**4P(LEC1/LEC2):**

**The People:**

Stakeholders:本校学生，团队成员，三国杀官方，本授课老师

Team Leaders：文浩羽

Software Team：ALL members

Agile Teams：ALL members

Software Team Organizations:Controlled decentralized,**Hybrid Structures（混合结构 2类——**Use hierarchies of organizational structure.**）**因为我们成员各有所长，特长差异明显：故通过集体讨论分配任务，leader跟进协调。

***此处最好有个图!!!!!!!!!!!!!***

***1***

***/ \ \***

***2 3--4***

***我们组的混合结构是这样的***

***层内运作灵活***

**Communication & Coordination Techniques (沟通与协调技术)** =

### ****Formal, Impersonal Approaches (正式、非个人化的方法)****

软件pre.docx下记录

1. **Formal, Interpersonal Approaches (正式、人与人之间的互动方法)**

每节课开展一次线下会议，每任务阶段开展一次腾讯会议或是微信会议，讨论完成进度和下一阶段目标

**3.Informal, Interpersonal Approaches (非正式、人与人之间的互动方法)**

微信群聊直接沟通

**The Problem/ Product：(两个stage，第一个分3个Step)**

**STAGE 1 : SOFTWARE SCOPE**

STEP1.Context

我们的目标是通过这款游戏让每一位师生都意识到，无论学术多么严肃，生活和互动同样需要幽默与轻松。

project name: 科大杀

project charter：本项目旨在本学期内针对澳科大学生和教师开发一款1V1 本地AI对战的仿《三国杀》游戏，无论学术多么严肃，生活和互动同样需要幽默与轻松。

**项目负责人、发起人和利益相关者：**

* **项目负责人（Project Owner）：** 文浩羽（项目经理，机器学习工程师，数据库搭建）
* **项目发起人（Project Sponsors）：** 黄一行（游戏开发总工程师，Unity引擎专家），孙诣然，张昊展（游戏美术设计师，音频工程师，图表设计 ，UI设计），
* **利益相关者（Stakeholders）：**
  + 本校师生
  + 本授课老师
  + 本团队
  + 卡牌类社团
  + 卡牌类游戏爱好者
  + 澳门地区他校学生
  + 三国杀官方

project justification/vision statement：以游戏作为课堂之外新的互动方式，拉近师生之间的关系。游戏将通过轻松幽默的对战促进学生和教师之间的互动与理解。通过这款游戏，用户能够在愉快的游戏过程中加强师生关系，同时提高他们的策略思维能力。通过这款游戏让每一位师生都意识到，无论学术多么严肃，生活和互动同样需要幽默与轻松。

project requirements：（放在需求文档request.txt 中！！！后面和需求可追溯性矩阵一起展示）

### ****游戏玩法要求****

✅ **1V1 卡牌对战模式**：游戏应严格遵守传统《三国杀》的基本规则，用户可以进行出牌、选牌、弃牌等操作，保证卡牌机制符合《三国杀》的原有规则。  
✅ AI对战**系统**：游戏中的对战模式应由AI系统负责对手角色的行为。AI作为玩家的对手，会模拟一个具备策略思维的玩家，具备与真人对战相似的决策过程

### ****用户界面与体验要求****

✅ **简洁易用的用户界面**：UI 设计应便于师生快速上手，无需复杂学习过程。  
✅ **游戏日志与记录**：用户应能够回顾和查看自己的对局记录，分析自己的游戏策略和结果。  
✅ **基础音效与视觉效果**：提供符合科大主题的音效和画面设计，以增加游戏的沉浸感，根据用户反馈添加“老师”武将牌以及音频设计

### ****AI 训练与优化****

✅ **强化学习**：要求通过游戏数据对 AI 进行训练，确保 AI 能够根据战局进行策略调整。

Milestone：

第1-2周：收集反馈，分析市场需求，确定游戏功能和玩法。确定游戏的设计方向和目标群体，编写初步需求文档。

第3-4周：设计游戏界面、系统架构，确定角色和卡牌设计。解析并导入音频素材，完成初步音效、画面设计。

第5-11周：开发并连接数据库与AI对战模型，确保AI能够智能化对战。完善游戏对局代码的编写，进行单元测试。

第12周：游戏正式上线，进行校园内推广。收集玩家反馈并开始进行持续的维护和优化。

**可交付成果（Deliverables）：**

第1-2周：

· **市场调研报告**：分析校园内玩家（学生、教师）需求，了解对《三国杀》游戏的兴趣与期望。.docx/pdf

· **需求文档**：确定游戏的基本功能需求，如AI对战、武将技能、卡牌机制等。.docx/pdf

第3-4周：

· **角色与卡牌设计**：设计游戏中的武将角色、卡牌和技能，附带设计图和描述。.png

· **音频素材文件**：导入的音效、背景音乐和角色配音，确保游戏的音效元素完备。

.mp3/wav

#### ****第5-11周：****

1. **游戏开发代码**：Unity引擎下的完整游戏代码，核心功能如1V1 AI对战逻辑、武将技能、卡牌互动等。.cs（C# 代码文件）、.unity（Unity 场景文件）、.prefab（预制体文件）
2. **AI对战模型**：实现AI根据牌序与武将技能智能决策的功能，确保1V1对战能进行。.py(AI)
3. **数据库连接文档**：与游戏数据的连接，包括玩家数据的存储和读取功能。.sql(MySQL database)

