

**Web-based Linux operating system**

**CS341 Operating Systems Laboratory**

D2

Wen Haoyu

Huang Yixing

Zhang Zhifan

A report submitted to fulfil requirements for Course CS341

School of Computer Science and Engineering

Macau University of Science and Technology

# Table of Contents

[Table of Contents II](#_Toc8264)

[List of Figure II](#_Toc24840)

[Abstract IV](#_Toc24840)

[Chapter 1 Introduction 1](#_Toc27599)

[1.1 Background 1](#_Toc30864)

[1.2 Project Objectives 1](#_Toc30864)

[1.3 Report Structure 1](#_Toc30864)

[Chapter 2 Methodology 3](#_Toc8895)

[2.1 Programming Environment 3](#_Toc6029)

[2.2 System Design 3](#_Toc6029)

[Chapter 3 Result and Analysis 4](#_Toc18240)

[3.1 Offline version of Linux 4](#_Toc15908)

[3.1.1 The code 4](#_Toc18495)

[3.1.2 Problems encountered 6](#_Toc2697)

[3.1.3 The solution to the problem 7](#_Toc2697)

[3.2 Web version of Linux 8](#_Toc15908)

[3.2.1 Web page implementation 8](#_Toc18495)

[3.2.2 Information transmission 9](#_Toc2697)

[3.2.3 Difficulties encountered 1](#_Toc2697)0

[Chapter 4 Conclusions/Recommendations 1](#_Toc23302)1

[References 1](#_Toc14321)2

[Appendix 1](#_Toc14321)3

# List of Figures

[Figure 3- 1 Library 4](#_Toc22985)

[Figure 3- 2 Function 5](#_Toc22985)

[Figure 3- 3 User View 1 6](#_Toc31088)

[Figure 3- 4 User View 2 6](#_Toc32645)

[Figure 3- 5 Error 6](#_Toc25494)

[Figure 3- 6 reason 7](#_Toc9251)

[Figure 3- 7 Solution 7](#_Toc27950)

[Figure 3- 8 Event Listener 8](#_Toc16170)

[Figure 3- 9 CSS 8](#_Toc6740)

[Figure 3- 10 Web View 1 9](#_Toc19500)

[Figure 3- 11 Web View 2 9](#_Toc2193)

[Figure 3- 12 Transmission mode 9](#_Toc18890)

[Figure 3- 13 Node.js 10](#_Toc7765)

[Figure 3- 14 Server 10](#_Toc9660)

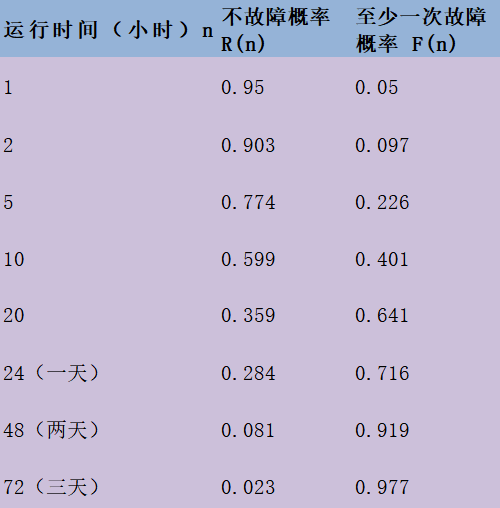
Chapter 9 Software Project Management

## 9.1 Software Reliability

1. Definition:

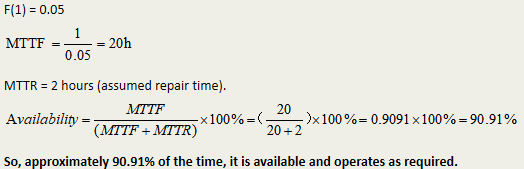
* Reliability = 0.95, in each complete game or interaction operation, there is a 95% probability of failure-free operation.

