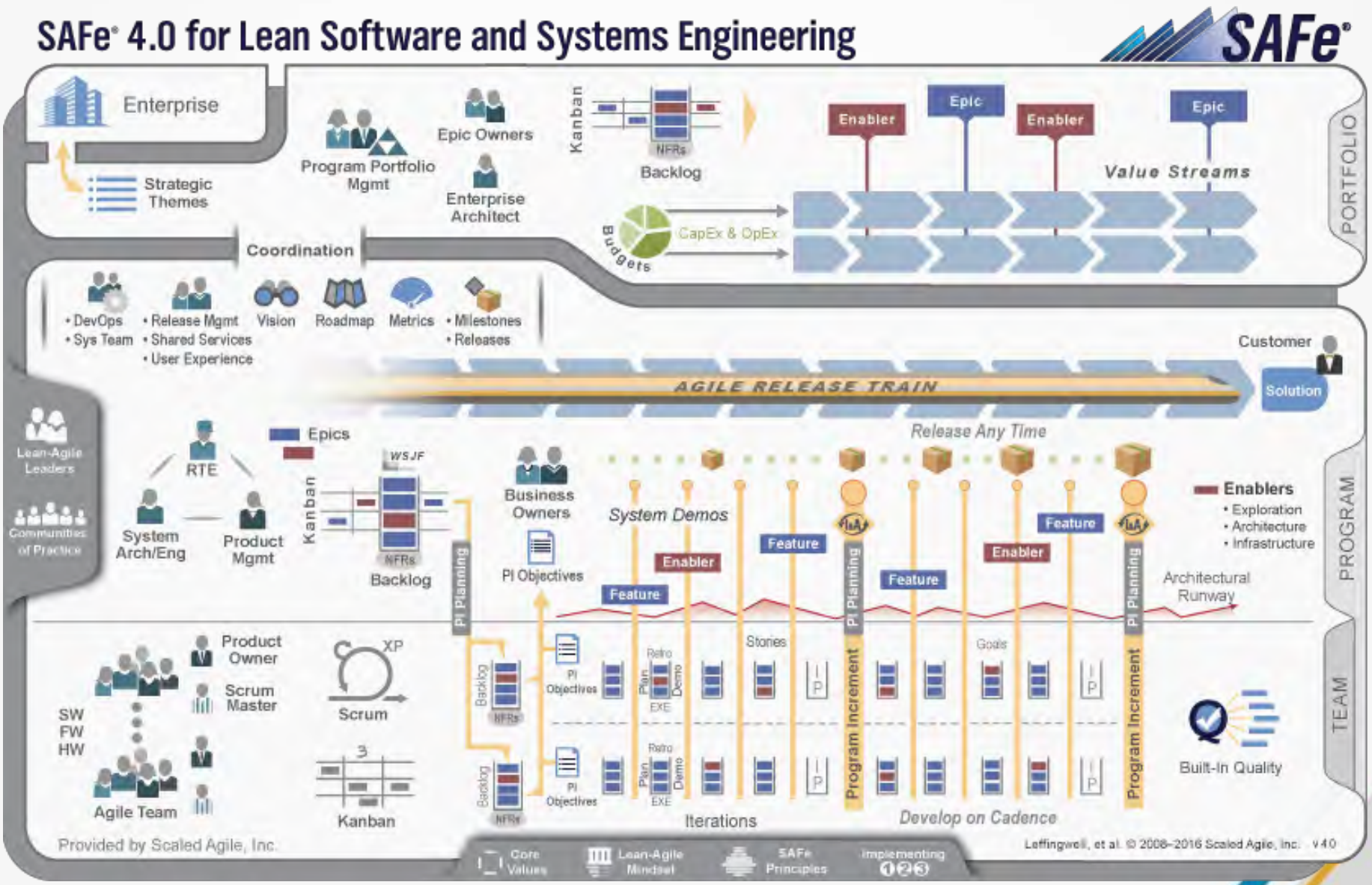
TiD 2017
质量竞争力大会

重构：

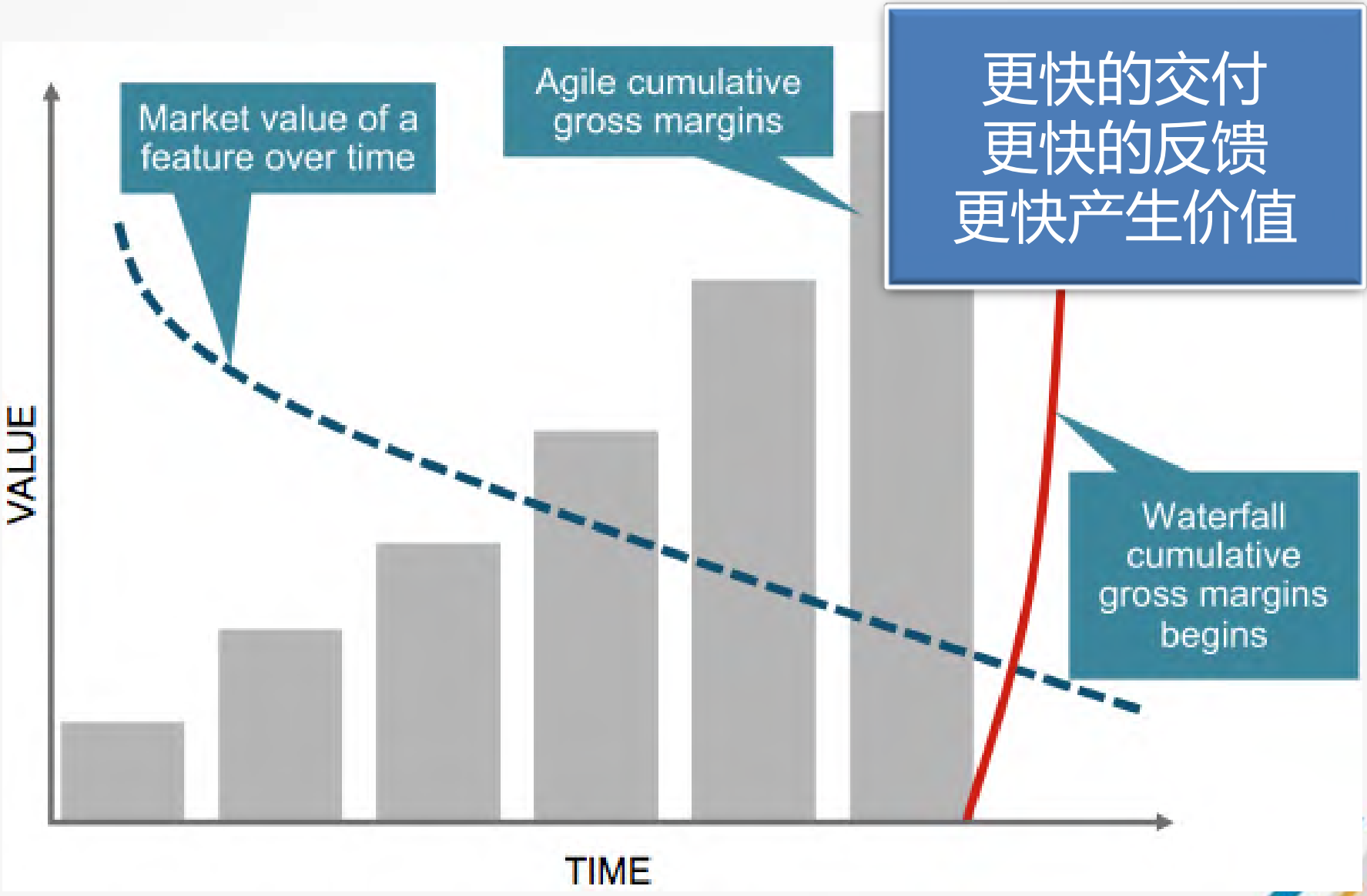
高质量产品的保证

范钢

TiD2017 SAFe规模化敏捷

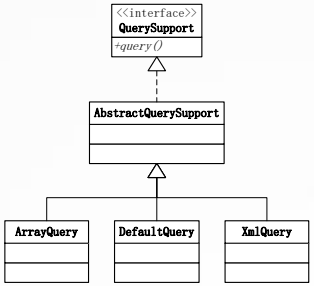


站在经济的角度看问题

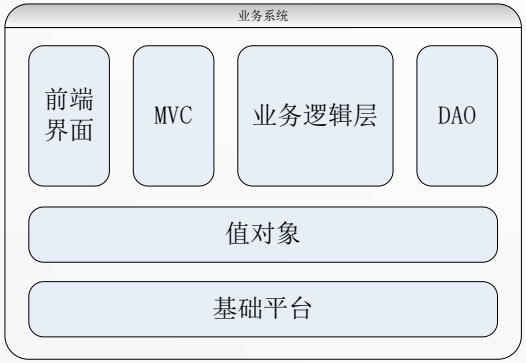

Market value of a feature over time
Agile cumulative gross margins
更快的交付
更快的反馈
更快产生价值
Waterfall cumulative gross margins begins
VALUE
TIME