第12周+：

1.游戏正式版本发布

2.用户反馈文档

3.更新与维护计划

非目标（Non-goals）

以下内容不属于项目范围，以避免项目范围膨胀或资源浪费：

1.多人在线对战：

该项目专注于1V1 AI对战，不涉及多人匹配、在线联机或实时服务器同步功能。

复杂的服务器架构：由于学生团队的资源限制，本项目不搭建云服务器，而是采用本地存储或轻量级数据库进行数据管理。

2.跨平台兼容：

游戏仅限于本地PC端，暂不开发移动端、web版本

3.高精度动画和3D建模：

游戏采用2D美术风格，不开发复杂的3D模型、骨骼动画或高级渲染效果。

4.大规模内容更新：

初版游戏仅包含基础角色、基本卡牌和AI对战机制，不会在首个版本内加入大量扩展内容（如额外游戏模式、游戏剧情等）。

5.商业化与盈利模式：

该项目的目标是提升师生互动，不开发商业化运营或付费系统（如内购、广告变现等）。

EEF（企业环境因素）:校园环境（非商业化）、技术限制（本地数据库）

OPA（组织过程资产~已有的资产）:《三国杀》规则、GitHub 开源AI代码、B站教程，数据库和AI模型开发经验，Unity开发经验

### ****Feasibility Check（可行性分析）****

本项目为\*\*《三国杀》学生团队开发版\*\*，可行性分析从\*\*技术（Technology）、财务（Finance）、时间（Time）、资源（Resources）\*\*四个维度进行。

#### ****1. 技术可行性（Technology Feasibility）****

* **数据库系统（SQLite/MySQL）**：技术难度低，适用于小型游戏，查询性能满足需求。
* **游戏开发（Unity）**：团队已有经验，支持2D/3D游戏开发，并提供强大物理引擎和动画系统。
* **AI对战逻辑（预计代码量待定）**：机器学习部分较复杂，但可通过现有算法优化。
* **美术与音效**：Photoshop/FaceSwapAI 等工具已确定，满足基本需求。

#### ****2. 财务可行性（Finance Feasibility）****

* **预算低，成本可控**：开发主要依赖免费工具和开源资源，可能的费用包括美术设计软件（月费 $7.90）和宣传/试玩活动（$0-$50）。
* **无需服务器成本**：本地存储或MySQL数据库方案，避免额外支出。

#### ****3. 时间可行性（Time Feasibility）****

* 项目预计开发周期为**一学期（约4-5个月）**。
* **关键任务周期**：
  + 数据库开发：1-2 周
  + 游戏核心功能（对战逻辑）：4-6 周
  + AI对战优化：4-6 周
  + 美术、音频及 UI 设计：4-6 周
  + 测试与优化：3-4 周

#### ****4. 资源可行性（Resource Feasibility）****

* **团队经验丰富，能够承担项目开发**。
* **硬件资源（电脑、开发工具）已具备**，无需额外投资。

## ****Resource Estimation（资源估算）****

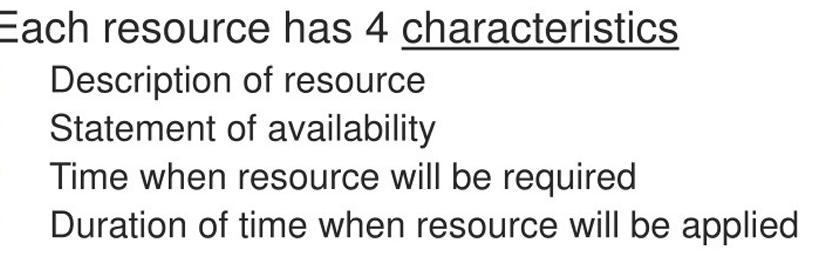
### ****1. 软件大小估算（LOC 方法）****

| **模块** | **预计代码行数（LOC）** |
| --- | --- |
| 数据库 | 150 行 |
| 对战逻辑 | 2000 行 |
| AI 逻辑 | 预估 800-1500 行（取 1200） |
| 游戏场景动画 | 400 行 |
| **总计** | **3750 行** |

### ****2. 资源估算（基于 Use Case + Decomposition 方法）****

资源估算按照**人员（People）、可复用软件组件（Reusable Software Components）、开发环境（Development Environment）三个维度，每个维度四个特性，并选择遵循Decomposition of Process**进行任务拆解。

