



United Business Media

POSTMORTEM TWISTED PIXEL'S THE MAW

DECEMBER 2008

# game developer

THE LEADING GAME INDUSTRY MAGAZINE

» INTERVIEW

YUJI NAKA TAPS  
HIS CREATIVITY

» AURAL FIXATION

COMPRESSING SOUND  
FOR THE IPHONE

» GAMESCAPE

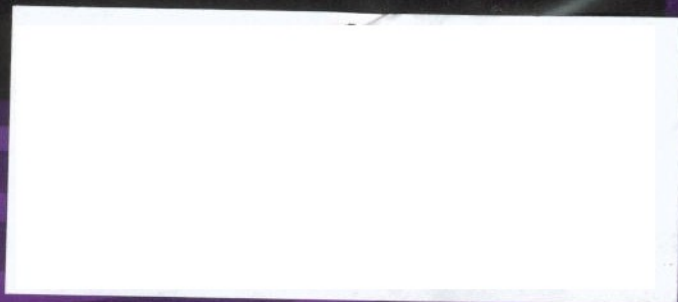
A HISTORY OF BOSTON  
GAME DEVELOPMENT



WHAT WENT WRONG

# RETURN

OF  
THE POSTMORTEMS!



>> brandon sheffield

# WHAT WENT WRONG?

## LEARNING FROM PAST POSTMORTEMS



POSTMORTEMS HAVE BEEN A STAPLE OF *GAME DEVELOPER* MAGAZINE FOR WELL OVER A DECADE.

With so many authors revealing what went right, and of course the car-wreck style appeal of what went wrong with the development process, patterns start to emerge. It becomes clear that a lot of the same mistakes are being made over and over again, and that some companies are even repeating their own mistakes.

With that in mind, we decided to round up every “what went wrong” entry from the last three years, and compiled the most frequently made mistakes (usually over five times each) into this cautionary feature. As the saying goes, those who do not learn from the past are doomed to repeat it. So here you have the top 10 “wrongs,” in no particular order, over the lifecycle of the current generation so far. No one is safe!

---

BRANDON SHEFFIELD is editor-in-chief of *game developer magazine*, and has worked on a few games himself. He still hopes to strike it rich through writing. What went wrong? Email him at [bsheffield@gdmag.com](mailto:bsheffield@gdmag.com).



# WHAT WENT WRONG?



1

**CONTENT ADDED TOO LATE.** Getting story and features right is difficult at the best of times, but when that content comes in just under the wire, not only does that content suffer, every element of the game that relies on that content suffers.

**TITAN QUEST (Iron Lore, Jeff Goodwill)** "We struggled getting 25 detailed design documents to the technical department on time. We were still working on first cut design documentation more than one year after the start of the project, so the technical team had to fly blind in the beginning and some systems had to be redone.

"There was a fairly constant issue of design documents not being detailed enough for technical implementation."

After this trouble, the company devised an approval process, which would have the producer, lead designer, technical director, and the publisher's creative manager sign off on each document before finalizing it. While this helped, it did slow down the company's process a bit, and it's worth noting that for all Iron Lore's good points, the studio is now closed. It stands to reason that a "what went wrong" from a shuttered studio would warrant extra scrutiny.

And to quote Riley Cooper, who had a similar problem with **TOMB RAIDER: LEGEND**, "This clearly serves to remind us that if you don't want any features in your game to under-perform, you need to do them 100 percent or not at all."

**BIO SHOCK (2K Boston, Alyssa Finley)** "We had many drafts of the story over the course of development, but the final draft turned out to be an almost complete rewrite.

"Competing demands for time and resources meant that, unfortunately, some of the important narrative details of the game weren't created until the final rewrite, and therefore required quite a bit of work to retrofit into an existing game."

In an RPG-like game, story is the guiding star of the project. While ideally the systems would be fully integrated into the story to the degree that each can affect the other, at the very least the scenario needs to be nailed down before full-on production. In a game like **BIO SHOCK**, which features loads of spoken dialog, implementation and pipeline problems can crop up at the last moment, if you don't deal with them early on. This is exactly what happened to the 2K Boston team.