TiD2017 传统的软件开发模式

需求分析



全面的设计



分层分模块

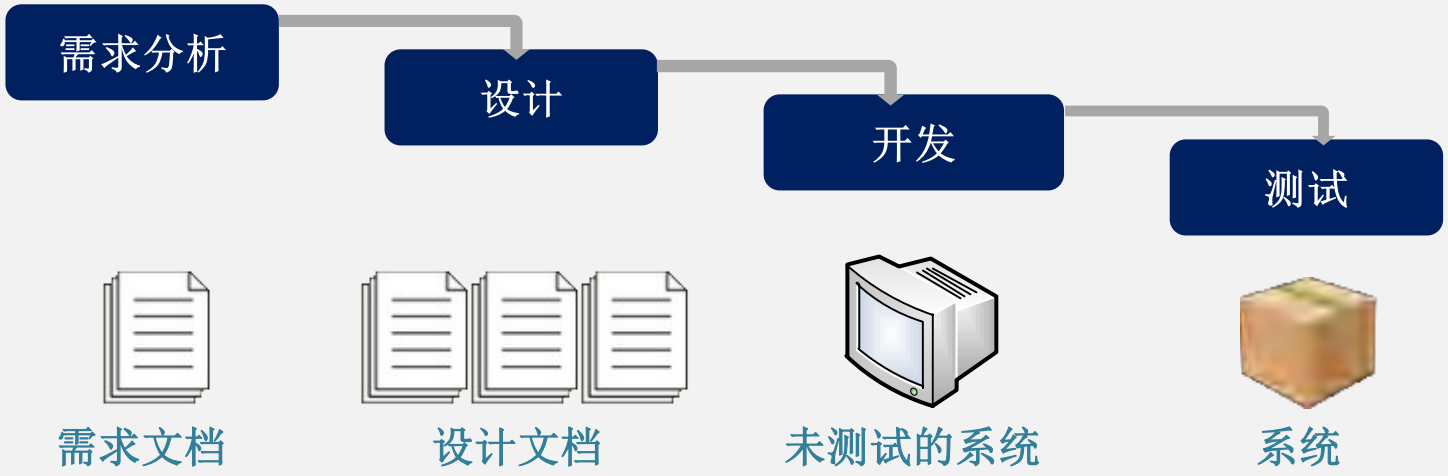
开发编码

集成组装

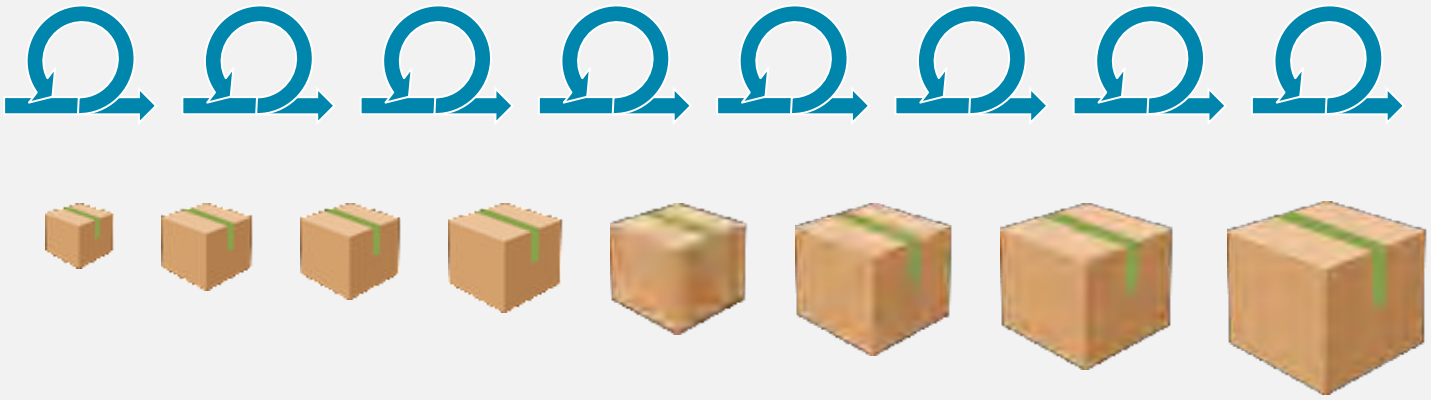
运行调试

敏捷软件开发

传统开发模式



敏捷开发模式



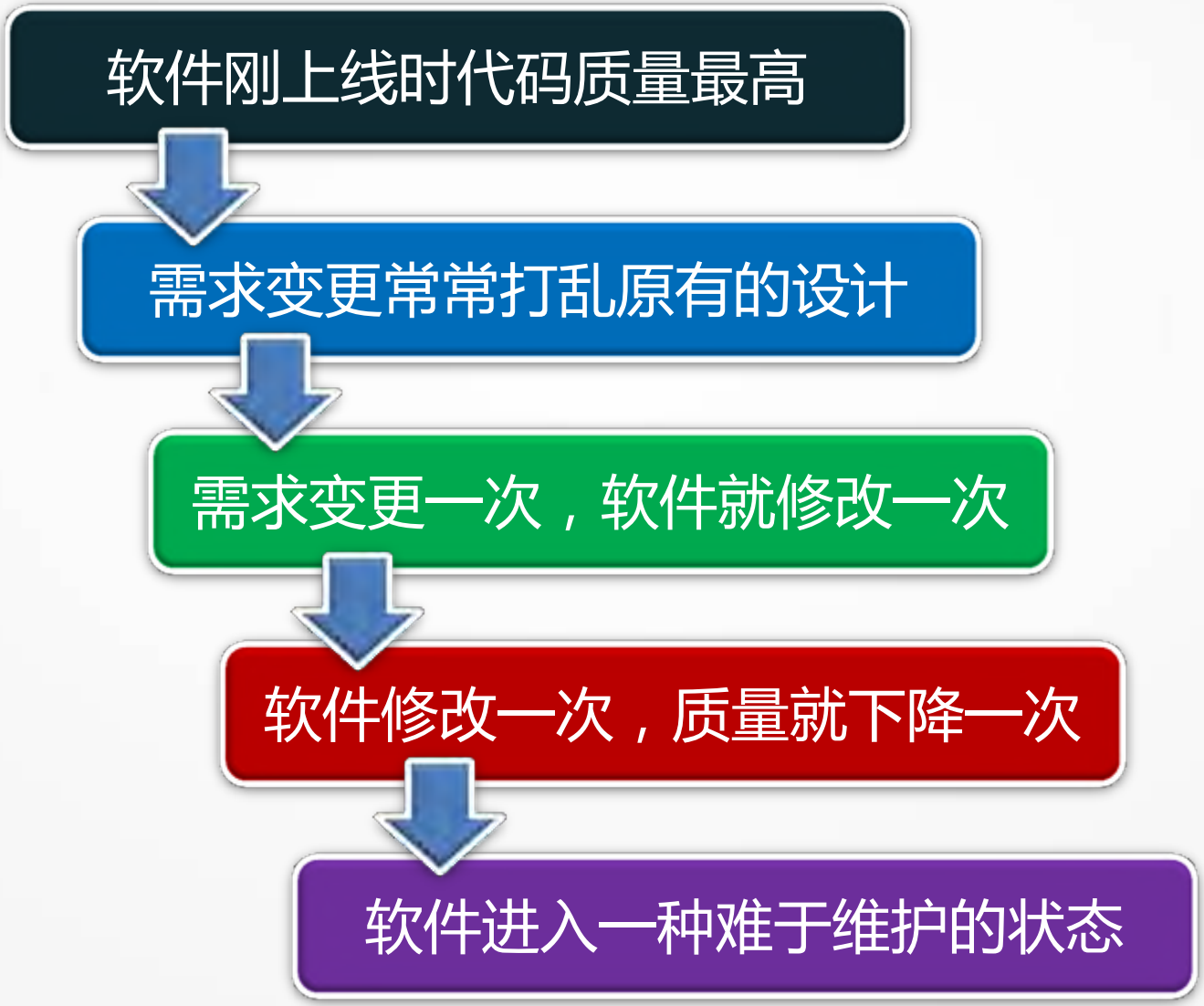
企业敏捷实践最大挑战

频繁地软件变更

Vs.

快速地软件退化

频繁的变更带来软件质量的下降



结论与思考

不解决软件变更的质量问题
敏捷开发就不能真正落地

频繁地需求变更

是软件退化的根源吗？

TiD2017 电商网站付款功能

```
public boolean payoff(Order order, String addressI
    //收集客户信息
    String customerId = SessionService.getUserId()
    Customer customer = dao.getCustomer(customerId
    order.setCustomer(customer);
    Address address = dao.getAddress(addressId)
    order.setAddress(address);

    //计算付款
    double amount = 0;
    for(OrderDetail detail : order.getDetails()) {
        amount += detail.getQuantity()*detail.getPrice();
    }
    order.setAmount(amount);
    Serializable orderId = dao.save(order);

    //跳转支付页面
    WebserviceFactory factory = new WebserviceFactory();
    PaymentWebservice payment =
        (PaymentWebservice)factory.getWebservice("aliPayment"
    return payment.payoff(orderId, customer, amount);
}
```

VIP会员：
- 金银卡会员
- 会员福利
- 会员特权

秒杀

预订

闪购

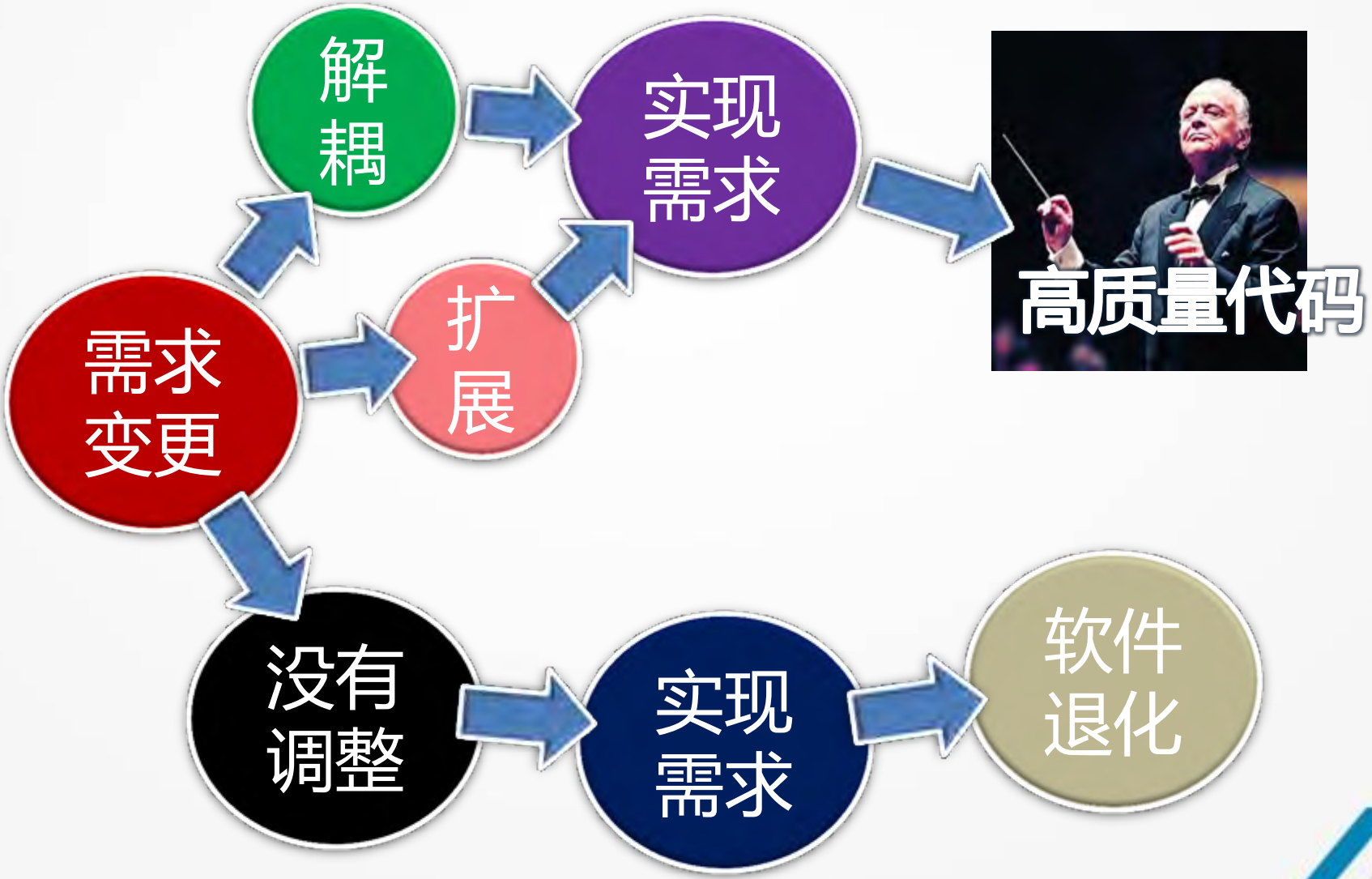
众筹

品折扣：
- 限时折扣
- 限量折扣
- 某类商品折扣

返券

支付方式：
- 支付宝
- 微信
- 银行卡

需求变更是软件退化的根源吗？



用慢动作的手法重新演练一次

- 需求变更到来时应当怎样应对？
- 正确的设计过程是什么？
- 什么才是高质量的代码？

TiD2017 电商网站付款功能

```
public boolean payoff(Order order, String addressId) {
      //收集客户信息
      String customerId = SessionService.getUserId();
      Customer customer = dao.getCustomer(customerId);
      order.setCustomer(customer);
      Address address = dao.getAddress(addressId);
      order.setAddress(address);

      //计算付款
      double amount = 0;
      for(OrderDetail detail : order.getDetails()) {
            if(hasTimeLimited(detail)) {
              //如果有限时抢购，此处省略100字
            } else if(hasQuantityLimited(detail) {
              //如果有限量抢购，此处省略100字
            } else if(hasClassifyDiscount(detail) {
              //如果有某类商品的折扣，此处省略100字
            } else if(hasDiscount(detail) {
              //如果普通商品折扣，此处省略100字
            } else {
              amount += detail.getQuantity()*detail.getPrice();
            }
      }
      order.setAmount(amount);
      Serializable orderId = dao.save(order);

      //跳转支付页面
      WebserviceFactory factory = new WebserviceFactory();
      PaymentWebservice payment =
            (PaymentWebservice)factory.getWebservice("aliPayment");
      return payment.payoff(orderId, customer, amount);
}
```

商品折扣：

- 限时折扣
- 限量折扣
- 某类商品折扣
- 普通商品折扣
- 不折扣

传统的系统维护方式

项目不断在维护中

已经打了无数补丁

补丁还在继续...

程序已经凌乱不堪

总是以需求驱动

总是使修改最小化

不能从全局去思考

两顶帽子的设计方式

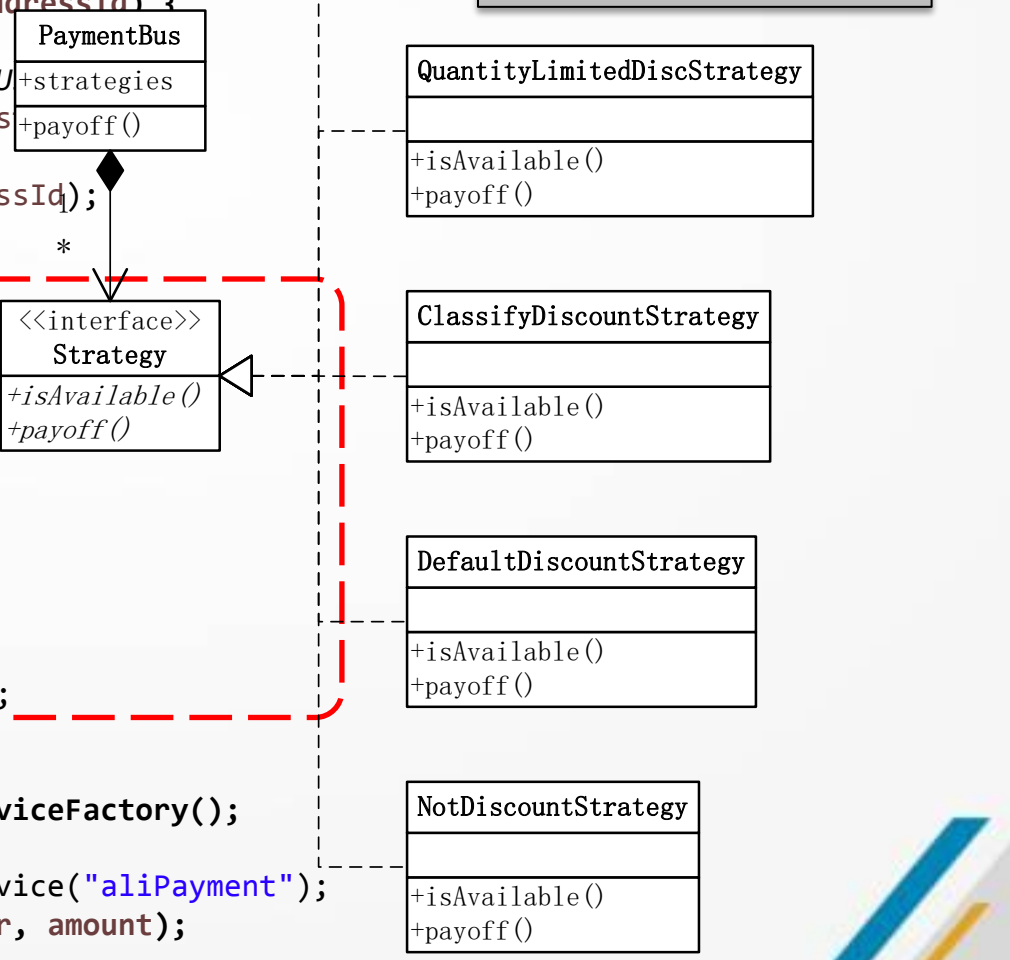
1. 在不添加新功能的前提下，
   重构代码，以适应新功能

2. 实现新功能

TiD2017 电商网站付款功能

```
private List<Strategy> strategies = null;
public List<Strategy> getStrategies() {
     return strategies;
}
public void setStrategies(List<Strategy> strategies) {
     this.strategies = strategies;
}
public boolean payoff(Order order, String addressId) {
     //收集客户信息
     String customerId = SessionService.getU
     Customer customer = dao.getCustomer(cus
     order.setCustomer(customer);
     Address address = dao.getAddress(addressId);
     order.setAddress(address);

     //计算付款
     double amount = 0;
     for(OrderDetail detail : order.getDeta
       for(Strategy strategy : strategies)
         if(strategy.isAvailable(detail))
           strategy.payoff(detail);
       }
       amount += detail.getAmount();
     }
     order.setAmount(amount);
     Serializable orderId = dao.save(order);

     //跳转支付页面
     WebserviceFactory factory = new WebserviceFactory();
     PaymentWebservice payment =
       (PaymentWebservice)factory.getWebservice("aliPayment");
     return payment.payoff(orderId, customer, amount);
}
```