1. Relationship of Software Reliability & Failure:



* F(n) = 1 - R(n)
* R(n): the probability of no failure occurring within a specified time period.
* F(n): the probability of at least one failure occurring within a specified time period.
* We use hours as the basic unit.  
   R(1) = 0.95  
   F(1) = 1 - R(1) = 0.05  
   R(n) = 0.95ⁿ  
  F(n) = 1 - R(n) = 1 - 0.95ⁿ

1. Availability = [MTTF / (MTTF + MTTR)] \* 100%



## 9.2 Software Safety

1. Definition:

* Identify & assess potential hazards that could lead to system failure, focusing on technological risks

1. Process

* Categorize hazards by criticality and risk
* Analyze severity & probability of occurrence

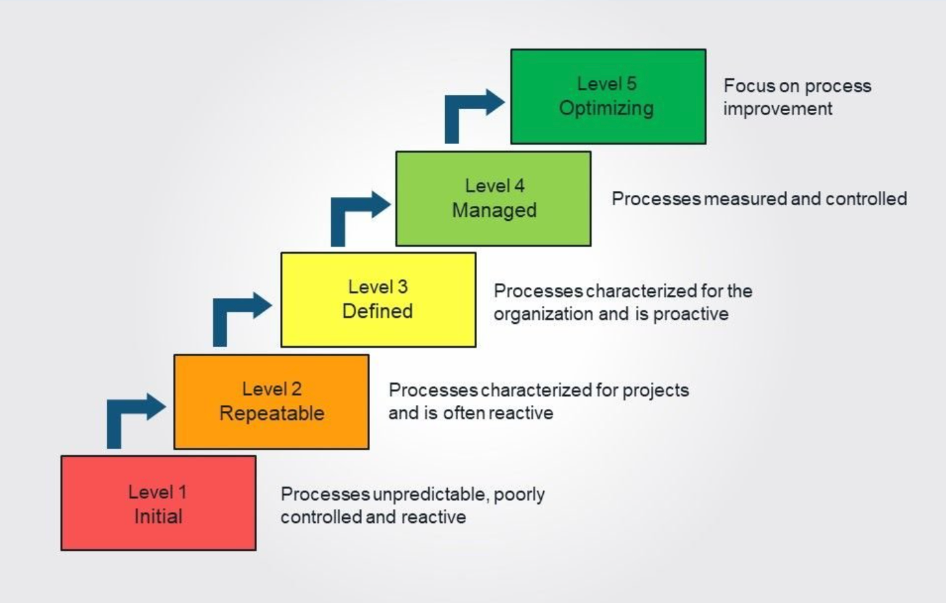
1. Key Difference from Reliablility:

* Reliability: Statistical analysis of failure likelihood
* Safety: Examines how failures create hazardous conditions

## 9.3 Quality Standard

* ISO 9001:2000: Specifies 20 requirements for quality management systems.
* ISO 9000-3: Guidelines for adapting ISO 9000 to software development processes.
* Certification: Requires third-party audits for compliance, with semiannual surveillance reviews.

## 9.4 CMM Model



1. **Our team adheres to CMM instead of CMMI, and we are at Level 2 - Repeatable:**

* We focus on the standardization and improvement of software development and process management, which aligns with the goals of CMM.
* There are no requirements for multi - domain integration. We concentrate on a single domain, namely game software development, without any in - depth integration across multiple domains.

1. **CMM Level**

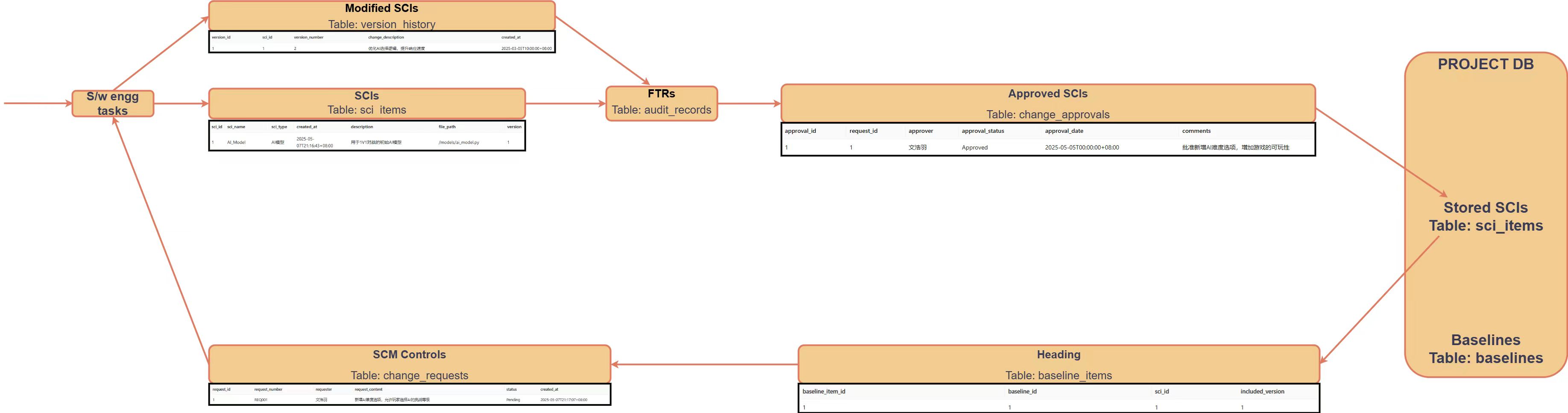
* INITIAL: The team has already carried out certain process management and documentation work.
* Repeatable: The team possesses certain project management capabilities. Clear management processes have been established in aspects such as resource and cost estimation, communication methods, and project charters, enabling the replication of successful project experiences.
* Defined: There is a certain degree of process standardization in project documentation, requirements analysis, and resource estimation.
* Managed: For example, there is no quantitative management and continuous dynamic optimization of the database, so this level has not been achieved.
* In summary, our team's CMM is at Level 3.

