优秀代码走读

4.5.2 如何进行能力值评估

1. 阅读代码
2. 结对 Review
3. 架构师反馈

4.6 代码规范性

4.6.1 概述

代码规范是指编写的代码符合简版规范，代码符合代码模板，无 PMD 扫描一级，二级问题，较少二级以下问题。

4.6.2 维度包含的知识点

1. 代码模板规范
2. 简版规范

4.6.3 维度应用场景

1. 代码规范是编码的基础，每个开发需要符合的，规范的基础上才能让代码可读性更好。

4.6.1 如何掌握这个维度

1. 掌握简版规范标准，代码检查。

4.6.2 如何进行能力值评估

1. 代码工具检查

4.6.3 如何考察学习结果

1. 抽查项目代码

4.7 代码设计理念和方法

4.7.1 概述

代码设计理念是代码设计维度表现出来的一些能力和经验的总结，包括代码中的领域模型清晰（与业务的匹配）程度、采用通用组件/框架/工具来辅助代码逻辑，减少重复造轮子的工作，同时代码设计不但给自身的系统代码带来明显的提升，同时设计理念、方法能够从一定程度上帮助改善技术部其他系统现状，成为代码设计范本。

4.7.2 维度包含的知识点

1. 领域模型编程
2. 通用组件/框架/工具运用
3. 错误码设计思想

4. 日志监控
5. 设计理念分享和传播

4.7.3 维度应用场景

1. 代码设计理念和方法是编码能力经验的总结和积累，在项目中可以覆盖较为全面的范围，对系统结构和代码层次能够有一定掌控。是编码能力成熟后在系统可维护，可降级，结构优雅层面体现。

4.7.1 如何掌握这个维度

1. 实践中总结分享
2. 定向培养，思维模式养成。

4.7.2 如何进行能力值评估

1. 历史分享
2. 架构反馈

4.7.3 如何考察学习结果

1. 结对 Review，交流
2. 代码结构考察

5 基础技术水平

基础技术水平主要是指系分人员应该掌握的基础技术能力，并能在系分活动中（技术选型、评估等等）熟练应用。目前采用的技术基本都是基于 java 的，所以在此我们主要考量是大家对 Java 类基础的技术水平。基础技术水平涵盖了一个初级开发人员到优秀系分需要了解或掌握的基础技术。

能力维度列表

维度编号	维度名称	备注
1	Java 基础	
2	J2EE 基础	
3	通用框架	

4	阿里云技术	

5.1 JAVA 基础

5.1.1 概述

JAVA 是目前开发中的主要语言，作为系分不仅要会使用这门语言,还要不断深入学习并在实际工作中运用它们，Java 基础技术是构成 Java 所有上层应用的基础。

5.1.2 维度包含的知识点

1. 使用 Java (70%)

- a) 基本编码（一些基本包的使用 lang、io、math、util 等等）、同时掌握面向对象、多态、接口设计等等
- b) 进阶编码，包括线程、Socket、NIO、Concurrent、泛型、正则表达式、security、变量进阶申明方式（ThreadLocal、SoftReference）等等
- c) 熟悉并能制定编码规范、走查代码
- d) 能使用 JUnit 或者 TestNg 来编写单元测试
- e) 学习并掌握 JDK 提供的工具，如 jstat、jconsole、javap 等等
- f) 学习并掌握 JVM 的参数设置，并理解为什么这么设置

2. 深入理解 Java (30%)

- g) 了解类装载方式
- h) java 内存分配机制、垃圾回收原理
- i) 了解中间代码语法及结构
- j) java 线程、锁实现机制
- k) Socket 深入了解（建链三次握手、断链四次握手、状态机变更、RTO、sendbuff 和 receivebuff 等等）
- l) 能通过工具分析 heapdump 和 threaddump 等
- m) 深入学习并理解 JVM（Hotspot、JIT）
- n) 学习并掌握 JMX