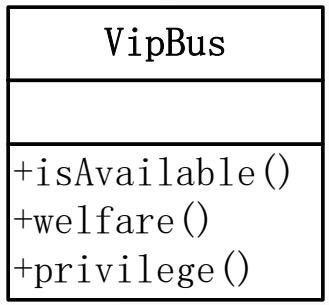
商品折扣：
- 限时折扣
- 限量折扣
- 折扣
- 折扣

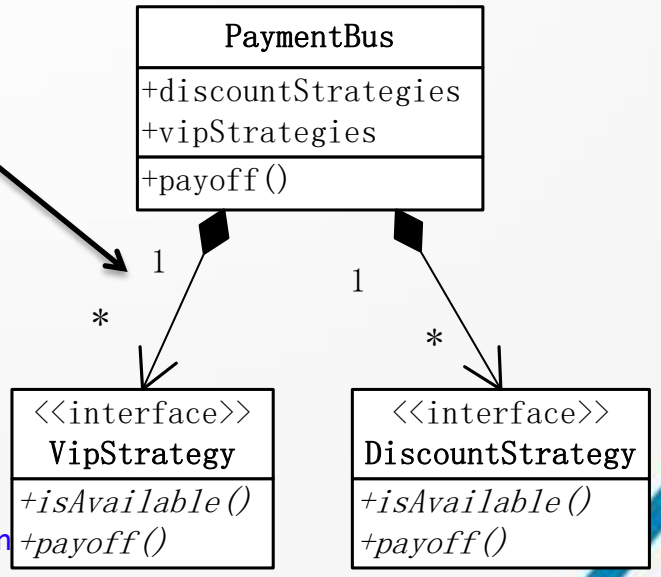


TiD2017 电商网站付款功能

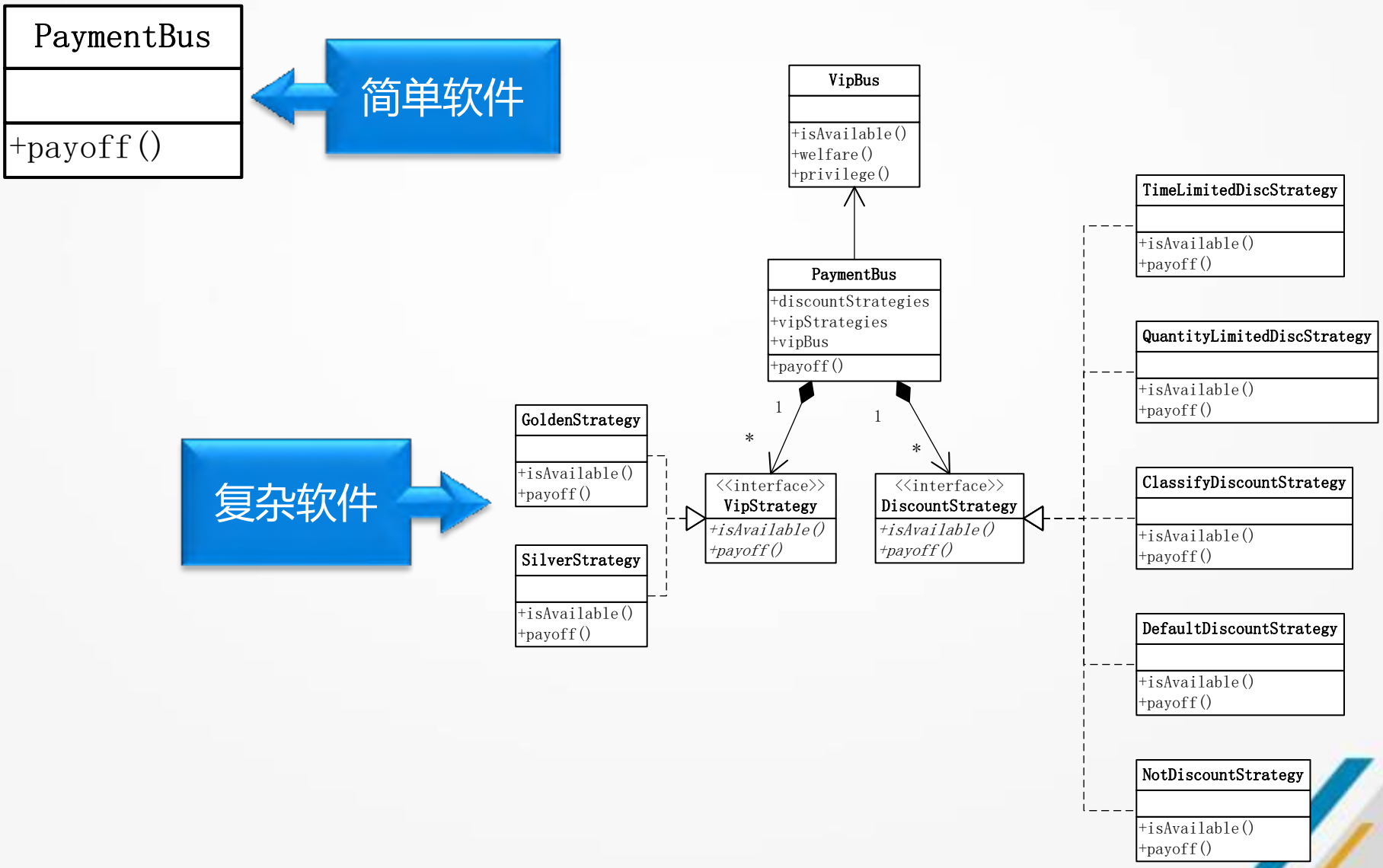
VIP会员：
- 金银卡会员
- 会员福利
- 会员特权

```
public boolean payoff(Order order, String addressId) {
    //收集客户信息
    String customerId = SessionService.getUserId();
    Customer customer = dao.getCustomer(customerId);
    order.setCustomer(customer);
    Address address = dao.getAddress(addressId);
    order.setAddress(address);

    //会员福利
    if(vipBus.isAvailable(customer))
      vipBus.welfare(customer);

    //计算付款
    double amount = 0;
    for(OrderDetail detail : order.getDetails()) {
      for(VipStrategy strategy : strategies) {
        if(strategy.isAvailable(customer))
          strategy.payoff(customer);
      }
      for(DiscountStrategy strategy : strategies) {
        if(strategy.isAvailable(detail))
          strategy.payoff(detail);
      }
      amount += detail.getAmount();
    }
    order.setAmount(amount);
    Serializable orderId = dao.save(order);

    //跳转支付页面
    WebserviceFactory factory = new WebserviceFactory();
    PaymentWebservice payment =
      (PaymentWebservice)factory.getWebservice("aliPaymen
    return payment.payoff(orderId, customer, amount);
}
```





TiD2017 简单软件 vs. 复杂软件

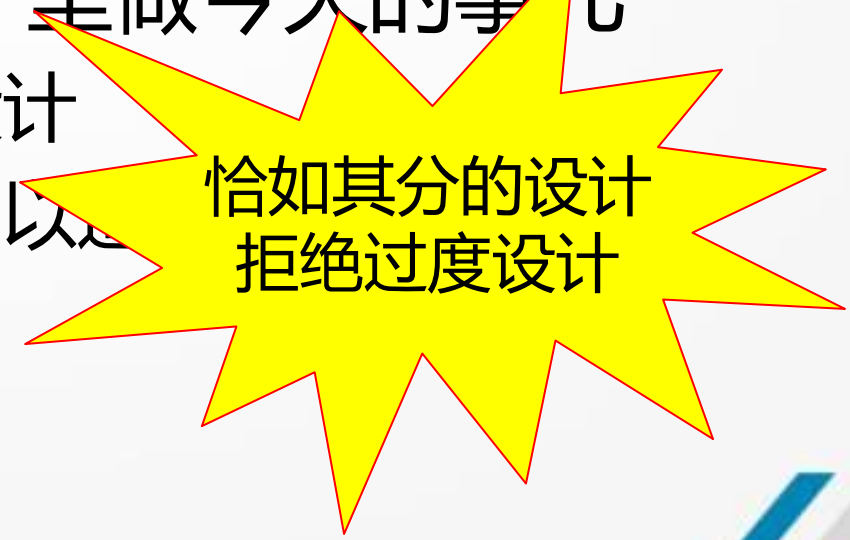


TiD2017 重构是软件发展的必然

- 结论
  - 软件总是由简单软件向复杂软件转变
  - 简单软件有简单软件的设计
  - 复杂软件有复杂软件的设计
  - 当软件开始由简单软件转变为复杂软件时我们就需要

TiD2017 两顶帽子的意义

- 过去，我们是这样设计的：
  - 总是担心未来的变更
  - 总是为未来的设计埋单
  - 总是在过度设计
- 今天，活在今天的格子里做今天的事儿
  - 只为当前的需求进行设计
  - 当未来变更时，先重构以适
  - 然后添加新的需求

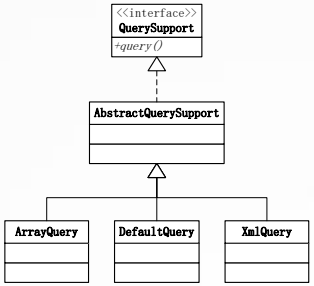


高质量敏捷开发

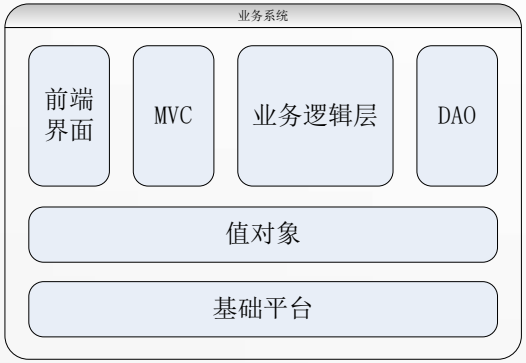
以往开发为何敏捷不起来
小步快跑的开发过程
我们在工程中的实践

我们以往是这样开发的...