2

**COMMUNICATION.** When deadlines get fierce, and everyone's chugging away, communication is most likely the second thing to go. The first, of course, is quality of life, leading to crunch, but you can bet that we'll get to that.

**AGE OF BOOTY: (Certain Affinity, Max Hoberman)**

"Over the course of the project there were numerous disconnects between the perceived state of the game and the actual state of the game.

"The hardest hit were the designers, who continued fine-tuning plans for sophisticated features like matchmaking and party support long after the programmers had already made huge simplifications (and often cuts) to these systems. A combination of lack of attention to the project, poor communication, and wishful thinking led to the design team believing that several features were far more advanced than we were actually able to implement, and they did not find out the reality until very late in the project."

This depressing quote is made moreso by the fact that this was an Xbox Live Arcade title, and thus had a relatively small team. The company was working on multiple projects at once, which widened the communication gap. But that's another element we'll get to in due time.

**FINAL FANTASY XII (Square Enix, Taku Murata)** "During the development of FINAL FANTASY XII, the pressure to succeed was at such a high point that we were on the brink of losing control during even the slightest misunderstanding. What happened was our team was given the freedom to make changes at various stages of development, but the adverse affect of this freedom was miscommunication, confusion, and disorder. How work was to be distributed was also often ambiguous, which contributed to the problem."

Management overhead is a particularly large problem in Japan, but that's not to say it can't happen at home, too. Often there are too many managers, but not enough actual management going on. Sound familiar?



3

**SCOPE AND SCALE.** Ambition is a double-edged sword. On the one hand, as necessity's sister, it is the aunt of invention. On the other, biting off more than you can chew means your project will bite back.

Schedules aren't always determined by developers, but they are agreed upon by them. Keeping the schedule and the scope of your game within reasonable limits while still doing the best you can is not easy. But it's absolutely critical.

**STRANGLEHOLD (Midway Chicago, Brian Eddy)** "Believe it or not, STRANGLEHOLD began its life as a free-roaming GTA-style game. But after about six months of development we convinced management that this would take far too long to develop, and was too risky to be Midway's first next-gen game on top of everything else we were tackling. We were then allowed to cut the game back to a more linear sandbox experience. But it still had an extensive feature list: the single player game, world interactions, four special moves, massive destruction, drivable vehicles with combat, and extensive multiplayer. We aggressively cut down the scope of each of these features as we progressed, but we kept them all in the game. After about a year of development it was clear we could not complete all these features at AAA quality, but we still kept trying. At the two year mark management set the hard end date. This let us cut drivable vehicles so we could concentrate on getting the rest of the features done at a high quality level. Ultimately cutting vehicles helped get the game shipped, but if we had made that decision a year earlier we could have put more time into mission variety and overall polish."

There's not much you can do when you're fighting against a parent that's beholden to stockholders. This just goes to show the importance of upper management pushing for realistic goals and cuts when necessary—or in fact making sure the scope is manageable in the first place.

**GUITAR HERO (Harmonix, Greg LoPiccolo and Daniel Sussman)** "We were really excited about including a freestyle mode so players could assemble their own crazy solos with divebombs, feedback, finger-tapping, and all the other adolescent guitar showboating moves that we so dearly love. We poured a lot of precious development time and resources into this feature, and sadly, had to cut it. It was very ambitious and we simply didn't have the



## WHAT WENT WRONG?

time we needed to both make it sound good and integrate it into gameplay. Some of us feel that it was a gamble worth taking. Others aren't so sure."

Proper managing of time and scale can prevent some, but not all cases like this. It's one thing to cut a feature when it's not working, but another when time is all it will take to improve it. Cutting partially-developed features primarily because of time doesn't make anyone happy.



**HIRING.** Humans are the most expensive aspect of game development, and also the most difficult to find. It was incredibly common for past postmortems to mention not only the reluctance to hire, but also the difficulty of integrating new hires into the existing culture and methodology.