#### ****人员资源估算****



| **资源** | **描述** | **可用性** | **需要时间** | **应用时间** |
| --- | --- | --- | --- | --- |
| 文浩羽 | 负责数据库、AI 及文档撰写 | 每周10-30h | 第1-14周 | 1-14周 |
| 黄一行 | 负责 Unity 游戏开发 | 每周25-35h | 1-14周 | 2-14周 |
| 孙诣然 | 美术/UI 设计 | 每周10-20h | 2-14周 | 4-14周 |
| 张昊展 | 音频、表格设计 | 每周10-20h | 2-14周 | 6-14周 |

#### ****（2）可复用软件组件资源估算****

| **资源** | **描述** | **可用性** | **需要时间** | **应用时间** |
| --- | --- | --- | --- | --- |
| Unity 物理引擎 | 物理碰撞、动画 | 高 | 即时 | 2-16周 |
| Unity AI 插件 | AI 计算 | 中等 | 第4周 | 第8-11周 |
| SQLite / MySQL | 数据存储 | 高 | 第4周 | 第8-11周 |

#### ****（3）开发环境资源估算****

| **资源** | **描述** | **可用性** | **需要时间** | **应用时间** |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Unity IDE | 游戏开发 | 高 | 即时 | 第2-14周 |  |  |
| Photoshop | UI 设计 | 高 | 即时 | 第1-5周 |  |  |
| VScode | 代码编写环境 | 高 | 即时 | 第1-14周 |  |  |

### ****总结****

* **软件代码量预计约 3750 行**，AI 代码量估算为 1200 行左右。
* **人力资源按 Use Case 进行分配**，确保关键任务（对战逻辑、AI、美术等）在学期内完成。
* **开发环境和软件组件可用性高**，预计不会成为瓶颈。

## ****成本估算（Cost Estimates）****

由于是**学生团队项目**，成本估算主要涉及**时间投入**和**基础资源**，预计不会产生高额资金消耗。1

### ****1. 人力资源成本****（基于团队成员的预计投入时间）

| **成员** | **角色** | **每周投入时间** | **总投入时间（学期内）** |
| --- | --- | --- | --- |
| **文浩羽** | 项目经理、数据库工程师、机器学习工程师、文书 | 10-30小时 | 约 150-200 小时 |
| **黄一行** | 游戏开发总工程师（Unity） | 25-35小时 | 约 200-250小时 |
| **孙诣然** | 游戏美术设计师, UI设计师，图像设计师 | 10-20小时 | 约 100-120小时 |
| **张昊展** | 游戏美术设计师，音频工程师，表格设计师 | 10-20小时 | 约 100-120小时 |

### ****2. 开发工具及软件费用****

| **资源** | **说明** | **费用** |
| --- | --- | --- |
| **Unity** | 使用 Unity 免费版 | **$0** |
| **数据库（SQLite / MySQL）** | 采用免费数据库 | **$0** |
| **音频处理软件（AdobeAudition）** | 免费开源软件 | **$0** |
| **美术设计工具（Photoshop /FaceSwapAI）** | 免费但限制流量 | **$0-$7.90/月** |

### ****3. 硬件资源成本****

| **资源** | **说明** | **费用** |
| --- | --- | --- |
| **开发设备（电脑）** | 团队成员自备 | **¥0** |
| **服务器或存储** | 本地存储，或使用MySQL | **¥0** |

### ****4. 其他费用****

| **资源** | **说明** | **费用** |
| --- | --- | --- |
| **校园宣传（传单，问卷）** | 可能打印成本 | **$0-$20** |
| **试玩活动** | 组织线下试玩可能涉及小额预算 | **$0-$30** |
| **团建** | 吃饭 | **$0-$100** |

除时间外，基本没有金钱成本

IF QUESTION ARISE:

1.项目文档：

基本 Proposal：存储在 project.docx/pdf，包含项目章程、目标、范围等。

详细信息：存储在 软件pre.docx，包括开发计划、技术文档、系统架构等。

代码与开发资源

2.代码需求：Google、GitHub（开源文件）

数据库：.sql 文件，存储游戏相关数据结构。提供开发工具与AI框架

3.游戏开发：Unity文件夹

AI 训练：PyTorch下的python文件

4.团队沟通平台

微信：用于实时讨论问题、同步项目进度、协调任务分配。

5.技术支持与问题解答

DeepSeek、ChatGPT：用于查询开发相关问题、优化 AI 训练、解决技术难点。

STEP2: Information Objectives (in scope ——stage 1)

**Information Objectives**：

* 输入数据：玩家选择的武将、当前双方手牌、AI的决策信息、剩余牌堆信息、玩家操作等。
* 输出数据：战斗结果（如胜败）、AI的决策（包括攻击、防御、选派等）、游戏结束时的提示信息

**STEP 3**:Function and Performance (~in scope in stage 1)

**软件系统需要执行什么功能（What function does the software system perform on i/p to produce o/p）？**

* + - 接收玩家输入的卡牌数据，并转化为血量、手牌和剩余牌堆的变化，交由AI进行战斗分析
    - 根据游戏规则，计算每回合的结果。
    - 根据血量结算战斗结果（胜败判断、玩家状态更新等）,记录到数据库。

