



# Requirements Engineering and the Creative Process in the Video Game Industry

David Callele, Eric Neufeld, Kevin Schneider  
Department of Computer Science

University of Saskatchewan, Canada

# Introduction + Motivation



NON-FUNCTIONAL  
REQUIREMENTS FOR  
GAMES?



HARD TO DEFINE  
"FUN" OR "IMMERSIVE"



COMMUNICATION IN  
TEAMS



DIFFERENT  
PERCEPTION OF  
LIMITATIONS

# Background



Translating expectations to requirements is hard

Different perceptions  
Hard to materialize



Traditionally creating requirements using goals:

Goal: Software should accomplish...  
Goal: Game should be fun



Team members with different specializations

**Diverse backgrounds**  
Need to construct a universal language

# Literature study



EMOTIONAL FACTORS



LANGUAGE AND  
ONTOLOGY



ELICITATION, FEEDBACK  
AND EMERGENCE

# Emotional factors

- *"Fun is not a property of software, but a relationship between the software and the users' goals at that moment"*
- *"Providing enjoyment is now a defining requirement of an important class of software"*
- Defining "fun"
  - Usability
  - Immersion
  - Motivation
- Successful games trigger a "flow" in user

# Language and Ontology

- Requirement engineering: “... translation from informal observations of the real world to mathematical specification languages.”
- Only partially true for game developers; no interest in mathematical representation
- Solution: common universal language
  - May be derived from statistical natural language processing



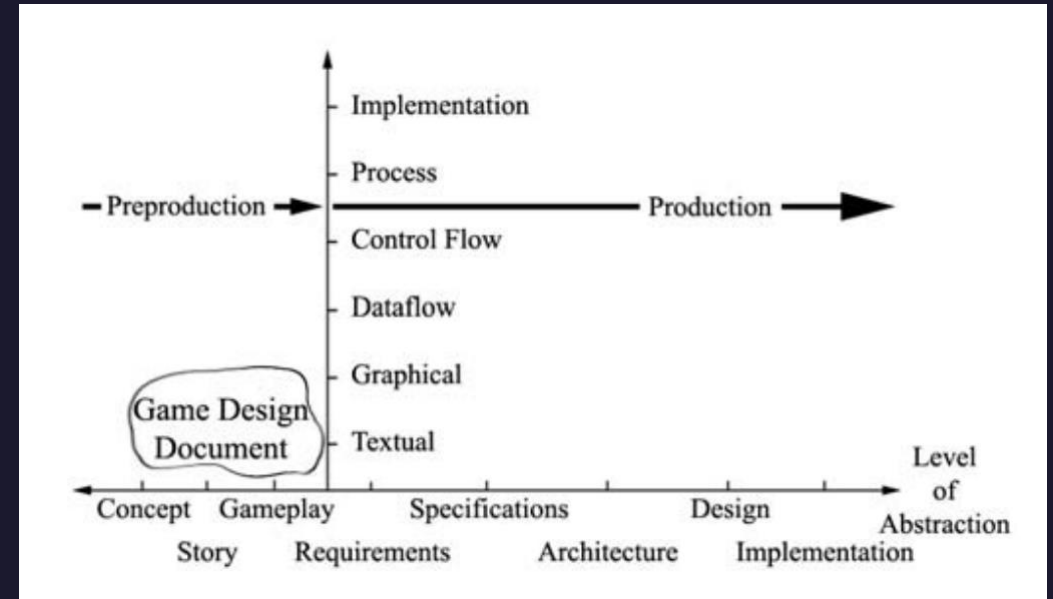
# Elicitation, Feedback and Emergence

- *“Feedback and feedforward go on all the time, at least in successful large projects”*
- Requirements may *emerge* under development
- Important with continuous communication between preproduction and production
- Very useful when production gives feedback on early prototypes from preproduction



# Video Game Development

- Differs from generic software
- Methodology: game design and production
  - Preproduction: Define wants and needs
- Game Design Document (GDD)





# Game Design Document

- Creative vision
- May be used as source for production:
  - Malformed; need to be structured
  - "Ad hoc"; relies on human memory
- Weaknesses
  - Need two separate sets
  - Style differs