**ROCK BAND (Harmonix, Rob Kay)** "The only way to complete the title on time was to grow the team. We put most of our staff onto the game, and expanded the organization to new levels. Our offices were bursting at the seams. With staff spread across three floors, and no room left for the new hires, we needed to finish the game, we bit the bullet and moved the entire company to a larger space mid way through Alpha.

Despite all of this, we still didn't hire aggressively enough. Many years making small, tightly focused games had ingrained an efficiency bias and 'smaller is better' mentality that was hard to shake. We were afraid of the additional management required to hire more people, and it resulted in a longer harder crunch for all of us."

The **ROCK BAND** team had further trouble later, and "fooled ourselves into thinking it was too late to integrate any new hires." But ultimately, they had to do so during Beta—the not the easiest time to be training new employees.

**CRACKDOWN (Realtime Worlds, Phil Wilson)** "Another shocking reality was that the number of coders yet to be hired was expressed in the order of tens. In a pragmatic development environment there would have been three choices: 1) increase the budget, 2) lower our ambitions, or 3) pull the plug. Unfortunately, the stakeholders were far from pragmatic, and rather than moving to jettison anything that wasn't in direct support of the clearly defined 'project pillars,' they used the scope discussions as a forum to add yet more ideas."

Keeping the team to measured goals is difficult when re-evaluating a project. Hiring more people and

then increasing the scope kind of defeats the purpose! But as a counterpoint, **TOMB RAIDER LEGEND** had the problem of too many cooks initially. With too large of a team in the pre-production phase, attempting to get 45 people to agree on the direction for the title was a challenge.



**JUGGLING PROJECTS, LACK OF LEADS.** These two problem areas have been lumped together because they yield very similar outcomes—people are stretched too thin, and hold too many responsibilities. This means people often can't see past their bottom line to check the larger trajectory of the game's progress or feel.

**AGE OF BOOTY (Certain Affinity, Max Hoberman)** "Despite our strong start, we made some serious mistakes. Much of this was due to the fact that we split our focus across too many projects and shifted staff onto and off of the game on numerous occasions.

"On the programming side, for instance, we pulled the original two programmers off of the project while we negotiated the deal, which took several months to finally get signed. We eventually got these two back on it by hiring two replacements to handle **LEFT 4 DEAD**, our other big project at the time. We hired a third programmer to assist these two while also contracting out some programming work. We then hired another full-timer and ended our work with the contract programmer. We handed the work the contractor was doing over to one of our **LEFT 4 DEAD** programmers, who volunteered to complete it in his spare time."

And it goes on from there. **Certain Affinity** then took on a third project, pulling the volunteer back off, and eventually putting the original two coders back on. This coder confusion, even on a smaller project, can make a game's development a bit schizophrenic. While **Certain**

**Affinity** felt as a young company it couldn't say no to projects, **AGE OF BOOTY** would've been better off without distractions.

**CATAN (Big Huge Games, Brian Reynolds)** "Our biggest mistake ... was failing to assign a full-time lead programmer or lead artist to the project.

"Not having a solid management structure meant that things tended to fall through the cracks. There was no one to set goals for the programming team or art group. There was no one to assert what needed to be done day-to-day, or week-to-week, or month-to-month. The employees sometimes drifted, unsure what they should work on next, spending too much time on assets that were unimportant, neglecting elements of the game that were actually critical."



As much as developers rail against management on occasion, leads are important for setting structure and scope. Left unchecked, an artist (just as an example!) will fiddle with an asset until doomsday.



**LACK OF TECHNICAL DOCUMENTATION.** This problem was more common than I imagined it would be. But it does stand to reason that determining the scale of your tools, with appropriate code review, is incredibly important to building out your game in the early stages.

**ROCK BAND [Harmonix, Rob Kay]** "Expanding our code department for ROCK BAND meant bringing in new programmers who were very talented but weren't all designer-programmer hybrids, or if they were didn't know they were allowed to be. Too often we jumped headlong into implementation of a new system without taking the time to properly examine the implications or test the edge cases of the design. This bit us in a few areas, notably online matchmaking, which had to be redesigned multiple times.

