How to keep your projects afloat

JOE'S PROJECT RUNS INTO A MINOR SNAG...
CANCEL YOUR WEEKEND PLANS, BOYS. THE BOSS WANTS HER SEAWORTHY BY THE END OF JULY.

A presentation by Jenny and Andrew

for PMI Mass Bay Chapter

September 20, 2007

Who we are...

Jenny's spent the last 15 years or so working in software quality

Andrew started programming in the 80s, and lost count of how many languages he's worked with.

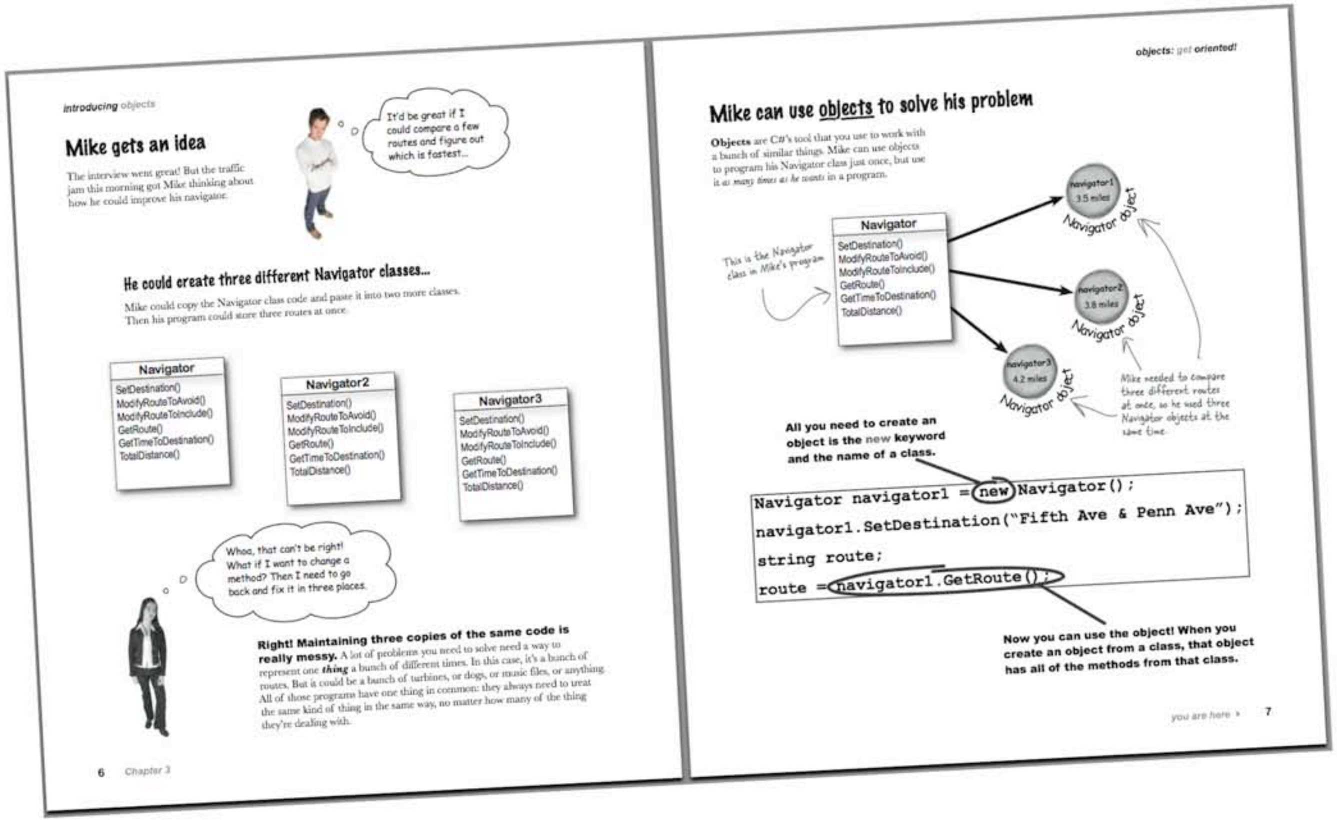
She's currently running a large distributed development team for an educational software company

He's led teams of programmers, requirements analysts and process engineers.