需求分析



全面的设计



分层分模块

开发编码

集成组装

运行调试

传统开发模式的问题

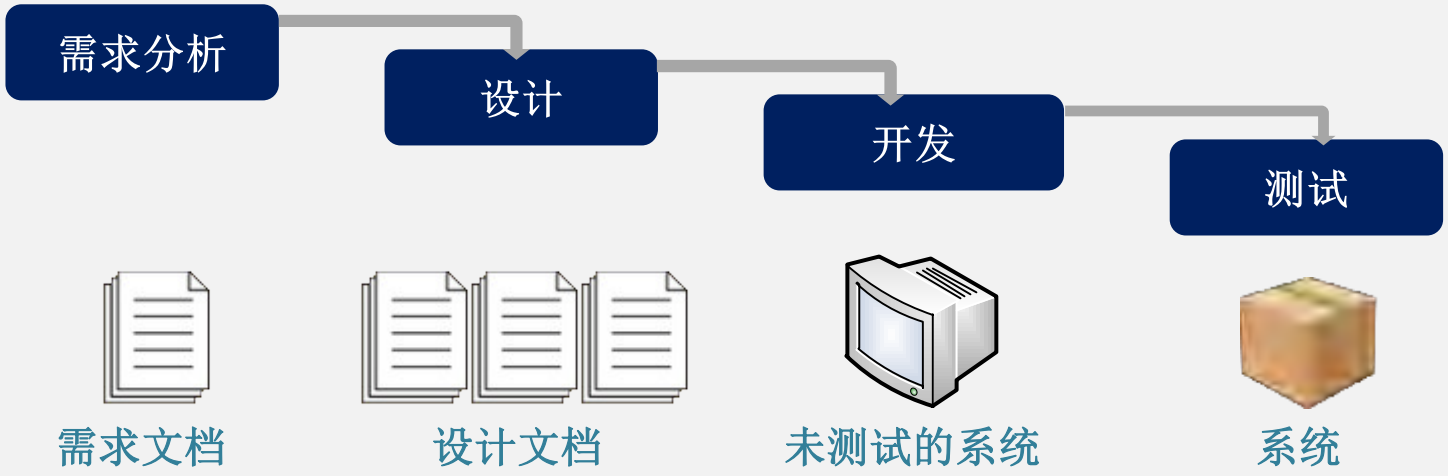
- 开发模式存在巨大风险
  - 传统开发模式不能立即给予验证
  - 系统集成、运行调试与问题定位都十分困难
- 各项功能不能得到有效解耦
  - 很容易做出复杂设计甚至过度设计
  - 很容易设计出大对象与大函数
  - 不能对各项功能拆解后进行验证
  - 为日后的变更带来设计难题

小步快跑的开发模式

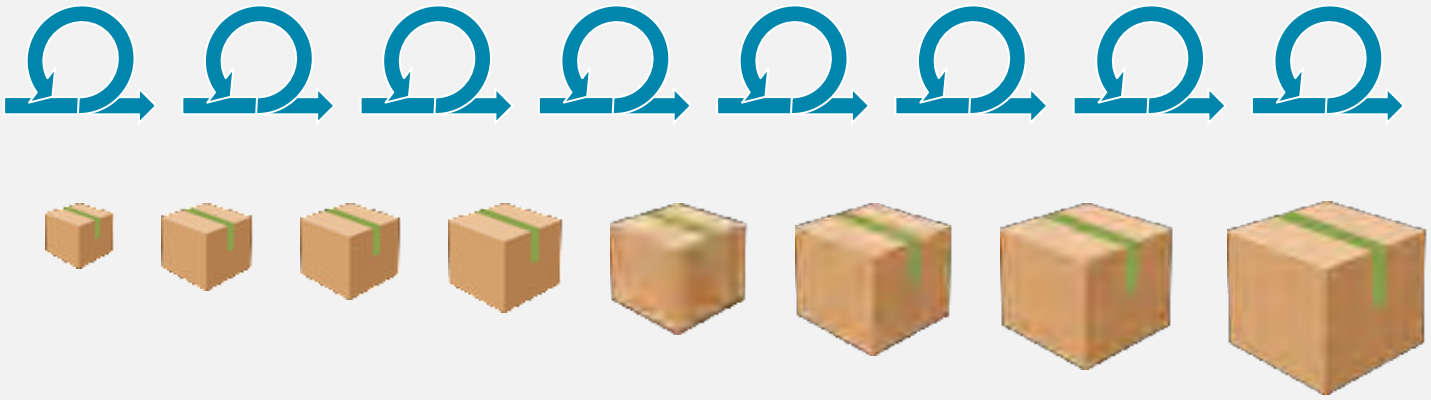
- 用最快的速度开发一个最核心的功能
- 让第一个版本运行起来并可以验证
- 在第一个版本的基础上不断添加新功能
- 每次只添加一个很简单、很单一的功能
- 对添加的功能进行验证并使其可运行
- 每个开发周期只有10分钟到半小时
- 每次添加新功能时只进行恰如其分地设计
- 下次添加新功能时如果需要应当适时进行重构
- 复杂的系统是由一次次正确的开发积累而成

小步快跑的开发模式

传统开发模式



小步快跑开发模式



我们是这样做的...

解决前沿技术开发的难题
实现高质量的敏捷开发
应当需求变更与技术变革

项目背景

- 我们做了基于税务数据的分析多年
  - 纳税评估
  - 风险控制
  - 绩效考核
- 我们有一个基于增值税分析的BI系统
  - 对增值税发票来源地与流向地分析
  - 对增值税虚开票进行风险控制
  - 从行业与地域等维度进行分析统计

TiD2017 增值税发票来源地/流向地分析

互联网转型为该项目带来新的机遇

- 网络实时开票
  - 可以更加实时地获得开票数据
  - 实现实时的数据分析与展现
- 电子底账
  - 更加准确与实时地获得跨省数据
  - 实现跨省的发票虚开风控管理
- 营改增项目
  - 可以获得更加全面的数据
  - 更加全面地展现各地区的经济状况

互联网转型为该项目带来新的挑战

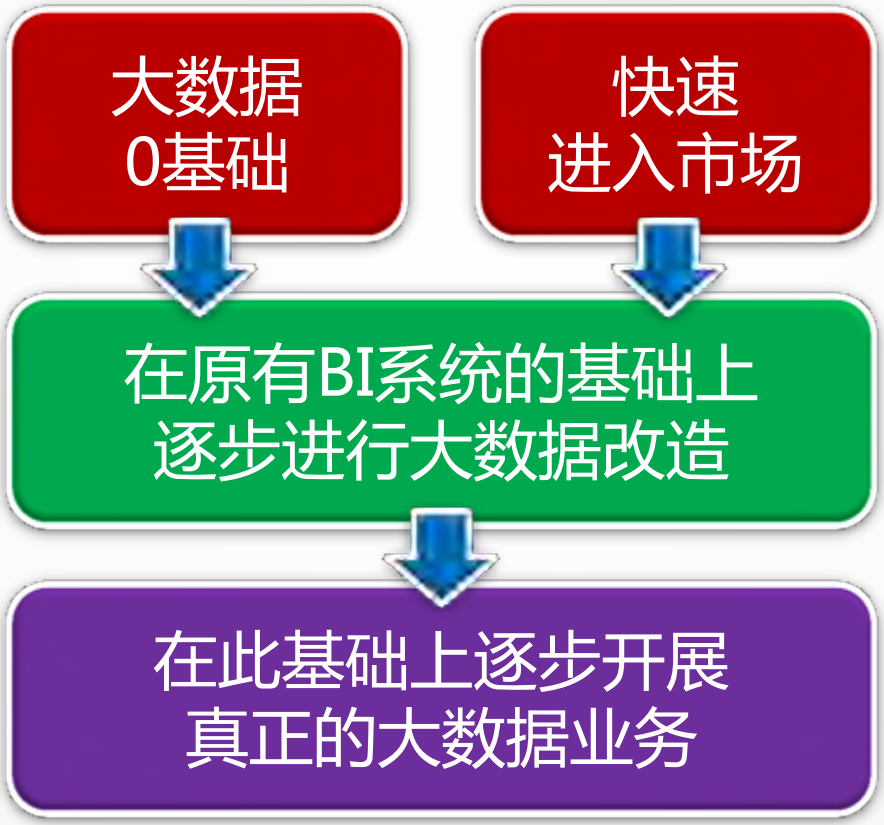
数据量数年才
积累几百个G

数据增量
1年就几个T

- 数据存储成本越来越大
- 数据查询速度越来越慢
- 数据扩展难度陡然增加

大数据转型

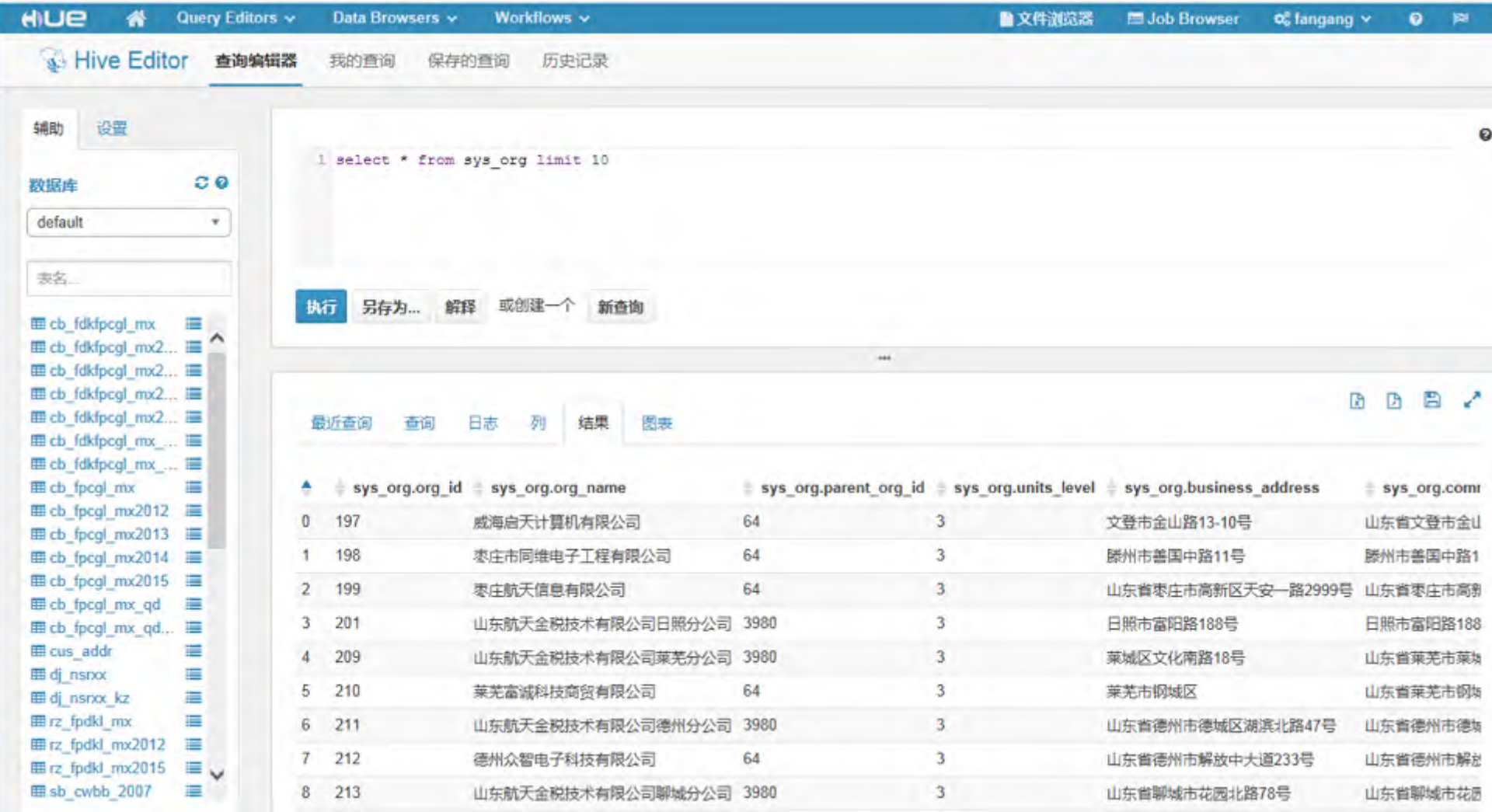
我们面临的难题与应对策略



TiD2017 项目组大数据0基础

- 大数据开发的难题
  - 开发人员不熟悉Spark/scala编程模式
  - 开发人员不熟悉分布式计算的设计原理
  - 设计与开发分布式计算平台很麻烦
  - 需要大量的业务数据需要分析与处理
  - 开发人员数据SQL语句
- 解决方案
  - 提供一个开发平台可以使用SQL语句编写分布式数据分析与处理程序
  - 可以将写好的SQL语句转换成Spark程序

