

Mission Critical Project Management

Vince Kellen

Vice President, Information Services

Instructor, School of CTI

DePaul University

November, 2004

Agenda

- Mission critical
- The conventional view is incomplete
- What minimizes risk?
- What to control
- How to motivate

Thoughts on mission critical

- The price for failure is high for the organization
 - Public awareness, lost market opportunity, business continuity loss, customer or shareholder confidence loss
- The effect on people's careers is high
 - Loss of management confidence, loss of personal confidence, career ending, firing possibility
- Factors that influence the level of risk
 - Tolerance of risk in the organization
 - Personal trust and commitment between employees
 - Level of collaboration
 - How can you tell when someone is collaborating?

What is project management's role?

Failure is not an option!

Are factors making the risk too risky?

- Is it a death march?
 - There will be casualties on a death march
- If so, have a resignation letter always on hand and let's begin...

Pre-conditions

1. The skill and knowledge in and the intestinal fortitude to endure risky project management must be sufficient in all project leaders and project managers. If not, don't begin
2. Somewhere, some executive management must want this to succeed and must make that known to you
3. You must have decent project monitoring tools and processes in place and understood

If you don't have the preconditions

- Don't begin the major project
- Instead, get the troops “battle-ready” with small skirmishes fought with spirit
 - Make these small projects *very* real
- Train people of project management concepts, tools and techniques
- Focus on “combat compatibility”
 - Can teams work well under pressure?

Good-to-have conditions

- A level of cross-department collaboration
 - If you have it, you are well prepared
 - If not, you can proceed, but put on a flack jacket
- Flexibility to change roles and personnel midstream, for whatever reason
- Ability to give people rest at the middle or end of project
 - Take advantage of ebb and flow
- Some flexibility somewhere
 - Speed of delivery, cost, quality
 - You can only pick two of the three!
- An IT organizational structure suited for project work

The ideal project manager

- Key traits
 - Insanely detailed, a hyper-communicator, quick as lightning, stubborn as can be, generally friendly, paranoid and distrustful but doesn't show it, very very smart about the technology, thinks like a user, shareholder, CEO and a technician, optimistic, would command God himself to stick to the project plan, understanding spouse.
 - Combination of serial-killer and saint.
- Skills
 - Would like to see a minimum of 2+ years engaged in significant thought about and experience in project management tools, techniques

There is but one concept to understand

- A mission critical project is usually mission critical because the chance of failure is high. While the decision to repaint all the delivery trucks for UPS could be a mission-critical decision, it might not be a mission critical project.
- What makes a mission-critical project risky is simple:

The Unknown

Let's dig into the unknown

- What is unknown about technology?
 - Actually, very little! Given enough time, one could know everything about anything with technology. Computers don't lie. So, it's not that the issues are unknowable, its that the time frame to know all the details is too short.
- *So what do we do about this?*

Know the unknown as fast as you can!

Ways to know the unknown

1. Go for the jugular

- Aggressively seek out the absolutely riskiest parts of a technology project as soon as you can or even before you can. Through architecture planning, you can proactively seek knowledge before it is needed for a project
- DO NOT COMPROMISE ON THIS

2. Think a hard, deep thought for a long time

- Most technical people give up 1/5 of the way through solving a deep problem
- The single hardest thing for most people to do is to think a hard thought for a very long time. Work at it!
- Think sustained deep thoughts early and often

Ways to know the unknown

1. Use the fastest thinking medium
 - Work in a single mind first, then several minds, then on paper, then with models
 - Build toy prototypes that prove the difficult concepts
2. Push people
 - Push them further in creative thinking, deeper thinking, boldly going where they have not gone before
 - The more they think about a difficult problem, the more at ease they are with it
3. Use the project plan to force thinking deep thoughts
 - Get people to develop detailed, meaty, *real* project plans early. Test these project plans early. Challenge early. Enforce discipline early

Believe it or not...