5.1.3 维度应用场景

1. 在系分时，需要评估某些业务的实现方式，需要扎实的基础知识，比如加解密，此时需要评估是否采用 java 自带的 security 包的工具类
2. 再比如再一些对性能要求很高的场景下，可以使用 SoftReference 来申明变量，这也需要对 java 基础有一定的了解
3. 在性能调优时，需要使用 jdk 提供的工具观察 gc、线程使用情况，通过了解 java 的一些实现原理来分析并调优性能

5.1.4 如何掌握这个维度

1. Java 使用通过自学的方式掌握
2. 深入理解 Java，可以安排一些公开课程（比如 GC、调优|内存分配原理等等）

5.1.5 如何进行能力值评估

1. 考试
2. 面谈

5.1.6 如何考察学习结果

1. 项目组反馈
2. 考试

5.2 J2EE 基础

5.2.1 概述

J2EE 核心是一组技术规范与指南，其中所包含的各类组件、服务架构及技术层次，均有共同的标准及规格，让各种依循 J2EE 架构的不同平台之间，存在良好的兼容性

J2EE 适用于创建服务器应用程序和服务。J2EE 是一种利用 Java 2 平台来简化企业解决方案的开发、部署和管理相关的复杂问题的体系结构

目前的系统也是基于 J2EE 的，所以了解 J2EE，对更好的学习和掌握技术有

一定帮助。

5.2.2 维度包含的知识点

1. J2EE 13 种核心技术（70%）
 - 1.1) 基础部分（30%）
 - a) JDBC
 - b) RMI
 - c) JSP
 - d) JAVA SERVLETS
 - e) XML
 - 1.2) 进阶部分（40%）
 - a) JNDI
 - b) JMS
 - c) JAVA IDL
 - d) JTS
 - e) JTA
 - f) JAVA MAIL
2. 了解并掌握主流、开源的 J2EE 应用服务器，并能分析其优缺点(30%)
 - a) Tomcat
 - b) WebService 等等

5.2.3 维度应用场景

1. J2EE 的很多核心技术，在平时日常工作中都会使用，比如 JDBC，XML，Servlet，JMS 等等
2. 了解并掌握 J2EE 基础，对理解分布式事务、消息通知都有一定的帮助；
3. 通过对分析不同 J2EE 服务器的优缺点，也能对今后系分时技术选型提供一定的依据

5.2.4 如何掌握这个维度

1. 自学
2. 专题培训

5.2.5 如何进行能力值评估

1. 考试
2. 面谈

5.2.6 如何考察学习结果

1. 项目组反馈
2. 考试

5.3 通用框架

5.3.1 通用框架技术认知能力概述

通用框架技术，是基于 Java 基础、J2EE 基础上做的一些框架性的开发、构建，是底层技术的包装和发布，用于解决某一特定的应用需求。

学习并掌握通用技术，对理解框架有帮助，同时通用框架也是系统同学在做技术选型时的基础素材，学习和掌握他们对技术选型有帮助。

5.3.2 维度包含的知识点

1. 相关框架了解并掌握

基础框架	网络/通讯	数据库	连接池	作业调度	缓存
Spring	mina	Hibernate	DBCP	Quartz	JBossCache TreeCache
OSGI	HttpClient	ibatis	C3P0		OSCache
	Ice		BoneCP		JCACHE
					Ehcache