TiD2017 Hive在Hue中进行数据查询

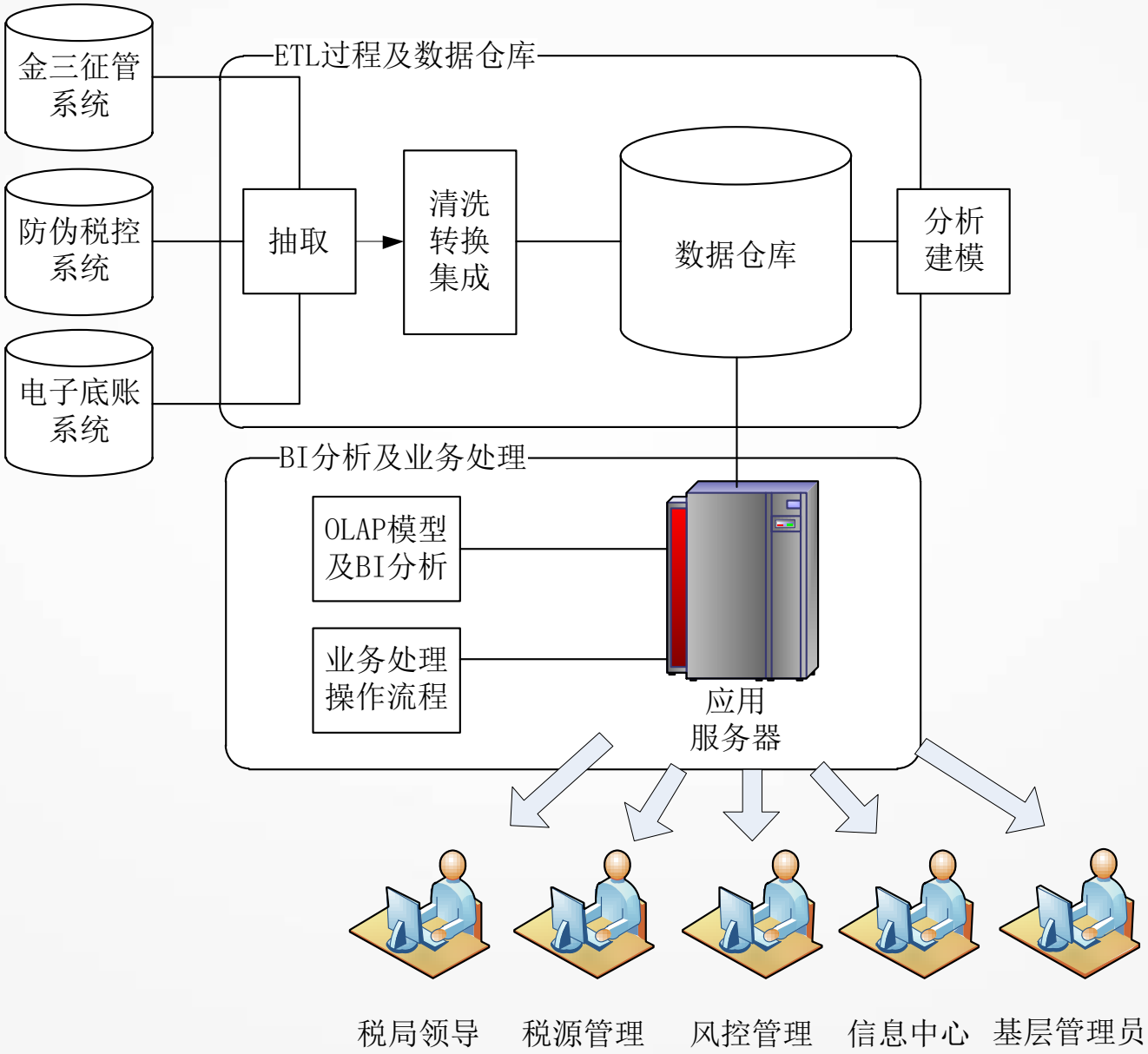


	sys_org.org_id	sys_org.org_name	sys_org.parent_org_id	sys_org.units_level	sys_org.business_address	sys_org.comm
0	197	威海启天计算机有限公司	64	3	文登市金山路13-10号	山东省文登市金山
1	198	枣庄市同维电子工程有限公司	64	3	滕州市善国中路11号	滕州市善国中路1
2	199	枣庄航天信息有限公司	64	3	山东省枣庄市高新区天安一路2999号	山东省枣庄市高新
3	201	山东航天金税技术有限公司日照分公司	3980	3	日照市富阳路188号	日照市富阳路188
4	209	山东航天金税技术有限公司莱芜分公司	3980	3	莱城区文化南路18号	山东省莱芜市莱城
5	210	莱芜富诚科技商贸有限公司	64	3	莱芜市钢城区	山东省莱芜市钢城
6	211	山东航天金税技术有限公司德州分公司	3980	3	山东省德州市德城区湖滨北路47号	山东省德州市德城
7	212	德州众智电子科技有限公司	64	3	山东省德州市解放中大道233号	山东省德州市解放
8	213	山东航天金税技术有限公司聊城分公司	3980	3	山东省聊城市花园北路78号	山东省聊城市花园

TiD2017 将验证后的Hive语句贴入SparkSQL

```
/**
 * 进项分析按纳税人聚合
 * @author hada
 */
object JxfxNsr {
  def main(args: Array[String]) {
  val task = LogUtil.start("dw.agg.jxfxNsr")
  try {
    val sc = SparkUtil.init("dw.agg.jxfxNsr")
    val hc = SparkUtil.getSqlContext(sc)
    hc.udf.register("getDateKey", (dateString:String) =>
         DateUtil.format(DateUtil.getTime(dateString),"yyyyMM").toLong)

    val result = hc.sql("SELECT getDateKey(MX.KPRQ) DATE_KEY,MX.GF_NSRSBH GF_NSR_KEY , "+
      " MX.GF_SWJG_DM GF_SWJG_KEY,NSR.NSRSBH XF_NSR_KEY,MX.XF_SWJG_DM XF_SWJG_KEY,MX.FP_LB,"+
      " SUM(QD.WP_SL) WP_SL,SUM(QD.JE) JE,SUM(QD.SE) SE,COUNT(DISTINCT MX.JXFP_ID) FPFS "+
      " FROM etl.ETL_JXFP MX INNER JOIN etl.ETL_JXFP_QD QD ON MX.JXFP_ID = QD.JXFP_ID "+
      " INNER JOIN dw.DW_DM_NSR NSR ON MX.XF_NSRSBH = NSR.NSR_KEY"+
      " WHERE (MX.ZFBZ='N' or MX.ZFBZ is null)"+
      " GROUP BY getDateKey(MX.KPRQ) , MX.GF_NSRSBH, MX.GF_SWJG_DM,NSR.NSRSBH,"+
      " MX.XF_SWJG_DM,MX.FP_LB,MX.XF_NSRMC,QD.SL")

    DataFrameUtil.save(result, "dw", "dw_agg_jxfx_nsr")
    LogUtil.end(task)
    } catch { case ex:Exception  => LogUtil.error(task, ex) }
  }
}
```

快速进入市场


金三征管系统
防伪税控系统
电子底账系统
ETL过程及数据仓库
抽取
清洗转换集成
数据仓库
分析建模
BI分析及业务处理
OLAP模型及BI分析
业务处理操作流程
应用服务器
税局领导
税源管理
风控管理
信息中心
基层管理员

TiD2017 系统重做 vs. 演化式重构

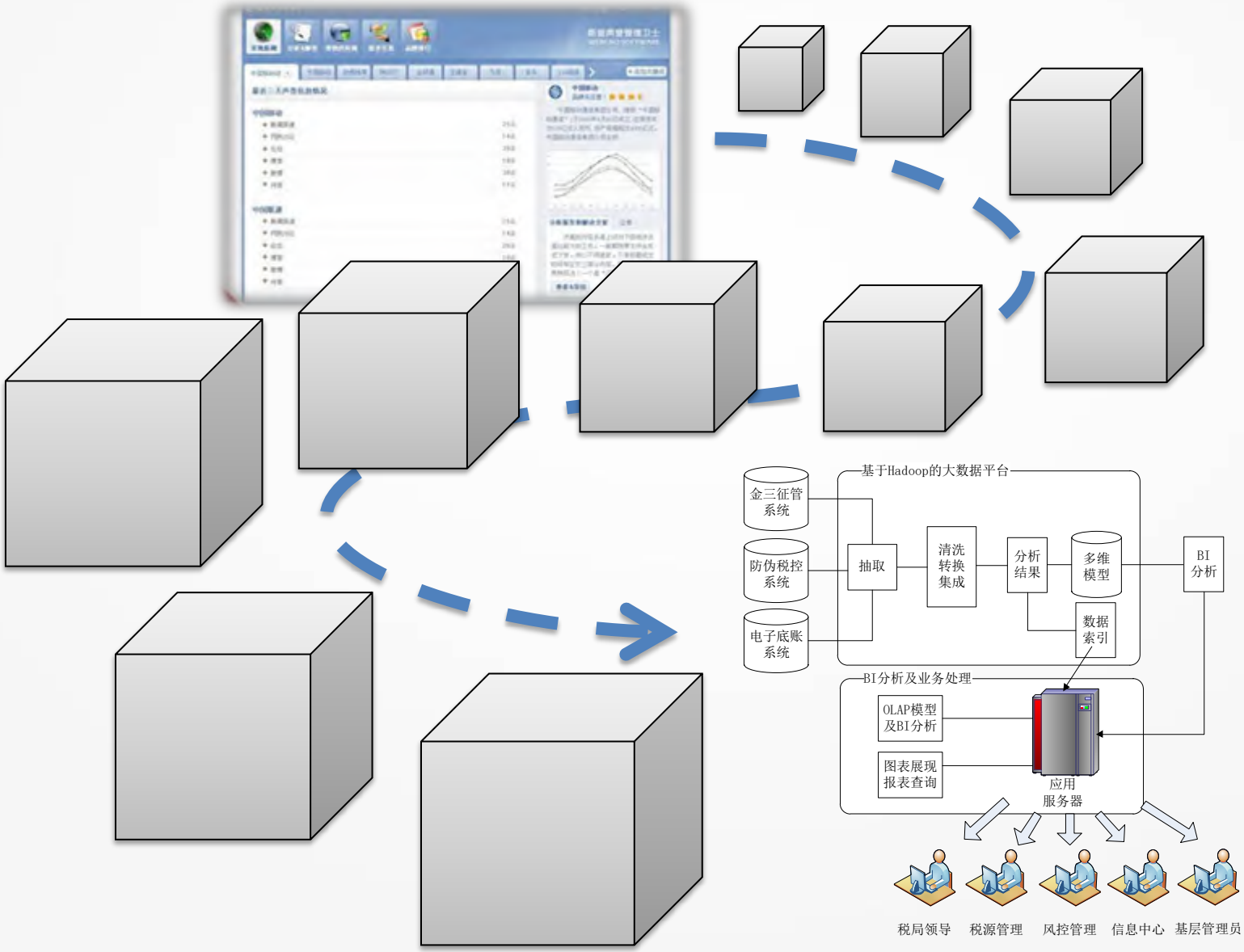
软件重做

抛弃原系统，重新搭建业务系统
很难准确识别原系统繁复的业务逻辑
不能有效替代原系统的功能

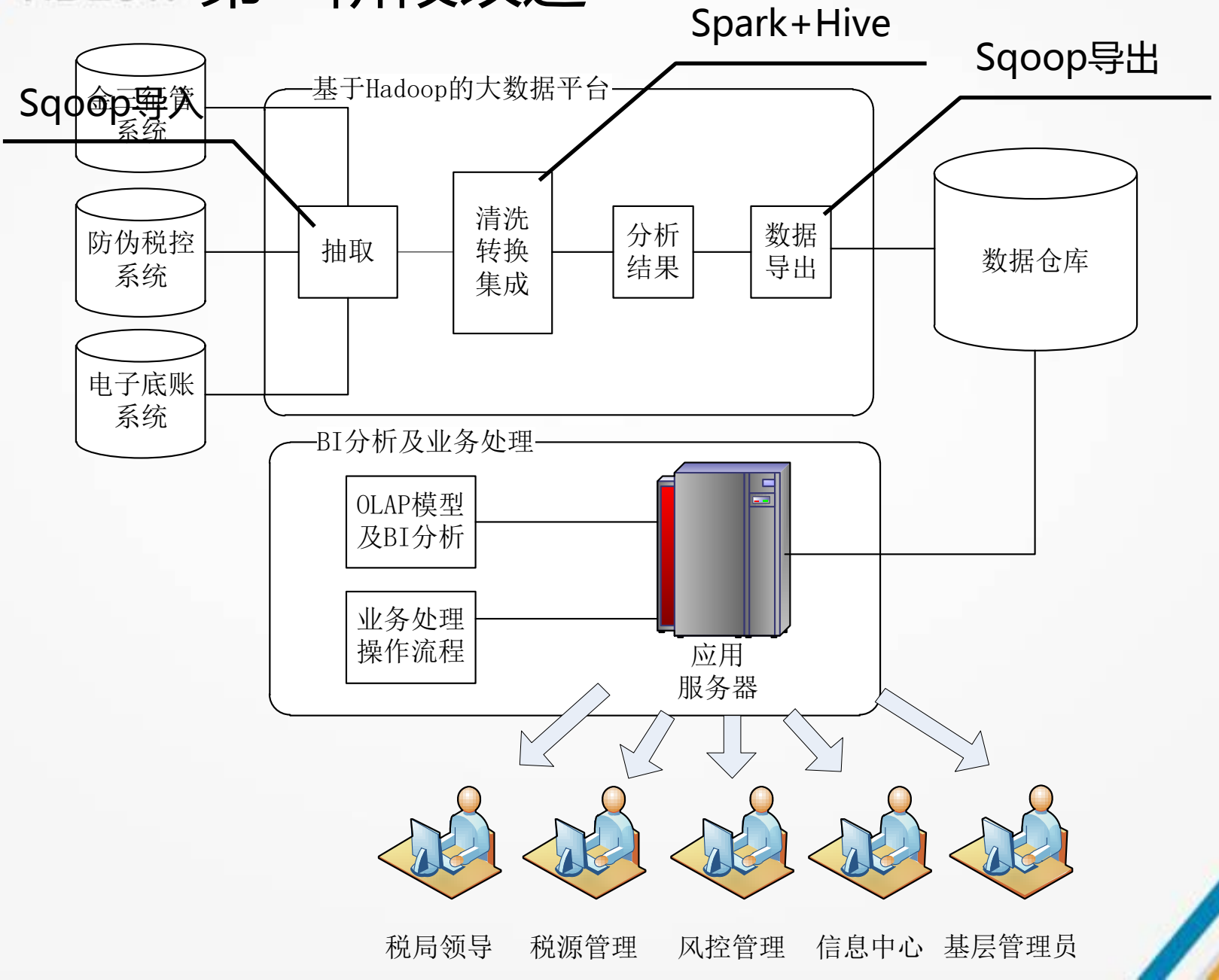
演化式重构

- 是在原系统的基础上一步一步地改造
- 每做一步就可以运行并验证正确性
- 每做一步都能保持原功能的一致
- 将系统改造的长周期变为短周期
- 随时都可以获得可运行的软件并发布

