Vanguard : Post Mortem

by Brad McQuaid (aka Aradune Mithara)

Pieced back together by Emperor of the Internet after being purged from the Wayback Machine.

Intro

I'm going to spend the majority of the time on mistakes we made in development and then what I'd do differently in my next project (e.g. the point of this is to learn from the past and share what I've learned here in the hopes that this information would be helpful not just to me going forward, but also to anyone else working on an MMOG). So the finger pointing is mostly going to be at me, not at our partners. I received a lot of email asking about what went wrong and what I've learned and how I'd apply that knowledge going forward with a new game. I've discussed most of these issues with others who were on the Vanguard team, and we don't always agree. Sometimes the disagreement is about the issue itself, and other times while we agree on the issue, we disagree on how bad the issue was or to what degree it affected development. So take everything with a grain of salt and please realize this is my perspective and not necessarily anyone else who was involved.

Part One: Get Everything in Writing

The first mistake that would have a serious impact later in development was the verbal agreement with Microsoft that Vanguard was to be a first rate, AAA title. In other words, we were going to get the funding we needed to compete with other AAA MMOGs, and that we would periodically evaluate the competition and adjust Vanguard's budget and/or release date if it made sense to us and Microsoft. And as time past by, we did increase the budget as games like WoW were released with very high development costs and a ton of polish. But then there was a regime change at Microsoft, and the people with whom we had this understanding and commitment were no longer there. The new hierarchy did not have the same perspective and commitment to Vanguard and when we needed more time and more money, the general reaction was that we were screwing up management-wise. So the moral of the story is one that should have occurred to us: get everything in writing, get it into the contract, because even a company like Microsoft can suddenly undergo significant changes to its management and teams. And when those changes do happen, you might as well be dealing with a new company – anything and everything can change, and change quickly.

One thought you might have reading this could be, "Would any publisher have signed a more open-ended deal?" And that would be a very good question. I do think, with 20/20 hindsight, that a more open-ended deal could have been made. I'm not talking about a contract carte-blanch, but something more than what went down. At this time I had multiple large publishers calling me on the phone, wanting to do a deal with me. So we were in a pretty good position.

Now, I said in the intro above that the majority of what I'm going to write is going to be focused inward, not outward. So my next blog is going to be about the plusses and potentially bad minuses of putting together an all-star team. That said, I'm going to approach this chronologically, and the above issue, IMHO, turned out to be our first significant mistake.

Without a doubt, everything conceivable should have been in writing. Like you touched on, if there is a change in the structure of organization, it can be a huge obstacle to overcome.

You needed to gain the trust of people you had little to no trust built with. When your allies and supporters left, those mutual "understandings" left with them.

Hindsight is always 20/20. This should serve as a lesson to anyone in business

---

Part Two: The All-Star Team

One of the most exciting prospects in terms of starting a new MMO-focused company was being able to hire the best and the brightest. Jeff and I were able to bring aboard who we wanted to be the founders of Sigil. Then the founders were able to recommend people with whom they'd worked. A shining example would be the art team David Gilbertson and Keith Parkinson put together – all sorts of top-notch people with a variety of skills and previous experience.

The majority of people hired were those with whom we'd worked with directly. Others were those that came with impressive resumes. Some were our friends, and while they may have lacked experience, they were a known quantity – often, one of us had known them for many years. Sigil has been accused of nepotism in the past, but I don't think it's nepotism when you bring a friend on board because he or she is someone you know and trust. Rather, I think it's really smart. Real nepotism, in my opinion, is when you hire a previously unemployable relative with room temperature IQ to do something he or she is hardly qualified to do. I do think we avoided this, though, and when and if I'm able to build a new company, I'll do it all much the same way. Well, much the same way except for the following:

All-Star people sometimes have all-star egos. And while these egos may or may not be justified in some abstract way, they don't work at all when building a team. And that's the key thing to keep in mind: you are building a team. And just like a soccer team full of egos who will only dribble the ball and never pass it, the inability to play well with others will almost always result in a dysfunctional environment. MMOG game development is ambitious no matter how you approach it or with whom. It demands a high degree of collaboration. Ideally you want that all-star team to consist of people with a variety of backgrounds, perspectives, and preferences. And the magic occurs when this group gets together and creatively comes up with something that is greater than the sum of its parts.

