```
@Override
public long createProductPlan(long affairId, long roleId, long userId, CreateProduceDTO form, Long date) {
 //...省略：模块初始化
    calendar.setTime(new Date(date));
    if (fromAnnId == null) {
        TimerEntity entity = new TimerEntity();
        entity.setAffairId(affairId);
        entity.setName("计划持续时间");
        calendar.set(Calendar.HOUR_OF_DAY, 0);
        calendar.set(Calendar.MINUTE, 0);
        calendar.set(Calendar.SECOND, 0);
        calendar.set(Calendar.MILLISECOND, 0);
        entity.setBeginTime(new Timestamp(calendar.getTimeInMillis()));
        calendar.set(Calendar.HOUR_OF_DAY, 23);
        calendar.set(Calendar.MINUTE, 59);
        calendar.set(Calendar.SECOND, 59);
        calendar.set(Calendar.MILLISECOND, 0);
        entity.setEndTime(new Timestamp(calendar.getTimeInMillis()));
        if (timerService.checkOverlap(affairId, -1, entity.getBeginTime(), entity.getEndTime())) {
            throw new ErrorCodeException(ResponseCode.EXIST_OVERLAP_ANNOUNCEMENT, 出品时间段重复");};
        timers.add(entity);
    } else {
 //...省略：与上面类似部分
            timers.add(entity);
        });
    }
    form.setAffairId(affairId);
    form.setCreateRoleId(roleId);
    form.setCreateUserId(userId);
    long annId = orderClient.createProduceAnn(form);
    timers.forEach(t -> {
        t.setAnnId(annId);
    });
    AbstractSession.currentSession().saveBatch(timers);
    //复用了某一发布
    if (fromAnnId != null) {
        copyProductPlan(affairId, roleId, fromAnnId, annId);
    }
    //创建采购结果记录
    PurchaseResultEntity entity = new PurchaseResultEntity();
    entity.setAffairId(affairId);
    entity.setAnnId(annId);
    entity.setState(ProductPlanState.DRAFT.getIndex());
    purchaseResultService.save(entity);
    return annId;
}
```