



## The Seven Deadly Sins of Game Development

Did you know that, of all the game development companies that have failed over the years, 69% have died from self-inflicted wounds? Also, that 13% of all statistics you read are completely fabricated?

Okay, so I made that up. But I actually wouldn't be surprised if the number of developers who failed due to their own mistakes far outweighed those who ended up victims of circumstances beyond their control. Further, when teams fail, the failures are not isolated events; they send ripples through the Business Space Continuum **that end up impacting us all in ways that are not always obvious** -- from the evolution of more onerous contract language to the need for companies to put themselves at risk to salvage a bad position created by someone else.

Some of the mistakes I will focus on this month should seem glaringly obvious and avoidable to experienced developers and business development professionals. But how often have we also seen expedience overcome experience? **No one has immunity**, so let's examine some of these wounds together in the hopes that more of us will avoid them ...

- **The Thief.** When is the last time you audited your machines for unlicensed software? The question presumes the innocence, or at least ignorance of management when raids occur, and once in a great while that may be true. Most of the time though, **everyone in the company is complicit in the use of the unlicensed software.**

I have heard every excuse in the book when folks get busted, from "XYZ should charge us less based on where we are located because everything is cheaper here than in the West" and "everyone does it so why are you persecuting us" to "we wouldn't be in business if we didn't do this." Now these same folks **would scream bloody murder if their game was pirated** and they got virtually nothing for their work.

Folks, we work in an intellectual property industry; certain things are required as part of the price of admission. And those who don't follow these ethical and business imperatives are not only unfairly competing with those who do, **they are also placing their companies and all of their customers at risk.** When raids occur, assets on projects in development are seized, as are the hardware and other means of getting those projects back on track. Publishers miss dates, lose marketing dollars and potential revenues, and other innocent developers are often called into a fire drill to clean up the mess. This one is avoidable at the macro level. Listen up publishers and developers and anyone who uses outside services to accomplish a project! Handle this one just like the issue of having proper insurance certifications was handled; **make it contractual and enforce it!** If developers and outsourcers have to provide copies of the licenses associated with every seat they have working on a project, everyone's problems except The Thief's go away. Seems obvious right? So why haven't we done it? Other self-inflicted wounds are more subtle and insidious ...

- **The Double Agent.** Fear is a great motivator, **but usually not with positive results.** Whether you are a publisher dealing with a developer, a developer dealing with an outsource company, or a manager dealing with your own employees and co-workers, if you try to get what you need through fear and intimidation, the chances are very high that you are creating Double Agents.

A Double Agent is someone who will tell you what you want to hear, even when it isn't true, because they feel you aren't interested in their issues and they must conceal the truth to protect their own self-interest. These rogue operatives stop trusting you long before you learn to stop trusting them, and very often the damage they do before they are unmasked ripples out through your other relationships. Rooting them out once they have been created is difficult, and if you have gone rogue yourself, you know how impossible it is to extract yourself from the situation cleanly. **So the best way to beat this wound is proactive avoidance.** Don't punish people for identifying problems and don't be afraid to do so yourself. Praise that behavior! Flag issues early and openly collaborate with your employees, vendor partners, and customers to resolve them. Make sure everyone knows that the only ones who get punished are the keepers of secrets -- the Double Agents!

- **The Know-It-All.** Clint Eastwood once said, "A man has got to know his limitations." All too often, people take on too many hats, including ones that clearly don't fit them. The most common justification given for doing so is that no one else is available or can do it. But there are other reasons as well: the need to control everything, failing to invest the time to train the folks around you who could do it if you let them, or being unwilling to invest in professional outside help. If you are constantly flying by the seat of your pants, **sooner or later you will find that they have caught on fire.**

I have seen too many game developers get tripped up by HR, legal, accounting, negotiating, or other business issues because they had simply spread themselves so thin they could not manage it all no matter how hard they tried. **Don't let frugality get in the way of efficiency.** Take advantage of skilled contractors who can do certain tasks more efficiently than you can. Figure out your strengths and stick to them, then find resources you can trust to cover your weaknesses.

If you are a business development-oriented person, then you had better make sure you have good production management to keep the services you are selling on track or pretty soon you won't have anything to sell. If you are a great production manager and hate marketing your company, admit it and offload that responsibility. If you don't, sooner or later that really challenging project will come along and consume all of your attention just to get it done. Then, all of a sudden, you deliver and **find you have nothing in the pipe behind it.**

Great leaders don't achieve that greatness by doing it all or doing it better than those around them. They do it **by surrounding themselves with great talent** and empowering that talent to do it better than they ever could have done it themselves.

