

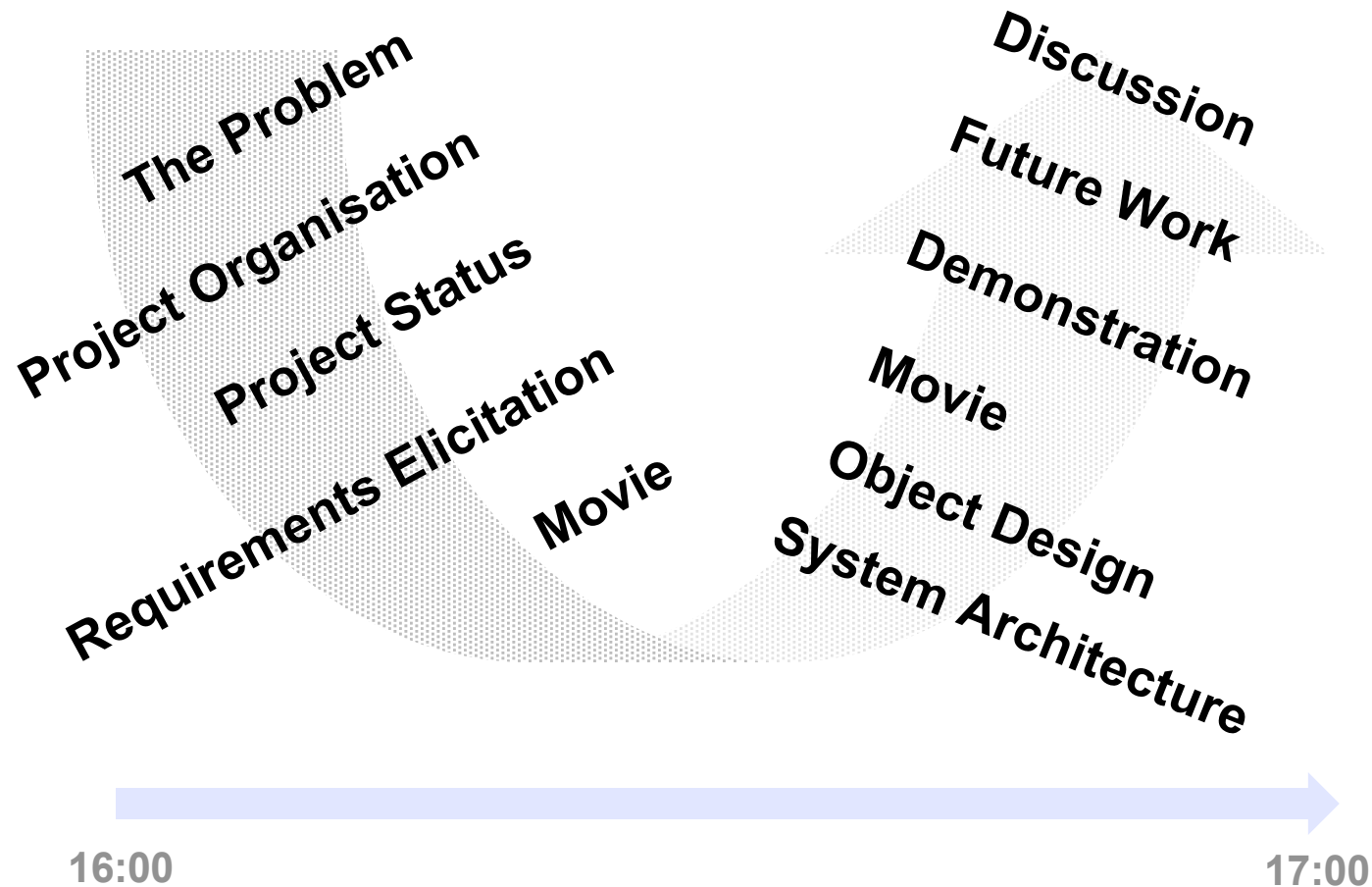
Client Acceptance Test

Michel Atoudem Kana



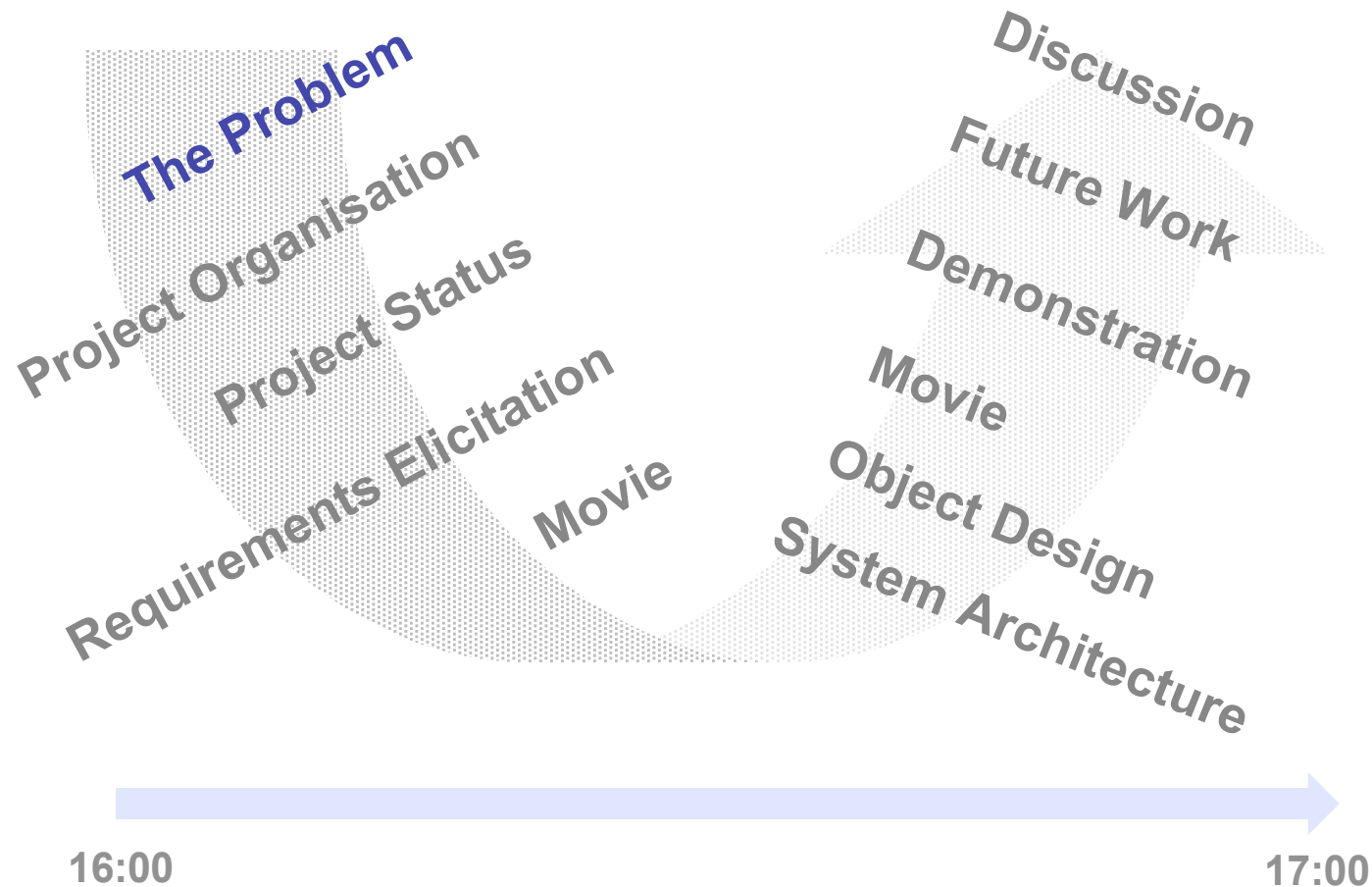


Content





Content





The Problem

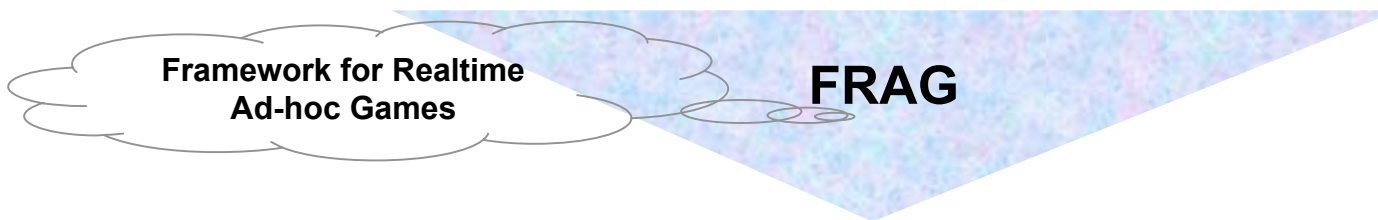
➤ **Multiplayer Online Games (MOGs)**

- client/server
- extensible maps, objects, weapons
- free movement

➤ **Peer-to-Peer networking (P2P)**

