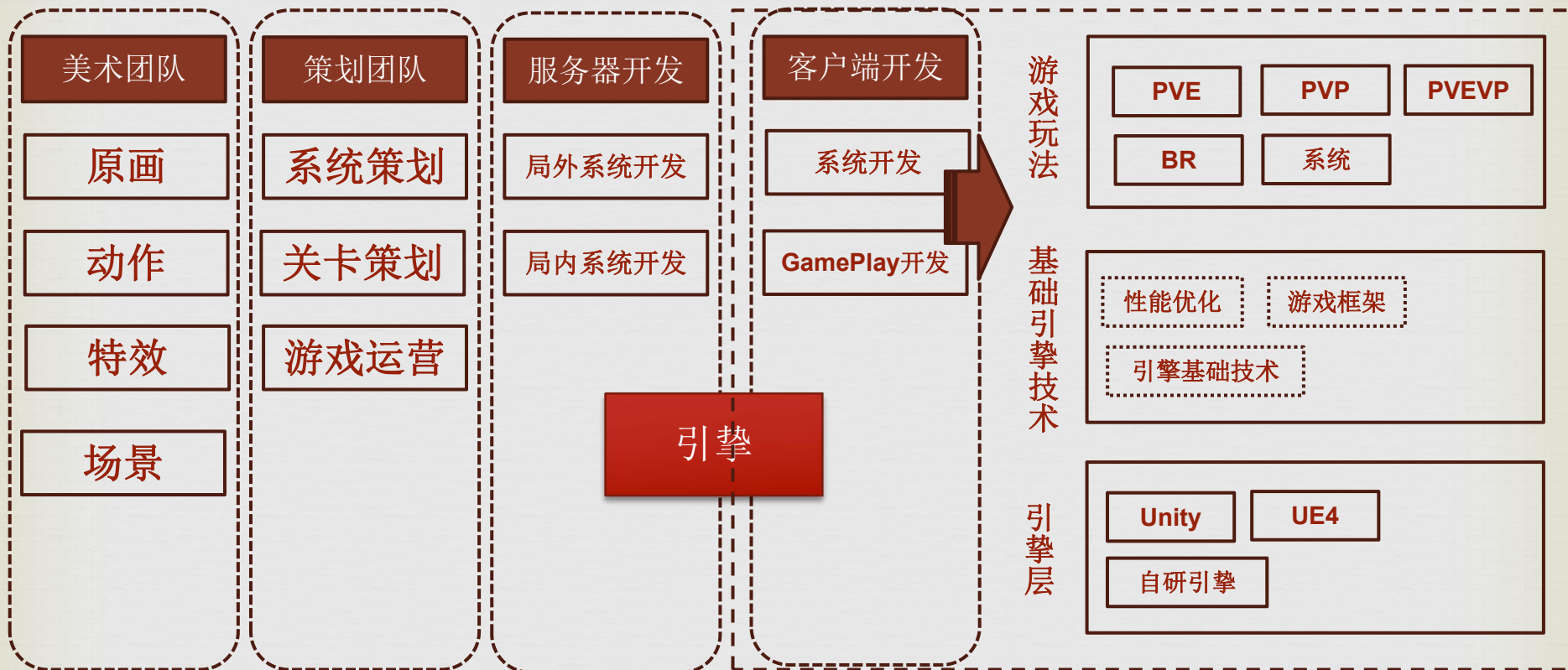


# 现代游戏开发工作流



一线大厂的游戏开发方式

# 传统游戏开发团队



# 引擎技术中台的建设背景



☞ 在手游3A化，对于品质和创意要求越来越高的今天，引擎技术成为了胜败的关键，以前的游戏开发团队都存在如下问题：

## 技术积累缓慢

- 需求驱动，人力解决
- 眼界局限，进步缓慢
