Jennifer Carlisle, PMP

The IT Laws of Physics

Practical Applications For the Private Sector

The Center for Program Transformation is a nonprofit organization designed to research ways to improve program management within the federal government. If you need convincing that government programs need help, consider this: the US Government Accountability Office (GAO) determined in July 2008 that over $25 Billion in federal programs were either poorly planned, poorly performing or both.

Among its other accomplishments, the Center has published a list of eleven IT Laws of Physics™. Although these laws were developed for the government realm, they ring true for the private sector as well.

First Law
Planning is a continuous process, not a one-time event.

Isn't this what project control is all about? Information Technology projects are notorious for cost and schedule overruns, often blamed on scope creep or unexpected integration issues. Agile Development methods are a great way to build iterative planning into the development process. While I am the first to advocate having a solid charter in place before beginning any project, I also believe that organizations are bound to be disappointed when they irrationally rely on initial estimates developed without full task level detail. They compound their problem if they don't adjust estimates based upon new information as it becomes available. How many of us have celebrated project completion knowing the team did a great job considering all the change thrown at them, but been a little miffed that we wouldn't get our bonuses because those were based on the original timeline and budget?

Second Law
Complexity kills IT projects since defects and security vulnerabilities increase nonlinearly with increased complexity.

There are a number of project elements that become more difficult as an IT project becomes more complex. Hardware, software, programming languages, vendors, physical locations, virtual teams, deployment mechanisms, data interfaces, compliance requirements, contractual obligations, etc., each layer of complexity can add more people to the team, more requirements to the project, more things to be tested, more risks to be realized. This is one reason some project managers state that their role is to prevent project change, because change is a source of complexity. Of course, wise PMs know that change is inevitable and their role is to manage it.

Complexity can be managed to some extent at the beginning of an IT project by minimizing the number of components (interfaces, vendors, etc.) involved; analyzing which are the most critical to meeting customer requirements; and understanding the risks associated with each new element as well as appropriate risk responses. For example, if multiple data interfaces across several software systems will be required, there is a risk that data elements will not be mapped

accurately. An appropriate response may be to create an easily accessible data dictionary library so that the analysts performing the mapping can be assured of accessing current, accurate data. Similar tactics can be used during the project to minimize the addition of new complexity.

Third Law
Schedules and project chaos create Event Horizons, from which a project cannot recover.

I have to admit that this one threw me for a loop. I had no idea what an event horizon was. Actually, I'm still not capable of articulating what I learned, other than to say it has to do with a black hole and it sounds really bad. I think the authors are referring to what I call the Perfect Storm Theory, which is that some projects become the Perfect Storm, and once they have reached that point, things will not end well. They might end; you might deliver the project; but no one is going to be happy. As this law states, schedules and project chaos have a lot to do with it, particularly fixed timeline schedules and poorly managed requirements.

To prevent the Perfect Storm in an IT project, use effective risk management techniques, paying particular attention to your risk response plans, as well as reducing complexity and closely managing requirements. The key, however, is to recognize the storm when it first begins to brew so you can stop it quickly. Mind your team members closely; when they no longer want to come to your meetings or their grumbling is no longer good-natured, it's time. Do whatever it takes; if you are at a loss, sit your core team down and brainstorm it through. They will appreciate you involving them, and, if you can prevent the storm, you'll all be champions. If the storm develops despite your best efforts, consider project termination—the strategic goals of the project may no longer be valid. It takes a strong PM to admit that, but I believe it is an ethical obligation.

Fourth Law
The initial requirements for any large system will be incomplete, independent of the resources expended to develop them.

Again, this is where many project slippages are blamed on scope creep, when they should be attributed to inadequate planning. The nature of IT projects, particularly those that involve large or complex systems, is of inevitable change. Accept it, plan for it, learn to embrace it. Include the expected change in your initial risk assessment and rate the impact based upon the size and complexity of the involved systems. Note the phrase "independent of the resources expended to develop them." This means that no matter how hard the project team works, change is still inevitable. Define requirements to the best of the team's ability during planning, but plan on keeping your requirements staff engaged throughout the project.

Fifth Law
Unvalidated requirements pave the road to project failure.