And while most Sigil team members did indeed play well with others, we did have a few that didn't. And even though it was only a few, it eventually led to some big issues, including negatively affecting morale and productivity. It wasn't only that the problem person had issues themselves, it also led to good team players not wanting to give it their all. But I was blind to this, especially early on. I was so excited about bringing all-stars on board and the potential greatness that could come of it, I downplayed the ego problem. Sure, I saw it as an issue, but also as something we in management could deal with in the future. We'd work with these people and, over time, most if not all of them would hopefully come around. There would be some pain involved, but it would be worth it. In fact, during the earlier stages of team building I'd even do press releases, announcing the all-stars we were bringing aboard (especially if they had been members of the 'original' EQ team). Heck, I thought, I might as well get the public as excited as I was.

But despite very serious (and often prolonged) attempts by management, including me, to address these issues, they rarely got better. And sometimes, even after a problem person eventually left the company, scars were left behind. Some bad feelings and habits persisted.

So what would I do differently? Well, I'd make an effort to resolve the situation, but if that effort failed, I'd let the person go. I wouldn't let what they could potentially bring to the team and project blind me. The benefits one gets from a smoothly running team or department without the egos is simply too great. And after all, despite how much fun it is to create games, it's still ultimately a business. You are creating a product and a service, not a community with great mental health counseling.

Ultimately I realized that I'd rather hire somebody less qualified, but who thrived in a team environment, than somebody with unparalleled experience and talent, but also an untamable ego. Thankfully we in management only made a few of these blunders, otherwise who knows what could have negatively occurred. But then, like I said, it only takes a few to have a real impact. And that impact was felt.

Part Three: The Downside of Being a Geek

Not many people know, but EverQuest was initially a software-rendered game. We first started working on EQ back in 1996 using Pentium 133mhz machines with Matrox cards. SISA had licensed the Pyrotechnics engine and had people already working with it on Tanarus and a Spawn PC game (later cancelled). We borrowed the engine, network code, and a level editor. It wasn't very long before we had something running and online, albeit very primitive.

As time passed, we bought faster PCs, updated the level editor, added a simple interface, etc. Then news about the Voodoo 1 3D card began to float around. We hired John Buckley, who had worked on the engine at Pyrotechnics, and he started modifying it to work with 3D hardware. At that point we were planning on releasing both a software renderer version of the game and also support for the Voodoo 1.

A year or so passed and I remember distinctly being in a meeting with the EQ leads and Smed. We were discussing min specs and hardware requirements. After a while, Smed starting pushing the idea of going hardware only, 3dfx Voodoo 1 required. I remember being wary of the idea, but also intrigued by it. Such a decision back then was a big one and we knew there would be very few games out when we launched that would also be hardware-only. But when we considered all the more we'd be able to do with the game world, we fairly quickly agreed with the idea and the decision was made.

By the time we released we were running two Voodoo 2 cards in tandem and thought the game both looked and performed well. And when we released and the game caught on like wildfire, we were pretty sure we'd made the right call. Ironically, about the only thing we didn't pull off technically was a seamless world (something else Smed had really wanted to see).

Fast-forward a few years and I was at Sigil talking with the Founders about the tech level we wanted to achieve with Vanguard. Many of us, coming from EQ, wanted to make another MMO using state of the art graphics. We felt it would make the world more immersive, and, quite honestly, we were graphics tech-heads and very excited about what hardware and Direct X would be pulling off in the years to come.

With EQ we were 3D and hardware accelerated, while our biggest competition at the time was UO, a 2d tile-based game. And we'd trounced UO pretty well. So looking at what might be Vanguard's competition, we felt we had to visually trounce them as well. Any concerns about performance were mitigated by the belief that by the time the game came out, there would be plenty of PC horsepower readily available. And Vanguard players, if they didn't already have the necessary horsepower, would undoubtedly upgrade their PCs. Heck, a LOT of people had bought Voodoo cards just to play EQ.

Of course, looking back now with 20/20 hindsight, we were very wrong. Over 80% of the people who bought Vanguard and tried to play it quit by level 2 or 3. What could be the reason? Well, given how fast it was to level the first couple of times, what could be so horrible that people would quit so quickly? My bet is crappy framerate (due to rendering too many polygons and too many and too big textures) and bad hitching (being the result of the world being seamless and having huge art assets). We also released early and didn't have a chance to optimize the code (but this is a subject I've already addressed in my blogs and not at all the only reason we had issues).