Jenny and Andrew truly believe that with better development practices and good project management habits, we can all build better software.

What's keeping us busy right now...

Our current project is **Head First C#**, the next big release in the acclaimed "Head First" series from O'Reilly. It should be in bookstores in November.



introducing objects

Mike gets an idea

The interview went great! But the traffic jam this morning got Mike thinking about how he could improve his navigator.

It'd be great if I could compare a few routes and figure out which is fastest...

He could create three different Navigator classes...

Mike could copy the Navigator class code and paste it into two more classes. Then his program could store three routes at once.

Navigator
SetDestination()
ModifyRouteToAvoid()
ModifyRouteToInclude()
GetRoute()
GetTimeToDestination()
TotalDistance()

Navigator2
SetDestination()
ModifyRouteToAvoid()
ModifyRouteToInclude()
GetRoute()
GetTimeToDestination()
TotalDistance()

Navigator3
SetDestination()
ModifyRouteToAvoid()
ModifyRouteToInclude()
GetRoute()
GetTimeToDestination()
TotalDistance()

Whoa, that can't be right! What if I want to change a method? Then I need to go back and fix it in three places.

Right! Maintaining three copies of the same code is really messy. A lot of problems you need to solve need a way to represent one ***thing*** a bunch of different times. In this case, it's a bunch of routes. But is could be a bunch of turbines, or dogs, or music files, or anything. All of those programs have one thing in common: they always need to treat the same kind of thing in the same way, no matter how many of the thing they're dealing with.

6 Chapter 3

objects: get oriented!

Mike can use objects to solve his problem

Objects are C#'s tool that you use to work with a bunch of similar things. Mike can use objects to program his Navigator class just once, but use it *as many times as he wants* in a program.

This is the Navigator class in Mike's program

Navigator
SetDestination()
ModifyRouteToAvoid()
ModifyRouteToInclude()
GetRoute()
GetTimeToDestination()
TotalDistance()

Mike needed to compare three different routes at once, so he used three Navigator objects at the same time.

All you need to create an object is the new keyword and the name of a class.

```
Navigator navigator1 = new Navigator();
navigator1.SetDestination("Fifth Ave & Penn Ave");
string route;
route = navigator1.GetRoute();
```

Now you can use the object! When you create an object from a class, that object has all of the methods from that class.

you are here › 7

Not all failures are easy to spot...

...but some projects do fail spectacularly.

The Tacoma Narrows Bridge project failed before the first yard of concrete was poured.

There was nothing wrong with the construction. Poor design and badly planned cost cutting in materials led to an unfortunate end.

"This time it's different..."

There's an old saying about how there are a million ways to fail, but only one way to be right. When it comes to projects, nothing's further from the truth. Projects fail the same few ways over and over again.

Don't go in the basement!

Software projects are a lot like cheesy horror movies. After you've seen a few of them, you know that the first guy to leave the group is going to get an axe in his head. Projects are the same way. People keep making the same mistakes over and over, and it keeps getting their projects killed.

You know you're on a failed project when...

A judge in 1964 said, "I don't know how to define pornography, but I know it when I see it." And the same goes for failing projects - we all know when we're on one that's sinking.

What does a failing project look like?

You know your project failed if it got aborted and everyone was laid off. But there are other, less obvious kinds of failure:

- ★The project costs a lot more than it should.
- ★It takes a lot longer than anyone expected.
- ★The product doesn't do what it was supposed to.
- ★Nobody is happy about it.

Sometimes failure seems normal

Nobody sets out to fail, but for some reason people just accept that a lot of software projects won't deliver on time, under budget with the expected scope intact. But talking about what causes failure makes people uncomfortable, because nobody wants to give or take that kind of criticism.

A show of hands, please...

We've never met a single professional software engineer with more than a few years of experience who hasn't been on at least one failed project.

Are there any here?

