

能力框架	能力项目	能力标准							
		T1		T2		T3		T4	
		关键词	行为标准	关键词	行为标准	关键词	行为标准	关键词	行为标准
通用能力	创新能力	有创新意识	有创新意识，工作中对常规工作方法提出疑问和挑战，提出合理化建议。	以创新的角度思考问题，灵活变通	1、能恰当地质疑已存在的解决问题模式，以创新的角度进行思考，对当前业务领域的技术或方式提出创新的方法或思路； 2、在一些问题上能灵活变通，积极响应创新。	进行创新性尝试，提升工作效率	1、思路敏捷，富有创意，能从全新的角度进行思考； 2、乐于在工作中进行创新性尝试，并能对相关业务领域或工作产生显著的良性影响，使工作效率提高。	全新突破	1、思维突破常规的、传统的思维模式的限制，提出具备可行性的新方案（这里“新”是指有实质性差异或是指全新）； 2、使工作效率有大幅度的提高，并且能有效控制创新带来的风险。
	解决问题	快速解决问题	具备快速定位、解决工作范围内出现问题的能力。	先思考再解决多方案选最优	1、具备良好的解决问题的方法论； 2、有意识思考问题的本质，制订多方案，并择优处理。	确认根本原因避免次生问题	1、从机制、原理层面分析问题的根本原因，采取治本的方案处理问题； 2、全面评估问题的影响，避免次生问题的产生。	全局思维解决疑难问题	1、攻坚型人才，能从软件的各个角度思考，包括不限于业务需求、可扩展性、安全、性能、可维护性、运维、稳定性等方面系统性思考解决问题，在解决问题的同时，真正提升产品的竞争力； 2、具备解决跨系统、跨环境、系统性的疑难杂症。
	学习能力	指导下进行学习	1、有学习愿望，能够在指导或者要求下进行学习； 2、能够通过指定的学习资源掌握做好自身岗位工作所需要的知识、技能、工具和信息等。	寻找学习机会，学以致用	1、积极和善于寻找学习机会，关注培训机会，结合成长规划，适时地为自己安排培训和学习，保持专业知识技能的更新； 2、积极地学习态度并且注意学以致用，不断探索改善和提高自身的工作效率； 3、在工作中和平时的学习积累过程中找寻有价值的信息。	总结提炼，成果分享，帮助他人学习	1、了解专业领域的发展情况，关注行业内新技术新方法的应用，并尝试在工作中运用； 2、能够运用所学知识，举一反三； 3、不断总结自己过去和他人的实践经验，从中汲取有价值的帮助； 4、与团队成员交流和分享相关知识、经验； 5、帮助他人了解更好的学习方式和学习机会。	营造学习氛围	1，超越岗位工作需求，学习本业务及相关业务领域知识，利用内外部资源提高团队业务知识、技能； 2，注意总结团队和个人在工作中的经验，使之成为团队和个人发展的财富，营造团队持续学习的良好氛围。
	协作能力	与人协作完成本职工作	1、在项目团队内，能够合理安排个人工作计划，确保工作项保质保量完成； 2、积极识别关联工作项，发现潜在风险并及时采取措施解决； 3、有意识的与人合作，积极反馈关联工作项，配合整体项目的成功。	胜任专项负责人	1、熟悉项目管理基础知识； 2、掌握基本任务分解能力和识别关键工作项的能力，合理制定分工计划； 3、识别风险和问题，积极参与制订有效的风险规避措施或解决问题的方案； 4、能有效跟进项目的进展，主动识别项目偏差并积极采取措施及时修正。	胜任单系统项目负责人	1、能有效管理和影响各方干系人的需求期望和变更（包括重大变更和突发事件），包括单系统内部和与之相关联的外部需求，为项目的成功提供必要的条件； 2、具备指导项目团队成员保质保量完成工作的能力； 3、能够有效平衡项目需求、成本、时间、质量的关系，积极协调各方资源，确保项目成功； 4、具体良好的项目管理软技能，提高项目团队的战斗力，确保项目和产品的成功。	具备项目指导能力胜任大/多项目负责人	1、能胜任项目管理教练的角色。提炼有效的项目管理方法论，指导他人，促进他人项目管理能力的提升； 2、能胜任整个产品线的项目管理工作，有效识别多项目的优先级，协调各方资源，确保整体产品的成功。