Why didn't our techno-geek approach work with Vanguard when it had worked with EQ? Probably there are a few reasons, but the big one I think is the perceived difference between the games and their competition. First, EQ didn't have a lot of competition and many people were new to MMOs. This wasn't true with Vanguard – there were many released MMOs when it came out. Also just about anyone, techno-geek or not, perceived a big difference between a 2d tile-based UO and a truly 3d game like EverQuest. It was like going from cassette tapes to CDs – just about everyone perceived a huge difference.

EQ was also a fairly hard-core game (although it's mellowed through the years). EQ's players were into the game big time, but also into the technology. They were willing to upgrade their machines to play this amazing looking (at the time) and amazing playing game. EQ topped out around 500k players, and I think it's safe to say that the majority of them were at least fairly hard-core.

Now compare Vanguard and WoW. Vanguard, technically, is far more advanced than WoW. But perception-wise? It's not like going from a cassette to a CD; rather, it's like going from a DVD to a Blu-ray disc. Videophile that I am, I totally prefer 1080p to 480p. But my wife? She shrugs at the difference and gets on my case about buying expensive blu-rays all of the time, upgrading my movie collection whenever a new disc is released.

Then you have more casual MMO gamers. Not only to them is the graphics technology not a big deal, but they're also far less willing to upgrade their PCs with expensive new CPUs and GPUs. Blizzard was brilliant – they created a mass-market MMO that could be played by the mass-market. Where their game lacks in technology, they make up for it with the quality of art and overall polish. And, as a result, they have millions and millions of players, not 500k (yes, I know there are other reasons as well, and I'll undoubtedly cover many of them in future blogs).

Now for those of you who read my pre-Vanguard launch posts, you know I'd already considered some of this. We knew that we were making a more hard-core game, and we also knew if we could even get 500k players that the situation would be very profitable. So why didn't the more hard-core gamers upgrade their machines, like they had with EQ?

As mentioned, the perceived extra quality and extra immersion was not nearly as great as with EQ vs. UO. Also, even if you had a pretty buff machine, Vanguard still ran poorly in many situations. EQ had some performance problems, even on dual Voodoo 2s, but not nearly as great. Also, a 3D world was a relatively new experience in 1999, especially an online 3D world. In 2007, 3D wasn't a novelty but rather a standard. And if you didn't like one MMO, you had others to try-out. In 1999, you didn't really have that luxury.

So where did we screw up? Were we just victims of changing technology and changing standards? No, I don't think so. I think we were blinded by where we saw technology going and all of the cool things we could achieve by harnessing that technology. We used the EQ experience to bolster our confidence that people would upgrade for a great game. We looked into the future, using Moore's law and the like, confident that while the game in development was a dog, that by release graphic cards would be both powerful enough and cheap enough that system reqs wouldn't be a big issue. This, obviously, was not the case. Lastly, I certainly encouraged my graphics programmers to attempt more and more. They'd integrate something new, say high dynamic range lighting, and I'd go 'ooh' and 'ahh', patting them on their shoulders and full of encouragement.

Looking back, and then looking into the future, I hope that I wouldn't make these same mistakes again. Whether I work on a more mass-market game or something targeted and niche, I still need to remember that forcing new technology on people had better offer the player something night and day better than what they are used to. I need to remember not to get caught up in the allure of 3d graphics technology and what new cards and new versions of Direct X promise. High dynamic range lighting, a 50 mile clipping plane, and a massive seamless world do help with immersion, but not at the cost of performance and playability. Immersion has more to do with a pleasing quality of art and polish while simultaneously achieving a playable framerate. And the lessons learned with EQ, while invaluable, are not necessary the be-all and end-all of lessons to be learned. Having the EQ experience behind me is a huge advantage when it comes to building an MMO, but I need to remember that EQ was 1999 (and, for that matter, Vanguard was 2007). Developing a 2014 MMO will rely on lessons learned in the past, but also on understanding the audience and the market at that future time. And while I disagree with those who assert EQ's success was all about timing, I do have to agree that timing was a big part of it. Times have and will continue to change.

Part Four: [Unknown Title For This Part]

We wanted a huge world at launch. We knew we were going up against other MMOs that were already out there, several with expansions already released. Also, with our goals of a seamless world and a view that went on for miles we knew the size of the world would be key. We also wanted to lay out the majority of the world, including what would be released as expansions.