**What level of performance is required /**Acceptance Criteria **？需要什么级别的表现**

**· 首先 正确运行：正确的将玩家操作转化为具体的状态变化**

· **响应时间**：玩家输入应该在**1秒内**得到响应，系统快速加载游戏界面和战斗数据。

· **AI对战决策时间**：AI决策过程应该不超过4**秒**，保证流畅的对战体验，避免玩家长时间等待AI做出决策。

· **资源使用**：游戏应当能够在校园机房计算机上流畅运行，不要求高性能的硬件支持，内存占用尽可能低，CPU利用率适中。

**Problem Decomposition(stage2)**

***This activity is at core of requirements analysis）***

**之前在scope中阐述了：游戏的需求是**：

* **核心玩法**：实现1V1对战模式、卡牌使用。
* **AI对战系统**：让玩家能与AI进行对战，AI根据局势作出决策。
* **数据存储与管理**：设计数据库，管理玩家数据、对局从操作记录等信息。

对此，我们将这个大项目拆解成四个部分：

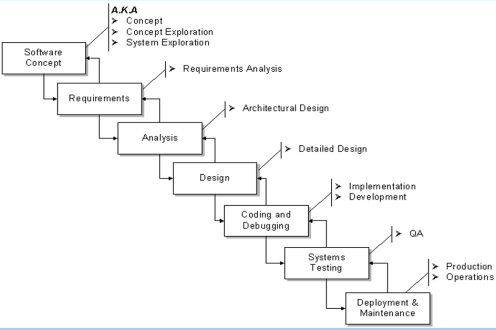
1. 游戏核心机制：Unity开发的游戏核心代码
2. AI对战系统：pytorch训练的AI对战模型
3. 数据储存与管理：MySQL支持下的数据库创建与管理
4. 客制化音画设计：根据用户需求设计的“老师”形象与音频模拟

Process：？？？？？？？？？？？？？？

瀑布模型、敏捷开发、Scrum、V模型等？？？？

要根据 客户需求 产品环境 产品特性 项目环境（组内工作周期阶段的环境~）

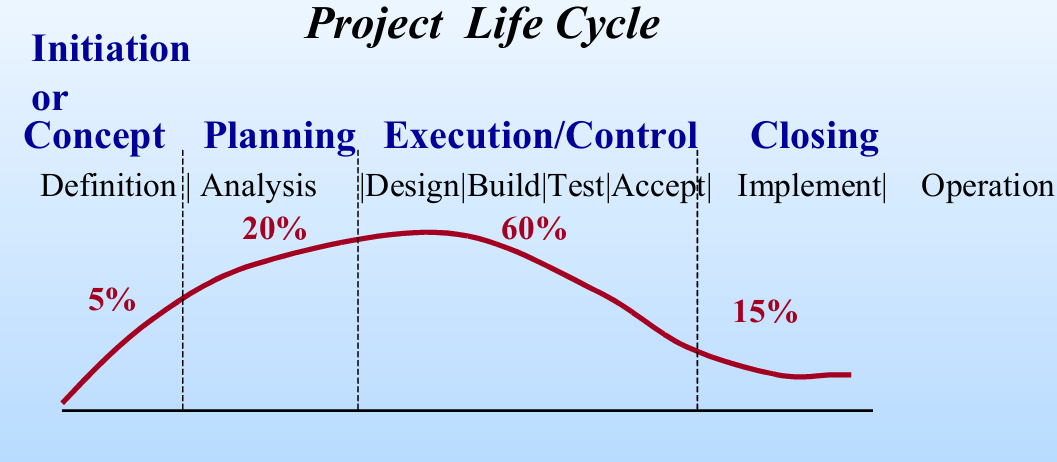
Project planning begins once model is selected



Process decomposition？？？？？：分解并阐释！！每个流程框架内的任务 ~~取决项目复杂度

**The Project~~组织内协同工作各个阶段分隔、阶段功能指定 （第4个P of 4P’s theory）：**

**最好能够使用现有的软件组件**



这我们在之前的Milestone介绍过：可以看出我们有着类似的时间分配，并且有着严格scope来避免project risk：（具体结合解释一下怎么避免risk）

第1-2周：收集反馈，分析市场需求，确定游戏功能和玩法。确定游戏的设计方向和目标群体，编写初步需求文档。

第3-4周：设计游戏界面、系统架构，确定角色和卡牌设计。解析并导入音频素材，完成初步音效、画面设计。

第5-11周：开发并连接数据库与AI对战模型，确保AI能够智能化对战。完善游戏对局代码的编写，进行单元测试。

第12+周：游戏正式上线，进行校园内推广。收集玩家反馈并开始进行持续的维护和优化。

W5HH principle中的五个：

目的，阶段性计划，职能分配，资源分配，以及“如何解决技术和管理问题” 都在scope已阐明

关于Where are team members如何找到我们的成员呢？我们不擅长逃课，每节课都可以在这找到我们，或者email 1220008351@must.edu.mo；至于我们组织内用腾讯会议和微信就够了