	程序开发（开发语言和框架工具）	掌握程序开发	1、熟练使用开发工作对应的操作系统和开发程序框架等，比如linux，mysql，redis，Yii2等； 2、掌握所使用开发语言； 3、掌握常用数据结构； 4、掌握良好的编程风格； 5、掌握code reivew方法； 6、掌握代码注释技巧，遵循编码规范，了解http协议。	熟练掌握	1、能够深入了解对应的开发平台，基于现有的操作系统和开发平台快速定位和解决问题，并熟悉其类库或者插件的使用等，掌握基本配置，独立完成开发环境搭建； 2、熟练所使用开发语言； 3、熟练常用数据结构与算法； 4、熟悉面向对象，基本设计模式，了解soa，同步 / 异步，阻塞 / 非阻塞服务模型等，熟悉http协议，能够独立完成业务模块开发。	精通	1、精通至少一个类UNIX操作系统使用和应用开发，有丰富的开发经验，存储组件，框架等，对某些组件有丰富使用经验； 2、精通所使用开发语言，熟悉各类设计模式，熟悉了解业界主流和新的一些应用服务组件等，能够独立进行核心业务模块的设计架构和开发，有一定的业务简化和抽象能力，良好的代码架构能力。	全面	1、能够掌握开发框架发展趋势，对最新技术保持敏感度，跨技术栈了解其他专业的技术框架，在技术选型和布局上能够有全局性指导； 2、跨技术栈知识全面，熟悉操作系统原理和特性，在熟悉业务的基础上，能够很好的简化和抽象业务逻辑，优秀的代码架构能力，从而为整个业务开发降低复杂度和提高开发效率，降低维护成本； 3、能够根据业务特点选择合适的语言进行构建。
	高性能高可用性设计和实现	关注性能指标，和业务监控指标	关注性能指标，和业务监控指标。	能解决性能问题	能够定位性能问题并解决，了解mysql，cache等存储特性，能够根据这些特性来更好等实现业务逻辑。	设计时考虑性能	在设计阶段能够通过缓存，弱化事务，分布等提高系统负载能力，能够通过观察系统性能指标（CPU，磁盘io等）和代码正确分析和解决性能问题，设计上能够对业务做隔离，避免过载，雪崩从而影响整个系统等，深入了解系统和各存储组件等；熟悉监控点，并能主动添加，有运营意识。	掌握现有服务，了解业界方案，指导日常工作	对负责业务系统等运作过程必须深入理解和掌握，了解现有系统的可能性瓶颈点和坑，能够根据实际情况适时推动优化，了解业界最新技术方案等，可以指导日常工作，努力提高自己和所在部门等技术架构水平；推动团队和产品进行监控丰富和智能化。

专业能力	软件架构能力	了解软件架构知识	了解基础的软件开发、编译、发布、运维等知识，了解OO，SOA等基本软件架构知识。	熟悉软件架构知识	熟悉掌握优秀的软件架构思想、方法和技巧。	精通软件架构知识	1、对软件研发过程中的各种思想和方法论都能够精通掌握； 2、拥有较强的建模能力，能够通过对业务建模，将复杂的逻辑简单化呈现，并降低系统实现的复杂度； 3、能够从数据结构、接口设计等几个方面对系统进行解耦，使得系统便于分别开发、集成和测试；能对业务逻辑进行归类，能针对不同的类别业务能形成自己相应的解决方案； 4、理解基本的敏捷思想，具备快速迭代和灰度发布的能力。对软件的复用性有了解，能合理使用公共组件，不重复造轮子。	深刻理解软件架构知识推动团队成长	1、深刻理解在业务开发过程中如何运用软件架构相关的知识快速优美的解决实际问题。对于常见的业务逻辑都能熟练建模，并知道如何简化表达和处理； 2、在领域内能有通用的设计模式和规则，能将自己的经验和模式复用到整个团队，使得整个团队在方案的选取和设计方面具有一致性； 3、能够通过建模将业务系统进行层次、数据流方面的解耦，使业务能在不同团队之间进行较好的合作开发。
	业务知识	了解产品业务流程	了解所参与研发的产品，产品提供的服务和典型业务流程、触发场景。	了解产品特性和主要技术架构	了解产品的核心业务特点，并掌握相对应的后台技术架构设计和实现方法，能够评估产品的修改产生的影响。	熟悉产品的特性、需求等	熟悉产品特性、需求等，能够对复杂产品需求提供完整解决方案，对产品、架构的未来发展有一定的思考。	深刻理解产品核心理念、合理规划产品技术架构、能和产品团队一起trade off	深刻理解产品业务特性，可以给出业务解决方案及建议。针对产品的未来发展，从技术架构、技术选型方面有合理的规划，对业界的未来发展（包括产品、技术架构等）有一定了解，能够指导平时工作，能和产品团队一起进行trade off。能够主动对同类业务特性聚类分析，推动产品平台化和服务化。