The game designers used map making software and produced this huge world. It was crazy big, although it was hard to tell exactly how big because we were just looking at a map. We hadn't tried to create any of the world in-game yet. The art team needed time as did the graphics engine (taking the Unreal 2.5 engine and making it seamless world capable took some time). We then decided on three major continents, Thestra, Qalia, and Kojan, and then what races would start out where.

We knew that those three large land masses would accommodate a lot of people. Travel time could be an issue, so we made sure we'd offer vehicles at a relatively low level (we used the term vehicles loosely and to include horses, ships, etc.). On message boards and in the FAQ I let people know about the size and, although we were not fans of teleporters, I posted that they may be needed to some extent and that we'd determine that in beta testing.

This led to two major errors.

First, when it became time for the artists to start building these land masses, it took longer than was expected and hoped. The easy solution at that point would simply be to shrink the world. But when this problem occurred to us, when Thestra was already mostly done and Qalia in progress, it wasn't an easy fix. People were saying 'let's shrink Kojan' or 'let's just cut Kojan completely'. But shrinking it too much would make it much smaller vs. the already built Thestra and Qalia. Cutting it completely would mean we didn't have a home for the races that started there. So we ended up cutting parts of it away, and the game launched with Kojan being smaller than the other two major continents.

Then, the art team went back and polished Thestra. The tools were a lot better at that point, and the artists better versed on how to build a good looking world. And even with re-visiting Thestra, if you really look, I think Kojan is the best looking continent by far. But the bottom line was that we spent too much time building three continents and making them look good. We should have had fall-back positions, enabling us to make a smaller world with more starting areas in fewer continents.

The second major error had to do with under-population.

It's hard, perhaps impossible, to launch with exactly the right amount of content and world size. With EverQuest the problem the majority of the time was over-population. There was only a certain amount of content, of dungeons, of outdoor areas. When the game took off being a much bigger hit than we had assumed it would be, over population was the big issue. Too many people per server meant too much fighting over limited resources. Players grew frustrated.

The fix for EverQuest, though, wasn't all that hard to implement. We came up with an upper limit in terms of server population during peak hours. If a world server was exceeding that limit, we'd launch another world server and then encourage people to migrate. We also would split servers if necessary.

What happened with Vanguard, however, was very different. Under-population is much worse than over-population. As mentioned, if a world server was over-populated, you can add another server and/or split a server into two worlds. And until you did so, people would complain (and rightfully so) that the world was two crowded and the fight over limited resources (zones, items, etc.) would grow too intense. That said, players usually would not quit the game.

But under-population creates an empty world in terms of other players. A big part of Vanguard (and EQ, for that matter) was about grouping. If we were going to push grouping, then people needed to run into other players. They could then group, get to know each other, and feel part of the world. Obviously, there was a certain population density needed for this to reliably occur. And, even with the game offering soloable content, players still want to feel like they are part of a living world. They may prefer not to group, but they still want to see other players, duel them, trade with them, etc. And, unlike the case of over-population, people would indeed quit because of under-population.

So if we knew because of our bigger world that we would need to support a larger number of players per world, what went wrong? Server performance. With EverQuest the world server would become over-populated in terms of resources and crowding before the actual server would become overloaded. I think this gave us a false sense of security and when, during beta, we discovered that the software and hardware limited us to too few per world, we really didn’t know what to do.

So we launched with a huge world and unparalleled in-game views of that world, but with hardly any other players around with whom to share that world. Adding teleporters helped, but not enough. Community, which is an important glue when it comes to MMOs, didn’t build because of under-population. The world seemed empty, it was hard to find groups, etc.

Looking back I think it would have been a lot better to make the world more modular. The design, the content, and the racial starting areas should have been laid out such that we could shrink the world without it interfering with our plans. And then we should have created test areas much earlier on that would have allowed us to artificially populate a region with lots of players. This would have revealed a population cap lower than what we had assumed. Then we could have reduced the world size accordingly. I think the fact that we correctly and relatively easily dealt with EQ’s over-population lured us into a false sense of security. We were simply so ambitious that we were blind to a lot of this, and by the time we realized what these problems were it was simply to late to go back and conduct a major overhaul, especially given that we were short on development time anyway.

Quellen

Developer Rant on Vanguard: Saga of Heroes

By Sony System's Designer Vincent Napoli

Link to Fires of Heaven Post

You know, as much as I hate having to carefully craft (AKA, lie through my teeth) an answer to "What was Vanguard's biggest failing?" in job interviews, I realized after reading that rather disappointing article how proud I am of it.

