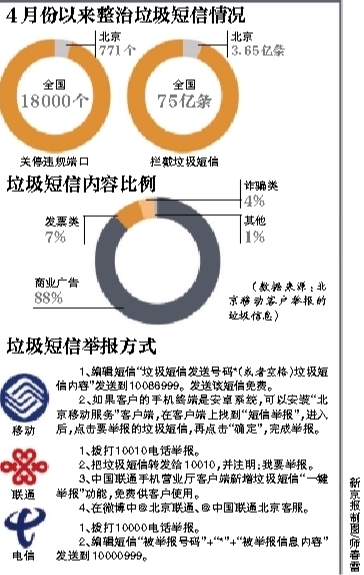
# 1背景

当前市场上主要的营销模式非常有限，市场营销对于产品或者服务非常重要，也是唯一让消费者快速认识、享用、信任、分享的一个过程，所以无论什么服务都无法离开市场营销，而宣传在市场营销当中又是重中之重。也是使消费者快速认识产品或服务的一种关键环节。而当前环境中针对中小型企业的宣传模式非常有限。无非分为两大类，传统媒体、新兴互联网媒体。传统媒体的投入较大，但是针对性不强，属于大海捞针形式的。而新兴互联网媒体逐渐成为现下越来越关键的媒体形式。

新型互联网媒体的方式也是多种多样。有短信营销、微信营销、网络广告、搜索引擎竞价排名。而其中网络广告和搜索引擎有着高投入的特点并不适合所有企业，特别是中小型企业。而短信营销并是不新型模式，但是其高投入、针对性强、直观、效果好等特点备受企业的喜爱。但由于近年来国家对于垃圾短信、营销短信的治理等原因，导致传统的短信营销这条快捷之路也不是很好走。今年4月，工信部启动了深入治理垃圾短信专项行动部署，移动、联通和电信三大企业均采取措施进行垃圾短信专项治理工作。据市通信管理局信息安全处相关负责 人介绍，专项行动启动以来，三大企业在中国关停违规端口18000个，拦截垃圾短信75亿条。其中包括关停北京违规端口771个，拦截垃圾短信3.65亿 条。12321举报中心共收到垃圾短信投诉17.3万余件次，同比下降34.2%。



从上述消息来看，通过以为短信平台的方式进行短信营销以及不可取了。现有市场上的短信平台已经无法使用，运营商一度退出的企信通业务也关闭，市场上短信平台大部分是骗子或者是触发类短信推送或者是伪基站，但是伪基站针对性不强，不予采用。

分析上述的营销模式，我们可以针对短信营销模式进行针对性优化，使其达到比以为更好的效果。现在随着智能手机的普及，以及人们对智能手机的依赖，可以让我们使用智能手机作为突破点，采用点对点的方式进行优化。

该种模式是指充分将每个人手中的智能手机利用起来，通过大批量的终端接入来消化短时间发送大量短信的目的。当然我们也要对发送的短信进行合法性过滤，严谨发送违法、色情等违反国家法律法规的信息。我们只针对营销短信进行发送。

# 2架构设计

1. **简介**

* 范围

本文用于定义SmsSender短信营销平台第一个版本的设计。用于生产指导第一个版本的开发、部署、以及测试工作。

* 目的

本文档是SmsSender1.0.0版本的设计说明书，用于指导版本的开发、测试与资料写作。

本文档的读者为项目开发组、测试组、资料组的成员。

1. **概述**

* 大版本描述

大版本通常用于大的版本，从用户功能角度上讲有较大的用户体验改善，通常周期较长。通常对用SmsSender1.0.0中的1。

* 中版本描述

中版本通常用于交付一个版本中有较大的系统架构或者是组网方式的变化，但是对于用户来说是没有太大的用户体验上改变，通常会有大版本的发布，通常大版本的周期为一个月左右。通常对用SmsSender1.2.1中的2。

* 小版本描述

L版本通常为迭代版本，用于将用户的大量需求分周期进行交付，以此较低交付风险，满足按期、高质量的交付。L版本的周期较短，通常为1~2周。通常对用版本中的SmsSender1.3.4中的4。

1. 业务简述

随着越来越多的营销需求和电信运营商国家政策法规对垃圾短信的治理，常规的短信营销已经不可取。退出SmsSender智能终端的普及，接入大批量终端实现短时间内发送大量短信的需求。