Process介绍管~敏捷开发 scrum model~要有！！！！

用leangoon写sprint截图

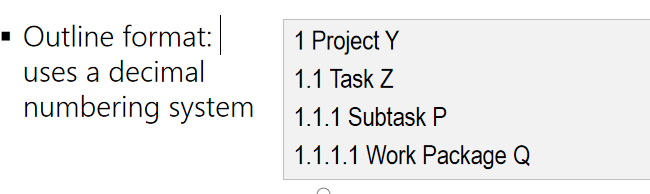
基于use case 预算

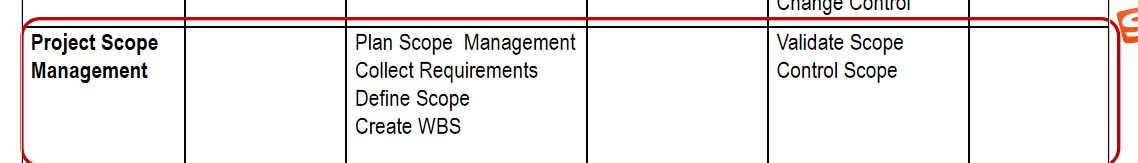
**Construct** a WBS using the analogy, top-down, bottom-up, and mind mapping approaches

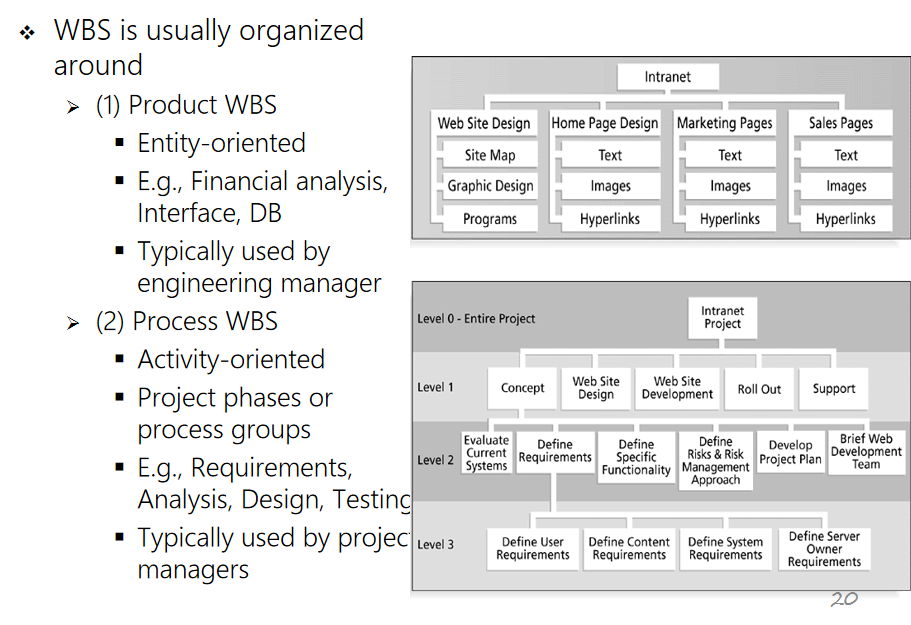
创建工作分解结构 (WBS)（管project） – 将主要项目可交付成果和项目工作细分为更小、更易于管理的组件

树状图/列表

A WBS may be ~~a diagram (graphical tree) or~~ a text list(outline-format)我们用下面图示这种