"No doubt about it, jumping into development of complex new systems without a technical plan upfront was a flat-out mistake."

Harmonix's solution going forward is to include code review and a technical design document before implementing new systems. It seems obvious in some ways, but the frequency of this problem in postmortems indicates it's not yet at the top of everyone's list.

**INDIGO PROPHECY [Quantic Dream, David Cage]**

"Another classic error we committed was trying to develop generic tools with a view to possible future productions, rather than tools dedicated to the experience of the game we wanted to create. The initial scripting tool was supposed to enable us to script both an FPS and a tennis game. The reality quickly proved to be different from the theory.

"A generic tool enables management of a great variety of cases, but none of them very effectively. The prospect of reusing a tool as-is for future productions is usually a pipe dream that costs time and money in the short term, with no guarantee of profitability in the long term."

Quantic Dream managed to realize this mid-project, and adapted the tool—but not without some lost work. Naturally, what would help here is some proper technical documentation of the project's specific needs.



**OUTSOURCING.** Outsourcing is becoming increasingly common in game development, with art being the most common target. It can save money, but it's tough to get right, and requires a lot of senior bandwidth.

**SAM & MAX [Telltale Games, Telltale Team]** "Once we settle on which contractors we're going to use, even more time goes into explaining exactly what we need. Since the SAM & MAX schedule is very tight, we don't have a lot of time to account for corrections or mistakes due to inadequate contractor talent or miscommunication. For example, we felt a lot of pain when a contract studio sold us an A-team, then once the contract was signed, gave us a B-team whose work fell short of the standards we agreed to.

"Communication became even more difficult when the contractors weren't local [we had some on the other side of the planet]. When the contractor's workday takes place during the middle of the night for us, what would normally be a 10-minute conversation turns into a two-day exchange."

Due diligence is incredibly important in the case of contractors, but it's tough to foresee the A/B-team problem, which incidentally was also mentioned by the STUBBS THE ZOMBIE team. Speaking of which ...

**STUBBS THE ZOMBIE [Wideload Games, Alexander Seropian]** "We underestimated just how much time was required to manage contractor submissions. We knew it would be time consuming, but even with that expectation, the combination of art directing and art production was more work than we had time to do. We were short on producers and our artists were scheduled to produce content on their own. We didn't have enough bandwidth available for reviewing submissions in a timely manner. We

realized too late that our production phase requires an intense focus on the work coming in from the contractors. Focusing our internal efforts on the contractor feedback loop should have been a higher priority for our art direction team."

Now we get to the senior management overhead. Without a team dedicated to managing contractors, a lot of work is going to go unused. Wideload now has a section of the team devoted to this, and subsequent production has reportedly been smoother.

CONTINUED ON PG 12





# WHAT WENT WRONG?

CONTINUED FROM PG 11



8

**POLISH** The devil is in the details, but when we talk about "immersion," that's often the result of that particular Beelzebub. Running out of time in the polish phase often means a game that's rough around the edges, or which doesn't fall in line with player expectation.

**TITAN QUEST (Iron Lore, Jeff Goodwill)** "We always told ourselves that six months would be the right amount of time to balance and polish the game. With more than 30 hours of gameplay at each of three difficulty levels and entirely new technology, we needed all of that scheduled time and then some. Just balancing the skill system, with all the combinations of masteries and skills, not to mention the thousands of pieces of equipment with hundreds of thousands of variants, was going to be an epic task.

"In the end, we only had three months. Luckily, we were able to whip the game into shape, but we also had pages of polish feature suggestions that we simply ran out of time to implement."

The team had to compromise, focusing on the first half of the game, and letting polish of the higher difficulty levels go until near the project's end. Not necessarily the best option for player retention.

**GUN (Neversoft, Scott Pease, Chad Findley)** "What's worse than a game that's too short? One that's too short and too easy. Due to a tight schedule and inexperience with the genre, we took a very simplistic approach to game difficulty, putting the standard 'Easy, Normal, Hard, and Insane' selection at the front of the game. Then we sat back and depended on the players to select the appropriate challenge level for their particular tastes. What the hell were we smoking?"

