

Student: abdalhadi haidar

chapter 13: Stages and Methods of Development.

- **Stages of Development:**

- Concept/Contract: Pitching the game idea to a publisher based on the team, project plan, and idea.
- Preproduction: Small team verifies the idea's feasibility, creates prototypes, and proves risky elements.
- Production: Longest and costliest stage, executing the functional vision with interdisciplinary teams.
- QA/Polish: Focus on refining the game, fixing bugs, and polishing before launch.
- Ongoing Production: Post-launch updates and maintenance based on user feedback.

- **Real World example "FLOW":**

- Small team's challenges in developing controls, adapting to the evolving PS3, and transitioning from 2D to 3D.
- Emphasis on simplification, focusing on core actions while maintaining the game's essence.

- **QA Process and Ongoing Updates:**

- QA testing involves creating a test plan, tracking bugs, and resolving issues systematically.
- Triage meetings prioritize and resolve bugs. The goal is to reach "gold code" with minimal remaining issues.
- Ongoing production includes post-launch updates based on user feedback, maintaining game quality over time.

- **Key Points - Agile Development:**

1. **Agile Methodology Basics:**

- Agile development is adaptive and people-centric, focusing on short-term goals and close teamwork.
- Teams follow a flexible approach, addressing priority features in short iterations called "Sprints."
- Regular meetings, called Scrum meetings, involve progress updates, goal setting, and issue discussions.

2. **Critical Agile Techniques:**

Working in Sprints: Fixed-length iterations, typically weeks, producing playable builds for evaluation.

Scrum Meetings: Short daily meetings led by a Scrum Master, updating on accomplishments, setting goals, and addressing obstacles.

Prioritization: Features are prioritized based on input from players, clients, and stakeholders, facilitating ongoing adjustments.

3. Example of Agile Team:

- Small team (designer, programmer, artist, producer) working on a mobile puzzle game.
- Two-week Sprint focused on implementing the core mechanic.
- Regular Scrum meetings for progress updates and issue resolution.
- Simplistic example applicable to both small and large projects.

4. Scalability of Agile:

- Agile can be used in projects of varying sizes, from small games to large complex endeavors.
- Larger projects may involve multiple interdisciplinary teams, each working on specific aspects.
- Communication is quick and efficient in small teams, helping identify and resolve dependencies.

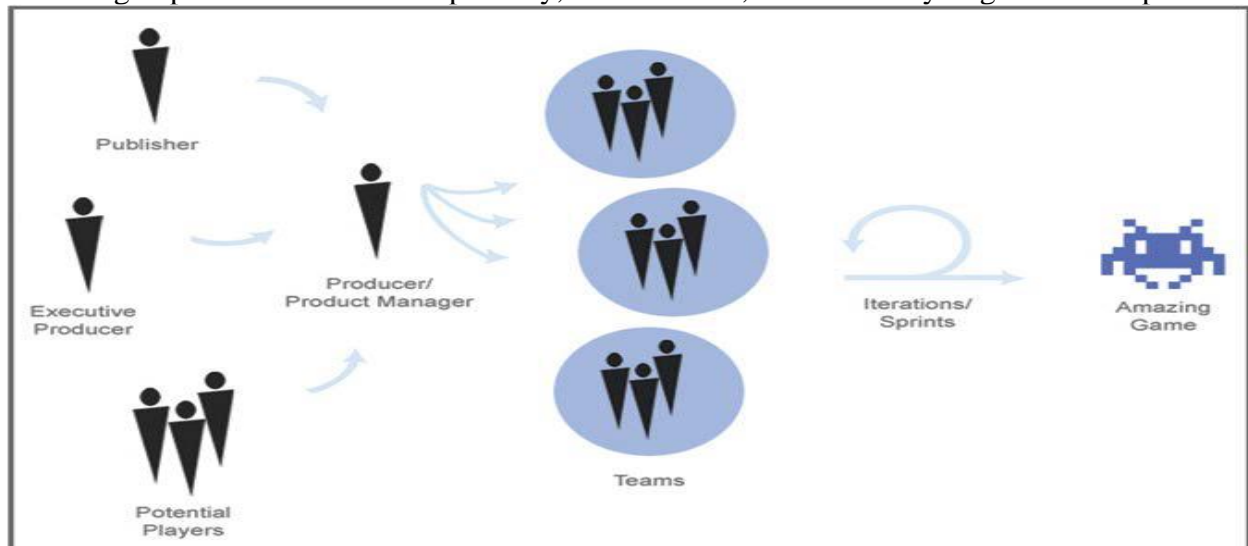
5. Implementation Tips:

- Agile methodology can be practiced by structuring project plans around Sprints.
- Holding daily Scrum meetings and practicing ongoing team-based planning are fundamental aspects of Agile.

6. Further Learning:

- More detailed management techniques in Agile and Scrum can be learned through courses and books.

These Agile practices enhance adaptability, collaboration, and efficiency in game development.



- **Conclusion:**

- 1. Technical Complexity of Games:**

- Games are among the most technically advanced software globally, presenting continuous challenges as technology evolves and player expectations rise.

- 2. Crucial Role of Process:**

- Implementing a robust process is essential for game development success.
 - Understanding and anticipating each stage of development is key to navigating the complexities of creating games.

- 3. Contributing to Success:**

- While designers may not be the primary creators or managers of project plans, understanding the development process empowers designers to contribute effectively.
 - Informed designers enhance team collaboration and become valuable team members.

Understanding the intricacies of game development processes positions designers to play a more significant role in contributing to their team's success, fostering collaboration, and meeting the challenges posed by advancing technologies and heightened player expectations.