1. 系统总体描述

* SmsSender网络位置



说明：

**CTMS：**特指内蒙古本地定制终端管理系统，专用于弥补统一版定制终端管理系统无法适用于内蒙古地区的终端销售流程而定制，也是本次交付的重点。

**CTRM：**集团定制终端营销管理系统。

**CBOSS：**负责OpenChannel和CTRM之间消息转发器。

**OpenChannel：**统一版定制终端管理系统，主要负责定制终端数据管理，CTMS需要考虑和OpenChannel的结合。

**OA：**内蒙古地区定制终端管理系统登入系统，营业员可以通过OA系统登入CTMS进行定制终端的管理。

**PC：**各销售渠道和角色可以通过PC登入CTMS进行定制终端相关业务操作。

**WorkStation：**各销售渠道和角色可以通过工作站登入CTMS进行定制终端相关业务操作。

**Mobile：**各销售渠道和角色可以通过移动设备登入CTMS进行定制终端相关业务操作。

1. 系统上下文



说明：

**Web：**通过W3C，浏览器形式向各销售渠道和角色展示CTMS系统的重要业务功能。

**BPS：**CTMS核心业务处理模块，以RestFul/Json形式发布CTMS核心服务，共Web、Wap和移动终端调用，完成统采流程、供货商采购、集采流程、铺货流程、申领调拨、审批流程等，所有Web端接入的业务逻辑处理，以及和CTRM、OpenChannel之间的交互。

**DMS：**CTMS后台管理模块，通过W3C浏览器形式完成CTMS数据的管理，通常包括配置参数管理、用户信息角色定义、产品库维护、信息发布、安全库存维护等。

**TTS：**CAB数据库分库服务器。存储记录数大于亿条的表，包括各种联系人信息表、联系人分组表、联系人及分组对应关系表、用户订阅表、用户授权表等。向所有的DSServer、CABServer和CABTimer开放数据库访问JDBC接口。

**OpenChannel**：统一版定制终端管理系统，CTMS通过调用其标准Web Service接口完成同CTRM之间的数据交互以及数据入库功能。

**Browser**：用户通过浏览器访问CTMS Web模块和DMS模块，分别用于定制终端销售业务操作和CTMS数据管理操作。

**Mobile**：用户通过手持设备或移动终端接入访问CTMS BPS，分别用于定制终端销售业务操作和CTMS数据管理操作。

**Report**：报表系统，通过FTP接口从CTMS获取标准数据话单，根据数据话单生成报表。

**CTMSMaterDB**：CTMS主数据库，存放CTMS基础表，数据量较小的表，主库一个CTMS实例有且仅有一个主库。

**CTMSSlaveDB1…n**：CTMS分数据库，存放CTMS业务表，数据量较大的表，可以有1到n个，可以实现方便扩展。

# 3系统设计

主要描述系统所功能构成，主要的阅读者为开发人员和测试人员，开发人员应当以该部分为依据进行SRS需求分析，测试人员应当以该部分为标准进行测试方案和测试用例的写作。该部分的描述主要分为以下几个章节：系统架构介绍整个系统的开发使用框架和软件环境，系统开发状态/类型主要描述对应该版本的产品设计规格的系统开发状态。系统运行概念是指按照逻辑功能来区分进行大致范围的界定，并且定义了最初的系统模型，开发人员和测试人员的用例和代码都应该以此组织结构为依据进行用例写作或是开发以及管理。

1. BPS系统架构



* BPS架构

说明：BPS基于Java开发，作为CTMS系统的核心模块，该模块复杂所有的Web端，以及和OpenChannel之间的交互以及业务逻辑处理，应当以提高稳定性、高可用性、高性能为目标，BPS可以通过对等集群和双击设计达到高可用性和高性能。BPS设计应当考虑接口的一致性和代码结构的分层架构，以此提高BPS的可维护性和对外接口的一致性。

1. Web系统架构



Web架构

说明：Web基于Java开发，采用Apache Tomcat作为其J2EE应用服务器，是CTMS系统的一种接入方式，主要承载W3C浏览器的接入处理，Web模块需要考虑到权限控制、参数校验，业务逻辑处理交给BPS。Web面对较大压力时，也可以采用对等集群的方式进行水平扩容，但是要考虑更加复杂的问题Session管理（这一块还不成熟，后续补充）。