Four basic ways projects can fail

There are plenty of ways that you can categorize failed projects. We like to think of them like this:

- ★ **Things the software does (or doesn't do)**
  How your project doesn't quite meet the needs of the people you built it for
- ★ **Things the team should've done**
  Once in a while, it really *is* the team's fault
- ★ **Things that could have been caught**
  …but weren't until it was way too late.
- ★ **Things the boss does**
  Classic management mistakes that can damage the project

Things the software does (or doesn't do)

It seems pretty obvious that you should know what the software's supposed to do *before* you start building it... not that that stops us.

- ★We only find serious problems after we've built them into the software
- ★We have big, useless meetings that fail to figure out what the software's supposed to do
- ★Scope creep
- ★90% done, with 90% left to go.

IT'S NOT THAT I DON'T LOVE YOUR PORK CHOPS, IT'S JUST THAT I WANTED MEATLOAF.
ONE MORE COMMENT LIKE THAT AND I'M STICKING THIS KNIFE IN HIS EYE.

Things the team should've done

The team could have done the work more efficiently, if only we'd taken the time to think it through.

- ★ Padded estimates compensate for unknowns.
- ★ Project teams will just pick a deadline and stick to it, no matter what basic reason and common sense tell them.
- ★ Somehow non-programming tasks always seem to get cut when the deadline gets closer.
- ★ Misunderstood predecessors lead to cascading delays.

I RAN OUT OF TIME, SO I DIDN'T FINISH BAKING IT. HOPE YOU DON'T MIND A LITTLE SALMONELLA.

Things that could have been caught

Which would you choose: a well-built program that doesn't do what you need or a crappy one that's irritating to use and does?

- ★ Getting a few tech support people to "bang on the software" is ***not*** testing.
- ★ Maybe we could've caught that design problem *before* the code was built.
- ★ Maybe we could've caught that code problem *before* we went to test.
- ★ "Beta" does not mean "use at your own risk."

WHY YES, IT DOES SAY THAT YOU HAVE SCABIES. BUT REALLY, IT'S JUST A BUNCH OF TESTS, RIGHT?

Things the boss does

Some problems start with senior managers, others start with ***us PMs***. But they can all sink the project.

- ★Unmanaged changes
- ★Micromanagement
- ★Over-reliance on gut instincts
- ★Tunnel vision
- ★An artificial "wall" that the business puts up to disconnect from the engineering team

THE TEAM SAID IT'D TAKE EIGHT WEEKS. I'LL PROMISE IT IN FOUR - THAT'LL LIGHT A FIRE UNDER THEIR ASSES!

What you can do about it

Some easy ways to make sure your project doesn't fail:

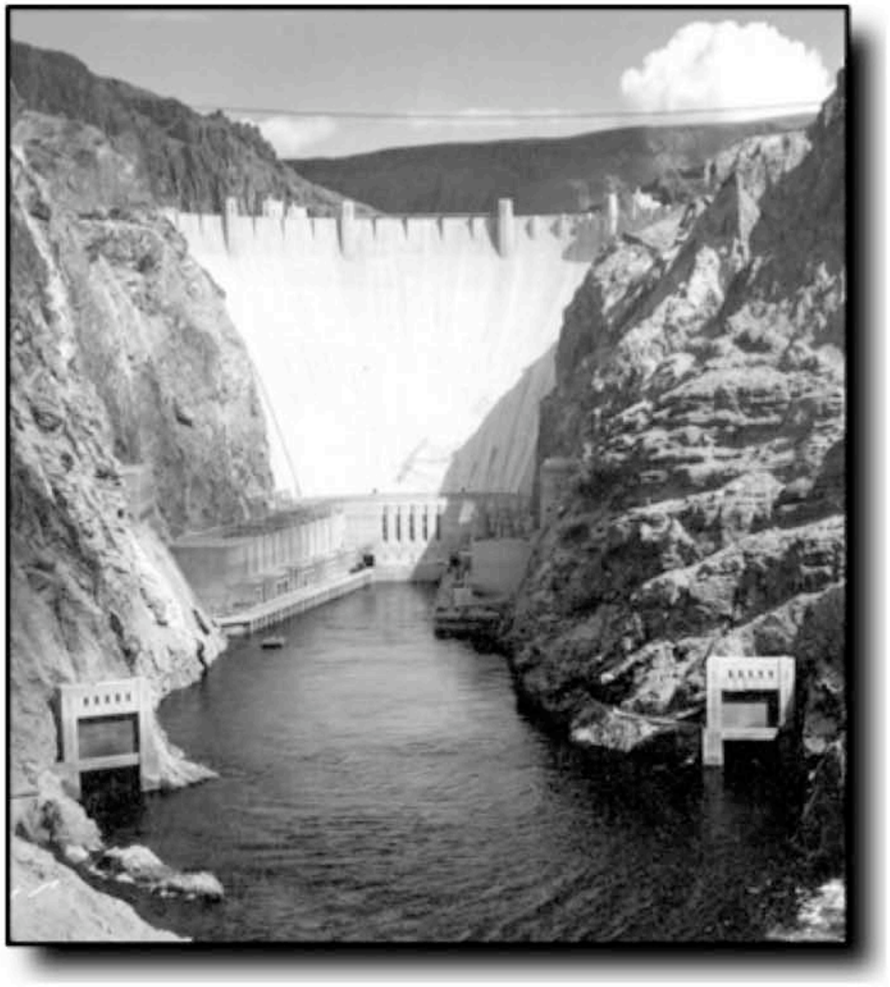
- ★ Tell the truth all the time
- ★ Trust your team
- ★ Review everything, test everything
- ★ All developers are created equal
- ★ The fastest way through the project is the right way through the project

The talent is there... the project management's not.

Hoover Dam was finished two years early, and under budget. Software's not so different that we can't engineer it just as well.

- ★ Our problems have, for the most part, been solved.
- ★ Over and over and over again. Seriously.
- ★ We just have to stop ignoring the solutions.
- ★ Also, hire awesome consultants who know what they're doing and have solved these problems before.
- ★ (us.)

Do you think your project will take more effort than this one?



Repeat after us: "Practices, practices, practices."

Every single one of the problems we've talked about has a time-tested solution, usually one that's not too hard to implement.

- ★ Manage your changes with a good change control system.
- ★ Build trust between the team and the manager using a repeatable estimation process.
- ★ There are all sorts of reviews that can help improve quality.
- ★ Scope management helps you make sure everyone knows what will (and won't) be built

JOE PERFECTS HIS PROCESS FOR THE PERFECT DATE.
HE'S SO DREAMY!
STEP 37: LEAN IN AT A 48 DEGREE ANGLE

Software projects need software practices

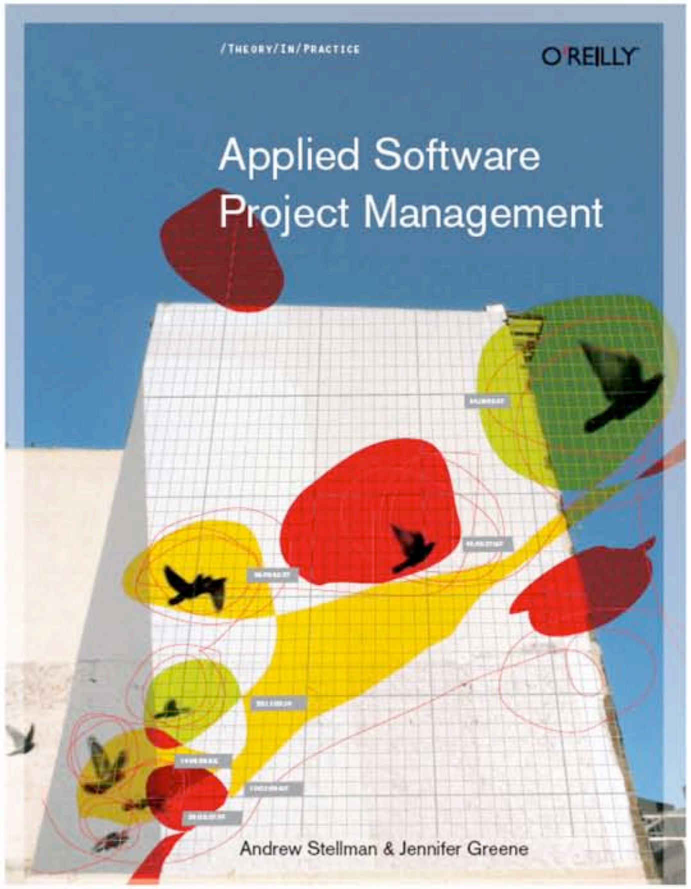
There are a bunch of great practices tailored for software.