2. 框架分类对比

能对同种类型的不同实现的框架进行分析、对比，给出其优缺点。

5.3.3 维度应用场景

1. 许多开源框架，在平时日常工作中都会使用

5.3.4 如何掌握这个维度

1. 自学
2. 专题培训

5.3.5 如何进行能力值评估

1. 考试
2. 面谈

5.3.6 如何考察学习结果

1. 项目组反馈
2. 考试

5.4 阿里云技术

5.4.1 概述

使用技术主要从框架、网络通讯、数据、消息、语言、安全这些维度来描述，这些都是平时系分工作中需要学习和掌握的。

通过了解这些技术的细节，也为系分同学的技术选型、方案选择时提供帮助。

5.4.2 维度包含的知识点

技术主要包括如下方面：SLB、ONS、MNS、EDAS、RDS、OSS、redis 等

5.4.3 维度应用场景

1. 独立搭建新系统，需要了解这些知识
2. 大量应用都是基于这些基础组件，在系分时要依据场景判断到底采用哪类基础组件。

5.4.4 如何掌握这个维度

1. 自学
2. 专题培训

5.4.5 如何进行能力值评估

1. 考试
2. 面谈

5.4.6 如何考察学习结果

1. 项目组反馈
2. 考试

6 软性能力

软性技能属于非专业技能之一，良好的软能力是系分不可或缺的，是系分综合素质体现。

能力维度列表

维度编号	维度名称	备注
1	沟通协作能力	
2	技术写作表达能力	
3	项目管理能力	
4	个人时间管理能力	
5	学习能力	
6	执行能力	
7	自我调节能力	

6.1 沟通协作能力

6.1.1 概述

沟通协作能力对于系分来讲，是一项非常重要的基本技能。它决定了我们能否在工作中顺畅的与团队各成员有效协作，对提高工作效率和质量有着重要的意

义。

6.1.2 维度包含的知识点

1. 迅速且准确地获取别人的信息、并挖掘潜在信息
2. 迅速且准确地表达自己的信息
3. 项目进行顺利时或者有突然事件发生时（比如资源出现问题、质量出现问题等与 PM、测试、前端、DBA 的沟通协作能力）与项目组和团队成员的沟通
4. 项目中有突发事件发生时（比如风险发生、需求变更、质量出现问题等）沟通并协调相关人员才能起到最好的效果。
5. 利用团队力量解决问题
6. 帮助团队解决问题

6.1.3 维度应用场景

1. 贯穿项目始末，所有与项目相关人员沟通协作的时候。如：系分评审、变更处理、发布计划评审等。

6.1.4 如何掌握这个维度

1. 需求理解和信息挖掘案例分享
2. 如何准确快速获取自己想要的信息的案例分享
3. 平时和突发情况下，与项目组人员的沟通技巧培训
4. 利用团队力量解决问题达到很好效果的典型案例分享
5. 帮助团队解决问题达到很好效果的典型案例分享

6.1.5 如何进行能力值评估

1. 准备一个实际应用场景，提供部分信息，然后由我们去理解当前信息，并找到能提供需要挖掘信息的相关人员通过自己的沟通如何快速拿到结果
2. 准备一个实际维度应用场景，项目出现问题的时候，作为系分的我们如何去通过找不同的人去沟通协作最终解决问题

6.1.6 如何考察学习结果

1. 对场景模拟中的案例，理解是否准确；沟通问题的时候，问题是否能很快拿到正确的结果，问题是否及时得到解决。

6.2 技术写作表达能力

6.2.1 概述

在整个项目过程中会涉及很多的文档，而如何通过写好这些文档来准确清晰表达自己的意思给别人知晓就需要系分同学具备良好的写作表达能力。

6.2.2 维度包含的知识点

1. PPT 写作及讲解能力；
2. 技术文档写作表达能力（系分文档等）

6.2.3 维度应用场景

1. 项目中各文档的编写

6.2.4 如何掌握这个维度

1. 文档工具使用培训
2. PPT 演讲技巧培训；、
3. 技术文档写作表达能力培训；
4. 系分文档等最佳实践分享（文档框架、排版、内容、图文结合等）

6.2.5 如何进行能力值评估

1. 系分文档编写考察，工具的基本使用功能考察

6.2.6 如何考察学习结果

1. 系分文档或者其他文档的表达能否准确清晰表达出书写人的意思，工具的基本功能是否熟练掌握

6.3 项目管理能力

6.3.1 概述

在项目管理体系中，系分承担了非常重要的角色，对于项目研发过程中的进度、质量、风险把控都有很高的要求

6.3.2 维度包含的知识点

1. 项目管理工具的使用（cq、project、xmind、wbs）
2. 项目风险意识和识别能力（风险评估、风险预案）
3. 任务分解和工作量评估的能力
4. 开发进度管理能力
5. 项目突发情况处理能力(需求变更、进度问题、质量问题、政策问题、资源问题)