"In the video game market, asking players to set their skill level before they've even played your game is a freaking naive way to go about it. ... our gut reaction to the reviews was that too many people played at the wrong difficulty level—and it was our fault."

Though the Neversoft team focus tested the difficulty levels, choosing difficulty before playing is a daunting task for many players. This is another problem that more time can fix! But how to get that time is a much deeper problem.



9

**POOR TOOL IMPLEMENTATION.** Building tools for your game is horrifyingly difficult, and the problem becomes larger when the use of these tools is not obvious, or when they become too big for their britches. This is the cousin of the lack of technical documentation problem, and no less common.

**RESISTANCE: THE FALL OF MAN (Insomniac, Marcus Smith)** "The flipside to homegrown tools and technology is that our tools changed quickly and our ability to properly train people on all the changes proved impossible. Building assets while simultaneously building the tools needed to create them is akin to trying to build a house on quicksand. Artists would literally open their tools one day and discover new interface buttons and have no idea what they were or how to use them. Many assets needed to be rebuilt, re-lighted and/or re-animated because of changes to our builder tools.

"Adding to the confusion, only a small number of programmers had the knowledge required to debug the problems, and these people were overwhelmed with requests for help. If it weren't for their inhuman effort and long hours hunched over their keyboards, we would have never hit the launch date."

Insomniac had a related problem, wherein the company couldn't maintain stable builds without a great deal of effort. Making your tools accessible is absolutely essential.

**UNCHARTED: DRAKE'S FORTUNE (Naughty Dog, Richard Lemarchand)** "We realized that we'd bitten off too much with our tools approach—we'd tried to be too clever, coming up with convoluted approaches that were intended to solve every last problem that anyone had ever had with each kind of tool. To make it worse, we'd gotten distracted by our lofty aspirations, and had left it very late to implement a build pipeline that let people actually run and play levels.

"The moral that we took away was that even though it's good to aim high with your tools, you should choose your battles, and shouldn't try to solve every last tools issue that your team faces. In some cases, simple work-arounds are better and free up more time to work on the game itself."

The goals were noble, as these problems were originated in a desire to share technology. The lesson remains the same—reign in scope until the project's goals are completely clear.



10

**CRUNCH.** Well here it is, the inevitable crunch discussion. Everyone knows it's a problem, but it keeps on keeping on. Crunch is usually a side effect of most of the other problems on this list, and then exacerbates those problems in turn. Some studios go well out of their way to avoid it—others can't seem to help themselves, for reasons of ambition, money, or any number of things. It's never good.

**DRAWN TO LIFE (5th Cell, Joseph Tringali)** "Consistent overtime is not fun, nor is it something that should be assumed when developing a game. We ended up working late weekdays and Saturdays for the entire second half of the project.

"Our studio had the perfect storm of factors that contribute to crunch—lack of experience on the platform, an ambitious game, and a schedule that was too short. Combined with so little pre-production, we were playing catch up from day one, taking far too many shortcuts to implement the required game features."

Playing catch up from day one is the key phrase here, and proper schedule and project management is the solution. Easier said than done!

**STUBBS THE ZOMBIE (Wideload Games, Alexander Seropian)** "(In spite of using contractors), we crunched for a solid three months, which isn't too bad relative to past experiences, but is way worse than zero. Because we let some major components slip into post-production and were four months behind schedule, and we let ourselves get behind on contractor approvals, we ended up with more than post-production tasks during our post-production phase."

This example should be a reminder to us all that contractors don't necessarily eliminate crunch. A healthy post-production timeframe is very useful, but shouldn't be used as a cushion when other phases fall behind.

## WHAT WENT WRONG?

In the end, a lot of the same problems will keep cropping up in development. People fall into their old working habits and let best practices slide. The jury remains out on methodologies like Scrum, but certainly a very conscious management of projects can help identify problems before they become critical. Let's keep sharing those best practices, and dare I suggest they be shared in *Game Developer* magazine? Do drop us a line! ✨