- 前沿技术的预研储备

## 人才升级

- 引擎人才偏少
- 培养与管理配套
- 工作室引擎技术的全覆盖

## 开放与传承

- 对外合作，合力共建
- 项目经验传承

# 引擎技术中台的定义

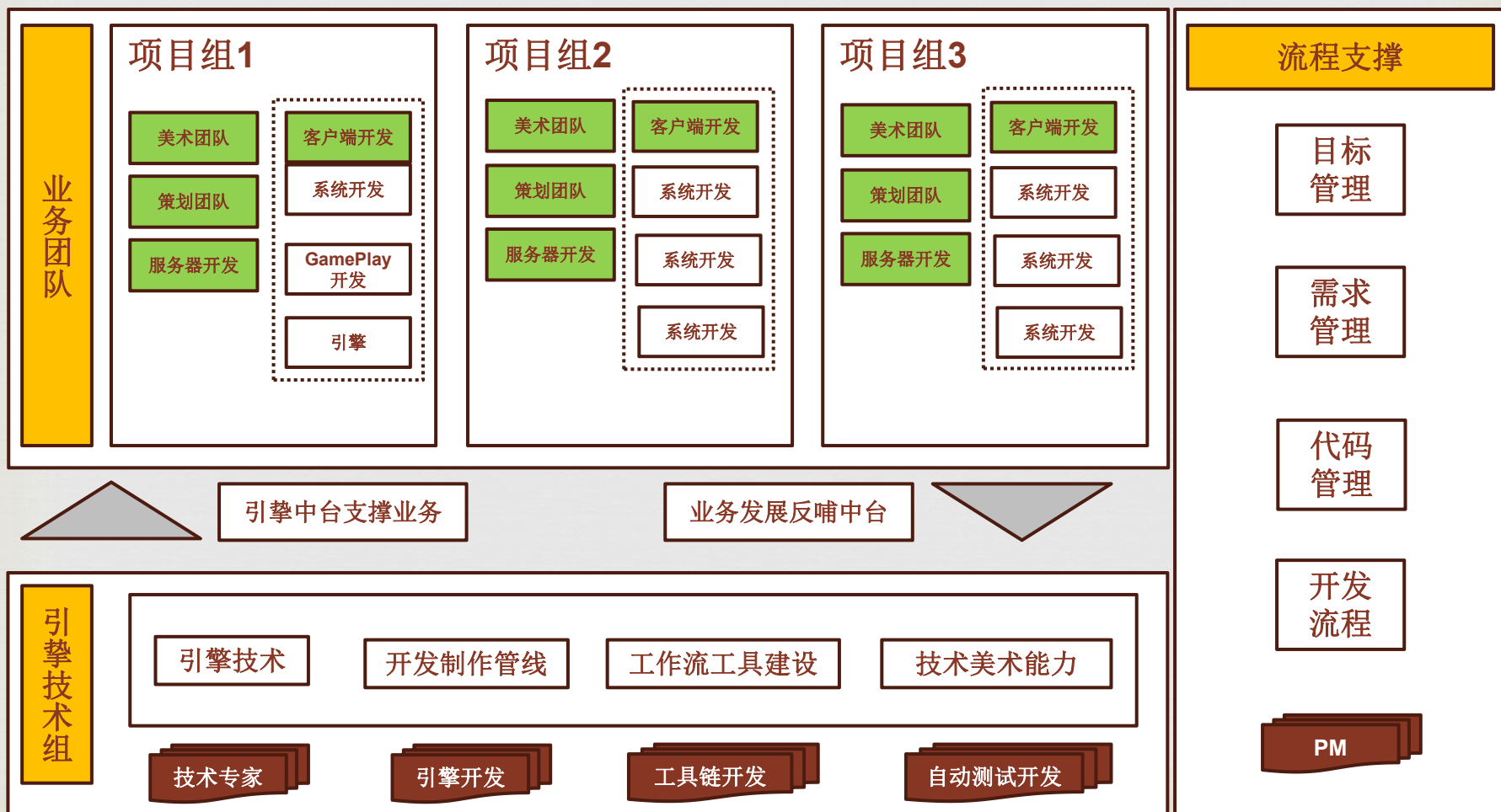


中台不仅仅是技术上的中间平台或中间件，而是一整套集创新组织架构，敏捷研发流程以及前沿游戏技术的综合生态