## 9.5 SCM

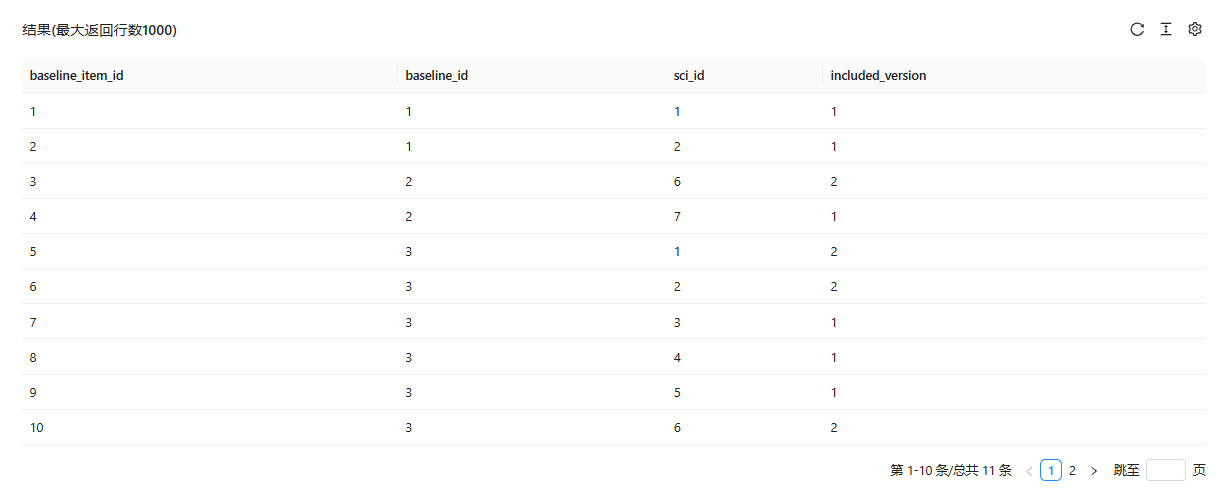
1. SCM Fundamentals

* Identifying s/w work products that are likely to change
* Establishing relationships among them
* Defining mechanisms for managing different versions of these work products
* Controlling changes
* Auditing and reporting on the changes made