软件项目最多 2-3 周，最少 1 天

使用“一对二”规则：

不要太小以至于无法进行微观管理

我们用自上而下的方法：从项目中最大的项目开始，然后将其分解

如何搞WBS:1.按客户有要求的guideline大纲 ；2.自己之前项目的WBS；3.自上而下的方法

从项目中最大的项目开始，并不断将其分解为下属元素（best if 问题不新颖，易理解，类似之前的项目，普遍适用）4.自下而上 5.头脑风暴

一个工作单元应仅出现在 WBS 中的一个位置。

WBS 项的工作内容是其下方 WBS 项的总和。

WBS 项仅由一个人负责，即使可能有许多人正在处理它。

W​​BS 必须与实际执行工作的方式一致。

项目团队成员应参与制定 WBS，以确保一致性和认同。

必须记录每个 WBS 项，以确保准确了解该项目中包含和不包含的工作范围。

P30 :每个WBS要有Dictionary：简单的 WBS 字典描述了每个 WBS 元素中的内容，并可能说明元素中不包含的内容，可以交叉引用

WBS之后 验证scope validate范围和scope control控制范围要包括！！！

scope validate：

Control Quality：

* AI 训练的准确性和合理性 达到80%以上
* 游戏界面无重大 Bug
* 音效、美术资源符合标准，音质图像清晰

开发后，要求验证validate：

· **游戏 UI/UX 设计**：界面是否符合玩家习惯，是否容易理解。

· **AI 逻辑测试**：面对用户， AI 能正确执行操作，不会出现异常行为。

· **玩法验证**：确认 1v1 规则、机制、和武将设定符合预期，并得到用户认可

scope control：

· **需求变更管理**：如果开发过程中发现某个（老师）武将血条设计不符合人物设定，需要提交变更申请，经过讨论后再决定是否修改。

· **功能范围控制**：限制游戏只做 1v1 AI 对战，而不扩展到多人模式，以保证项目按期完成。

· **资源管理**：确保美术、音频、AI 训练等资源投入不会超出计划范围，多采用开源免费软件，如需改变需更改cost记录。

**Metrics For Process & Projects**

**1. 功能点计算（Function Points Calculation）**

**选择武将、游戏对战、AI决策、游戏结算、游戏设置、历史记录**

**· 数据通信：系统是否依赖数据通信（例如，网络通信、数据传输）。 3 ——**Value Adjustment Factors

**· 分布式处理：系统是否涉及分布式处理（如跨多个服务器或地区的计算）。 0**

**· 性能要求：系统是否有严格的性能要求（如响应时间、处理速度等）。 2**

**· 用户界面：系统是否需要复杂的用户界面。 3**

**· 处理逻辑复杂性：系统的核心处理逻辑是否复杂。 4**

**· 事务处理：系统是否需要高频率的事务处理。 2**

**· 并发用户数量：系统支持的并发用户数量是否很多。 1**

**· 安全要求：系统是否涉及复杂的安全要求（如加密、访问控制）。 1**

**· 接口与集成：系统是否需要与其他系统或设备进行复杂的接口和集成。 4**

**· 数据质量要求：系统是否有严格的数据质量要求（如数据一致性、完整性等）。 3**

**· 可恢复性：系统是否需要有高可用性和灾备恢复的能力。 2**

**· 可维护性：系统是否需要易于维护和扩展的架构。 4**

**· 外部环境变化：系统是否需要应对快速变化的外部环境（如法规变化、市场变化等）。 2**

**· 部署平台：系统是否需要在多个平台上运行（如跨操作系统、硬件平台等）。 1**

CAF=0.65+(0.01×32)=0.65+0.32=0.97 （32就是14个Value Adjustment Factors的总得分）

· **选择武将**：这是游戏的一大功能，涉及武将选择、技能展示、确认等操作。可以算作 4-5 个功能点。

· **游戏对战**：包括玩家对战的核心功能，如出牌、发动技能、结束回合等操作。这个功能复杂度较高，可以算作 8-10 个功能点——function points。