	业务系统的设计与开发	能在他人指导下从事复杂业务逻辑的开发	能在他人的指导下进行复杂业务逻辑的开发，并从中学习到相关的理念和思想。	能独立进行负责业务系统子模块的设计和开发，并处理模块中的难点	能独立进行复杂业务系统子模块的设计与开发，并解决其中的重点和难点，在设计文档，数据设计和代码上清晰呈现，避免产生次生复杂业务，注意数据库等存储性能。	有复杂业务系统的设计和解耦能力，能处理模块间通讯，解决模块实现的重难点	1、能发现并解决开发过程中的重难点，能够通过合理的设计和架构较完美的解决难题；能够结合业务实际情况进行优化和重构。能够抽象并简化业务逻辑，减少业务复杂度； 2、不单纯接受产品经理的思路，能够审视和反思，并反向推动产品经理简化业务逻辑。	能反向梳理产品逻辑，设计有度，建立全团队的复杂业务能力	既不过分设计系统，也能够为将来的优化和重构预留空间；并提升全团队的复杂业务处理能力。
	质量	编写代码符合规范，能够通过常用工具排查出代码低级错误	编写代码符合规范，掌握常用的调试工具和基本的调试方法 / 自测 / 掌握基本的单元测试 / code review可以发现一些明显问题。	编写代码满足业务需求，能够通过单元测试、代码审核发现代码中的问题并提出解决方案	代码逻辑符合产品需求，能够通过单元测试，code review等可以发现一些逻辑问题。	能够编写出高质量的代码，并能编写、使用自动化测试工具保障代码质量	代码逻辑、结构清晰，能够编写高质量、可维护代码，能发现系统的结构性问题并推动解决，对测试有一定理解，能够编写、使用自动化测试代码、工具保障代码质量。	在架构和代码设计层面，具备质量意识，带领团队打造产品质量体系	架构和代码设计有质量意识，产品设计满足需求的同时容错性高，稳定可测，性能好，对测试有深刻理解，能够建立产品质量体系。
	安全	遵循规范	了解安全漏洞，遵循规范，避免出现安全事故。	了解前后台安全知识	了解安全漏洞，遵循规范，了解前后台安全知识，避免出现安全事故。	主动避免	理解安全问题背后的原因，有丰富经验，在系统设计实现时能够主动避免。	架构安全	全技术栈了解安全问题（前后台，运维，网络安全等），理解安全问题背后的原因，有丰富经验，在系统设计实现时能够主动避免，在设计架构业务时能够有隔离意识，避免安全隐患蔓延全系统。
组织影响力	方法论建设	总结归纳	能够定期（月度、季度）进行个人工作总结，与导师或者直属上级讨论，不断优化工作方法。	提炼规律	能从工作中总结与提炼共性的规律，把岗位的工作心得或案例沉淀总结并输出成果，形成可复制的经验与模式，优化工作效率。	方法论沉淀	通过标杆研究及内部实践，在本专业领域内沉淀出切实有效的方法论,并推广应用。	方法论沉淀，创新方法论与工具开发	1、具备对某个领域发展最前沿的了解和分析见解，掌握最前沿的方法论与工具； 2、能够开创性地进行一些创新性方法论与工具的研究与开发，对公司相关专业领域提升起到积极作用。
	知识传播	参与	积极参与部门内技术分享。	公司&部门内的知识分享	能积极参与公司与部门内的知识分享和交流。	公司内&跨部门的知识分享	1、积极参与跨部门的分享和交流，具有精华原创输出； 2、能为部门建立起良好的知识学习氛围。	外部的分享交流	1、能进行外部的知识分享和交流，具有精华原创输出； 2、能为公司建立起良好的知识学习氛围。
	人才培养			简单辅导	能够简单的指导或帮助同事完成工作任务	系统辅导	能够系统地辅导初级员工，指导其完成工作任务，并帮助改进工作效率和工作质量，提升能力。	主动关注人才识别与发展	1、主动关注团队人员的能力情况进行针对性的培养； 2、主动关注团队中后备人才的识别和培养； 3、在培养过程中注重传授思维理念和工作技巧。