渐进式大数据转型


基于Hadoop的大数据平台
金三征管系统
防伪税控系统
电子底账系统
抽取
清洗转换集成
分析结果
多维模型
BI分析
数据索引
BI分析及业务处理
OLAP模型及BI分析
图表展现报表查询
应用服务器
税局领导
税源管理
风控管理
信息中心
基层管理员

第一阶段改造


Spark+Hive
Sqoop导出
Sqoop导入
基于Hadoop的大数据平台
合一征管系统
防伪税控系统
电子底账系统
抽取
清洗转换集成
分析结果
数据导出
数据仓库
BI分析及业务处理
OLAP模型及BI分析
业务处理操作流程
应用服务器
税局领导
税源管理
风控管理
信息中心
基层管理员

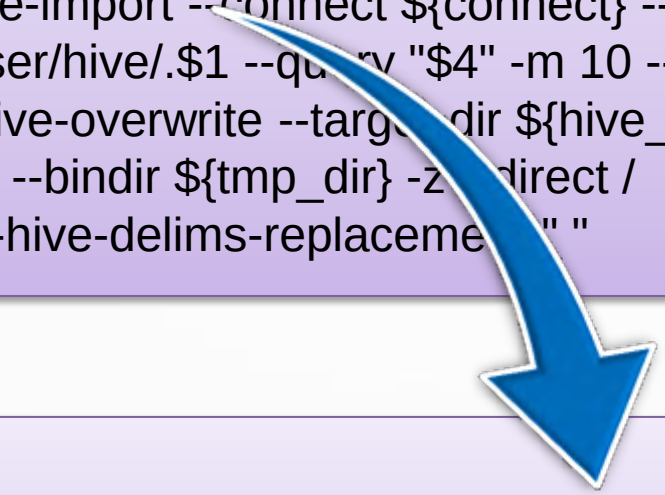
TiD2017 Sqoop数据导入程序

```
sqoop import --hive-import /
--connect jdbc:oracle:thin:@192.168.1.24:1521/SDHTXX /
--username admin --password-file /user/hive/.admin /
--query "select * from dzdz_fpxx_hyzp t where \$CONDITIONS" /
-m 10 --hive-database dzdz --hive-table dzdz_fpxx_hyzp /
--hive-overwrite /
--target-dir /user/hive/warehouse/dzdz.db/dzdz_fpxx_hyzp /
--outdir ../tmp --bindir ../tmp -z --direct --null-string '\\N'  /
--hive-delims-replacement " "
```

```
sqoop import --hive-import /
--connect jdbc:oracle:thin:@192.168.1.24:1521/SDHTXX /
--username admin --password-file /user/hive/.admin /
--query "select * from dzdz_fpxx_zzsfp t where \$CONDITIONS" /
-m 10 --hive-database dzdz --hive-table dzdz_fpxx_zzsfp /
--hive-overwrite /
--target-dir /user/hive/warehouse/dzdz.db/dzdz_fpxx_zzsfp /
--outdir ../tmp --bindir ../tmp -z --direct --null-string '\\N'  /
--hive-delims-replacement " "
```

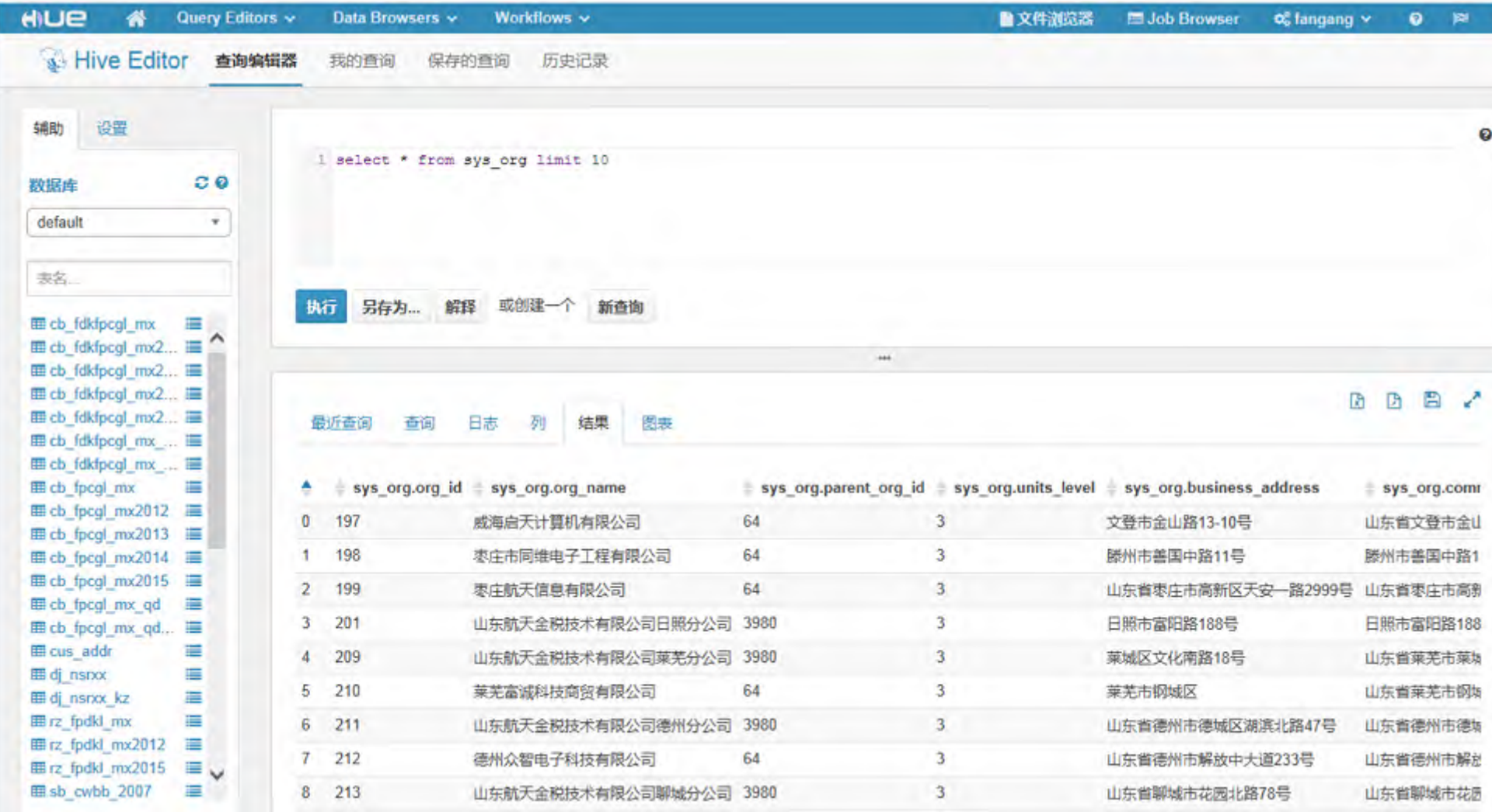
TiD2017 Sqoop的重构与整合

```
#!/bin/bash
cd /home/aisinobi/bin
connect=jdbc:oracle:thin:@192.168.1.24:1521/SDHTXX
hive_dir=/user/hive/warehouse
tmp_dir=../tmp
sqoop import --hive-import --connect ${connect} --username $1 /
--password-file /user/hive/.$1 --query "$4" -m 10 --hive-database $2 /
--hive-table $3 --hive-overwrite --target-dir ${hive_dir}/$2.db/$3 /
--outdir ${tmp_dir} --bindir ${tmp_dir} --direct /
--null-string '\\N'  --hive-delims-replacement " "
```



```
#!/bin/bash
sql="select * from dzdz_fpxx_hyzp t where kprq between
to_date('$1','yyyy-mm-dd') and to_date('$2 23:59:59','yyyy-mm-dd
hh24:mi:ss') and \$CONDITIONS"
map=SL=DOUBLE,SE=DOUBLE,JSHJ=DOUBLE
bash SqoopJdbc.sh DZDZ dzdz DZDZ_FPXX_HYZP "$sql" $map
```

TiD2017 Hive在Hue中进行数据查询



	sys_org.org_id	sys_org.org_name	sys_org.parent_org_id	sys_org.units_level	sys_org.business_address	sys_org.comm
0	197	威海启天计算机有限公司	64	3	文登市金山路13-10号	山东省文登市金山
1	198	枣庄市同维电子工程有限公司	64	3	滕州市善国中路11号	滕州市善国中路1
2	199	枣庄航天信息有限公司	64	3	山东省枣庄市高新区天安一路2999号	山东省枣庄市高新
3	201	山东航天金税技术有限公司日照分公司	3980	3	日照市富阳路188号	日照市富阳路188
4	209	山东航天金税技术有限公司莱芜分公司	3980	3	莱城区文化南路18号	山东省莱芜市莱城
5	210	莱芜富诚科技商贸有限公司	64	3	莱芜市钢城区	山东省莱芜市钢城
6	211	山东航天金税技术有限公司德州分公司	3980	3	山东省德州市德城区湖滨北路47号	山东省德州市德城
7	212	德州众智电子科技有限公司	64	3	山东省德州市解放中大道233号	山东省德州市解放
8	213	山东航天金税技术有限公司聊城分公司	3980	3	山东省聊城市花园北路78号	山东省聊城市花园

TiD2017 SparkSQL第一个版本的实现

```
/**
 * 增值税进项发票ETL程序，从增值税专用发票表(DZDZ_FPXX_ZZSFP)中抽取数据，进行集成、清洗、转换操作，
 * @author Fangang
 */
object ZzsfpJx {
  def main(args: Array[String]) {
    val conf = (new SparkConf).setMaster("master").setAppName("zzsfpJx");
    val sc = new SparkContext(conf)
    val hc = new HiveContext(sc)

    val dateFormat =  new SimpleDateFormat("yyyy-MM-dd HH:mm:ss")
    val now = dateFormat.format(new Date())

    hc.udf.register("getJxfpId", (fpdm:String, fphm:String, kprq:Date) =>
         if(null==kprq) fpdm+"X"+fphm+"X" else fpdm+"X"+fphm+"X"+kprq)
    hc.udf.register("fillZero", (swjg:String)=> if(null==swjg) swjg else swjg+"00")

    val result = hc.sql("SELECT getJxfpId(D.FPDM,D.FPHM,D.KPRQ) JXFP_ID,D.FPDM,D.FPHM,"+
      "'YB' FP_LB,D.JE JE,D.SE/D.JE SL,D.SE SE,D.XFSBH XF_NSRSBH,D.XFMC XF_NSRMC,"+
      s"D.GFSBH GF_NSRSBH,D.GFMC GF_NSRMC,D.KPRQ,D.RZDKL_BDRQ RZSJ,D.BSSWJG_DM SWJG_DM,'$now
      "fillZero(D.GF_QXSWJG_DM) GF_SWJG_DM,fillZero(D.XF_QXSWJG_DM) XF_SWJG_DM,D.ZFBZ,D.SKM
      "FROM dzdz.DZDZ_FPXX_ZZSFP D ")

    val path = "/user/hive/warehouse/etl.db/etl_jxfp"
    result.save(path, SaveMode.Overwrite)
  }
}
```

TiD2017 开发下一个功能

```
/**
 * 增值税进项清单发票ETL程序，从增值税专用发票表(DZDZ_HWXX_ZZSFP)和进项发票表(ETL_JXFP)中抽取数据，
 * @author Liruiming
 */
object ZzsfpJxQd {
  def main(args: Array[String]) {
    val conf = (new SparkConf).setMaster("master").setAppName("zzsfpJx");
    val sc = new SparkContext(conf)
    val hc = new HiveContext(sc)
    val dateFormat =  new SimpleDateFormat("yyyy-MM-dd HH:mm:ss")
    val now = dateFormat.format(new Date())
    hc.udf.register("getJxfpqdId", (fpdm:String, fphm:String, kprq:Date,
      qdbz:String, id:Double) => if(null==kprq) fpdm+"X"+fphm+"X"+"null"+"X"+qdbz+"X"+id
         else fpdm+"X"+fphm+"X"+kprq+"X"+qdbz+"X"+id)
    hc.udf.register("getJxfpId", (fpdm:String, fphm:String, kprq:Date) =>
      if (null==kprq )fpdm+"X"+fphm+"X" else fpdm+"X"+fphm+"X"+kprq)

    val result = hc.sql("SELECT getJxfpqdId(D.FPDM,D.FPHM,R.KPRQ,D.QDBZ,D.ID) JXFPQD_ID,"+
        "getJxfpId(D.FPDM,D.FPHM,R.KPRQ) JXFP_ID ,D.ID HH,'YB' FP_LB,D.HWMC WP_MC,"+
        "D.DW WP_DW,D.GGXH WP_XH,D.SL WP_SL,D.DJ DJ,D.JE JE,D.SLV SL,"+
       s"D.SE SE,R.RZSJ RZSJ,'$now' CZSJ,R.KPRQ KPRQ,D.QDBZ "+
        "FROM dzdz.DZDZ_HWXX_ZZSFP D JOIN etl.ETL_JXFP R "+
        "ON (D.FPDM = R.FPDM AND D.FPHM = R.FPHM)")
    val path = "/user/hive/warehouse/etl.db/etl_jxfp_qd"
    result.save(path, SaveMode.Overwrite)
  }
}
```