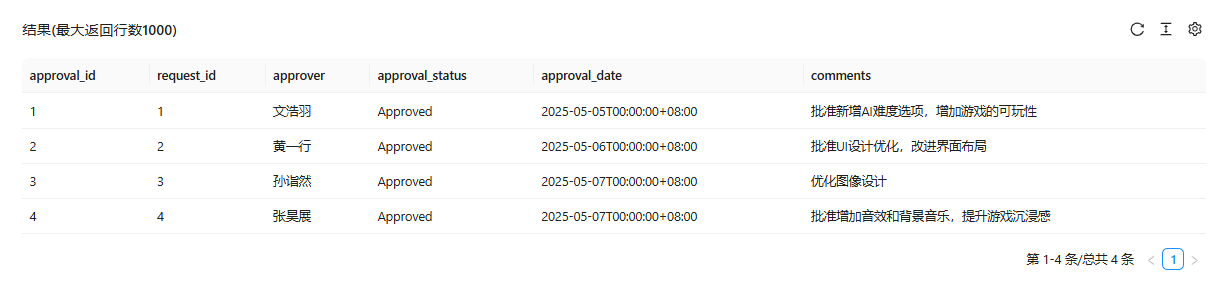
1. SCIs & Baselines

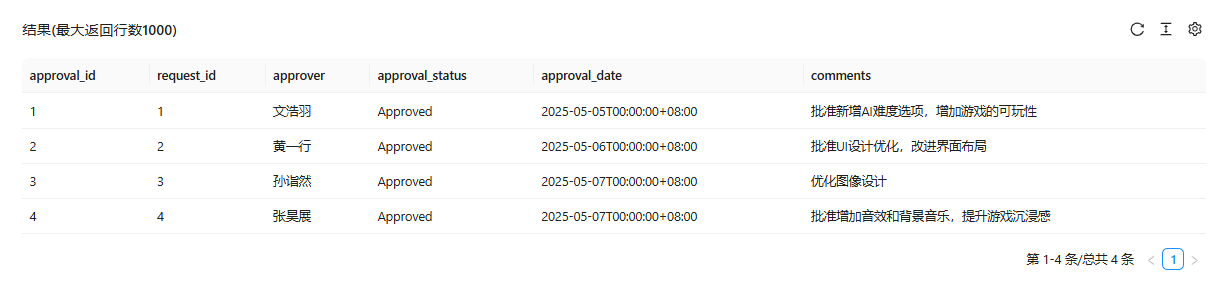


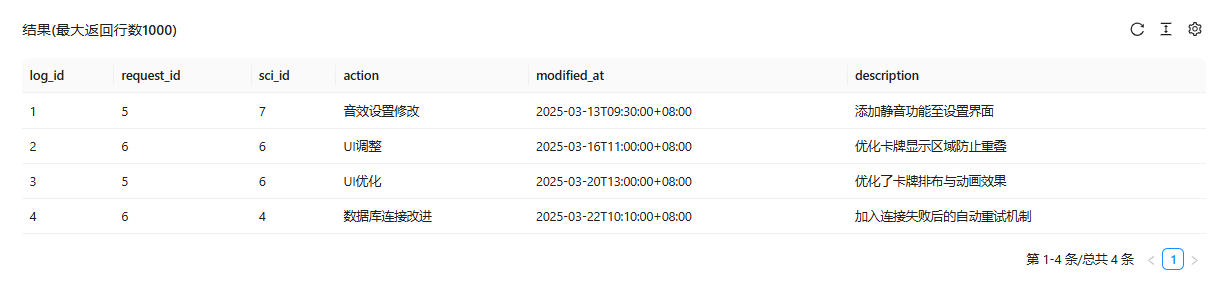




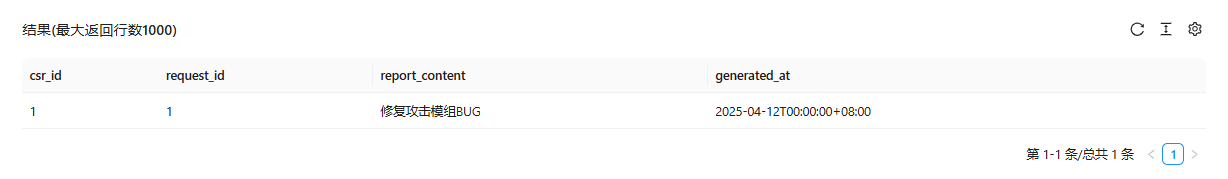


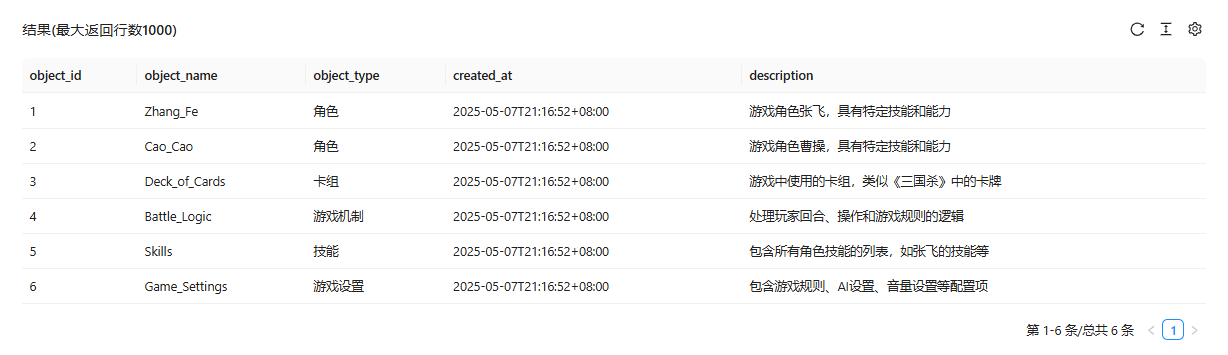