· **AI决策**：如果AI模型需要做决策（例如选择出牌、使用技能等），这部分复杂度较高，估算为 6-8 个功能点。

· **游戏结算**：包括游戏结束后的结算，判断胜负，记录结果等，约为 4-5 个功能点。

· **游戏设置**：调整音量、游戏规则、语言等设置，估算为 2-3 个功能点。

· **历史记录**：存储和显示历史战绩，涉及数据管理和展示，可以算作 3-4 个功能点。

总计约**33 个**

FP=33×0.97=32.01<50

**小型项目**：功能点（FP）小于 50

优化后的功能点方法：

### ****第一步：计算未调整的功能点数（UFP）****

整理数据功能和交易功能：

#### ****数据功能****

**内部逻辑文件（ILF）**：

* + 卡牌信息：1
  + 武将信息：1
  + 历史记录：1
  + 游戏设置：1
  + **总计：4**

**外部接口文件（EIF）**：

* + 数据库：1
  + AI接口：1
  + **总计：2**

#### ****交易功能****

**外部输入（EI）**：

* + 出牌：1
  + 弃牌：1
  + 选牌：1
  + 确认行动：1
  + 游戏设置调整：1
  + **总计：5**

**外部输出（EO）**：

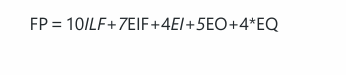
* + 游戏结果：1
  + 提示信息：1
  + **总计：2**

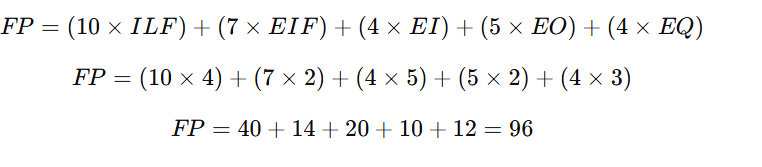
**外部查询（EQ）**：

* + 查询卡牌信息：1
  + 查询武将信息：1
  + 查询历史记录：1
  + **总计：3**

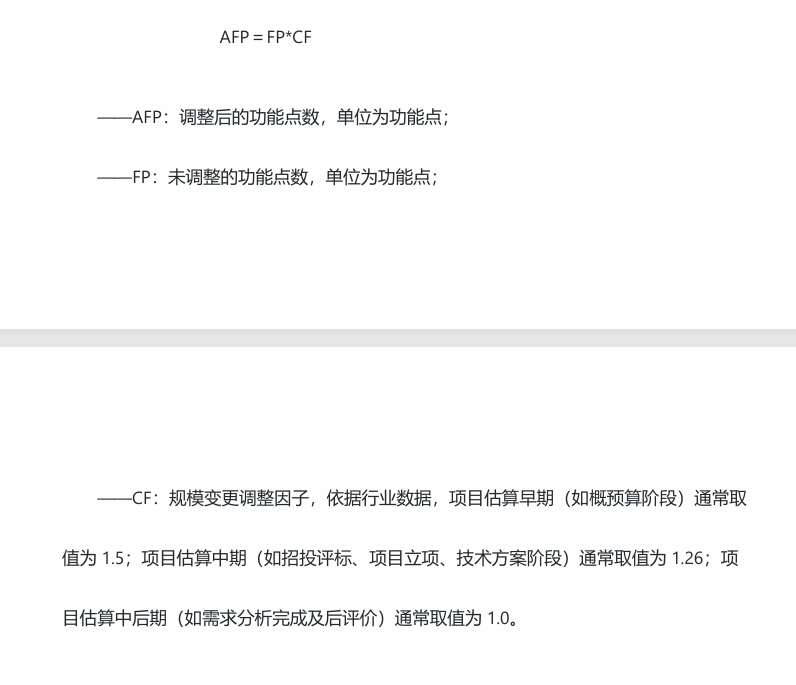
### ****第二步：计算 UFP（未调整功能点数）****

使用估算功能点计算公式：

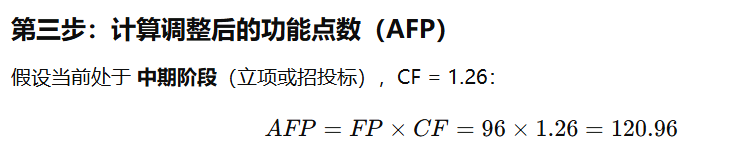




### ****第三步：计算调整后的功能点数（AFP）****



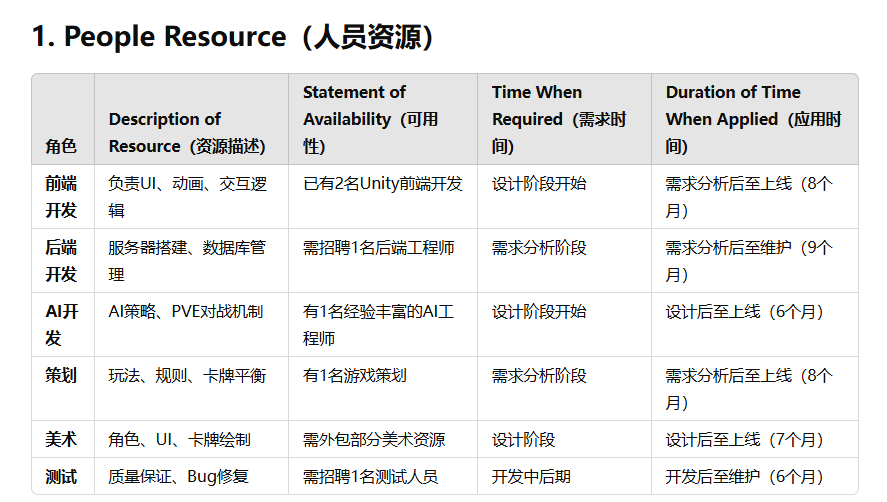
假设当前处于 **中期阶段：**CF = 1.26：

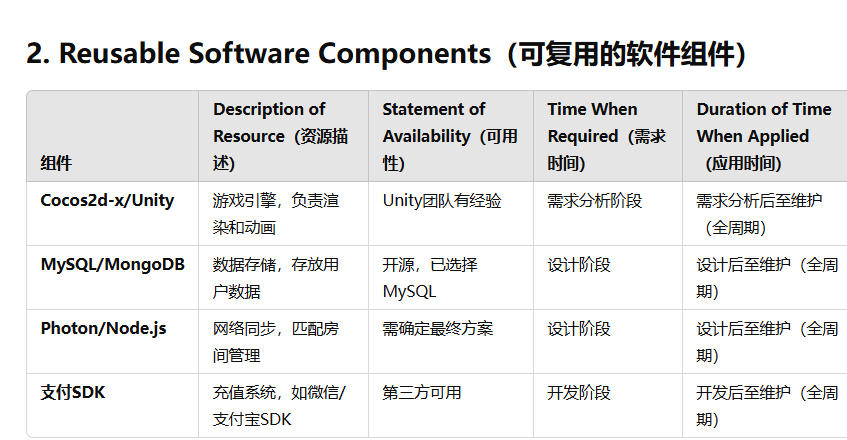


### ****Feasibility Check 分析（基于三国杀项目）****

本分析涵盖以下三大资源类别，每个资源包含**4个特性**：

1. **People Resource（人员资源）**
2. **Reusable Software Components（可复用的软件组件）**
3. **Development Environment（开发环境）**