- messaging
- file exchange

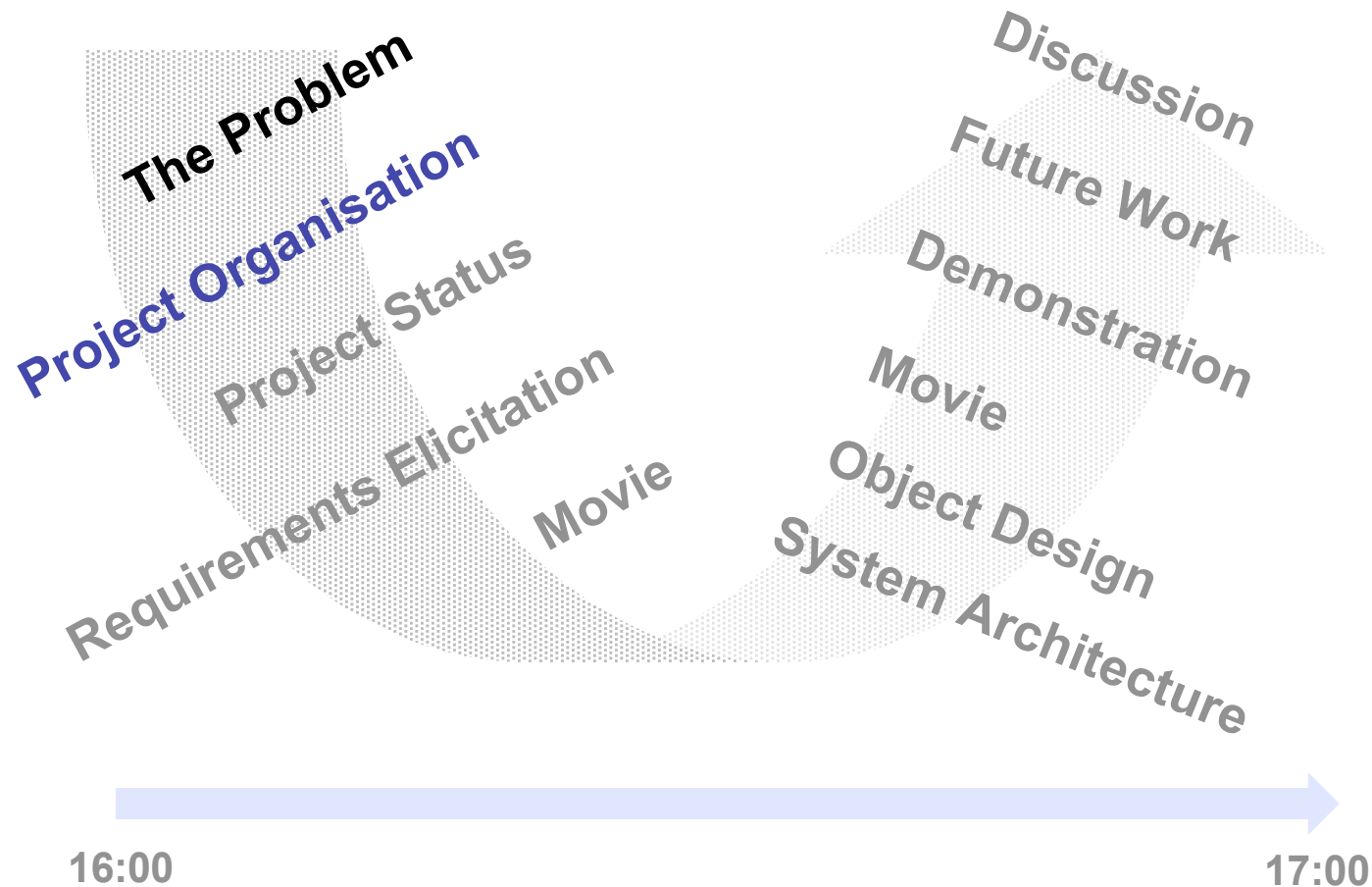
**Broadband
communication
networks**



**develop a peer-to-peer multiplayer realtime online game
(SWORD)**



Content



ARENA Project Organization



Client Team



Manager Team

**Cross-functional
Team**

Developer Team

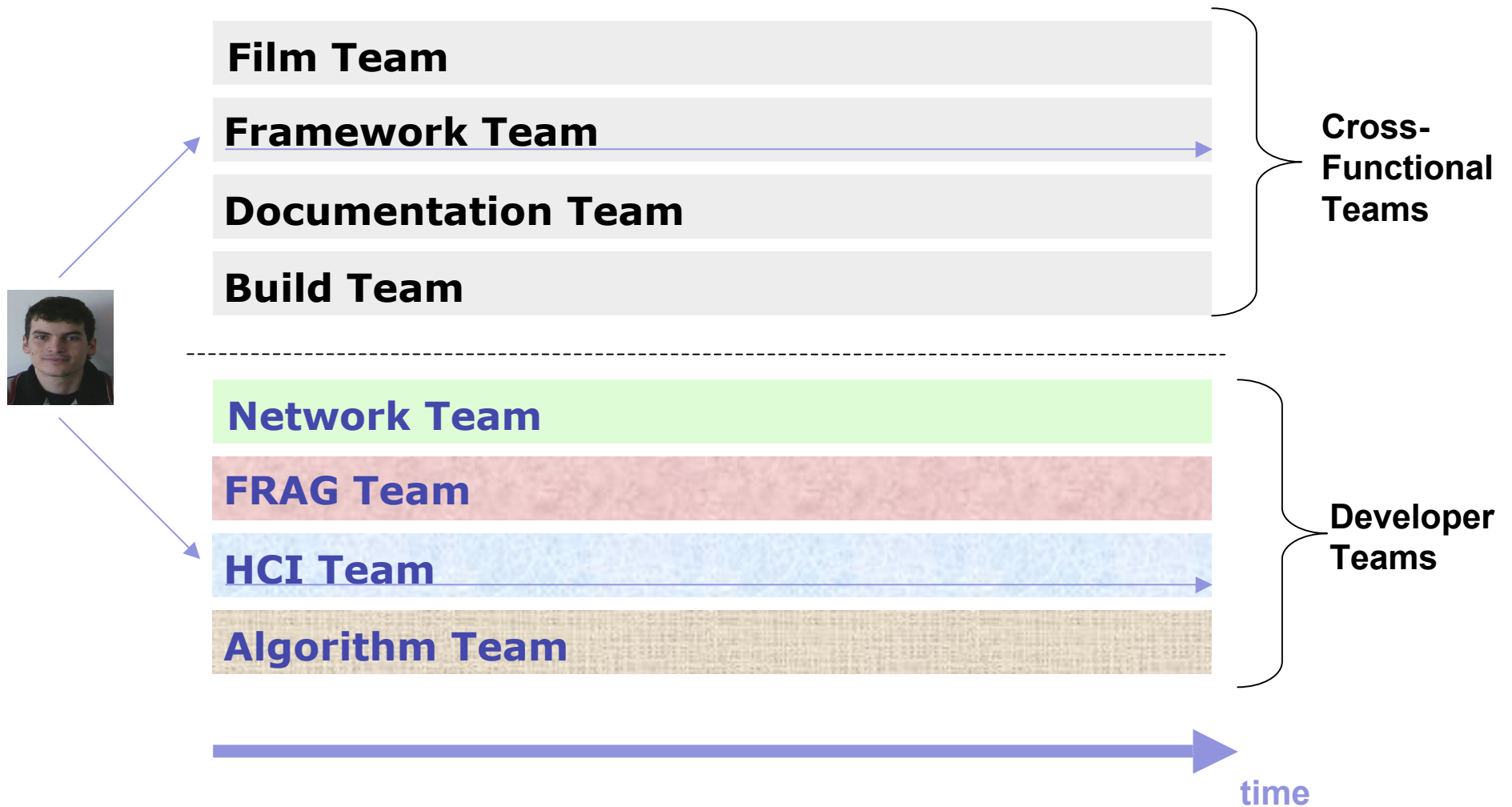