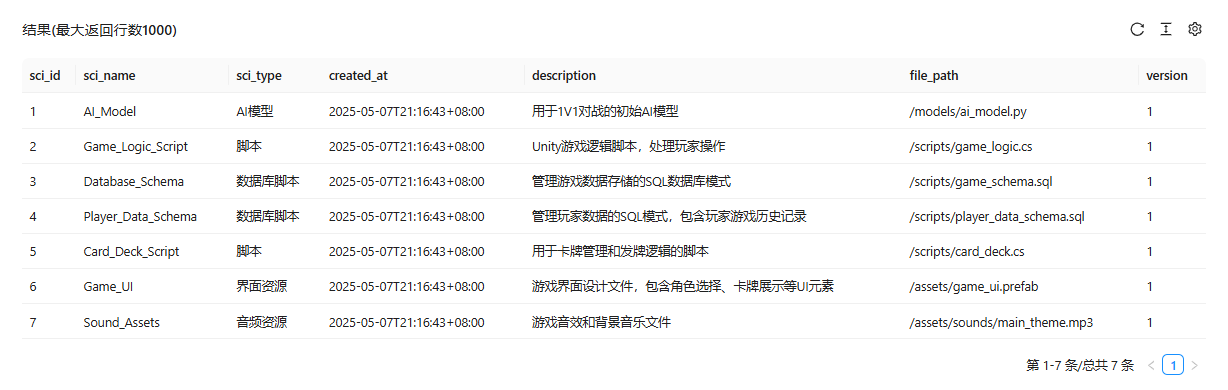




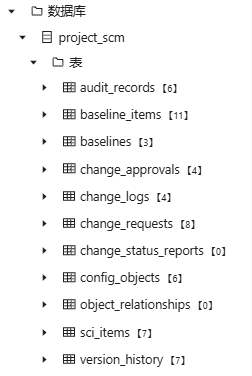




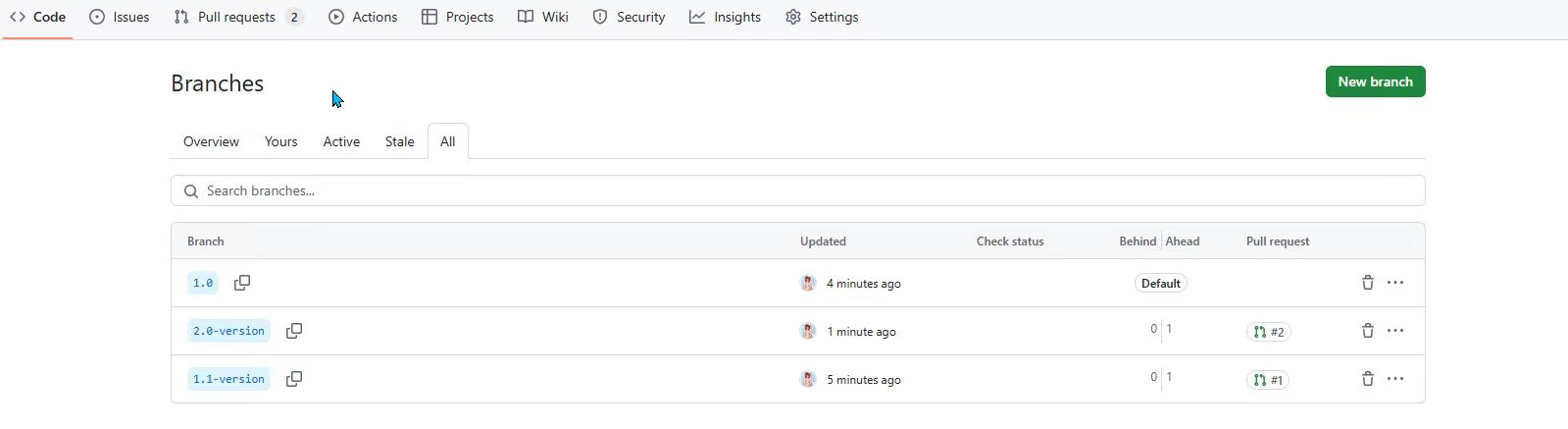


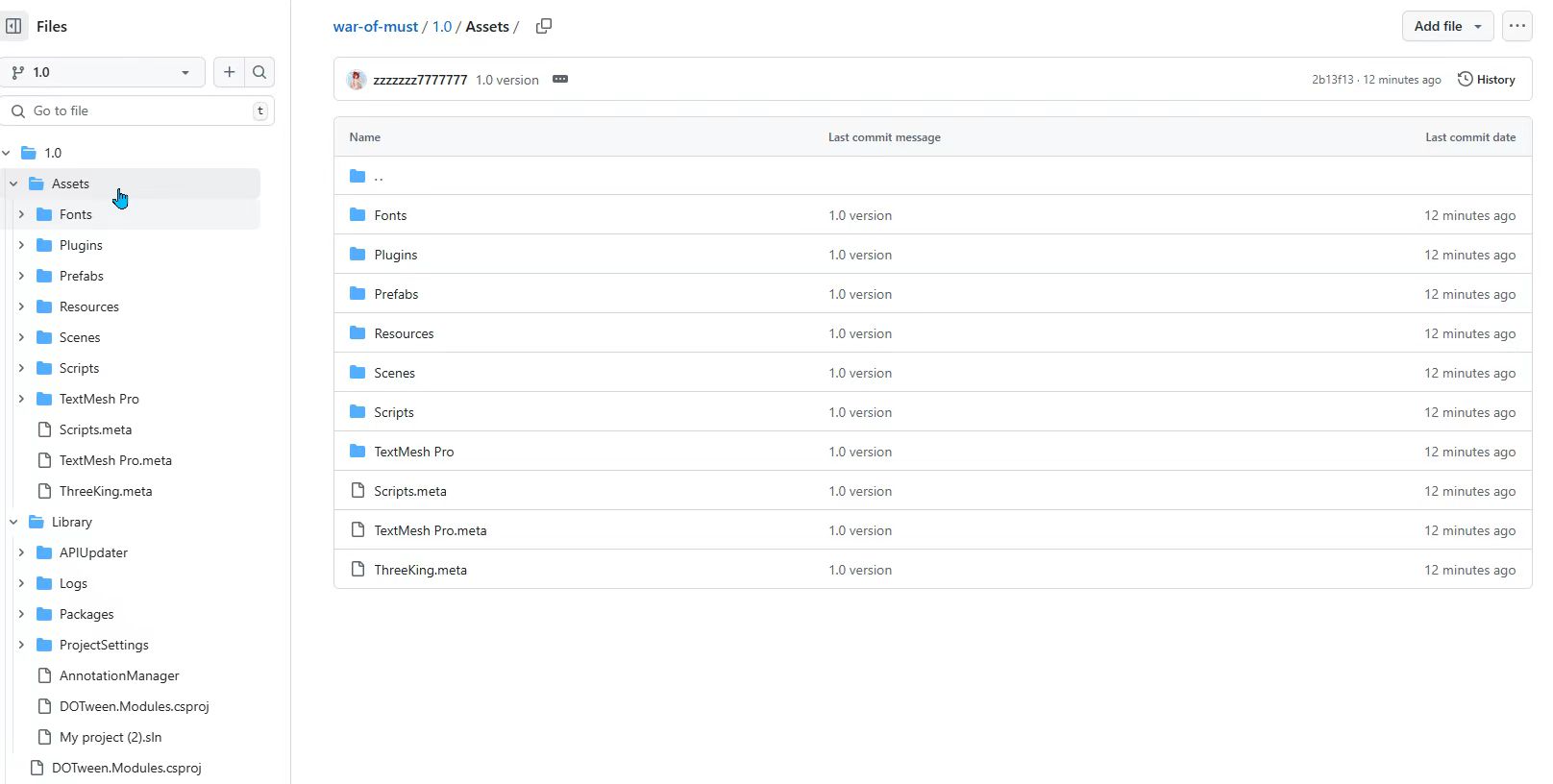


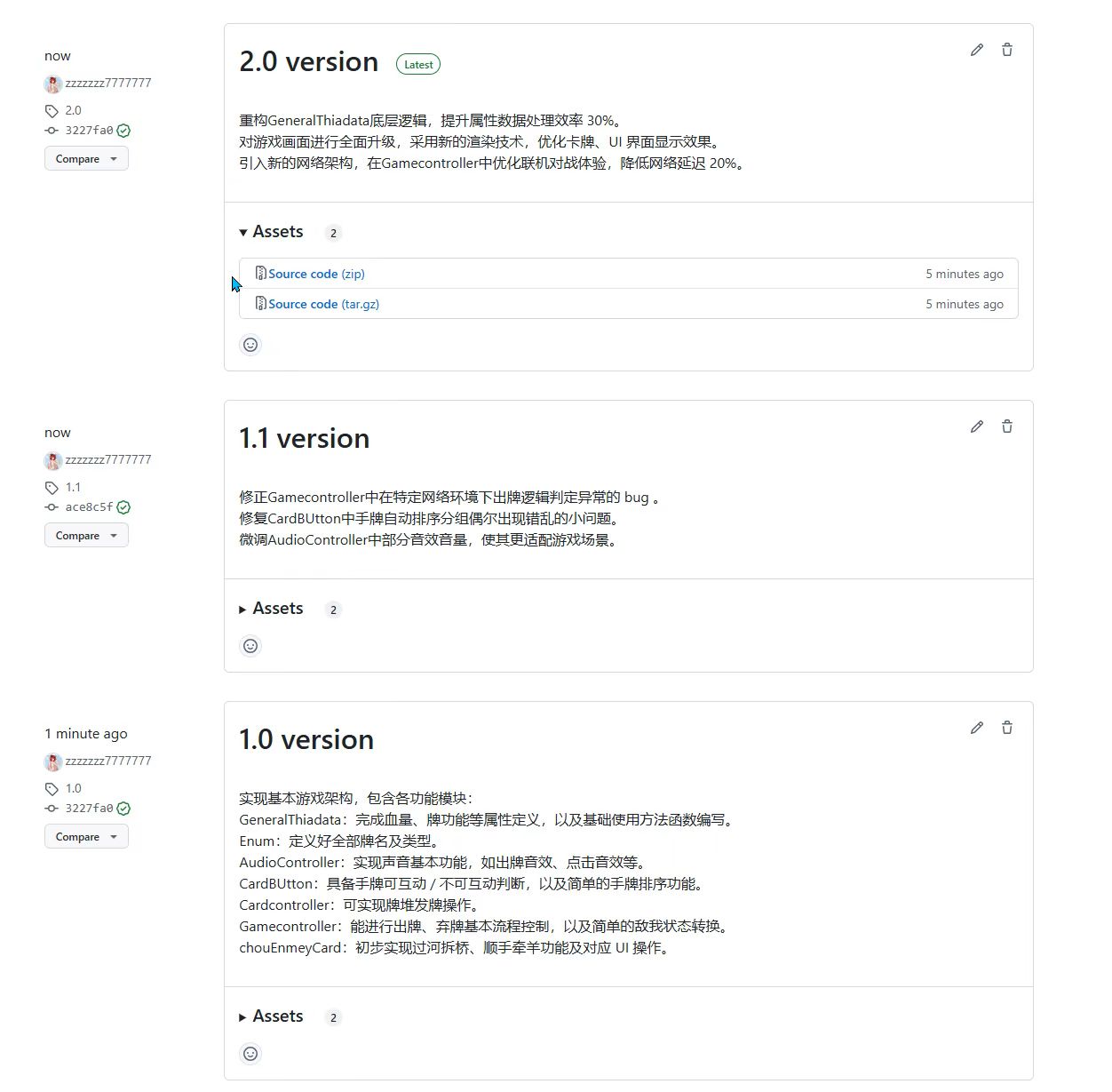