1. DMS系统架构



Web架构

说明：Web基于Java开发，采用Apache Tomcat作为其J2EE应用服务器，每个运行实例都是一个JVM单进程，在小容量局点中采用双机方式部署，在大容量局点中采用N+1集群方式部署，并且每个运行实例提供相同的功能，可以处理任意压力的业务请求。

1. TTS系统架构

s

TTS架构

说明：TTS基于Java开发，采用Apache Tomcat作为其J2EE应用服务器，在CTMS系统中有且仅有一个实例，由于定时任务一般对数据库的扫描是全表扫描，所以对于TTS更重要的是定时任务的执行时间和周期的控制，对于实时性要求不高的定时任务最好都是在系统闲时（每天晚上）执行，对于实时性要求较高的定时任务，采用间隔扫描处理的方式，定义最大处理批量，提高单笔的吞吐量，尽量减少数据库连接。

1. **需求分析**

需求分析阶段主要是针对客户提出的需求进行需求分析，还要完成概要设计、项目计划、SRS（需求分析文档）文档输出，输入主要是该迭代过程中需要完成的客户需求。目前我们开发的现状基本都是先动手写代码，容易造成由需求理解不透彻或不完整引起的反工。其实需求分析应当放入编码之前，这样开发人员可以不用关心具体实现，精力放到需求分析上，以及测试人员、客户等多方人员的参与评审，可以将需求了解更透彻，不容易发生偏差。

1.项目计划

根据客户需求进行任务划分、人员安排、时间评估，整个迭代生命周期严格按照此计划执行，后续的SRS、编码、上线、维护都应该遵从本计划定位的范畴。

2.SRS写作

SRS写作非常重要，这个是作为开发、测试后续流程共同输入，SRS写作质量关系到需求是否分析清楚，完成的功能是否能够基本满足客户需求。要提前制定SRS文档写作规范。

3.SRS评审

SRS评审阶段以SRS文档作为输入，产出SRS评审文档，以及对SRS的修改。其中测试人员的评审必须要具备，测试人员主要从可测试性、可炎症性、可维护性、可扩展性等方面对SRS文档进行评审。

4.测试方案写作

该过程是以SRS终稿作为输入，以测试用例作为输出。我们现在这块比较欠缺，测试没有形成规范，测试方案和功能测试文档几乎都是开发人员完成，这样的测试文档质量难以保证，也会造成后期测试的不充分。测试用例作为后续功能点的稽核标准，很有维护的价值和意义。

## **开发过程**

该过程中，开发人员主要工作是编码、单元测试、自测，主要输入为SRS终稿文档、以及一些复杂需求的概要设计、详细设计，输出为可测试的完整功能代码。测试人员主要工作是测试用例写作，主要输入为测试方案、SRS终稿文档以及一些复杂需求的概要设计、详细设计。

1.编码/测试用例写作

需要制定统一的编码规范，以及静态错误清零，以避免长期以往的技术债务。

2.持续集成

开发工程中应当结合持续集成，持续集成结合checkStyle、FindBugs、圈复杂度检查，提交触发构建以及定时构建。将错误尽可能早的发现，因为越早发现，修复错误的成本越低。在就是通过持续集成构建持续增强开发人员对版本质量的信心。

测试报告

3.代码Review

视代码量而定，代码量大采用代码抽检，发现问题责令其修改相关一系列问题，代码量少，可采用代码互检。该阶段输入为代码，输出为CodeReview报告。

4.提供版本

## **测试阶段**

该阶段以版本作为输入条件，最终输出测试报告，以及缺陷。该阶段测试人员主要工作为依照测试用例进行功能测试，产出测试报告，以及BUG。开发人员针对BUG进行修复。该阶段BUG数可作为对开发人员的考核。

## **发布/上线**

该阶段主要是发布版本后进行升级上线，以及上线后测试工作。

## **迭代总结**

每个迭代结束后都应当对本次迭代展开总结，总结可有多种形式，版本质量、缺陷回溯、案例输出、管理经验输出，可以将一些好的做法加强、宣贯，持续到下个迭代，不好的提出规避和解决措施，以免下个迭代再犯。常此以往将会形成良性循环，经过多次迭代后形成适合自己本地项目的开发管理经验。主要输入为该次迭代story，输出为项目总结、案例总结、管理经验、各个环节规范文档。