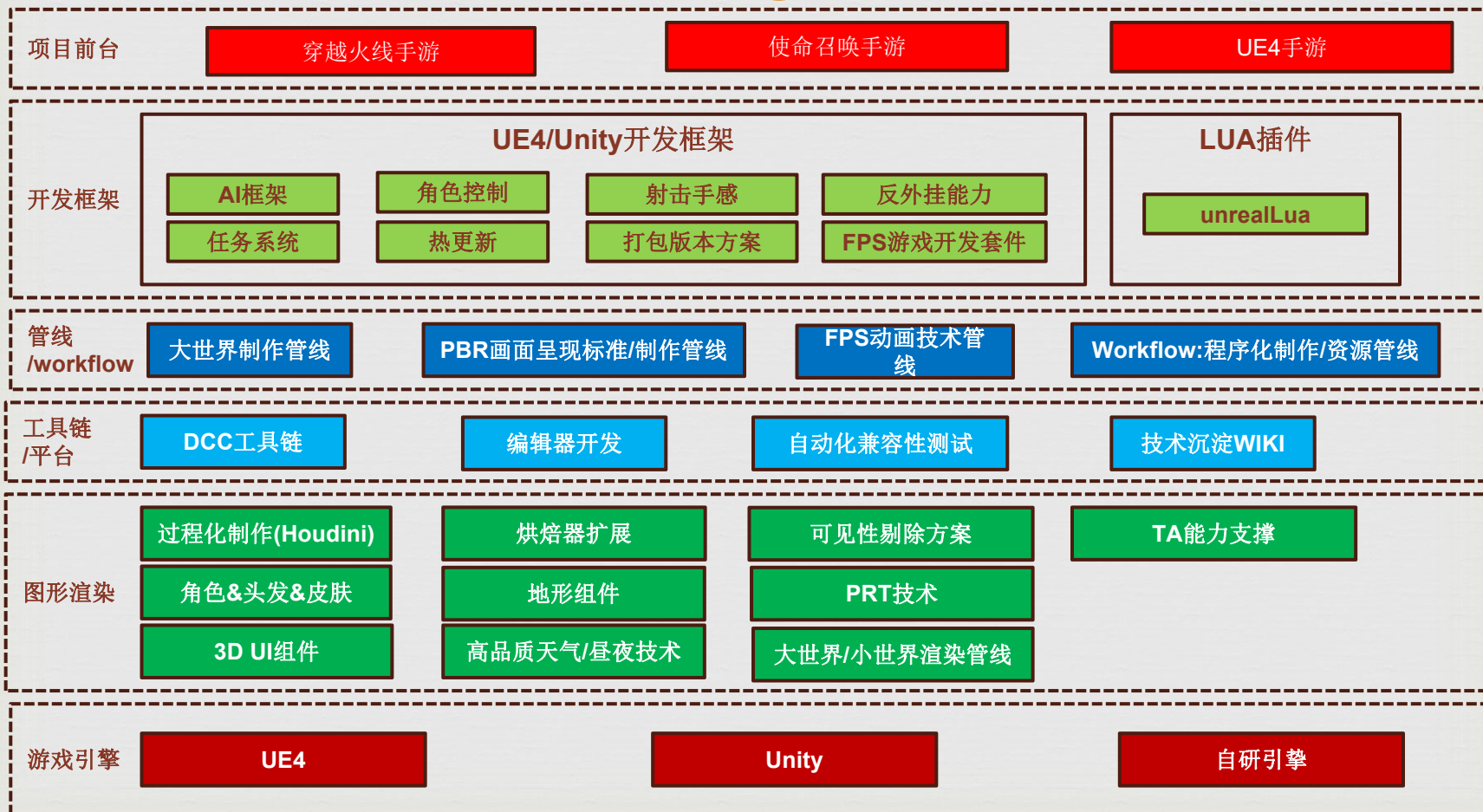
- 组织维度：营造环境—支持项目快速创新，规模化创新，提供完整的持续交付工具链
- 技术维度：赋能业务—根据技术前瞻提前组织资源，利用技术先发和复用的优势，赋能业务真正做到：多（同一时间支持多个项目），快（持续快速交付），好（品质保障），省（技术与人才充分复用）的逐个落地