1. Configuration Objects



## 9.6 Version Control





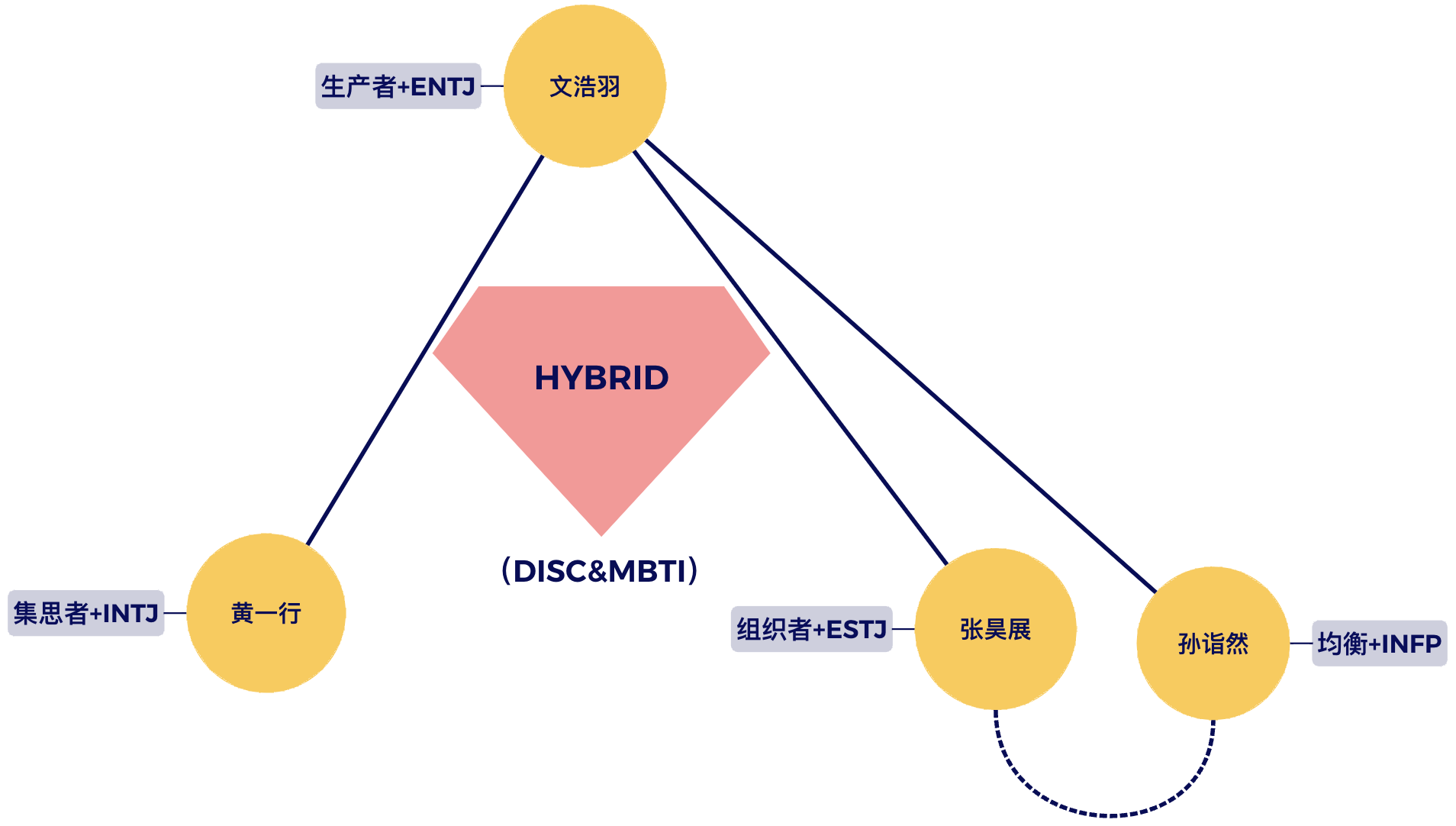


Chapter 10 Software Team Organization and Teams

## 10.1 Team structure

* Stakeholders : The MUST students & teachers
* Team Leaders ：Wen Haoyu
* Software Team ：ALL members
* Agile Teams ：ALL members
* Software Team Organizations : Controlled decentralized & Hybrid Structures（Type II —— Use hierarchies of organizational structure.）

## 10.2 Personality test



## 10.3 Contribution-based evaluation

|  |  |  |
| --- | --- | --- |
| **NAME** | **SCORE** | **Completed Tasks** |
| Wen Haoyu |  |  |
| Huang Yixing |  |  |
| Sun Yiran |  |  |
| Zhang Haozhan |  |  |