# 3本地化项目生命周期

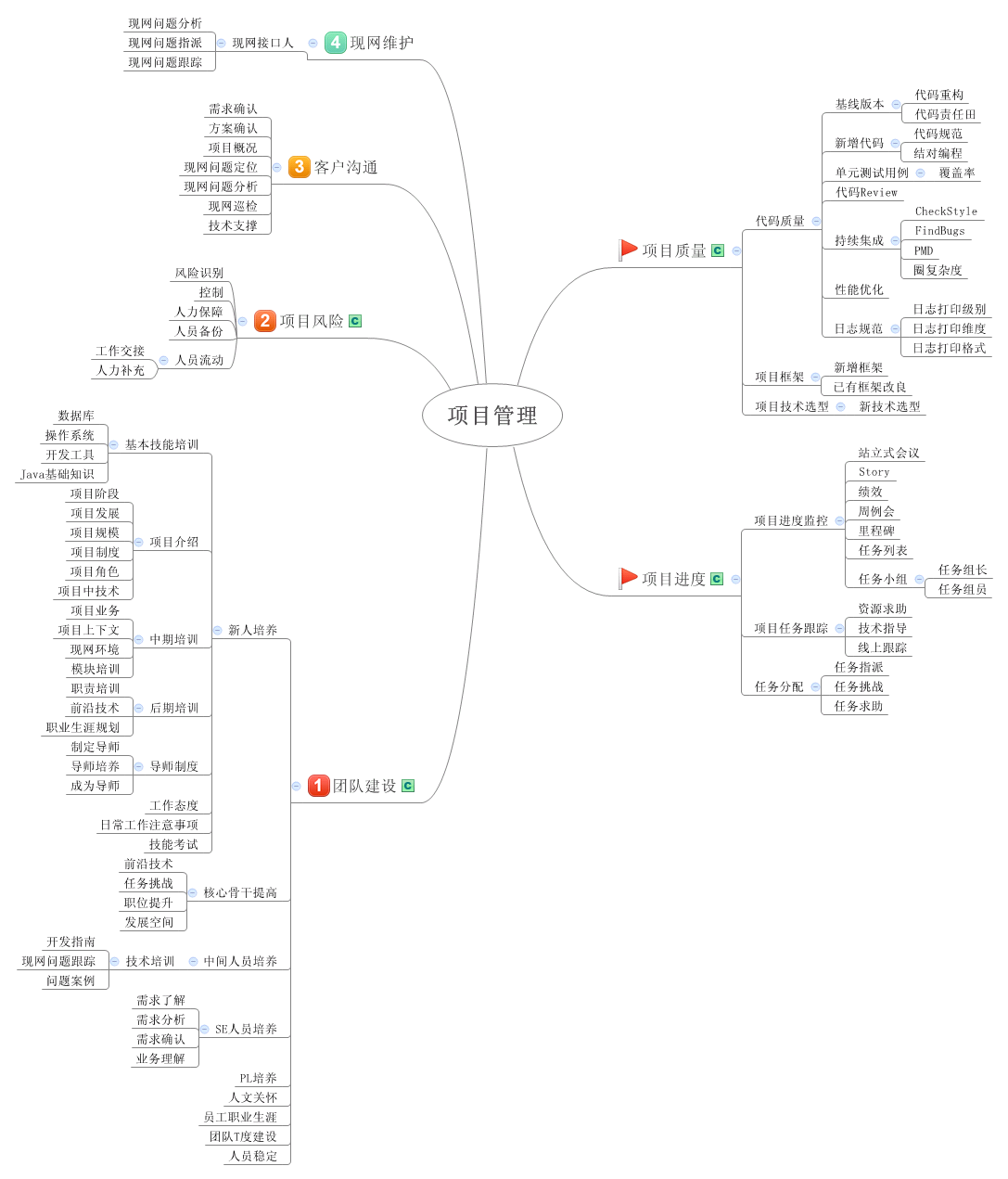
针对我们应当制定出迭代计划，该迭代计划以一个月为周期进行时间划分，以一个11月为例，时间划分。对于生命周期环节这里没有进行过多的裁剪，这里认为这是一个开发项目所必须具有的生命周期，所以无法再裁剪，但是时间划分可根据实际情况在调整。



根据以往经验开发和测试比例应当达到1:1比较，其中开发中需求分析和编码工作比例要达到1:1或1：1.5，才能够保证版本质量，但是考虑到现场发布/上线工作量比较大，所以讲发布上线时间调整，压缩整体的测试开发比较，但是相对开发和测试比较保持不变。