- Mastering the unknown is 80% of the problem
 - It alleviates stress, improves collaboration because trust increases, it cures the common cold and prevents grey hair
- For this reason, a project plan is probably better as an illuminating tool rather than a controlling tool
- The other 20% involves monitoring, adjusting, reallocating resources, gap filling, reporting negotiating, etc.

Project risk management

- Early on aggressively write down those things that can cause the project to run amok
- For each risk, write down the steps you will take to mitigate them, the date you want the risks to be mitigated and the means by which you will know mitigation has occurred
- Discuss risks regularly, but do so in a way that turns the worry (an unknown) into a known through conscious action
- The world is full of nay-sayers. A good PM puts them to work managing risk or puts a box around them

Project planning and estimating

- The basics:
 - activities, start date, duration, resources assigned, allocation of time, predecessor activity, resource costs, vacation and holiday schedules
- The conventional approach spends a lot of time modeling dependencies. In fact it spends a lot of time modeling everything under the notion that it must in order to predict an end date. It has an assumption that one can and should have perfect knowledge
- I have issues with this

Project planning issues

- Just because you can add more elements to your model doesn't mean you should. Using Occam's razor as counsel, one should use the least amount of information for successful project modeling
- The tension between
 - Complete knowledge versus partial knowledge
 - Causal reasoning versus correlated reasoning
 - Logic versus statistics

My project planning philosophy

- Ignore dependencies
 - Why? Too many errors in dependency reasoning to make it worthwhile. Small changes in logic creates huge differences in end dates. Dependencies come and go as people learn more and change their designs or approaches. Dependencies can be negotiated or worked around. Dependencies require too much knowledge in the wrong hands.
 - Instead, model a small number of dependencies in the form of *project phases*

My project planning philosophy

- Projects are nothing more than the sum of objects. Objects are tangible things that three people can get together and agree to score the object as “simple,” “normal,” or “complex.” Objects are listed in the project charter as those things being delivered that demonstrate delivery
- Tasks (or activities) are units of work done by resources (people) to deliver an object. Objects may have many tasks.
- Tasks are assigned an owner who takes responsibility for estimating, monitoring and completing that task, even if multiple people work on the task

Task philosophy

- Tasks are independently estimated by those who know the object the best
 - Average task size of 8-16 hours, with 4-8 as ideal.
 - Estimates are 50/50 estimates. Not padded, not aggressive
- Tasks are assigned to an object and a phase, have a start date and an end date
- All IS staff enter time daily against tasks

What does this mean?

- Projects will have between 50 and a few thousand line items (tasks). Each task is about a days unit of work.
- Tasks are associated with objects and objects comprehensively describe a project
- Tasks have as much chance of going over budget as they do in going under budget
- Errors in estimate self-correct. This leads to the funny conclusion that statistically speaking, larger projects are easier to manage
- We are “statistically” managing project performance, not managing through manipulation of causal logic
- Actual project decisions get pushed further down to the teams

What do we do and NOT do?

- We do not manage dependencies
- Project managers and team leaders are given a block of hours to complete the work and have total freedom to rearrange those hours if it helps finish the work on time
- PMs do not spend time modeling fast-moving dependencies
- Staff is incented to “play it up the middle” when entering time. It’s a zero-sum system. Can’t beat it
- We don’t sweat the cost side of human capital. Its not really manageable at the level anyway
- We look for signs of troubled projects

Signs a project is in trouble

- Total hours that should have been spent to date is getting awfully close to total hours actually spent, or is over
- Variation in actual to estimate difference is high
- Average task size is high
- Tasks that are completed have been largely over budget
- Tasks in progress are over budget
- Remaining hours left in the project exceeds calendar time

Intervention!

- Zero in on troubled tasks, troubled objects
- Look for design problems or redesign
- Consciously go OVER budget in design more with the expressed goal of going UNDER budget in implementation
- Review project managers that can't produce regular numbers. Experience has shown that project plans that produce regular numbers usually go smoother. (Zero-sum game, remember)
- Build out the risk mitigation plan and act on it
- You should know 1/3 to 1/2 of the way through a project where it's going to end up