Know why? Because I can honestly say with 100% validity: I'm a big reason for Vanguard's failure. Not Brad Mcquaid - not Microsoft. Me. And Guess what? I'm really kind of proud of it.

Brad Mcquaid didn't do shit. (News Flash?) He's had an opiate addiction for years now, which only got progressively worse as the project failed. His cumulative face time with sigil designers in the most crucial final years of development? Approx: 15 minutes. And some of the time was spent begging for legitimately acquired narcotics (Or in times of desperation, jacking them from people's desk).

The lead designers didn't do shit. (News Flash?) Sigil fired all of their golden-boy, EQ-Genius designers (Save some who would walk away in disgust) who this board once speculated simply "left." It wasn't even secretive. It all happened on the same day.

Sony didn't do shit. The extent of sony's help was 2 designers who ended up writing some diplomacy quests in Tanvu and some adventuring quests in Tursh. I think there was an artist that came in 2 days a week or something for about a month also. Thom Terrasas (sp?) is the only Sony employee that ever directly affected the direction of that game.

The only part Sony really played in Vanguard's destiny was to let its life unnaturally and undeserving-ly continue. And apparently, it's simply because they were naive enough to think this project was worth their cash. Hah! Even the staff at sigil was left wondering why the hell Sony would buy us. Dozens of lunch hours were spent trying to figure out why.

"What profitable web of intrigue and mystery was big'ol Smed spinning with this crazy move(????)," we'd often cry

It was pretty shocking (and just lame) to hear John Smedly actually get angry and complain to people after the layoff's that he, "didn't know what he was buying." He even expressed anger at Jeff and Brad for bamboozlin' him. Poor guy. Maybe next time tough-guy Smed decides to spend several million dollars on something he'll expend some brain power figuring out what it is first.

Dave Gilbertson DID do some shit. (News Flash!) But this guy? Man, so much stuff I could say about this guy. He was truly unbelievable. Even when you thought his insanely unprofessional antics couldn't get any more outrageous, he'd go and do something like tell everyone they're getting a raise (to keep crunching) and then one by one call people into his office who WERE actually getting raises (but would never actually get them), how much they were going to get (VERY, soon). Unfortunately he would move through desk rows one by one and simply skip over the unlucky ones. It took a whole 5 minutes for the office to see through his brilliantly laid out scheme. He used the same plan for the lay-offs too. Classy huh?

He's literally never played a video game in his life, yet when Brad died off and Dave inherited the position of Vanguard Jesus, he decided he must be the final call on every design decision. I guess if you ride dirt bikes with a gamer god, his genius just wears off on you.

Fortunately, sometime this would result in getting played like a fiddle by whoever happened to be lovingly pulling the strings that day. But more often than not, this just meant people had to go around him to get something in, only without the help of (Place whatever department here) that was necessary for a game feature to actually turn out right. Imagine for a second people at Sigil actually knew how to do something right? (Believe it or not, we did on occasion) this guy would become the bottleneck to prevent that from happening.

If there was a ceremony for the Gamespy award, Dave would be accepting. For the sake of all our future video game consumer habits, let's hope this guy goes back to the only thing he's qualified to do, whatever that might be.

Anyway, enough of my blabbering. The most shocking reality that I don't think anyone really ever understood is that Vanguard was made (exclusively the design staff, I should say) COMPLETELY by amateurs. People who had been hired less than a week with 0 prior experience were tasked with designing entire newbie areas that shipped. People who had never produced a game in their life were asked to fix a 40 million dollar fuck up. People with no experience were asked to fix the item, diplomacy, ability, content, quest and pretty much every system in the game.

The game that exists now was designed in a single year by people with 0 experience. If that sounds too vague think of it like this: about 1 year from release we had 0 quests in the DB because the tool didn't exist yet. When I decided to split the team there was over 30,000 quest object entries. Yeah, explains a lot doesn't it?

What a huge let down indeed.

Oddly enough, the whole situation was probably a bigger let down to the designers than the consumers. I accepted a position thinking I was going to work with a bunch of experts - Masters of their craft - and really learn the ropes of game design. Instead, my fellow design associates and I were unwittingly tasked with trying to fix a failed video game that had literally been canceled twice before any of us were even hired. So in retrospect, despite everything, I guess I'm still pretty proud of vanguard. Every team member should be proud in spite of a truly pitiful and pathetic waste.