Coach Team



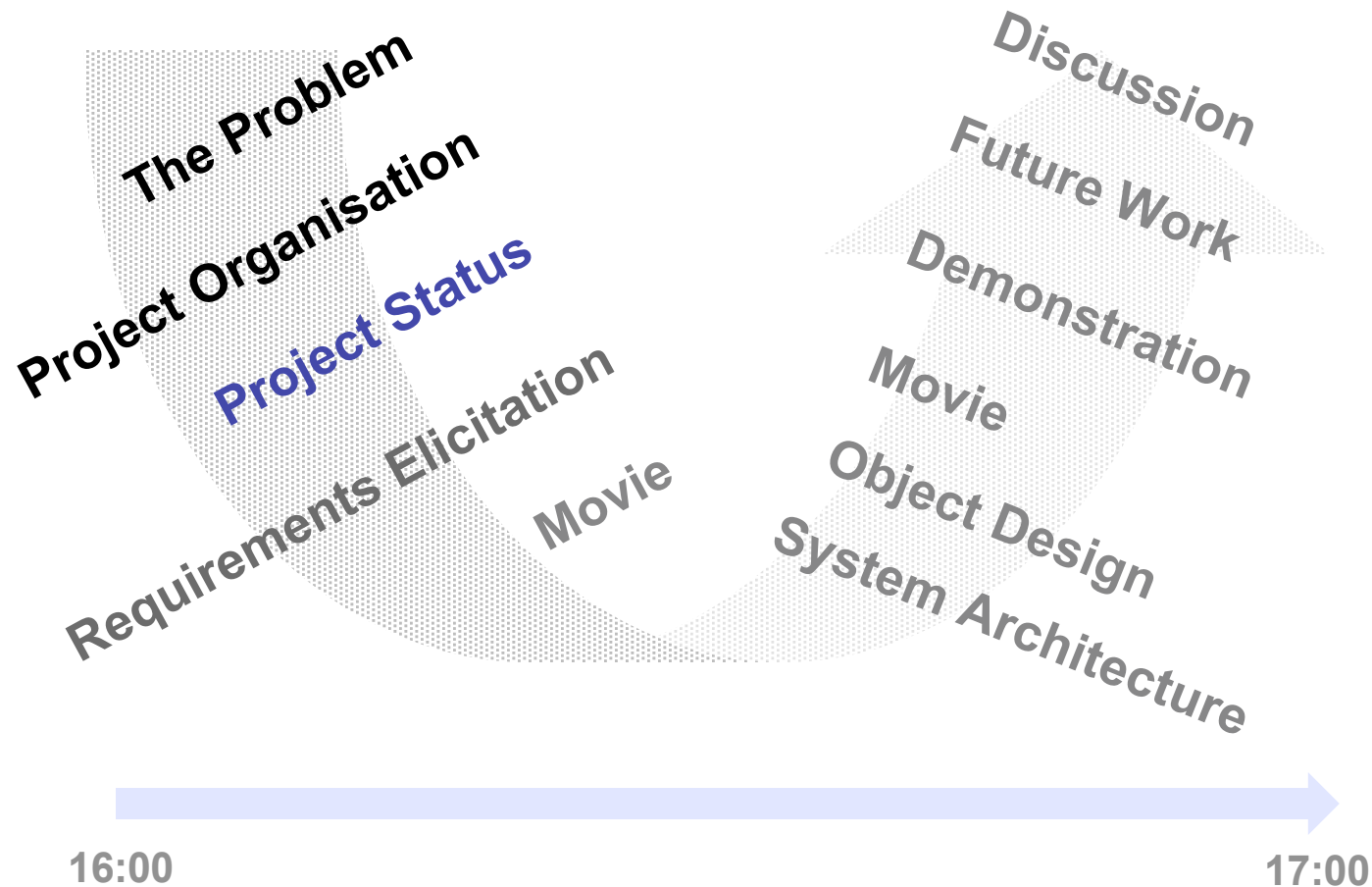


Teams Overview





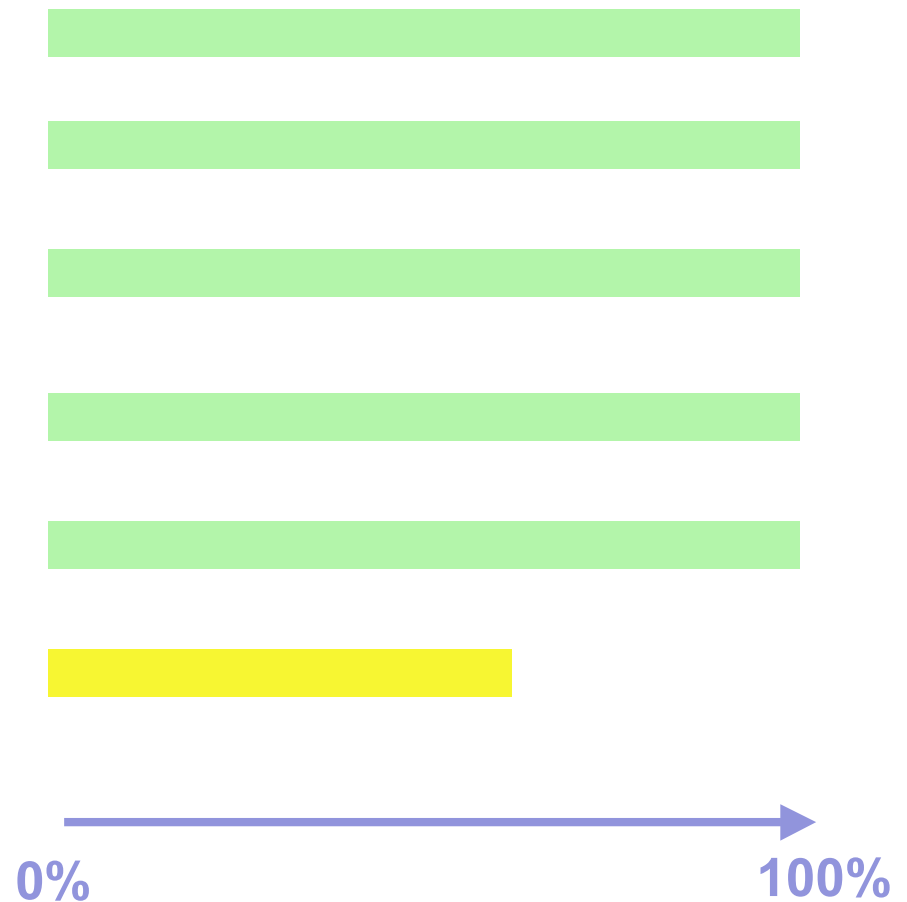
Content





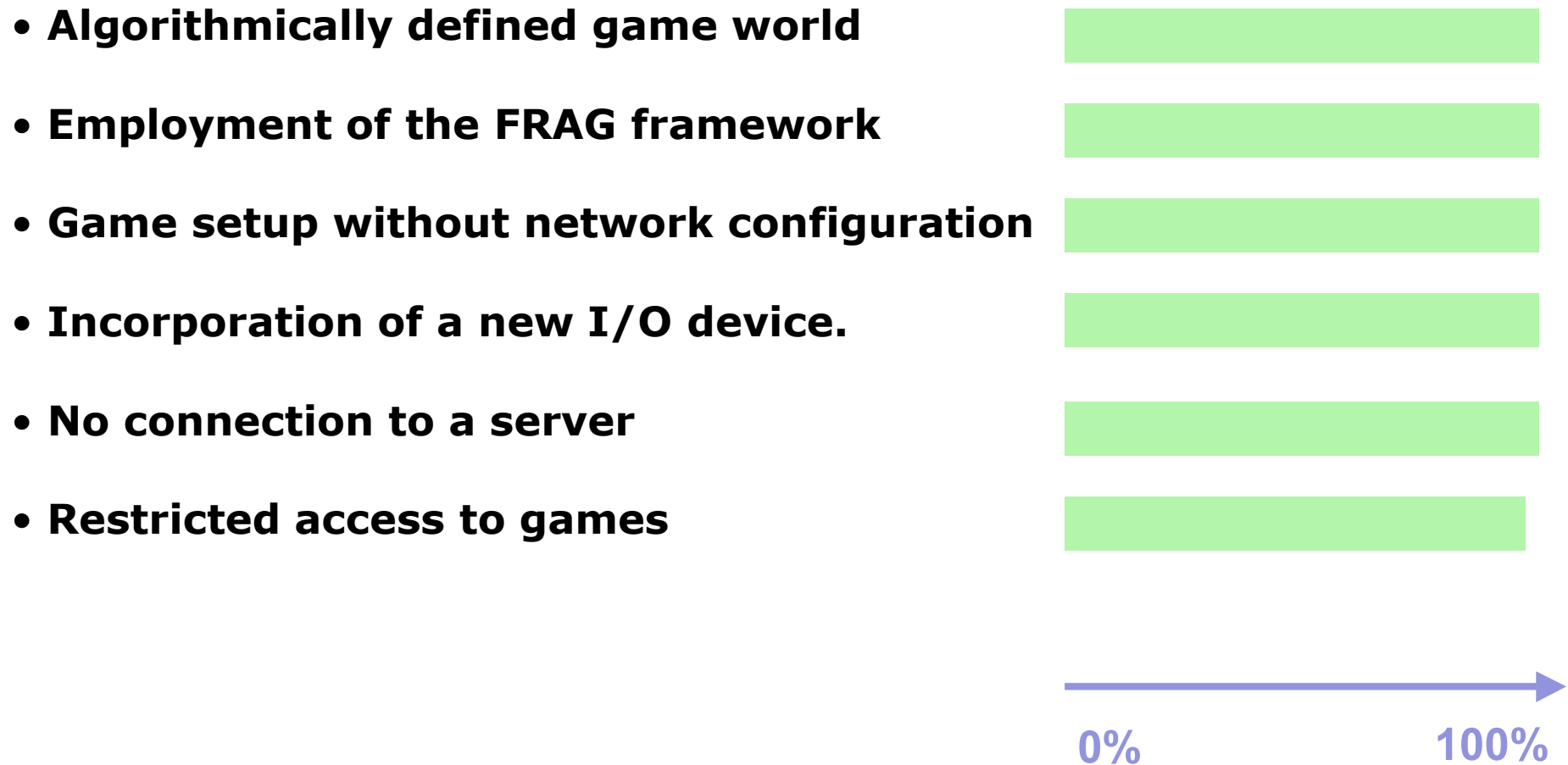
Status of Functional Requirements

- **Start sword**
- **Suspend**
- **Hand over Properties**
- **Hand over Game**
- **Broadcast game data**
- **Disconnect/Reconnect**