项目生命周期帮助我们更好的了解项目进度，但是项目质量保证还是需要另一方面项目管理的。项目生命周期相当于横向的那么项目管理则是纵向，穿插在整个项目生命周期中，时刻保证项目质量、团队建设。

# 4项目管理



项目管理是需要在日常工作中，每时每刻都要进行的，零散的充斥到整个项目生命周期中，更多的是需要管理者是组织、推广、策划，组员配合实施。要求对管理者要有一定的管理经验。

## 项目质量

项目管理中很重要一部分工作在于提高版本质量，稳定现有功能的基础上，保证新增功能的质量，逐渐的引导客户，方可持续发展。

项目质量工作不仅仅是管理者的工作，需要组员充分参与，调动每一个员工的积极性，才能够做好项目质量工作。

我认为管理者的职责在于发起一些质量保证措施，制定一些流程、规范，并宣贯、推广、执行，对管理者的执行力有较大的考验，因为新的事物往往不容易被接受，或者制度都代表束缚，要想做好这部分的工作，需要组员充分参与，制定制度、规范得到组员的认可，充分采纳认可组员的建议和意见，这样才能从根本上解决组员对制度、规范的反感。

## 项目进度

项目进度更多的是从生命周期处去把控，每个环节应当在时间结束，每个环节应当在什么时候开始，这些都是要非常清楚，并且在管理的日常工作中强调，让组员建立生命周期的概念，对于自己的工作任务的进度延迟提高警觉性。除了要了解整体进度外，还需要针对每个组员的进度进行充分的了解，并了解其工作量是否饱和，是否存在阻塞性问题或困难，以便及时协调资源进行解决，避免问题拖到后期带来的灾难性后果。

1.项目进度监控

项目进度监控，可以通过站立式会议、Story、周例会等多种渠道进行了解，其中站立式会议尤其明显。每天进行10分左右的站立式会议，可以充分了解到各个组员的进度以及困难，以此来看整体进度是否有延迟。周例会主要针对整体进度进行知会。以及项目过程中的问题进行讨论。

2.任务分配/跟踪：

## 团队建设

团队建设是管理者很重要的一部分工作，管理者应当将重心放于此处，因为此处关乎到人员稳定性，团队整体能力，组员的能动性等等多个方面，这些方面往往能够决定一个项目的成败。团队建设主要分为新人培养、核心骨干提高、中间人员培养。团队建设应当注意T度，合理搭配人员，一般核心骨干1到2个，中层人员3-6个，新员工3-6个。这样形成一个良好的团队T度，每个阶层的组员都可以找到自己的目标以及优越感。

新人培养

新人的培养是很关键的一步，如何能够快速的让新人投入生产，如何快速的让新人进入状态，都很关键，新人的培养一方面是作为储备，一方面是项目组注入新鲜血液。新人培养工作，可以制定一些制度和措施，具体实施可以由管理者或核心骨干人员负责具体实施。

|  |  |
| --- | --- |
| 基本技能培训 | 数据库 |
| 操作系统 |
| 开发工具 |
| Java基础知识 |
| 项目介绍 | 项目阶段 |
| 项目发展 |
| 项目规模 |
| 项目制度 |
| 项目角色 |
| 项目中技术 |
| 中期培训 | 项目业务 |
| 项目上下文 |
| 现网环境 |
| 模块培训 |
| 后期培训 | 职责培训 |
| 前沿技术 |
| 职业生涯规划 |
| 导师制度 | 制定导师 |
| 导师培养 |
| 成为导师 |
| 工作态度 |  |
| 日常工作注意事项 |  |
| 技能考试 |  |

核心骨干提高

黑心骨干的提高也是非常重要的，通常针对个人情况进行刺激性其工作积极性，可以通过前沿技术、疑难杂症解决、有挑战的任务、发展空间、职位提升等几个方面进行。

3中间人员培养

中间人员更多的任务完成者，大部分的代码出于于他们的手，所以对于中间人员的技能提高、工作积极性的调动是非常重要的。可以参考核心骨干的提高。

## 项目风险

项目管理过程中，需要有一定的项目风险识别意识，以及对面风险及时采取措施，避免酿成大祸。大风险规避方面我们应当注意几个方面：风险识别、控制、上报、人力风险、

针对于人力风险，我们应当将工作做在平时，形成良好的人员备份，以免人力流动造成项目的损失。另外，面对风险，当超出自己的能力范围内，应当及时上报领导，以便第一时间商讨对策。