表要具体修改！！！



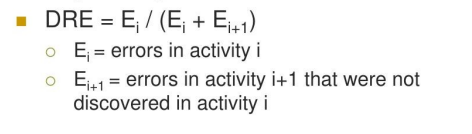
无支付系统！！

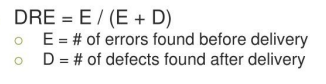


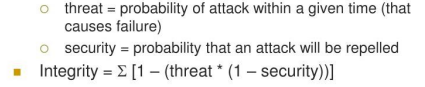
**Feasibility Check 结果：**

1. **人员资源**：
   1. 需招聘**1名后端开发、1名测试**，部分美术外包。
2. **可复用组件**：
   1. 采用**Unity + Node.js + MySQL**方案，第三方SDK降低开发成本。
3. **开发环境**：
   1. 服务器购买是最大成本，需优化预算

**DRE** Defect Removal Efficiency**由于项目还未发布，无法计算，我们无法预料开发中还会遇到哪些代码问题**







完整性=1−(威胁×(1−安全性))

**威胁**：假设攻击发生的概率为 **0.1**（10%）。

**安全性**：系统完全无法抵御攻击，因此安全性为 **0**（0%）。

完整性=1−(0.1×(1−0))=1−(0.1×1)=1−0.1=0.9完整性=1−(0.1×(1−0))=1−(0.1×1)=1−0.1=0.9

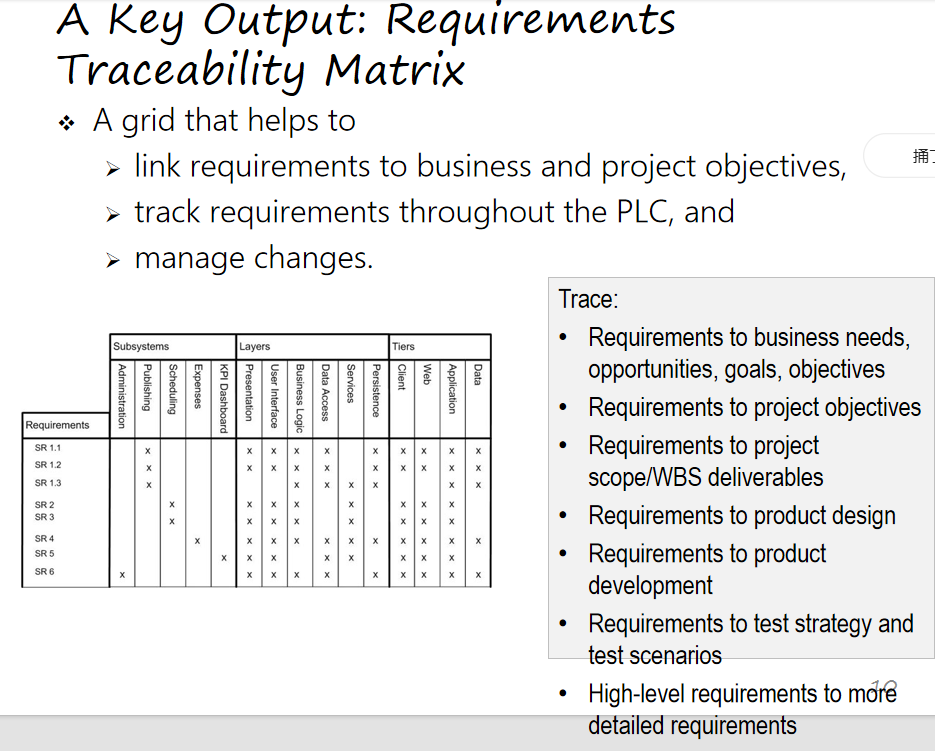
这意味着你的系统的完整性为 **90%**。虽然完整性仍然较高，但系统的安全性为 0，表明系统在遭受攻击时无法保护自身。（本地运行的你想改就改咯）

### Use-case oriented Metrics

10个主要用例来测试我们游戏的基本功能

应用程序我们保守估计？？？行代码

测试用例10个/per测试用例 共100个



管需求分析

网格有助于

将需求与业务和项目目标联系起来，

在整个 PLC 中跟踪需求，并管理变更

追踪：

对业务需求、机会、目标、宗旨的要求

对项目宗旨的要求

对项目范围/WBS 可交付成果的要求

对产品设计的要求

对产品开发的要求

对测试策略和测试场景的要求

对高级要求到更详细要求

***LEC4（共2个）和SCRUM未看***

***需求矩阵文档Traceability matrix***

**Construct** a WBS***（1.1 1.1.1~~）分product 和process 两个 （见Lec3——名为Lec02b P18页）***

***敏捷开发sprint截图-leango ~***