Status of Non-Functional Requirements





Status of Deliverables

- **Based on FRAG**
- **Functional prototype demonstration**
- **Design and implementation complete**
- **Complete project archive on DVD.**
- **Open source project portal**



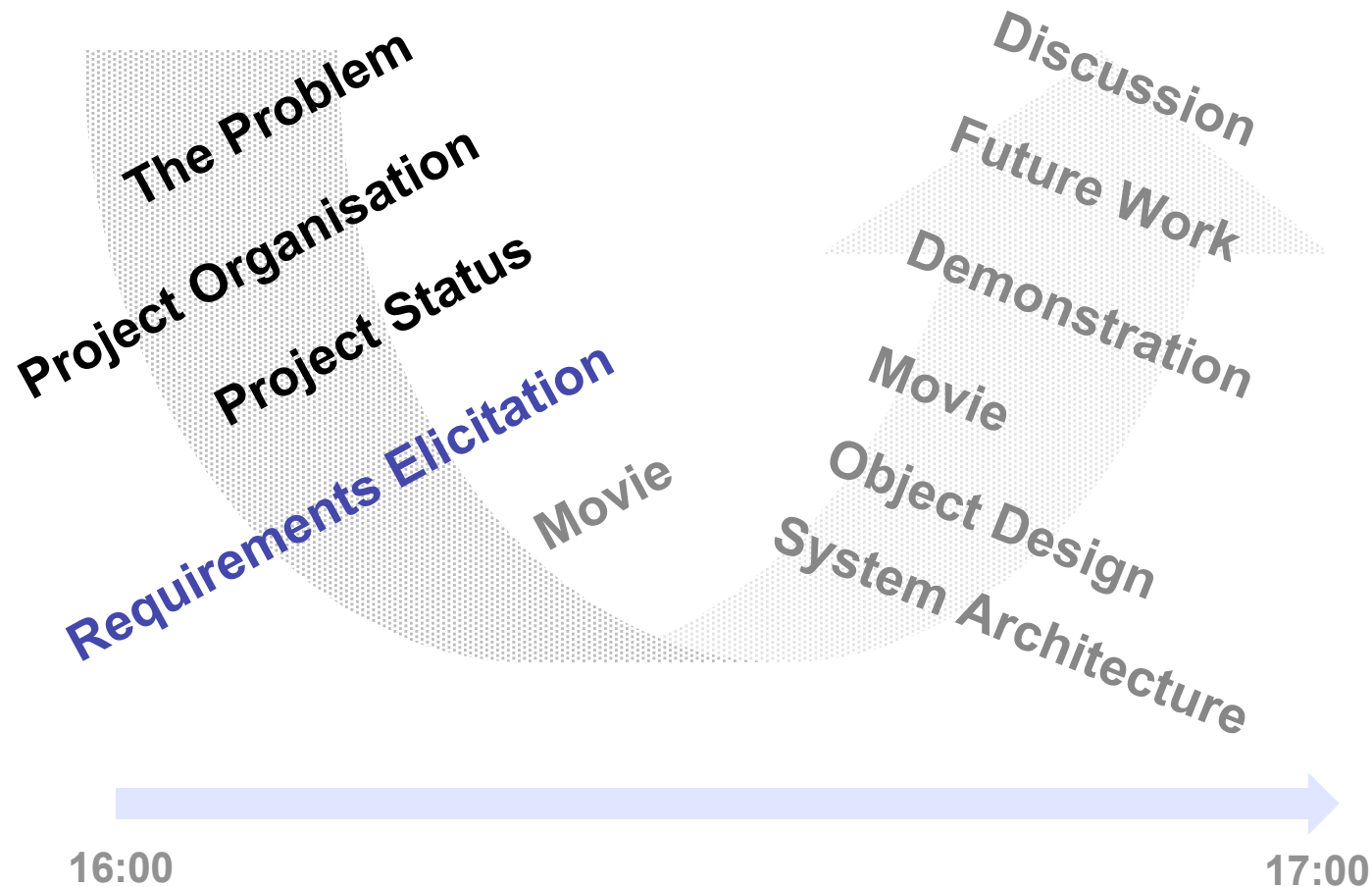
Client Acceptance Test


Banko Banov





Content



The word "ARENA" is written in a large, bold, serif font with a 3D metallic effect. It is set against a background of a sandy arena floor with concentric circular tracks.

ARENA Prototypical Gameplay

- Characters: torturer, witch, monk, troll, gnome...

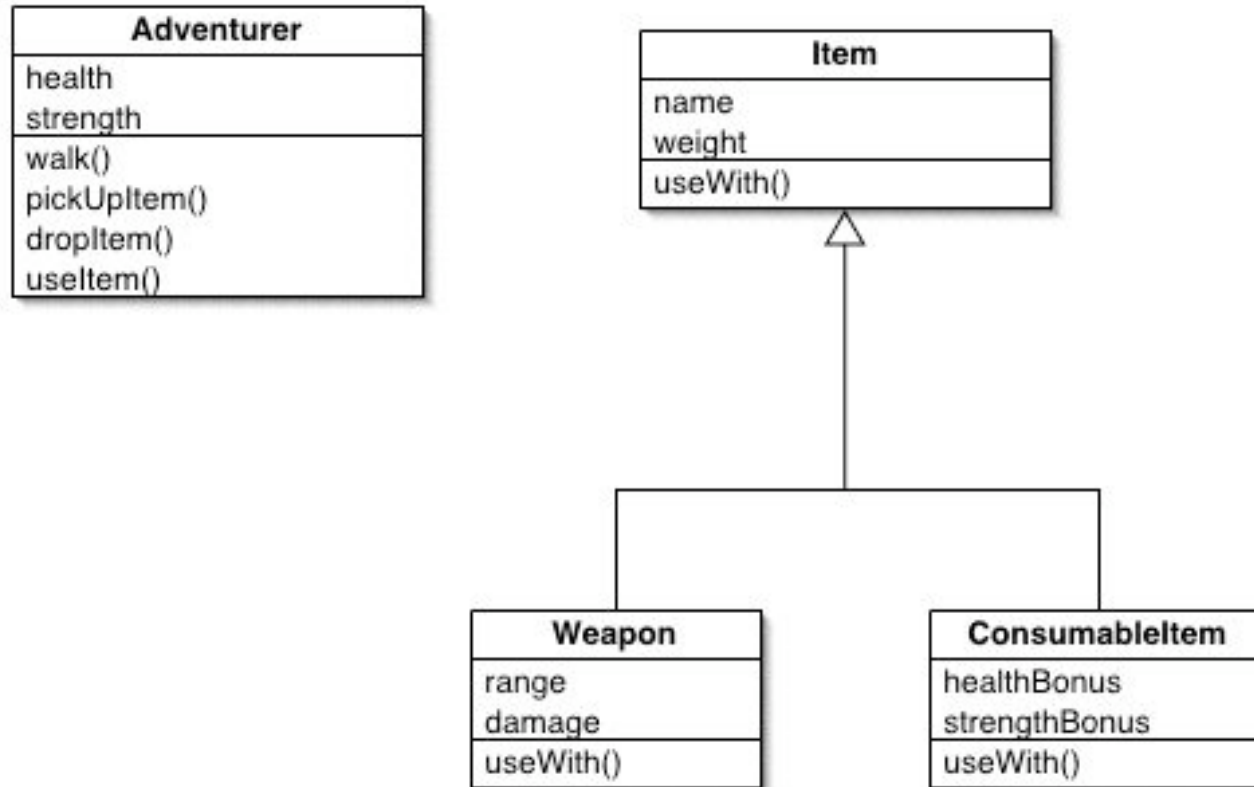
Abstracted to Superclass *Adventurer*

- Objects: sword, ring, apple, battle axe ...