# Game Design



# Analysis of games

Based on feedback reports; *Postmortems*

*"... Explain what 5 goals, features or aspects of the project went off without a hitch or better than planned.... Explain what 5 goals, features or aspects of the project were problematic or failed completely"*





# Analysis of games - Genres

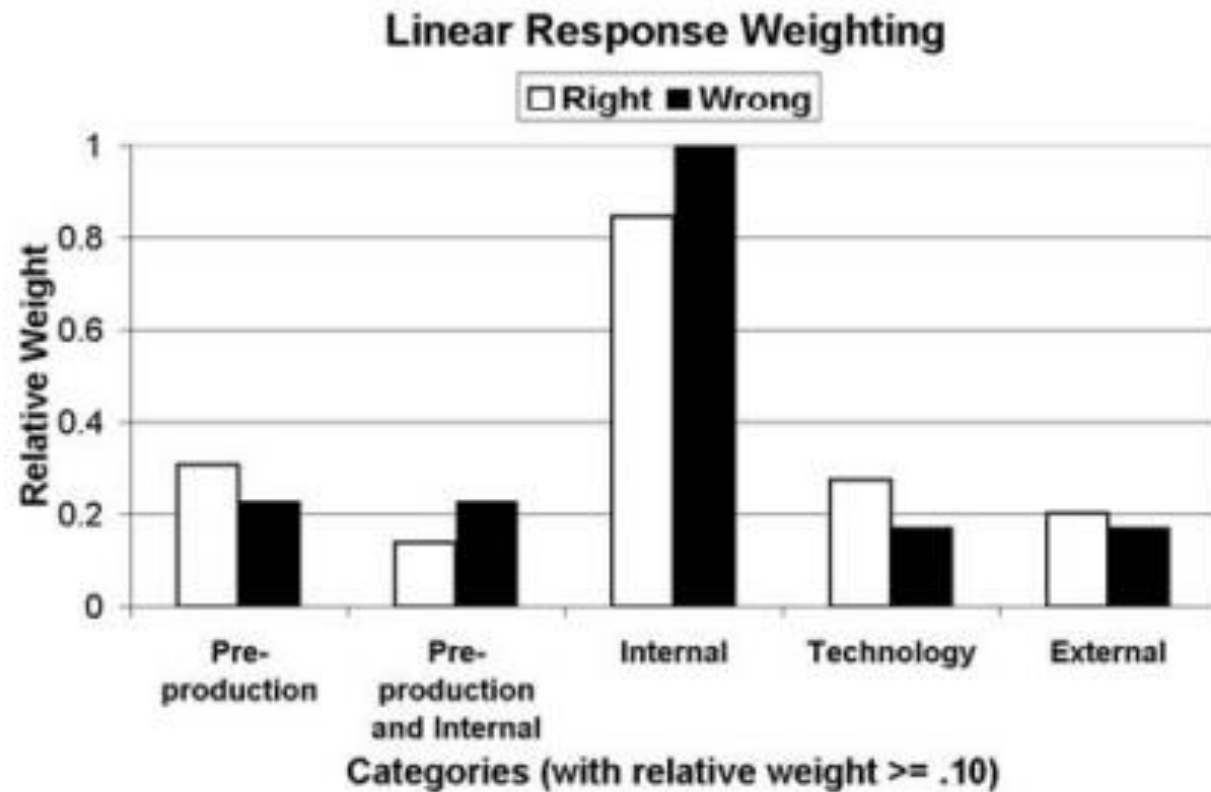
- Preproduction
- Internal
- External
- Technology
- Schedule

(Pairings allowed as well)



# Analysis of games - Results

- Most feedback tagged with internal
- Most problems linked to project management
  - Preproduction should get feedback on early prototypes as soon as possible
- Balance in categorical results; maximum deviation of 7.7%
- Production process had often positive and negative experiences




# Example analysis

Game Designer proposes story:

*"After her father, Bernard, died, Crystal did not know which way to turn – paralyzed by her loss until the fateful day when his Will was read."*



# Example analysis - Document transformation



1	Story	After her father, Bernard, died, Crystal did not know which way to turn – paralyzed by her loss until the fateful day when his Will was read.
2	Gameplay	The Player must visit Anna the Lawyer to receive a copy of Bernard's Last Will and Testament, thereby obtaining the information necessary to progress to the next goal.
3	Requirements	The Player must be represented by an avatar. Female Non Player Character required: Anna the Lawyer Inventory Item: Last Will and Testament (LWT) Player can not progress beyond Game State XYZ until LWT added to Inventory
4	Specifications	Could easily reach 50 pages