TiD2017 第一个版本的分析

```
/**
 * 增值税进项发票ETL程序，从增值税专用发票表(DZDZ_FPXX_ZZSFP)中抽取数据，进行集成、清洗、转换操作，
 * @author Fangang
 */
object ZzsfpJx {
  def main(args: Array[String]) {
    val conf = (new SparkConf).setMaster("master").setAppName("zzsfpJx");
    val sc = new SparkContext(conf)
    val hc = new HiveContext(sc)
    val dateFormat =  new SimpleDateFormat("yyyy-MM-dd HH:mm:ss")
    val now = dateFormat.format(new Date())
    hc.udf.register("getJxfpId", (fpdm:String, fphm:String, kprq:Date) =>
         if(null==kprq) fpdm+"X"+fphm+"X" else fpdm+"X"+fphm+"X"+kprq)
    hc.udf.register("fillZero", (swjg:String)=> if(null==swjg) swjg else swjg+"00")

    val result = hc.sql("SELECT getJxfpId(D.FPDM,D.FPHM,D.KPRQ) JXFP_ID,D.FPDM,D.FPHM,"+
      "'YB' FP_LB,D.JE JE,D.SE/D.JE SL,D.SE SE,D.XFSBH XF_NSRSBH,D.XFMC XF_NSRMC,"+
      s"D.GFSBH GF_NSRSBH,D.GFMC GF_NSRMC,D.KPRQ,D.RZDKL_BDRQ RZSJ,D.BSSWJG_DM SWJG_DM,'$now
      "fillZero(D.GF_QXSWJG_DM) GF_SWJG_DM,fillZero(D.XF_QXSWJG_DM) XF_SWJG_DM,D.ZFBZ,D.SKM
      "FROM dzdz.DZDZ_FPXX_ZZSFP D ")
    val path = "/user/hive/warehouse/etl.db/etl_jxfp"
    result.save(path, SaveMode.Overwrite)
  }
}
```

SparkUtil

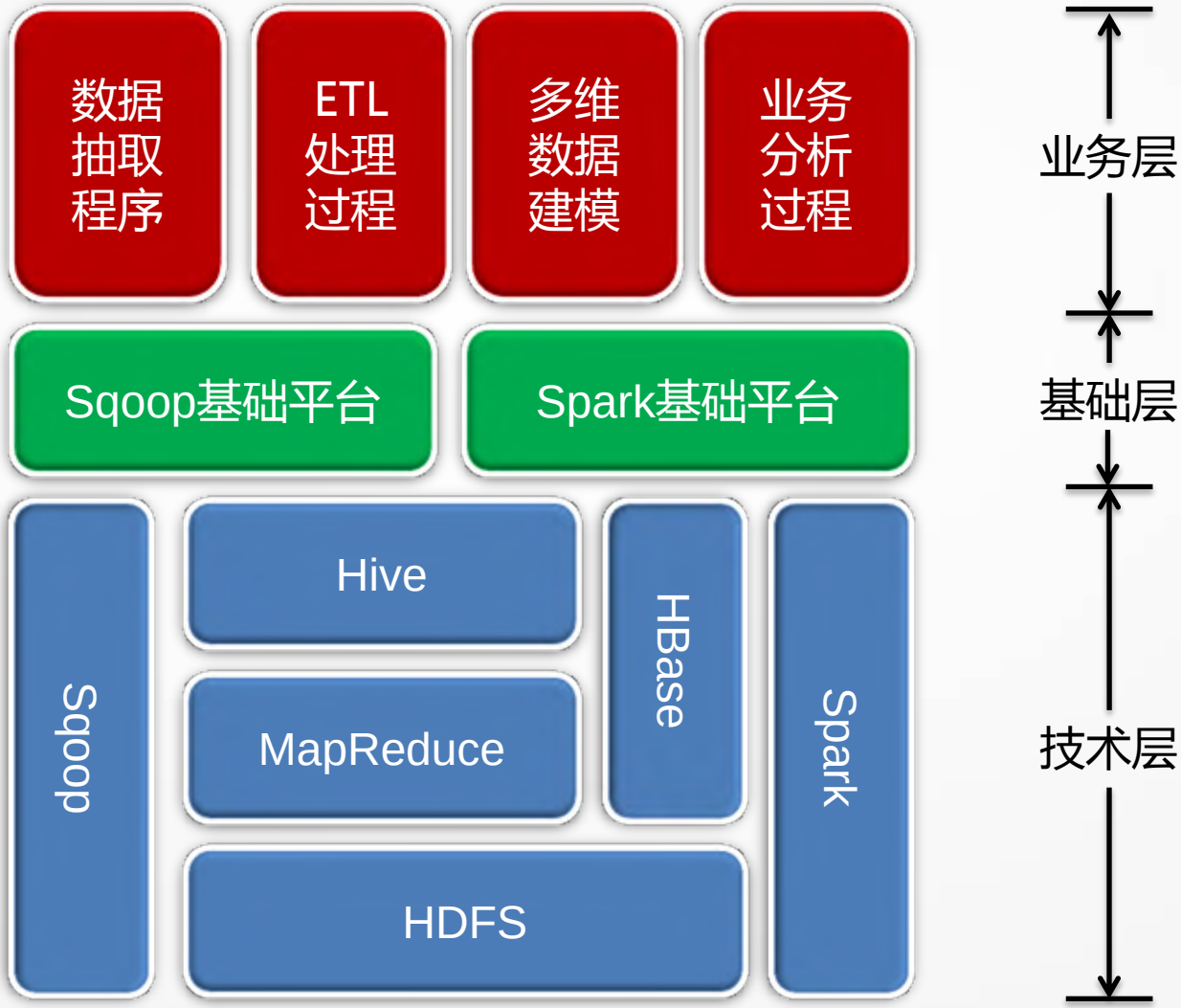
DateUtil

DataFrameUtil

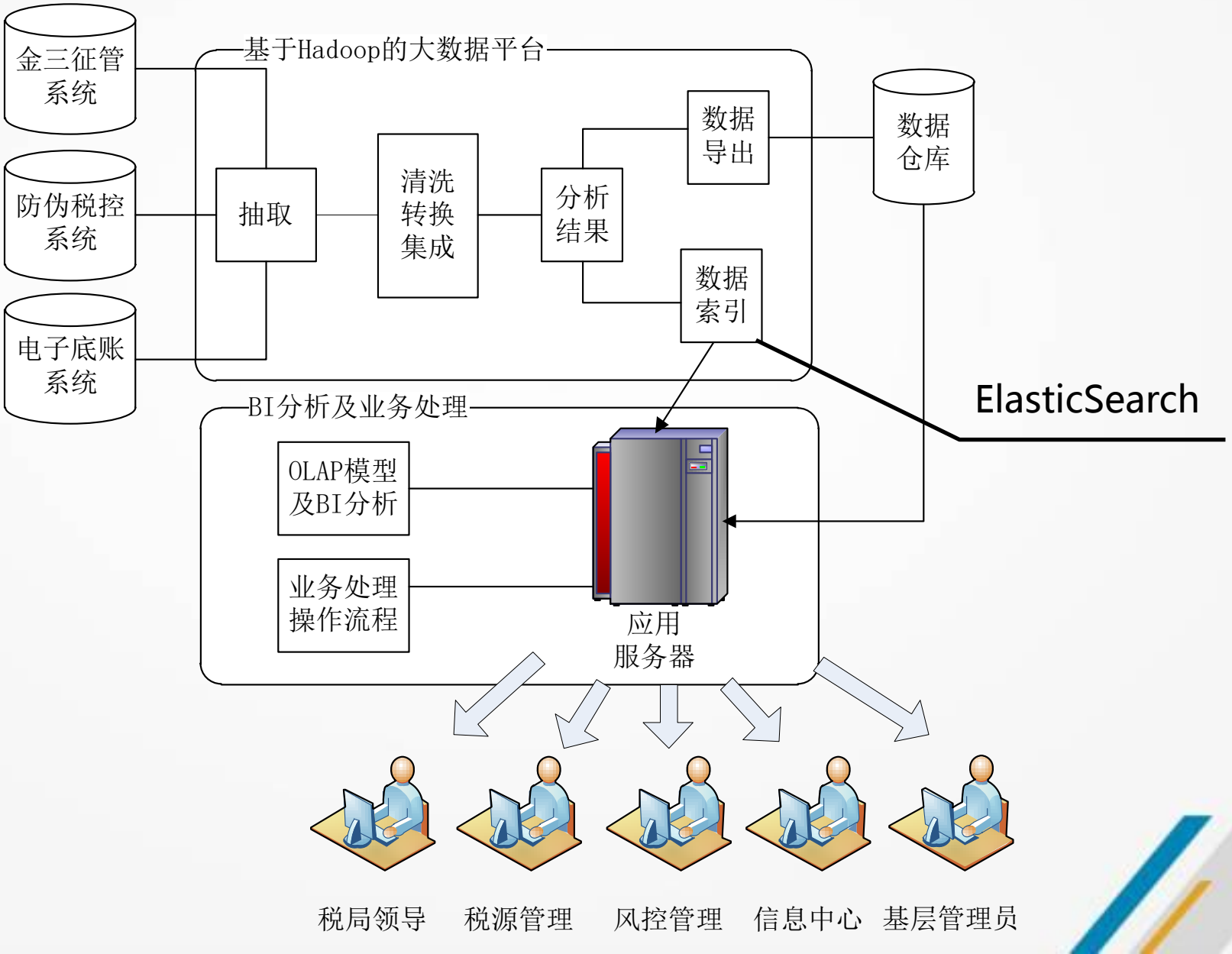
TiD2017 第二个版本进行重构

```
/**
 * 增值税进项发票ETL程序，从增值税专用发票表(DZDZ_FPXX_ZZSFP)中抽取数据，进行集成、清洗
 * @author Fangang
 */
object ZzsfpJx {
  def main(args: Array[String]) {
    val sc = SparkUtil.init("zzsfpJx")
    val hc = SparkUtil.getSqlContext(sc)
    val now = DateUtil.getNow

    hc.udf.register("getJxfpId", (fpdm:String, fphm:String, kprq:Date) =>
        if(null==kprq) fpdm+"X"+fphm+"X" else fpdm+"X"+fphm+"X"+kprq)
    hc.udf.register("fillZero", (swjg:String)=> if(null==swjg) swjg else swjg+"00

    val result = hc.sql("SELECT getJxfpId(D.FPDM,D.FPHM,D.KPRQ) JXFP_ID,D.FPDM,D.
      "'YB' FP_LB,D.JE JE,D.SE/D.JE SL,D.SE SE,D.XFSBH XF_NSRSBH,D.XFMC XF_NSRMC,
      s"D.GFSBH GF_NSRSBH,D.GFMC GF_NSRMC,D.KPRQ,D.RZDKL_BDRQ RZSJ,D.BSSWJG_DM SW
      "fillZero(D.GF_QXSWJG_DM) GF_SWJG_DM,fillZero(D.XF_QXSWJG_DM) XF_SWJG_DM,D.
      "FROM dzdz.DZDZ_FPXX_ZZSFP D ")

    DataFrameUtil.save(result, "etl", "etl_jxfp")
  }
}
```