6.3.3 维度应用场景

1. 需求分析评估
2. 风险分析评估
3. 项目开发计划管理
4. 任务分解和工作量评估
5. 项目突发情况处理

6.3.4 如何掌握这个维度

1. 项目管理工具使用的培训
2. 项目风险评估和预案准备分享
3. 项目任务分解和工作量评估能力培训
4. 突发情况下的项目协调处理能力分享和培训

6.3.5 如何进行能力值评估

1. 项目管理工具使用考察、风险评估和预案准备考察、突发情况下的处理方案准备

6.3.6 如何考察学习结果

1. 项目管理工具使用是否熟练，风险评估是否准确、预案准备是否充分，突发情况下处理是否得当。

6.4 个人时间管理能力

6.4.1 概述

作为系分、每天都会在大量并发事件的情况下去做事情，如何分清事情的轻重缓急、合理安排自己的时间去做当前最应该做的事情，对于系分来讲是很重要的。

6.4.2 维度包含的知识点

1. 分清当前并发事件的优先级和顺序
2. 合理安排个人的时间、在正确的时间做正确的事

6.4.3 维度应用场景

1. 系分每天的日常工作中

6.4.4 如何掌握这个维度

1. 并发事件优先级和顺序性判断案例分享
2. 在正确的时间做正确的事的案例分享

6.4.5 如何进行能力值评估

1. 模拟场景，有很多事情需要系分去完成，事情之间有优先级和顺序性、给一定的时间让系分安排自己的时间如何高效的去完成这些事，最终拿到好的结果。

6.4.6 如何考察学习结果

1. 事情优先级和顺序性判断是否准确，时间安排是否得当。

6.5 学习能力

6.5.1 概述

学习能力，是作为技术工作者一个必不可少的能力，对系分来讲更是很重要的一个能力，如果在尽量短的时间内更快的上手并做好一个事情，学习能力就显得尤为重要。

6.5.2 维度包含的知识点

1. 掌握正确的学习方法
2. 接触一门新技术可以快速的了解其底层原理并使用起来
3. 对于外部接口技术文档和新概念的业务文档，能快速学习并找到自己想要的部分

6.5.3 维度应用场景

1. 项目中需要用到新技术的时候
2. 项目中涉及新概念的业务产品
3. 有第三方技术需要使用的时候

6.5.4 如何掌握这个维度

1. 好的学习方法分享与培训
2. 快速阅读第三方技术文档并使用的案例分享

6.5.5 如何进行能力值评估

1. 给一份新技术、快速的把它使用起来
2. 给一份新概念的业务文档，快速理解业务产品的原理。

6.5.6 如何考察学习结果

1. 新技术、新业务学习是否迅速并准确。

6.6 执行能力

6.6.1 概述

执行能力，是作为系分在工作中能否快速拿到结果的保证，我们有了计划之后需要去执行，执行能力强弱就决定了计划是否最终能顺利得到保证的关键。

6.6.2 维度包含的知识点

1. 正确理解目标
2. 帮助团队理解目标
3. 推进一起解决

6.6.3 维度应用场景

1. 项目计划执行

6.6.4 如何掌握这个维度

1. 关于个人执行力的经典案例分享

6.6.5 如何进行能力值评估

1. 模拟一份项目计划执行，系分需要理解当前项目计划执行的关键点，并帮助团队同学一起理解，然后切实的推进每一个步骤都按计划顺利去执行。

6.6.6 如何考察学习结果

1. 最终项目计划是否被切实执行完成

6.7 自我调节能力

6.7.1 概述

自我调节能力包括在项目中与不同的人打交道遇到困难或者挫折时候如果快速的调整自己并最终拿到结果，还包括遇到较大项目内外压力的情况下，如果调整好自己的状态不受影响。

6.7.2 维度包含的知识点

1. 困难和挫折调节能力
2. 压力调节能力

6.7.3 维度应用场景

1. 各种工作场景

6.7.4 如何掌握这个维度

1. 困难和挫折调节经典案例分享
2. 压力调节经典案例分享

6.7.5 如何进行能力值评估

1. 暂不做评估

6.7.6 如何考察学习结果

1. 暂不考察