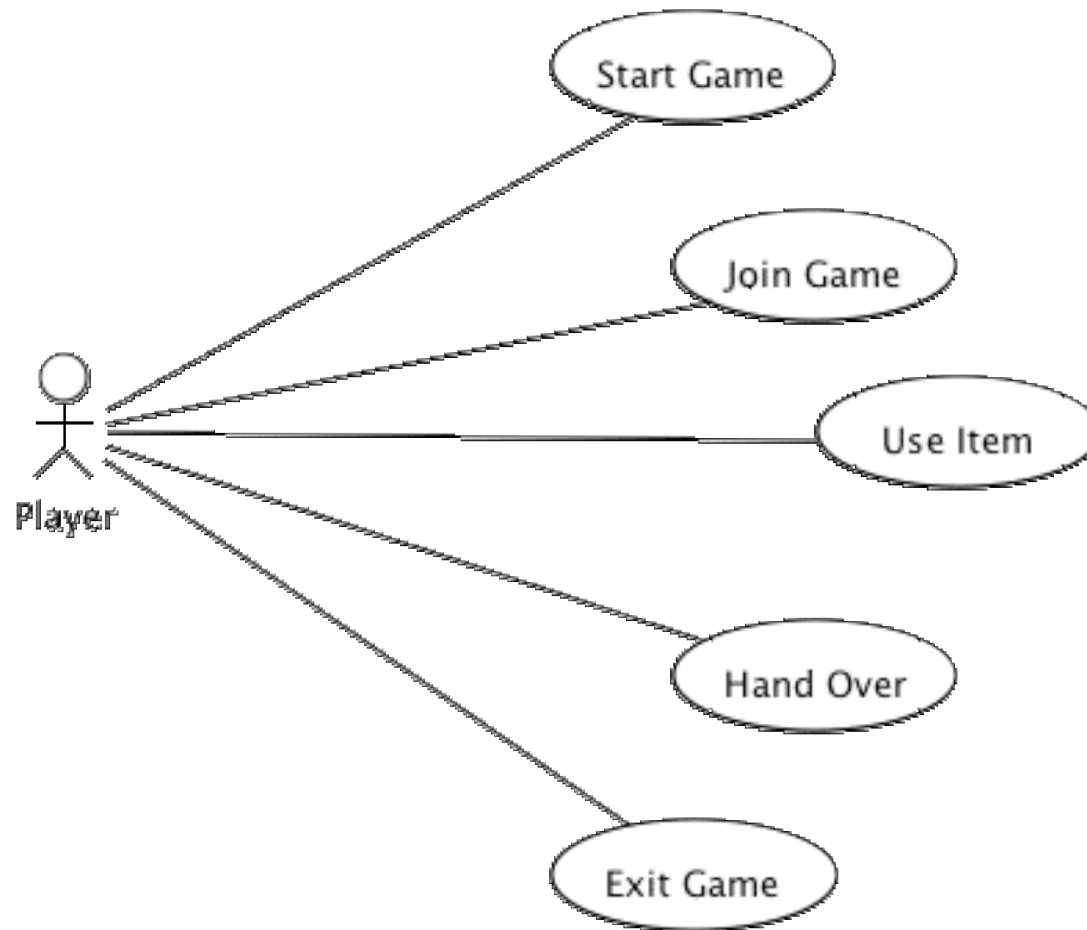
Abstracted to Superclass *Item*



Analysis model

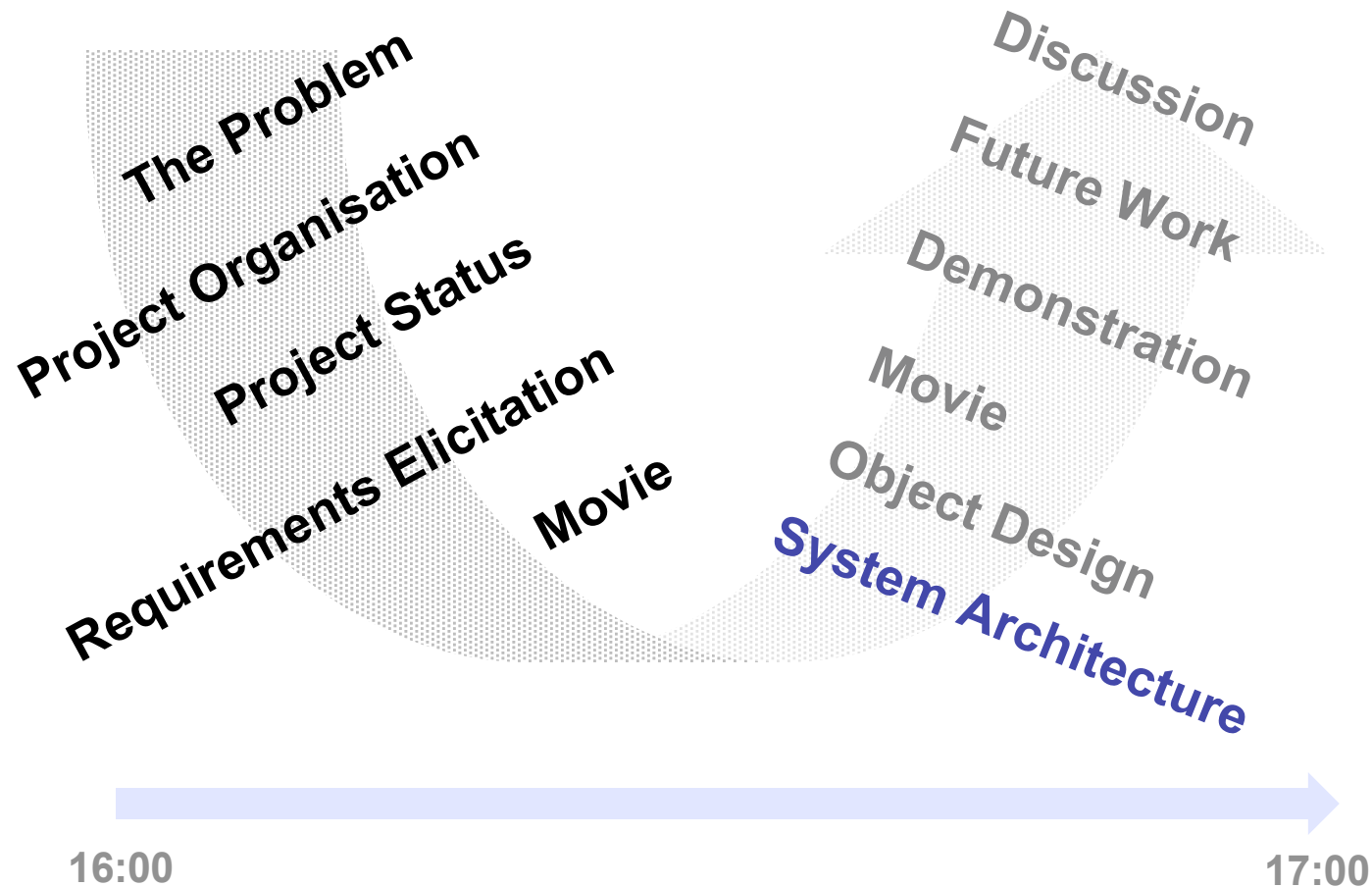


Use Cases





Content



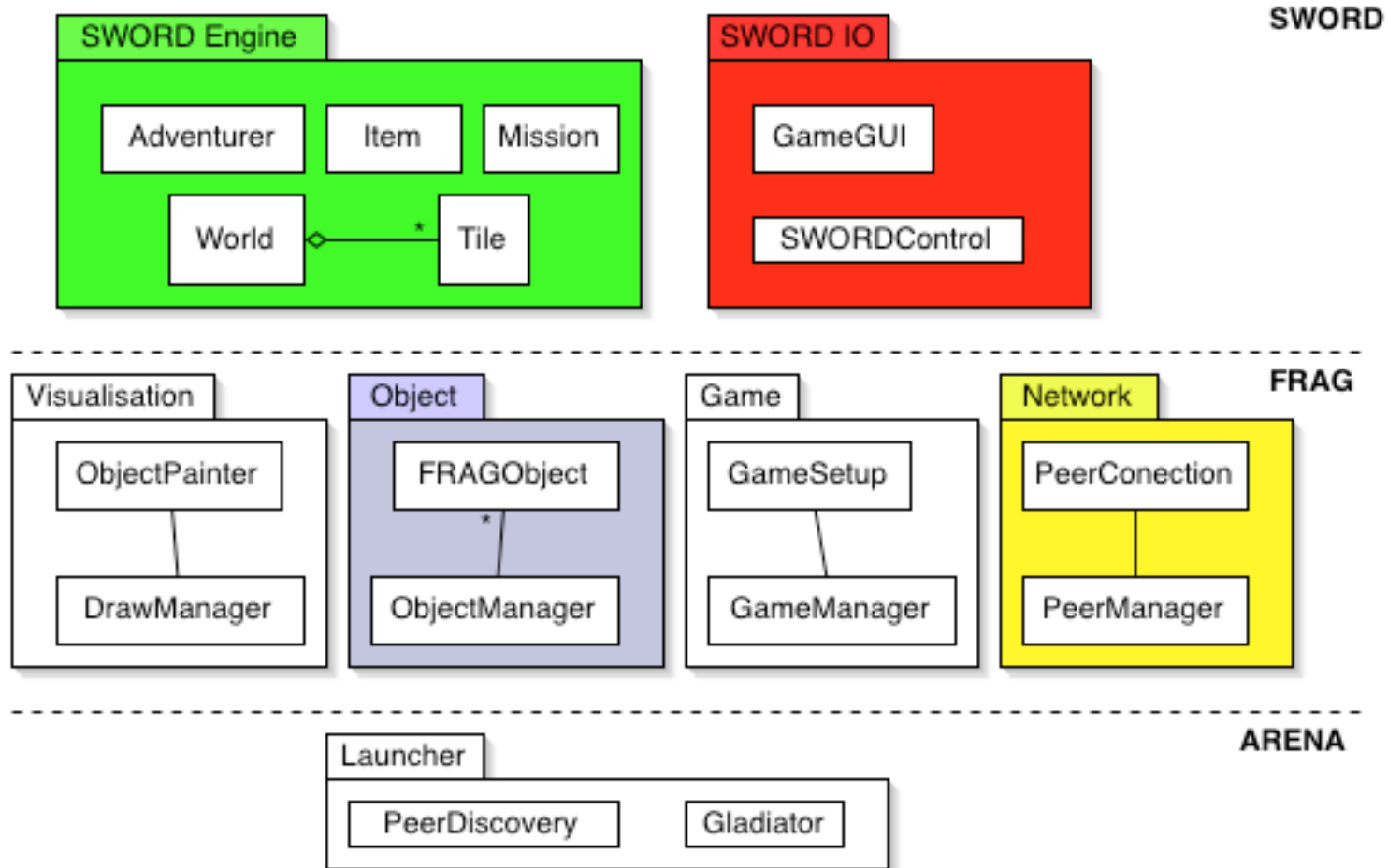


Architecture

- Subsystem decomposition
- Hardware/Software mapping
- Access control /Boundary conditions
- Persistent data management

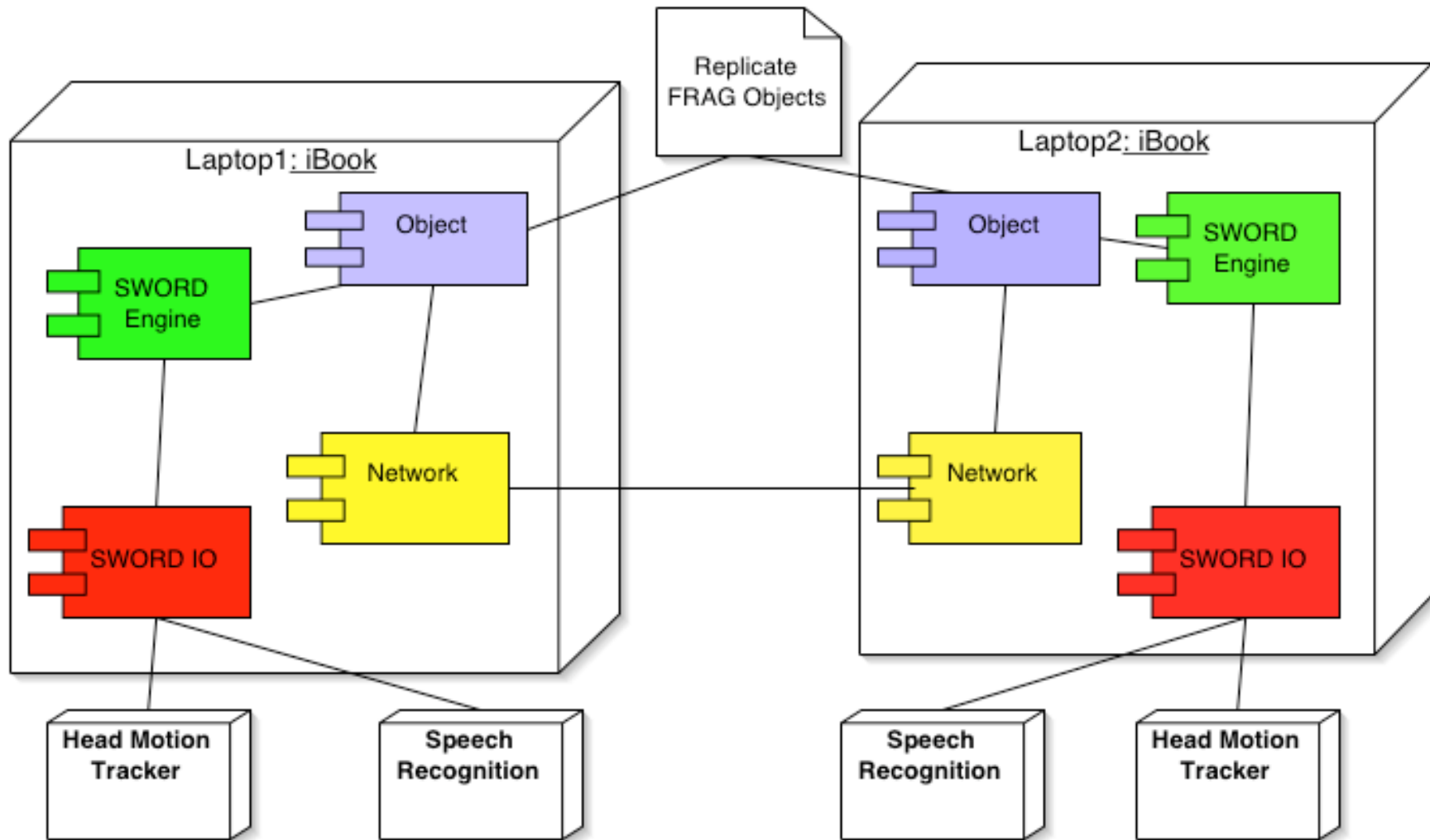


Subsystems





Hardware/Software mapping



ARENA Peer-to-Peer Gaming



Susan
“starts game,”
“downloads character
to the client” again
players auto-
matically and
“saves her character”

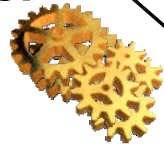


Synchronizing
game data ...



Steve received
Susan's game
with a
Snowman Frosty
playing with Steve
game Frosty again
alone

SWORD



Frosty received Susan's game
and
“loads a character”



Access control/ Boundary conditions

- Access control
 - optional password
- Boundary conditions
 - start-up:
no manual network configuration, mission, Adventurer
 - shut down:
saving game and/or character