I have seen many IT departments set themselves up for failure by waiting until User Acceptance Testing (UAT) to engage the end user. This is a classic example of unvalidated requirements. I have great respect for members of the International Institute of Business Analysis (IIBA) and holders of the Certified Business Analysis Professional™ (CBAP®) designation. I have performed business analysis tasks on my own projects, and am confident in my own skills, but strongly prefer to have one of these true requirements experts on my team. IT projects in particular need dedicated requirements management staff throughout the entire project, all the way through testing and closure.

Sixth Law
You can't manage what you can't see.

Visibility goes a long way towards empowering project team members to make the right decisions.

If the team doesn't know what they're being measured on, they don't know what the priorities are. If you're tracking against specific milestones or deliverable completion dates, they can align their efforts towards those goals. If you're tracking against targeted vendor performance measures, the team will pay attention to those. It's up to the project manager to create the visibility that will not only help them manage the project, but will help the team's sense of direction as well.

One major caveat: rebaselining, if abused, can reduce visibility. I have seen time and again PMs that use a schedule slip as a reason to rebaseline. This is totally unacceptable, so be sure to implement a solid change management process to prevent this type of misuse.

Seventh Law
Not controlling the right things assures failure.

One of my favorite speakers, Gopal Kapur, taught me this valuable formula:

Percent Complete ≈ Lie ÷ Fantasy

His point was that task progress measured by percent complete is disgracefully inaccurate, and I think this is particularly true in IT projects. Every person I know defines percent complete differently. How reliable can a metric be if no one can agree on how it should be calculated? Yet, because so many of us are under incredible pressure to deliver in a fixed time frame, and, let's face it, Microsoft Project makes it easy to track by percent complete, this is often the driver for many status reports. Even when the metric is valid, obsessive focus on this or any single aspect of a project potentially blinds the PM to other areas that need attention.

Therefore, it is critical that the PM identify not only what should be controlled in an IT project, but the most effective ways to measure those items. Best practices are an excellent way to begin. I think risk and requirements management are particularly helpful for IT projects, since both of these will point to critical success factors more often than not.

Eighth Law
Poor defect management causes high rework and leads to project failure.

Defect management takes on a life of its own in IT projects, and the path to success begins with effective requirements management (see the Fourth and Fifth Laws). After all, how can you tell if you have a defect if don't know your requirements? This is why I recommend keeping skilled requirements staff engaged until the end of the project—so they can work hand in hand with testing staff to help define appropriate test outcomes, clarify any unexpected product functionality issues that arise during testing, and assist the customer during UAT. A knowledgeable business analyst comes in very handy when a customer attempts to add scope during testing, especially when they have done a thorough job of requirement validation earlier in the project.

Unfortunately, many organizations give short shrift to the quality assurance discipline. Since it is at the end of the project schedule, there is tremendous pressure on the QA team to deliver. I have seen environments in which zero time is allocated to redevelopment, and every defect identified during testing is viewed as an unnecessary delays caused by the QA team (rather than the developer who delivered flawed work.) QA should be the final check to make sure everything works end to end and that new code didn't break old code; it should definitely not be the first time code is run through its paces.

Insist on unit testing during development, add a code review task, require vendors to demonstrate interface testing success prior to code delivery; these measures and more will reduce your risk of identifying defects at the last minute. Assess the risk that QA will

find defects in the product they receive (and they will always find some), and build rework time in accordingly. Finally, make sure that QA findings are appropriately categorized. Not every negative finding is a defect; other findings might include usability improvement recommendations, possible security concerns, customer enhancement requests, etc. Findings that are not true defects may be scheduled for a future release, or presented to the customer as a potential scope change.

Ninth Law
Unknown and untreated vulnerabilities originating in ineffectually implemented processes destroy IT projects.

Businesses sometimes consider technology a silver bullet to solve inherent business problems. What they fail to consider is that technology will never, let me repeat that, **never**, be successfully implemented without effective supporting business processes. What good is a new payroll system if no one knows how to use it, or worse, it is used with flawed processes to calculate checks incorrectly? Have you ever had an executive go to lunch with a sales rep and come back excited about a product that was going to "revolutionalize the way we do business?" Often that statement makes PMs cringe because innovation projects can be very exciting or very frustrating.

I'm not saying new technology is bad; just that its introduction requires effective change management. An organization that is immature in change management will be particularly challenged when tackling innovation. Add this to your risk assessment and build in extra time for process development. If at all possible, test new or revised processes in parallel to testing the new technology, or at the very least, use the training material during UAT to identify potential communication issues. Pilots and planned process revisions prior to full rollout are useful to ensure that new processes are fine-tuned, and therefore more readily accepted.