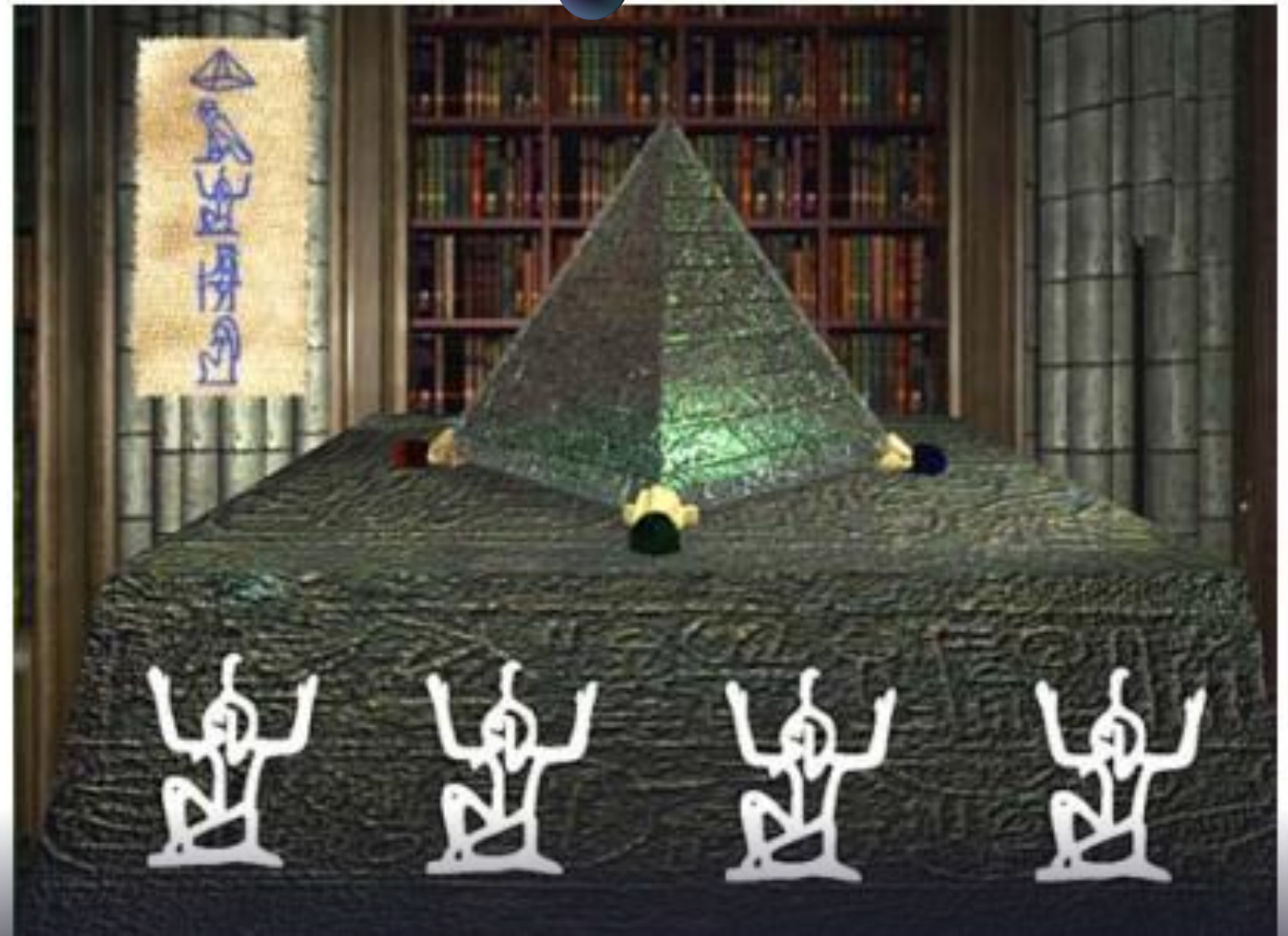
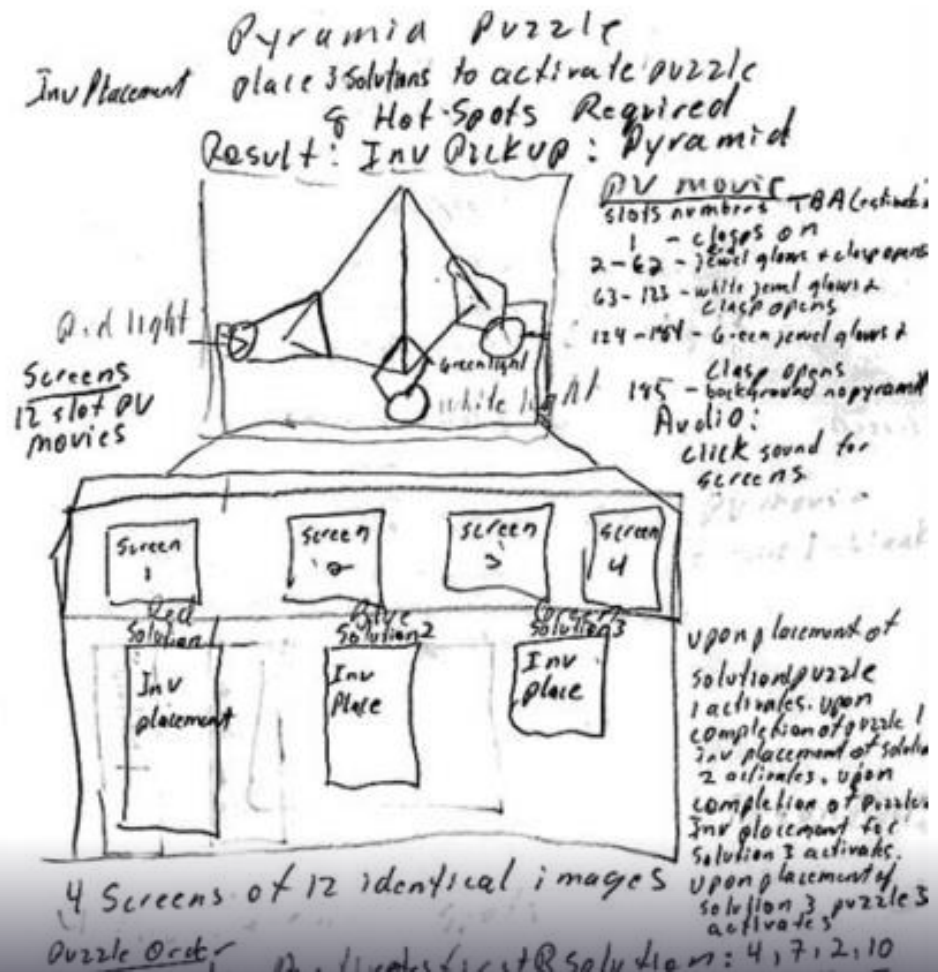
# Example analysis - Implication

- *"After her father, Bernard, died, Crystal did not know which way to turn – paralyzed by her loss until the fateful day when his Will was read."*
  - Level 1 (easy to derive): Implies existence of direct derivations
    - Examples: Crystal, Anna
  - Level 2 (captured by adept teams): Implies existence of game world and environment
    - Examples: Anna's office (with background sounds, visuals, other NPCs?)
  - Level 3 (captured by experienced teams): Implies existence of details and architecture
    - Example: World between Player and Anna (with visuals, paths, other tasks, NPCs?)



# Case - Pyramid Puzzle

From game *Apocalypse Spell*





# Case - Pyramid Puzzle

- Complex puzzle; player was given four clue scrolls as inventory items.
- Communication between preproduction and production; game engine not capable of displaying clue scrolls in puzzle interface!
- Suggestion: "Hang scrolls in puzzle", making scrolls visible.
- Problem: Low resolution!
- Compromise: One scroll is visible; player must switch clue scroll manually.

# Conclusion



Main challenge: Requirements and implications are not communicated clearly between game designers and production



Identifying and understanding implications is important for success



Project management has a huge say for outcome



Requirement engineering should be done continuously through dialog between preproduction and production