- **The Mad Scientist.** Mad Scientists come from all walks of life in this industry, but game designers are the most notorious. A Mad Scientist is any individual who is so passionately, slavishly devoted to achieving their personal mission irrespective of the potential consequences to the community around them that they endanger us all.

Consider a game designer or design team who creates something **so ambitious they can never possibly realize it with the resources a team can bring to bear.** Scarce resources are often wasted creating demos that fall short of the vision, and developers try to blow past the gaping holes with platitudes like "that feature hasn't been implemented yet." Either the project doesn't place because commercial minds prevail (and the team still suffers) or, even worse, someone buys into the vision and invests additional resources into the monster before eventually realizing that everyone has bitten off more than they can chew and "feature creep" looks more appealing than the creature itself.

For those who know your indie development history, the most famous instance of this involved neural net programming upon which no less than three publishers in sequence were hoisted aloft in the electrical storm and fried beyond recognition. Learn the lesson that babies are always most beautiful in the eyes of their creators, and don't be afraid to invoke child intervention services when necessary to regain control of a project ... **or kill ones that clearly should have never been born in the first place.**

- **The Glutton.** The Glutton just doesn't know how to say "no." Some Gluttons can't say "no" for the same reasons that people who lived through the Great Depression tend to be hoarders and pack rats. When the famine is over (read as "surviving a series of hand-to-mouth projects"), they try and store up as much as they can for the next famine. And so they over-commit their resources to too many projects and kill their quality and reputations in the process. Others over-commit just because they are greedy.

But there is a third kind of Glutton who is sometimes mistaken for a Double Agent. This is the one who can't say "no" because they so desperately want to please the customer that they agree to almost any request, however impossible it may be. And there are cultural twists on this one as well. In some cultures, it is considered rude to say "no" and, in others, the correct way of saying it is never going to happen is to say, "it's possible." **I call these the unwitting gluttons** because even though they don't mean to be gluttonous, the end result is the same: no one's expectations are met and, again, some innocent bystander is usually called in to clean up the mess. Beware the fat man in the Monty Python sketch who is offered the tiny thin petite wafer; an explosion will be coming.

- **Chatty Cathy.** One of the most common mistakes people make in a relationship-driven business like ours is not to know how to manage them properly. Developers and publishers alike spend tons of their time on conversations that are not going to bear fruit. Networking with well-thought-out objectives and an agenda you know is appropriate for your target is good. **Networking for networking's sake is bar-hopping.** It's a lot of meaningless conversation that is forgettable to both sides. You're lucky if you don't wake up with a headache trying to keep track of it all.

Introductory meetings are fine and usually polite, so the first land mine is **being unable to sort polite interest from real interest.** If you are a developer, you are the talent, and some acquisition folks are never going to say anything to upset the talent. So they behave like Gluttons, having difficulty giving you a clean "no." And you are Chatty Cathy, pursuing them relentlessly and repeatedly while they politely wait to see if you get any prettier over time.

Don't be Chatty Cathy and waste time; be a straight shooter and ask for the same in return. A quick "no" is always better than a prolonged "maybe." Focus your biz dev efforts on conversations with real momentum and, if you don't have enough of those, either you need to reassess what you are offering or find different folks who want what you have. **Going back to the same dry holes gets you nowhere.**

- **The Miser.** You've all heard that "it takes money to make money." No developers have unlimited resources, even the wholly owned studios, so you have got to learn how to manage those resources wisely. Misers threaten the long-term survival of their companies by being penny-wise and pound-foolish. If you want longevity and success, you have to bid your projects accordingly, plan for the future, and invest in it. Invest in the right people, then invest in them further by giving them vested interests in the outcome of their efforts. Give them the right tools they need to do the job and update them frequently. Invest time in your training and developing your staff as well so their skills stay ahead of the power curve, too. Pay attention to upcoming platform transitions and be as prepared as you can be for them. Invest at least 10% of your gross revenue -- not your profit margin -- back into research

and development of your proprietary tools, technologies, and processes.

And if you say you haven't got a 10% margin for this purpose, then your first investment should be in better business management and development services. **You have to be in a constant state of re-engineering with the pace at which technology moves.** And, from a business development standpoint, you need new stories to tell in order to market and position your company for future opportunities. Misers skimp on these things in favor of short-term, lower-impact items. And, in doing so, they slowly undermine their company's long-term potential.

So there you have it ... seven top ways that development professionals shoot themselves in the foot -- and quite often take a lot of good people down with them. Are there more sins we could discuss? Sure, but that will have to wait for another day. **Until then, Good Hunting!**