

Lastwords

Any significantly advanced
technology is indistinguishable
from a rigged demo

Er...ah...I mean magic

Rigged Demos?

- The investors were coming for a demo and the hardware had been having a bad day so we put the four investors in the cockpits that worked and had four of our people in the dead ones, yelling like they were having a lot of fun. Nobody noticed.
- *Of course* I don't recommend this (unless you're desperate or your managers are demanding way to many demos) but always be on the look out for people trying to do it *to you!*

Use off-the-shelf parts *and* software whenever you can!

Remember to figure in *all* the costs of going custom!

Be Careful of Custom Solutions

- We tried to build a multi-channel intercom twice even though off-the-shelf solutions were available. The final solution ended up MORE expensive than the off-the-shelf one and didn't work as well.
- When evaluating a custom solution vs off-the-shelf it's easy to forget the cost of design, test, installation and the cost of opportunities missed while developing it, These costs usually exceed the cost of parts by a considerable margin.
- Early cockpits used an Amiga as a second processor. We could buy them retail, throw out the bits we didn't need and still cost less than building something similar.

But when you can't avoid it...

- If there really IS no off the shelf solution, don't try to pretend there is, go ahead and make the tool.
- "World Builder" tools are often better custom.

Too many demos can kill you!

The problem with demos

- Some demos are necessary, but many high profile projects seem to have them every other day.
- Demos to investors and publishers should have time allocated for them in the schedule.
- Keeping software in a demonstratable state isn't always easy when it's being worked on.
- Managers need to understand asking everyone on the team to frequently "show someone what you're working on" will impact the schedule.
- If you take the time to save a copy whenever you have a "working build" managers and PR people can do their own demos.

Military simulators aren't as sophisticated as many people think!

But they are still worth studying!

Military Sims?

- Their prime focus is *accuracy*.
- The visuals and effects reflect this, many would be considered amateurish in a mainstream game.
- They tend to solve problems with dollars and bandwidth that games couldn't afford.
- All their work targets a very narrow market, "war fighting professionals".
- They don't have to worry about scaring off the novices, they have a captive audience.

But they are ahead in some areas

- Representing large amounts of terrain. (hundreds of square miles)
- Interoperability of unrelated simulations.
- Human factors research (they do a ton of this which is very valuable to read)
- Novel user interfaces and training techniques.
- Mixing simulation and reality.
- Blowing and drifting smoke.
- Real-time terrain modification (bulldozers that actually work!)

Beware the Million Man Scenario

- The military has been trying to do this for years, it's a lot harder than it looks!
- They have spent a lot more money and burned a lot more bandwidth and CPU cycles than you could ever afford, and it still doesn't work.
- Everyone I know of who has tried this has failed.
- Of course, it's not impossible, but it is a very, very hard problem so beware of projects promising they are going to solve it.

Thoughts for Managers

Remember, your tech staff are not just eccentric, caffeine swilling pizza eating geeks...they create the product that *pays your salary.*

Thoughts for Managers

- It's very bad for moral if management and marketing don't appear to work as hard as the technical staff. And this is *common!*
- Jackets, T-shirts and parties are moral boosters and help make people feel like a team, but *do not* think they are a substitute for a cash bonus!
- *Do not* force event attendance, remember not everyone likes the same things!
- When giving gifts remember not everyone drinks, not everyone eats meat and some people already have all the watches they need. Don't give gifts only half your people want, cash is universally well received.

More thoughts for managers

- People's opinions on open/closed office arrangements vary, but always separate noisy areas (kitchen, playtest, lobby) and people who must be on the phone a lot from your tech staff.
- When people are in crunch time working late into the night, the least you can do is have (good) food delivered. (it's also tax deductible!)
- If every other project your team has done took 18 months, don't force them to do the next one in 12.
- It's better to ship a product *when it's done*, than to ship on time with lots of bugs. You can sell crap if you put it in a pretty enough box, but only *once*!

Still more...

- Design achievable schedules and milestones so people can feel they are accomplishing things.
- Many of your employees could be making more money in industry, they have a reason for working in the games, find out what it is and encourage it!

Thoughts for the Tech Staff

Managers and marketing people aren't just stiff, suit wearing, cash obsessed MBAs...they SELL the product that *pays your salary!*

Thoughts for the tech staff

- Managers aren't idiots, their training is just different from yours!
- A huge percentage of the problems between tech and management staff is caused by lack of effective communications.
- You need to learn how to communicate tech issues in understandable terms, not just talk buzz-words till people's eyes glaze over.
- Be realistic in your communications and time estimates, when something isn't going to work, explain clearly why, don't just bitch to each other about it or problems will never be fixed.

Thoughts For Everyone

- Everyone schedules time for testing, but many people forget to schedule time to fix the bugs testing finds!
- The more people in your company who don't like to play your own games, the more you should suspect your design sucks.
- If your management and marketing department doesn't like playing your games, you may be working with the wrong people.
- Incentives and bonuses should be paid regularly and promptly, not used as "golden handcuffs"
- Don't hide your light under a bushel basket.

Don't forget the small stuff!

- We massively improved the quality of our sound system, simply by using thicker hook-up wire.
- Remember, people need to be able to easily install (and someday remove) your software.
- Every time we skimmed on (even basic) testing, we regretted it.
- Always try to use the fastest development hardware you can afford, it saves you time in the long run (*but don't test on it!*)
- It's often better to get a design working and *then* make it fast, than to obsess on speed from the start.

When you're under the gun...

- Remember to check your E-Mail and faxes, we sweated a bug all night before a demo and couldn't fix it, only to discover the investors had sent us a fax postponing the demo a day.
- What you think is cool isn't always the same as others. We sweated putting in a raft of new features for a demo but what wowed the investors was a debug feature that let us make a mech float up into the air.

General Discussion

- Questions?
- Random thoughts?
- Stories?
- Your own experiences?
- Throw rotten fruit?