# 中台架构



# 腾讯引擎中台技术全貌

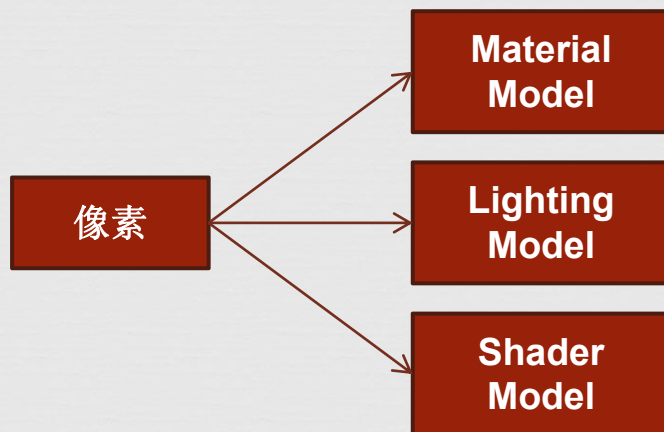


# 开发流程



- 画面标准建立
- 验证
- 管线
  - 游戏自动化兼容性测试框架
  - 大世界制作
  - 烘焙器深度定制
  - 程序化植被制作
- 性能
- 引擎

# 画面标准建立



真实感：标准建立，用同样的语言说话

- 物理意义的参数和真实物理定律(PBR)
- 全场景物体统一光照环境



# 画面标准建立



## 动态物体shader model

### ■ 高中低配上采用完整的PBR方案

### ■ 全实时计算 + shadowmap

- 直接光: GGX specular + lambert diffuse
- 间接光:
  - ibl cubemap (Indirect Specular)
  - L1/L2 Spherical Harmonics(Indirect Diffuse)

### ■ Occlusion

- Bake Ambient Occlusion (Diffuse AO)
- 使用AO + normal + Eye direction + smoothness计算SO (Specular SO)
- 直接用AO (Specular SO低配)

# 画面标准建立：小结



- 项目中所有物体统一使用**PBR**材质渲染，**BRDF**均为**Lambert + GGX Cook-Torrance**
- 实时光阴影：**Shadowmap**衍生技术(实时/预烘焙)
- 间接光：**Convolutated Cubemap + SH(Lightprobe)+Lightmap**

通用性：场景中所有可见实体均使用这套**shading model**的变种，其他**20%**(头发，皮肤，水体，植被)使用其他特殊方法

# 验证



## 标准场景IBL(Image Base Lighting)光照环境



### 1主场景

- 固有色温
- 不同粗糙度
- 金属材质



### 6辅助场景 检验:

- 室内环境（弱点光源）-暖色调环境下效果
- 强对比阴阳光照
- 典型室外光照下效果
- 固定点是否太黑

# 管线



## 游戏自动化兼容性测试框架（WeTest）

- ❑ 渲染结果backbuffer基准机器 对比
- ❑ 游戏Graphic API兼容性情况
- ❑ TOP200兼容性测试
- ❑ TOP200性能测试
- ❑ 新渲染特性/渲染管线评估

