



February 18, 2013

Agenda:

- Article of the Week – *Conducting a Project Postmortem*
- Showcase
 - Promotion
 - Purpose
- Year on Review/ Semester Feedback
- Collect Committed version of Games
- Final word
- Game Exhibitions

Article of the Week

Conducting a Project Postmortem *by Steve Pavlina*

http://www.gamedev.net/page/resources/_/business/business-and-law/conducting-a-project-postmortem-r977

Given all the responsibilities that game developers face, it isn't surprising that they seldom take the time to conduct project postmortems. A postmortem is a procedure whereby you summarize a project's history and analyzes its positive and negative aspects. The goal of a postmortem is to draw meaningful conclusions to help you learn from your past successes and failures. Despite its grim-sounding name, a postmortem can be an extremely productive method of improving your development practices.

The best time to conduct a postmortem is about two weeks after a product is released (or for certain products, after the project is cancelled). This allows you to regain your objectivity without forgetting the details. Your memories will still be fresh, and you'll have a good perspective to see the project as a whole rather than focusing too strongly on the most recent work.

Here are ten suggestions for conducting a good project postmortem:

1. Involve all contributors. If others were involved in your project, arrange a project review meeting. If you were the sole contributor, set aside time to reflect on your experience. Document the good, the bad, and the ugly aspects of the project. Note all grievances, and solicit ideas for future improvement. I recommend using a project review questionnaire with three parts. First, use a one to ten rating system to quantify subjective feedback. Secondly, include targeted questions for specific areas that might need improvement. And finally, use a free form comments section for open-ended feedback.

2. Document the postmortem in writing. Document the postmortem details in writing. While it is perfectly acceptable to create a simple text document for small projects, I recommend using HTML for



February 18, 2013

large projects. The document can then be published on an intranet, so everyone in the company has easy access to it. This way you can provide links to additional resources such as design documents, schedules, and archived files. It also makes it easy to create a collection of postmortems over time, with links between related sections of different documents. For a multi-release product, include additional data on each upgrade. A series of postmortems can ultimately evolve into a valuable record of your hard-earned wisdom.

3. Begin with a project overview. Begin the postmortem document with a brief overview of your project, including a design overview, estimated budget, and project start and finish dates. Describe the product's purpose, intended customer, and other general information.

4. Include project details. Document the quantifiable details of your project. Include your schedule and budget estimates as well as the true outcomes. List the number of lines of source code in the release as well as the lines of reused code. Document the size of the media and system requirements for the end user. List any known bugs or compatibility problems. Note which development tools were used and their version numbers, including any mid-project upgrades. List everyone who worked on the project, and document their contributions.

5. Document what went right. Document what went right with your project. Did the final product turn out better than expected? Did you experience an occasional stroke of luck? List at least ten things that turned out well for your project. For my last project, I had the most luck with quality assurance. I paid tremendous attention to code quality, and I was able to keep the program virtually bug-free throughout its entire development. Consequently, I was able to spend very little time on debugging, allowing me extra time to improve the quality of the product.

6. Document what went wrong. What difficulties did you encounter? What assumptions proved incorrect? Compile a detailed list of your project's shortcomings. List at least ten things that turned out worse than expected. Document unpleasant mid-project surprises as well. My last project's greatest challenge was flushing out the design. I was developing a unique new type of game, and I experienced extreme difficulty trying to get all of the gameplay elements to coalesce into a unified whole that would be fun to play. I spent about two-thirds of the entire project on very painstaking level design, far more than expected.

7. Assess your risk management. Document all the risks you took during the project, and note how effectively you managed them. Did you explore experimental ideas, use new technology or development tools, or develop for a new platform? Did your schedule slip from poor estimation practices? Did you take big risks, or did you play it relatively safe? Assess your risk management experience, paying particular attention to how you would change your approach for the next project.

8. Assess mid-project changes. What unanticipated changes occurred throughout the project? How did you respond to them? Did you incorporate some great mid-project additions, or was excessive feature creep a problem? I was able to manage change effectively on my last project by designing flexibility into the game engine. But once the level design reached a certain point, most new changes had to be locked out in order to prevent corrupting previous work.

9. Draw meaningful conclusions. What conclusions can you draw from your project history that will help you improve in the future? What should you start doing, keep doing, or stop doing? Describe new practices you should try for your next project. This is perhaps the most important part of the postmortem, so be prepared to spend more time here than in any other section.

10. Take action. It would be a great waste of time to create a beautiful project postmortem, archive it, and forget about it. The final step is to develop an action plan that can be applied to your next



February 18, 2013

project. Review your project history, reflect on the lessons you've learned, and specify new guidelines to follow in the future. I recommend creating a checklist for your next project. If you already have a checklist, then update it with new refinements.

Whether you are a lone-wolf programmer or part of a larger team, postmortems will help you maximize the benefits gained from your hard-earned experience. On large projects you can even conduct a mini-postmortem after each milestone, allowing you to profit from what you've learned as soon as possible. One final benefit of conducting a postmortem is that you gain a very gratifying sense of closure because you actually document all the personal growth you've experienced as a result of completing the project.

If you wish to view a sample postmortem from my company's last project, visit www.dexterity.com/postmortem. I included the complete text of the postmortem and well as a sample project questionnaire you can use in conducting your own postmortems.