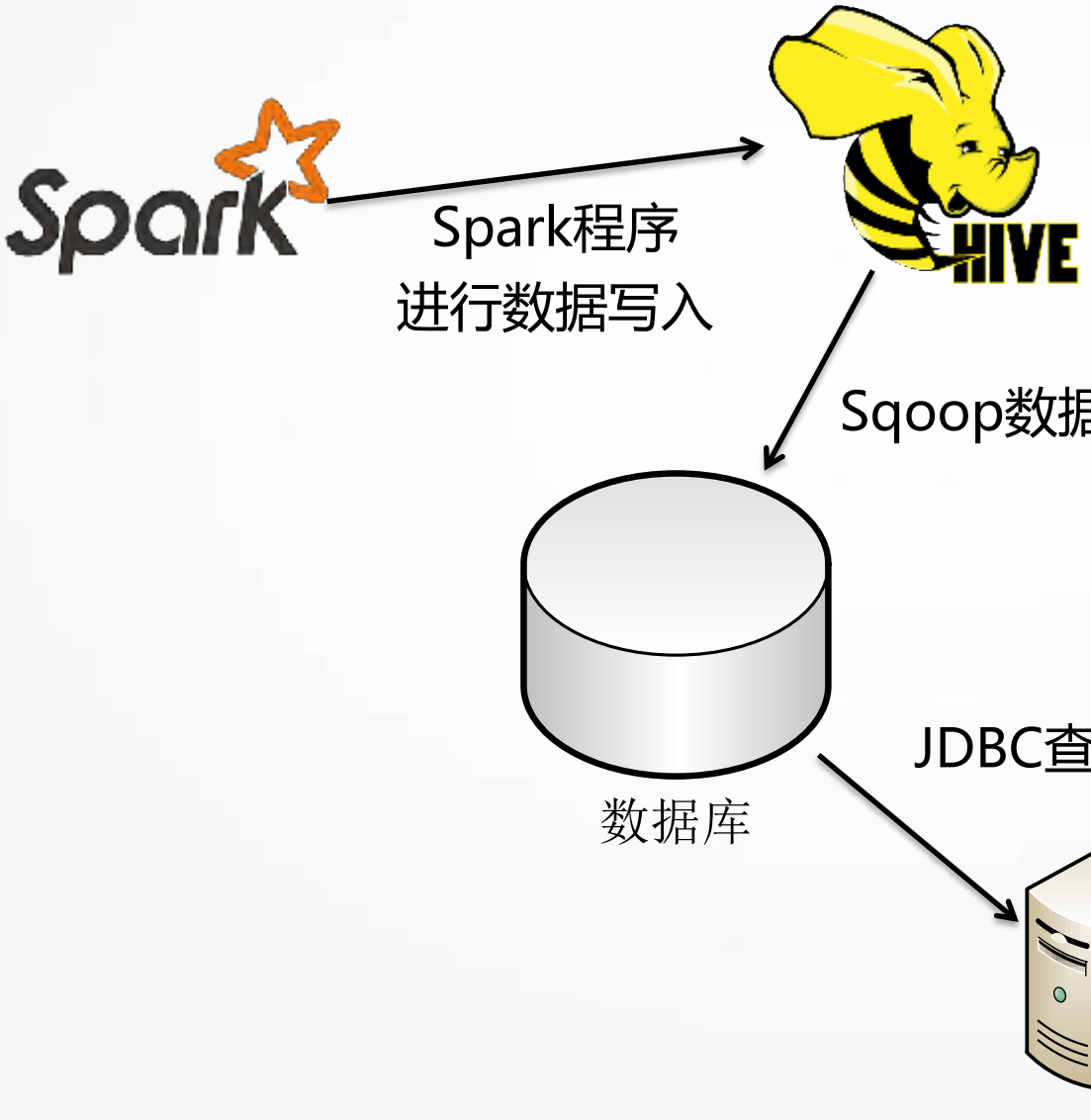
分层便于运维


数据抽取程序
ETL处理过程
多维数据建模
业务分析过程
Sqoop基础平台
Spark基础平台
Sqoop
Hive
MapReduce
HBase
Spark
HDFS
业务层
基础层
技术层

第二阶段改造


金三征管系统
防伪税控系统
电子底账系统
基于Hadoop的大数据平台
抽取
清洗转换集成
分析结果
数据导出
数据索引
数据仓库
ElasticSearch
BI分析及业务处理
OLAP模型及BI分析
业务处理操作流程
应用服务器
税局领导
税源管理
风控管理
信息中心
基层管理员

TiD2017 原有的架构设计



TiD2017 重构成ElasticSearch的架构设计

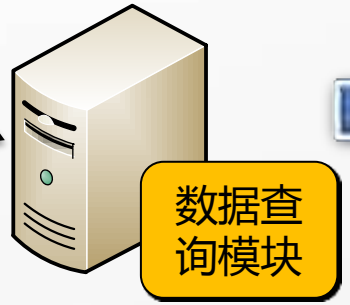
Spark程序
进行数据写入

映射成一个hive表

HDFS数据存储

elastic

通过接口进行查询

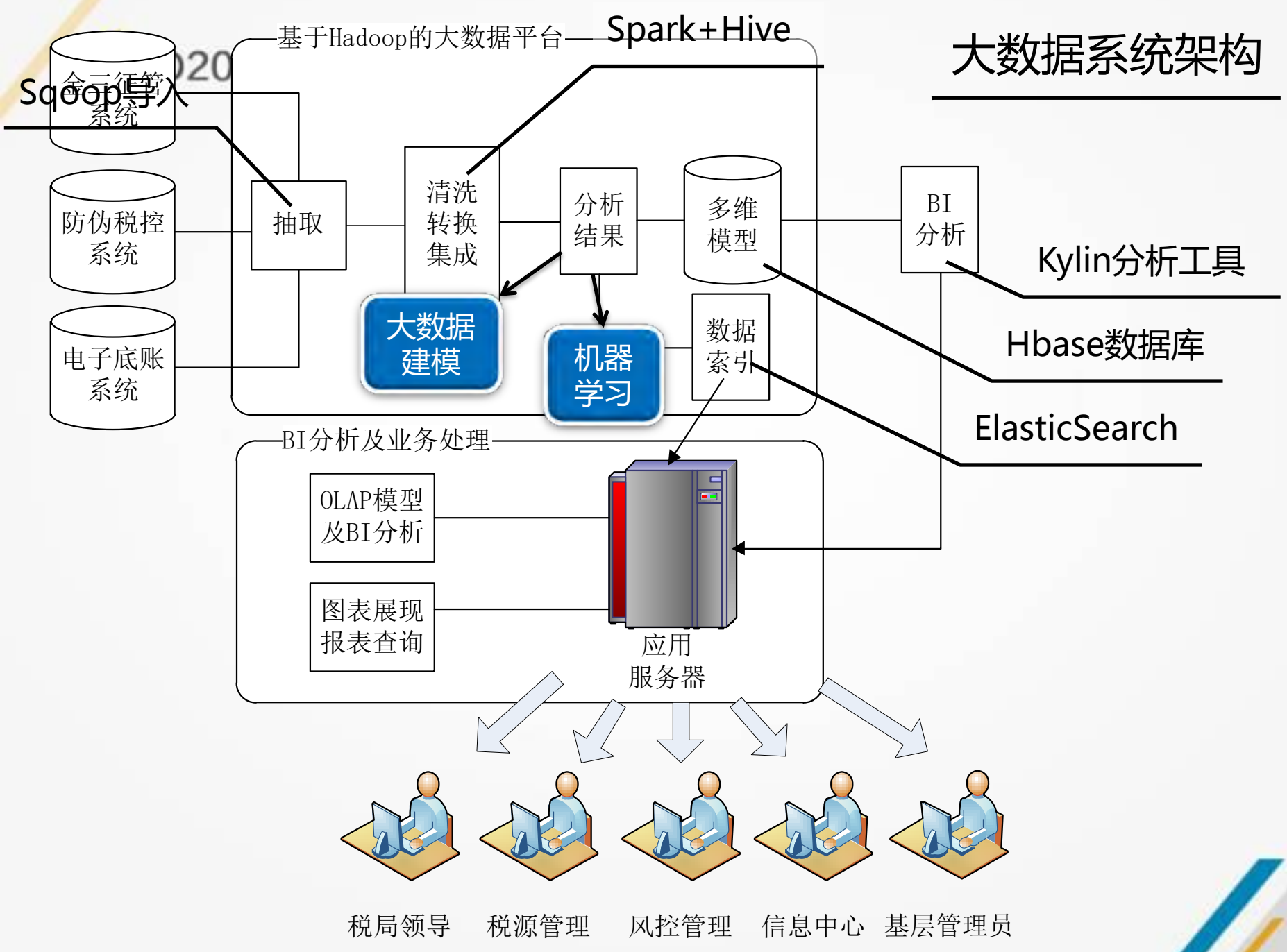


数据查询模块

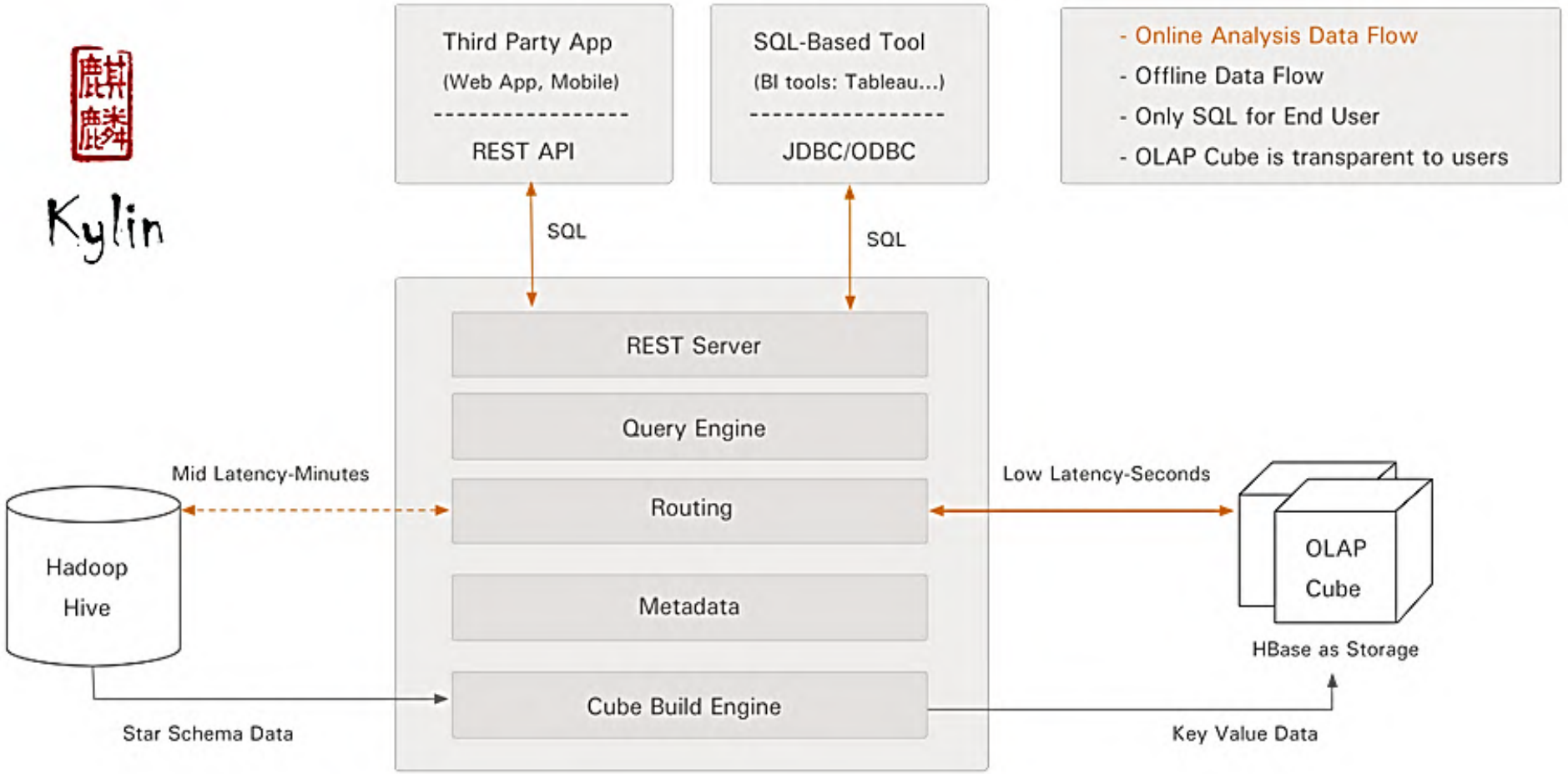
实时查询

数据展现

大数据系统架构


基于Hadoop的大数据平台
Spark+Hive
Sqoop导入
金三征管系统
防伪税控系统
电子底账系统
抽取
清洗转换集成
分析结果
多维模型
BI分析
Kylin分析工具
Hbase数据库
ElasticSearch
大数据建模
机器学习
数据索引
BI分析及业务处理
OLAP模型及BI分析
图表展现报表查询
应用服务器
税局领导
税源管理
风控管理
信息中心
基层管理员

Kylin的系统架构


Kylin
Third Party App
(Web App, Mobile)
REST API
SQL-Based Tool
(BI tools: Tableau...)
JDBC/ODBC
- Online Analysis Data Flow
- Offline Data Flow
- Only SQL for End User
- OLAP Cube is transparent to users
SQL
SQL
REST Server
Query Engine
Routing
Metadata
Cube Build Engine
Mid Latency-Minutes
Hadoop
Hive
Star Schema Data
Low Latency-Seconds
OLAP
Cube
HBase as Storage
Key Value Data

Tutorial_Cube	READY	1.50 MB	7,001	2016-03-16 04:12:30 PST	ADMIN	2016-03-15 13:22:19 PST	Action	Action

Grid | SQL | JSON(Cube) | Access | Notification | HBase

Cube Designer

1 Cube Info — 2 Dimensions — 3 Measures — 4 Refresh Setting — 5 Advanced Setting — 6 Overview

Model Name Tutorial_Model

Cube Name Tutorial_Cube

Notification List []

Description

Tips

1. Cube name is unique name of entire system
2. Can not edit cube name after created
3. Cube must belong to project which you have privilege to create

Next →

总结：大数据转型历程

sqoop

HIVE

Spark

elastic

Apache Kylin™

Extreme OLAP Engine for Big Data

TiD2017 小步快跑的特点与优势

- 特点
  - 快速开发出第一个版本
  - 总是不断添加新功能
  - 总是在测试与验证
  - 总是可运行
  - 总是在思考简单的问题
  - 总是在调整与优化中
- 优势
  - 可以实现恰如其分的设计
  - 可以适时调整与优化设计

小步快跑能解决的问题

- 复杂功能得到有效地解耦
- 代码编写总是可测试与验证
- 简化设计与思考的复杂度
- 每次开发只考虑此次的需求
- 不论第一次开发还是日后维护，都是在添加功能
- 每次添加功能时，能够适时重构
- 可以避免软件维护过程中的退化
- 让敏捷实践更加易于落地

交流时间