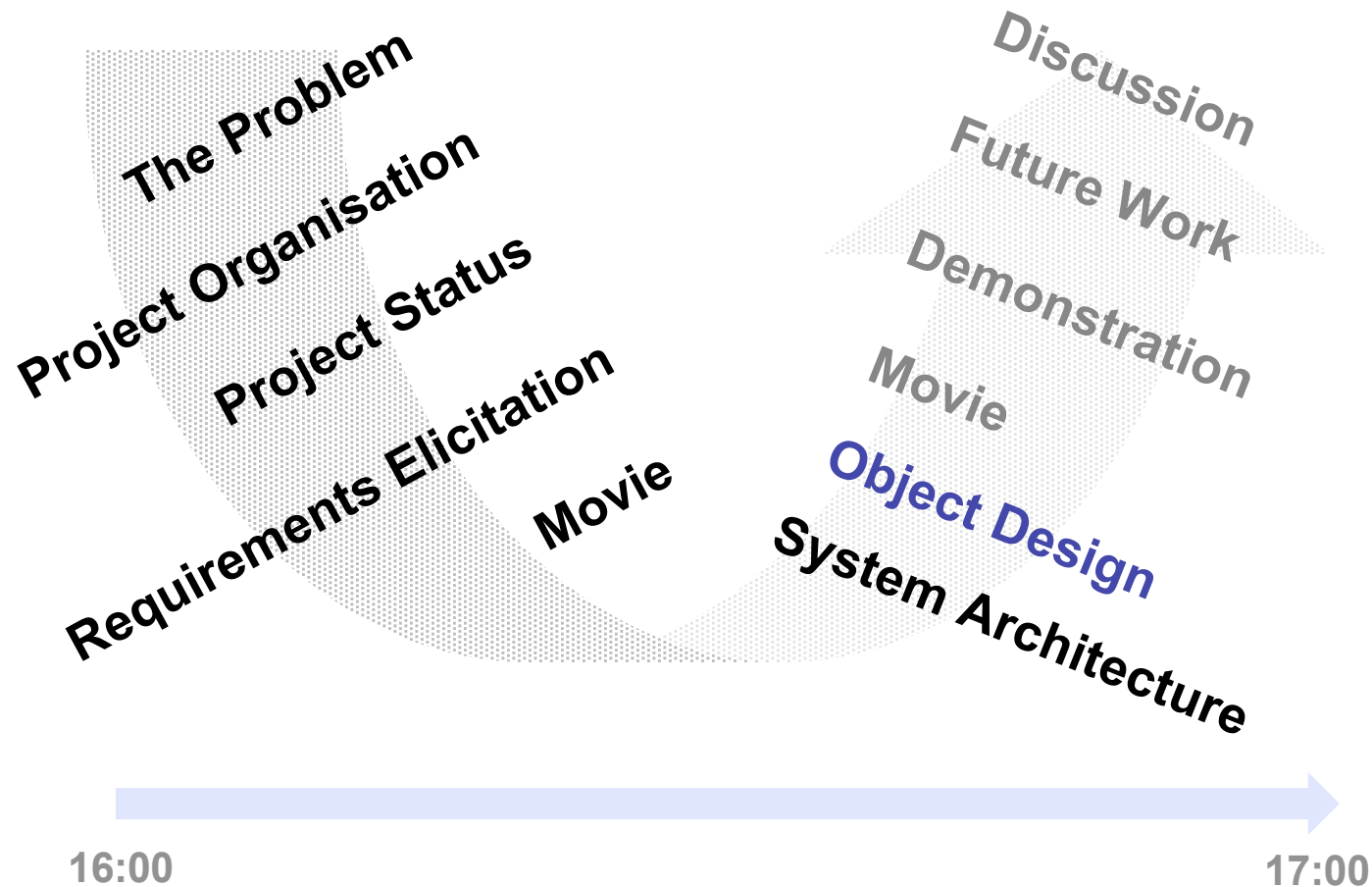


Persistent data management

- Mission:
 - XML file
- Gameworld:
 - seed for the generation algorithm - integer
 - Items - serialized objects
- Character:
 - serialized object