# 管线



## 大世界制作

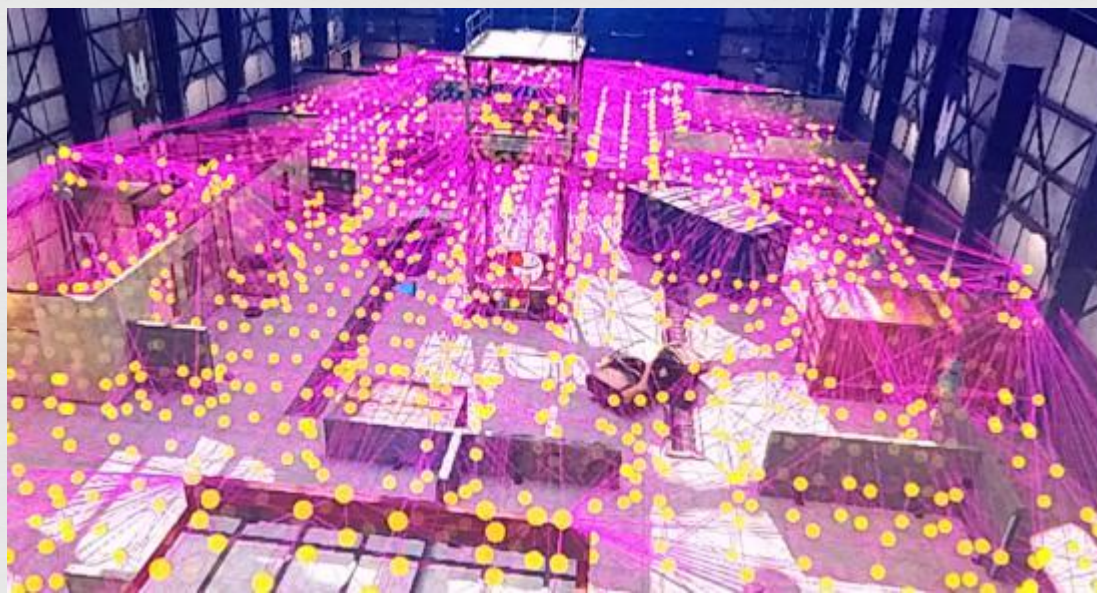
- ❑ 基于Houdini的工作流
- ❑ 同地貌Drawcall合并
- ❑ Vertex Fetch Texture
- ❑ 10+biomes
- ❑ Meterial地貌变化
- ❑ Distance LOD过渡改进



# 管线



烘焙器深度定制【程序化生成LightProbe】



# 管线



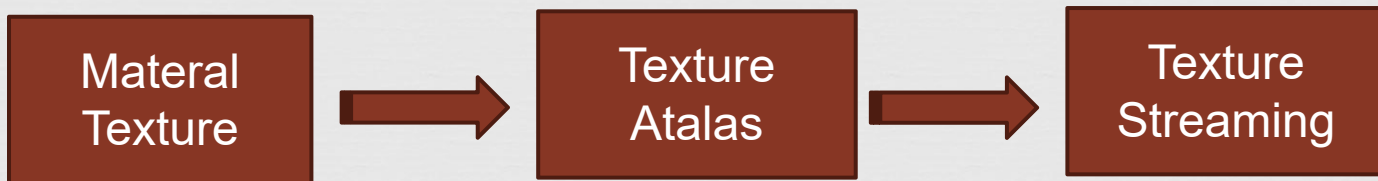
## 程序化植被制作



# 性能



美术制作，引擎，性能工具一体化



不中断 workflow，后续执行优化：

- **Texture Atlas:** 按照材质进行贴图合并
- **Texture Streaming:** 运行时优化，减小渲染带宽
- **Texture Lod**

# 引擎



## Unity引擎特性的升级

头发：各向异性高光

皮肤：**SSSS**

角色：完整的**3A PBR**制作标准



# 总结



## 引擎技术中台的能力

- ◆ 资源与经验重用
- ◆ 品质与性能保障
- ◆ 工作流与制作管线
- ◆ 引擎技术能力
- ◆ 创新赋能
- ◆ 梯队培养