Earthquakes in the project portfolio

by Dr. Graham Oakes, Graham Oakes Ltd, UK

Even back in the mid-1990s, when my experience with the industry began, a game could easily cost $2-million (R16-million) to develop. The company I worked for had about 60 such games under development at any one time.

My company, like most in the industry, had a problem. Projects slipped. They often slipped by months or even years. This didn’t do a lot to help our reputation with retailers, reviewers and customers.

Perhaps even more critically for people who cared about things like making payroll, it made it impossible to predict cashflow. I was part of a team that was set up to bring predictability to our project delivery.

Each member of my team was responsible for providing an independent view of the status of about ten projects in the development portfolio. Each week we looked at what our project were producing and tracked this against the original milestone schedule. We tracked the status of key risks. We read status reports.

Above all, we talked with the project managers, discussing the issues they were dealing with and listening to their concerns. Sometimes we offered suggestions or put them in touch with other people who were dealing with similar issues, but often we just functioned as a sounding board to help them think through what was going on.

We also produced a weekly report for senior management - the director of development and the chief financial officer. This consisted of a simple ordered listing of the project under development, ranked by our assessment of their level of risk.

We also wrote one or two sentences on each project, summarising our reason for its ranking. This report was openly published to everyone in the company, which gave plenty of opportunities to tell us where we’d got it wrong.

Interestingly, project managers generally reckoned their project was riskier than we’d assessed them to be.

Either way, people started to actively track the positioning of their project, and to tell us how our ranking of its status could be improved. By publishing our report openly, we created a very useful channel for this information.

After we’d been working with our projects for a while, we began to recognise a pattern. Projects would go through a couple of fairly formal investments-approval reviews when they were set up.

They’d then run quietly for nine or 12 months. Then, about three months before the date they were due to be delivered into testing, they’d start to slip. Often they’d have a big slip initially, followed by a series of smaller slips as they got closer to the end date.

The pattern of the slips was the same for just about everyone.

This pattern was remarkably consistent. Because we were working with a portfolio of 60 similar projects, we could draw graphs and start to see statistical trends. We found a strong correlation between the magnitude of each slip and the length of time left until the due date for delivery into testing.

Projects took three month slips when they were about three months from delivery, two month slips when they were about two months from delivery, one month slips when about a month from delivery, and so on. The company was developing racing games, adventure games, first person shooters, and so on.

Our teams were based in the UK, in France and in America. Everywhere it was the same: our apparently stable projects suddenly started to slip in their end phases. And the pattern of the slips, a large slip followed by a succession of smaller ones, was the same for just about everyone.

To me, with my original training in geophysics, this pattern looks a lot like an earthquake. Stress gradually builds up as tectonic plates move. Finally the rocks break, give off a loud bang, and settle into a less strained position. Then a series of aftershocks releases any residual stress.

So it was with our projects. For a long time people could ignore the small setbacks and minor slippages, perhaps hoping to make up the time later. Finally, as the end date loomed, they could no longer avoid reality. So they’d take a big slip to release all the built up delay. Then the stress would build up again, and they’d take a series of smaller slips to release it.

We monitored this pattern as we continued our reviews. After a couple of years, we found that the pattern had shifted in time. Projects were still slipping. The general correlation was still pretty much the same - a large initial slip followed by progressively smaller ones. But the slips were happening about three months earlier in the project lifecycle.

There were several reasons for this movement - people were monitoring status more closely; project managers could use the review team to back their judgement as to when a slip was needed; so had the confidence to make the call earlier; we’d got better at defining clear milestones.

Overall, however, we were simply having a much more informed dialogue about our projects. This helped us to identify and relieve the stresses earlier. It also allowed us to give everyone three months more notice about potential delays, meaning that the CFO could be a little more confident about setting analyst expectations, and about making payroll.

Of course, life’s never as simple as the case studies make out. In order to operate effectively, the review team needed to overcome a number of challenges. For example:

- Game development teams have very diverse skills - programmers, graphic artists, musicians, and so on. It can be difficult for a reviewer to understand the status of everything these specialists are working on. By doing small, frequent reviews we could often get a good feel for overall trends, but sometimes we needed to call on technical experts from other teams to help us understand what was going on.

This helped us to identify and relieve the stresses earlier. It also allowed us to give everyone three months more notice about potential delays, meaning that the CFO could be a little more confident about setting analyst expectations, and about making payroll.

Of course, life’s never as simple as the case studies make out. In order to operate effectively, the review team needed to overcome a number of challenges. For example:

- Game development teams have very diverse skills - programmers, graphic artists, musicians, and so on. It can be difficult for a reviewer to understand the status of everything these specialists are working on. By doing small, frequent reviews we could often get a good feel for overall trends, but sometimes we needed to call on technical experts from other teams to help us understand what was going on.
• Reviewers could sometimes get pretty isolated. They floated across several teams rather than belonging to any one project. Furthermore, they often had to back their own judgement and deliver bad news against the convictions of a project or line manager. So we needed to build our own internal mentoring and support structure.

• Everyone wanted to be the first to see our reports. Project managers naturally wanted a chance to fix any issues on their projects before we reported them more widely. The project managers’ line management wanted to know what we were saying before their executive management heard it. And, of course, the executives we reported to wanted to hear of any issues as quickly as possible. We had to evolve clear communication protocols to win everyone’s buy in.

Over time, I’ve come to realise that not all of life is like games development. Industries have different drivers. Companies have different strategies and approaches. People differ in all sorts of ways.

Games are very different to banking engines or to customer relationship management systems. However, there are many similarities too. Some of the lessons we learned on that games portfolio can be generalised. For example:

• People need a sounding board. Most IT projects are complex, with many people involved and lots of moving parts. Project managers are rarely given the time to sit down and reflect on what’s going on. Simply by creating space for this reflection, we helped them identify and solve many of the problems on their projects.

• Openly published information creates a conduit for dialogue. This dialogue helps improve the accuracy of the original information, and provides an opportunity to gather more information. Project, line and executive managers all need such accurate and complete information if they are to make effective decisions.

• Independent reviews can help validate the information provided by teams and project managers. They can also help make the above reporting and reflection processes more robust. I believe project reviews can add value in virtually all circumstances, but you need to tailor them to each situation.

Projects can only succeed when they deal with reality. Reflection, dialogue and independent review are key tools for helping our projects keep in touch with reality.

Acknowledgment
This article was published in the British Computer Society’s journal ITNow, January 2008 and is republished with permission. See www.bcs.org
Contact Dr. Graham Oakes, www.grahamoakes.co.uk