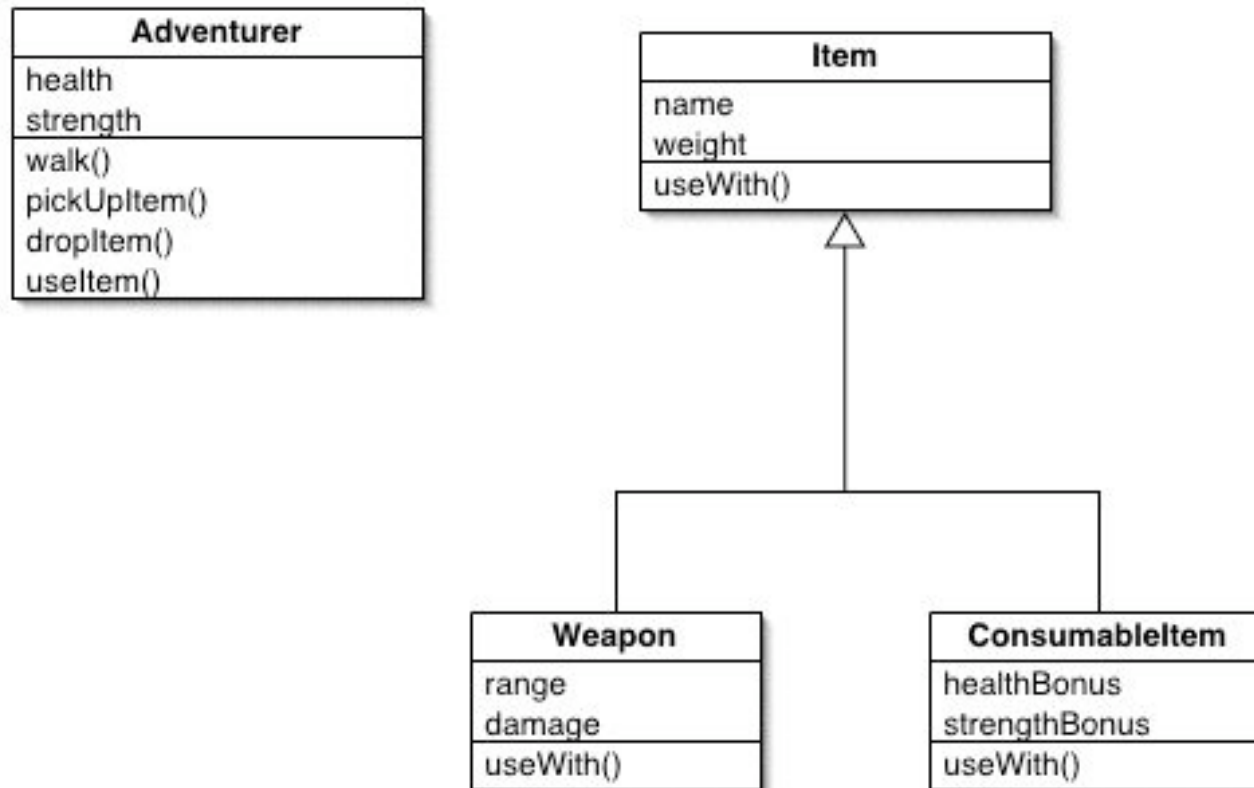


Content

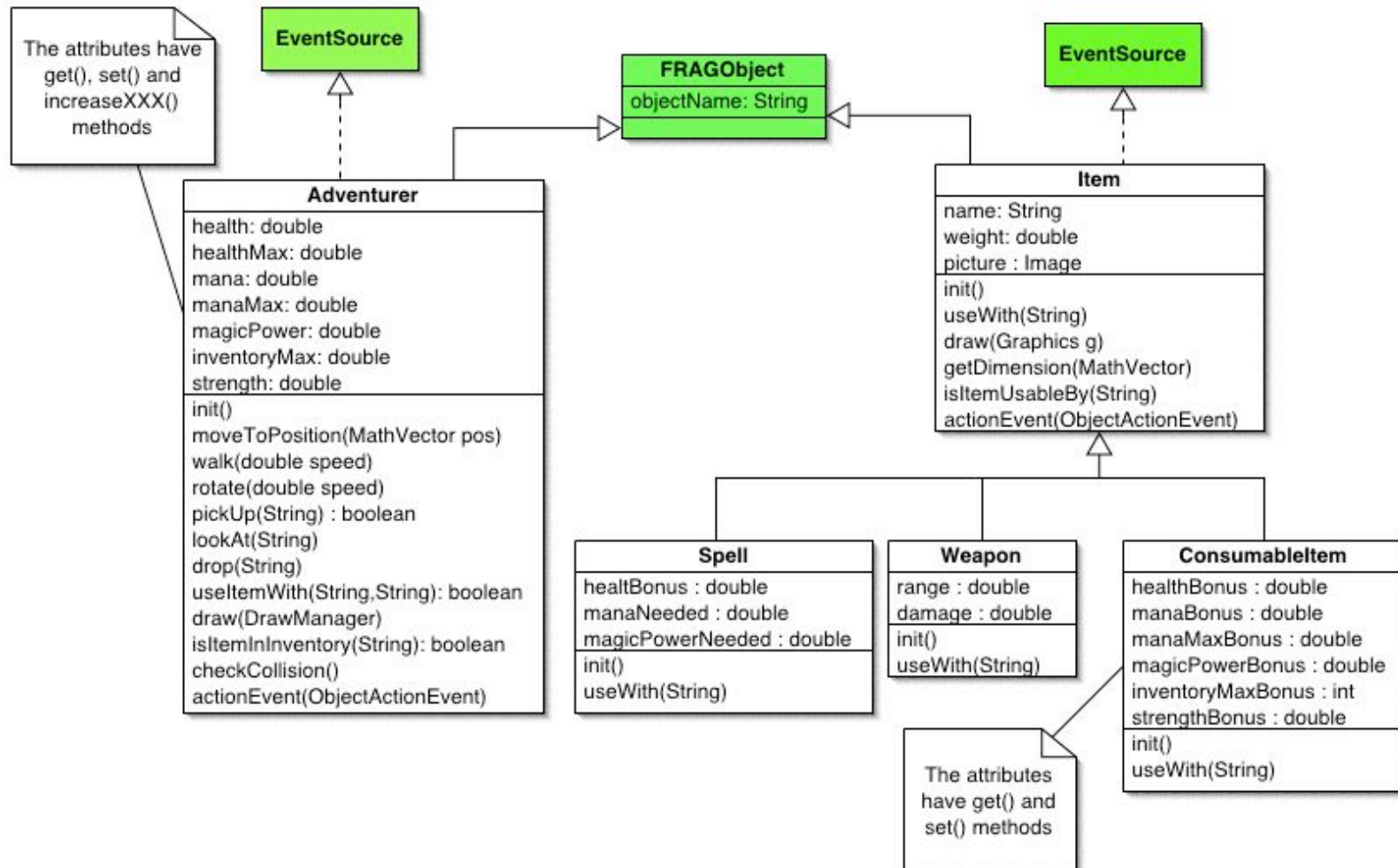




Analysis model

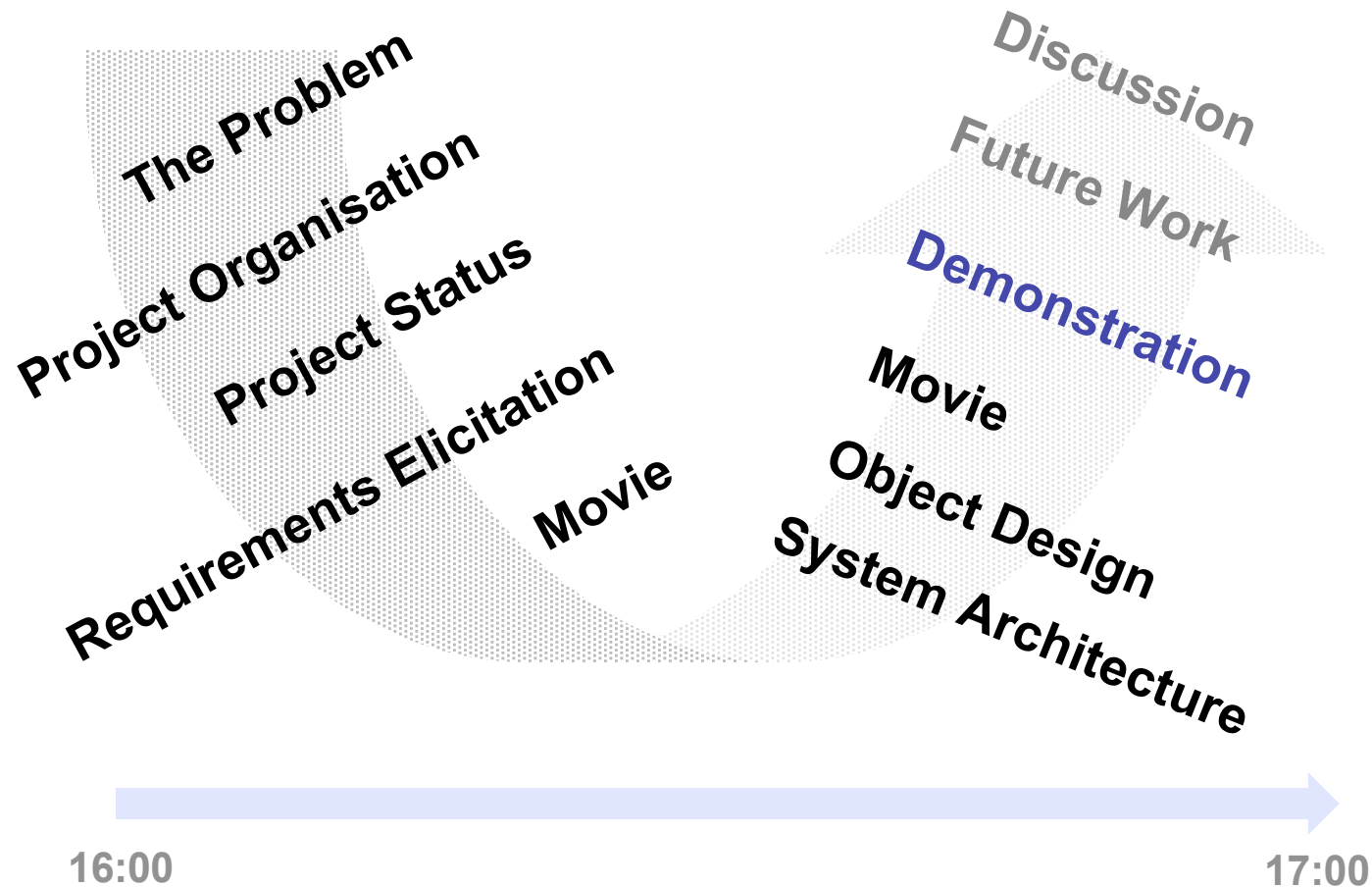


ARENA Object Design Model





Content



Client Acceptance Test

Petr Ossipov



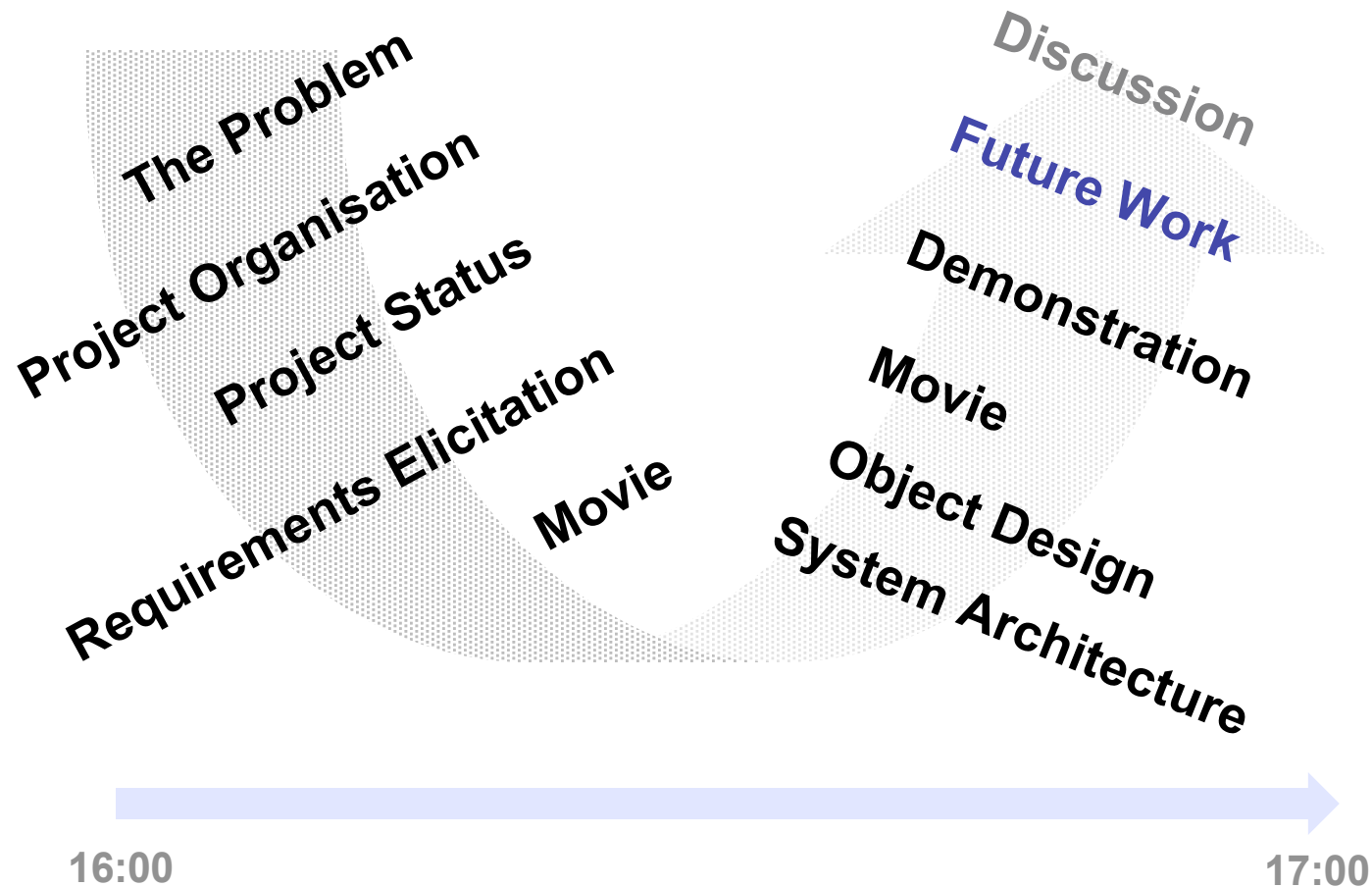


Demonstration

- Start game
- Join game
- Pick up and drop Items
- Use Items to affect other players
- Save character
- Voice commands: forward, backward, exit



Content





Future Directions

- Computer controlled Adventurers
- Algorithmic generated mission
- New gameplay perspective
- Hand-held devices
- Distributed computing applications

Thanks for your attention!

Salve Gladiators!