Tenth Law
Development Contractors will do what is in their financial interest, and government organizations may be led toward a project Event Horizon.

Vendor management is often very difficult for PMs, simply because they don't always get to negotiate their own contracts. The key to motivating vendors is to learn everything you can about their contractual obligations, chances (or hopes) for future business with your company or others in your industry, and what drives their costs. Then use this information every chance you get.

Vendor contracts should always be negotiated with an eye on what could happen during the actual project, including incentives for meeting critical project objectives and penalties for the reverse. IT service vendors, especially software development firms, love to charge by the hour. If the project takes longer than expected, they simply bill you more. Avoid this by using "not to exceed" clauses or performance awards for early delivery. Scope expansion is a vendor's dream; it can be a license to print money and drag the project out indefinitely. Try to include well-defined deliverables (screen samples and everything) in your contracts, even if you have to pay a small amount up front for requirements validation. It will save you a fortune in the long run. Then require that the vendor get your sign off on any scope change before beginning work.
If your vendor will be passing along costs for hardware, software licenses, or ongoing support, negotiate this according to your most beneficial strategy - it might make sense to roll this into a strategic sourcing contract, or you might want to include an increased cost-plus percentage bonus in the hardware contract for rapid delivery on the development work.

If you inherit a pre-negotiated contract, attempt to establish these principles as ground rules in an informal written agreement at project kick-off. Position it as a win-win proposition so that you and the

vendor can both look like heroes at project delivery and the vendor is likely to agree. Remember that they want more business, and their cooperation on this project can be a feather in their cap for the next.

Eleventh Law
Thoughtful, knowledgeable, committed people operating as a team are critical to IT Project Success.

Earlier, I mentioned involving your team in brainstorming if the project starts to go south. I think we as PMs sometimes think we have to stay in "management" mode all the time. I tell my teams that my job is to deal with the crap so they can do what they do best; to remove roadblocks, educate the executive team, work with the customer, iron out issues with the vendors, etc. A developer that I have worked with recently wrote this in a recommendation for me:

> *"I have worked with Jennifer on multiple projects and one thing you can always count on from her is that she will make your needs in her project her main focus for resolution. "*

Not only was I incredibly flattered, I was thrilled that I had succeeded in making this team member feel valued. This is how it should be.

Recognize your team as often as possible, and publicly! During the project, say please and thank you. Drop off little thank you notes on people's desks at the end of a rough week. Use any recognition tools your employer provides, from employee of the month to cash bonuses. At the end of the project, send functional managers notes about outstanding performers to include in their annual reviews. Remember that a good PM takes all of the blame and none of the credit.

In summary, the Center for Program Transformation is on the right track. Federal programs need help, and those of us managing private sector IT projects can learn from the IT Laws of Physics. Most importantly, understand that IT projects face unique risks that must be carefully managed throughout the project. You can reduce the degree of risk by minimizing complexity, practicing continuous planning, and closely managing requirements. Implementing careful methods for project monitoring, defect detection/resolution, vendor motivation, and team member recognition are also key to IT project success.

Sources:

Brown, N., Bulkin, D. A., & McGrath, F. (2008). The IT Laws of Physics—Baseline 2.0. Retrieved November 21, 2008, from The Center for Program Transformation Web site: http://uscpt.net/CPT_IT_LawsOfPhysics_Gov.aspx

Kapur, G. (2007). How to Be Productively Disruptive in a Crazy World (podcast). Retrieved November 21, 2008, from the PMI Sacramento Valley Chapter Web site: http://www.pmi-svcast.com/web/?p=6

Powner, D. (2008). Information Technology: OMB and Agencies Need to Improve Planning, Management, and Oversight of Projects Totaling Billions of Dollars. Retrieved November 21, 2008, from the GAO Web site: http://www.gao.gov/new.items/d081051t.pdf

Jennifer Carlisle, PMP, is serving her third term as PMI Dallas Chapter's VP Communications, and previously served the Chapter as Registered Company Coordinator for GE Medical Systems. In addition to her PMP certification, she holds a Green Belt in Six Sigma. She has over 20 years experience in project, program and portfolio management in IT governance and compliance in industries such as healthcare, financial services, hospitality, and direct marketing. Jennifer is a Lead Editor for *Project Landscape*, and has spoken at several regional and national events. She can be reached at jen@jcarlisle.us.