- ★ Use cases are a simple and effective requirements tool
- ★ Test-driven development and refactoring improve quality by delivering better code.
- ★ Continuous integration and integration testing make sure all the pieces work together.
- ★ Code reviews and pair programming keep the bugs out of the software.
- ★ Version control makes sure everyone's working on the right file.

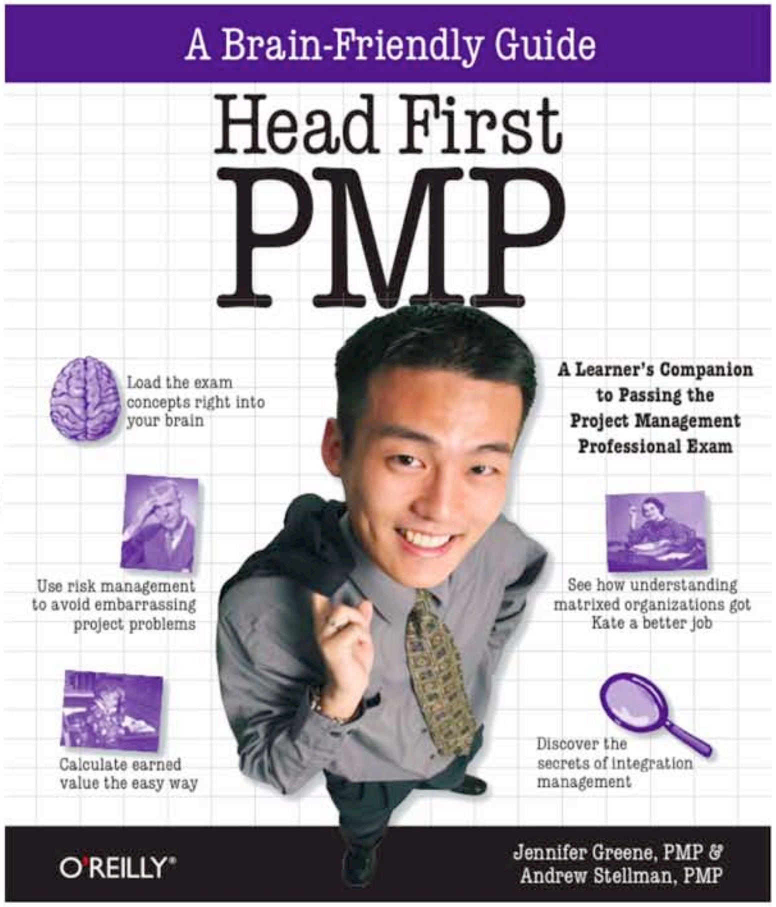
Super high tech equipment can't help you save a sinking project

Photo by Laughing Squid (CC license)

One last quick note from the O'Reilly marketing department


/THEORY/IN/PRACTICE
O'REILLY
Applied Software Project Management
Andrew Stellman & Jennifer Greene

Buy these books


A Brain-Friendly Guide
Head First PMP
A Learner's Companion to Passing the Project Management Professional Exam
Load the exam concepts right into your brain
Use risk management to avoid embarrassing project problems
See how understanding matrixed organizations got Kate a better job
Calculate earned value the easy way
Discover the secrets of integration management
O'REILLY
Jennifer Greene, PMP & Andrew Stellman, PMP

And check out our blog, "Building Better Software"

http://www.stellman-greene.com/

We'll post these slides on the site over the weekend!