



**Michael James**  
Michael James is a software process mentor and Certified Scrum Trainer, focusing on the engineering practices that enable Agile project management. Having worked in the software industry for more than 20 years as a software developer (formerly “architect”), he has experience in automated testing that predates the Extreme Programming movement; formal, phased, high-ceremony processes based on DOD-STD-2167A; chaotic non-processes of the dot-com era; and Agile processes including Scrum and XP.

# Flattening the Cost of Change Curve: Theory vs. Practice

Posted by Michael James on February 19, 2008 at 10:16 p.m.

Modern languages combined with the engineering practices derived from eXtreme Programming (Test Driven Development, constant refactoring, continuous integration, etc.) make it possible to flatten out the pessimistic exponential “cost of change curve” we learned in the early days.

The ability to \*do\* this involves learning skills and habits your team may not have yet.

There are a few areas in which no one’s learned how to flatten the cost of change curve completely, especially post-release.

Examples of things Danube has found harder to change (especially post-release):

1. Database schemas
2. APIs others have started depending on
3. UIs your users have become accustomed to
4. Application servers (JBoss, WebSphere, etc.)
5. Programming languages

These things can (and often should) be changed in the future, but the cost will be higher than a design change in a well written object-oriented program that has automated tests. Perhaps this shows limitations of our skills as an industry.

(The word *ScrumMaster* itself is another example. Many of us wish we’d chosen a term like *Scrum coach* or *Scrum facilitator* to more clearly convey that the ScrumMaster isn’t the team’s boss. But changing the term now would only increase confusion.)

This isn’t an excuse to dwell forever on future-proofing things like the database schema. There’s a reasonable balance between due diligence and analysis paralysis<sup>1</sup>. You can release a migrator with your next version. Sure there’s some rework, but rework is cheaper than no work.

--mj

**danube**

Danube Technologies, Inc.  
520 SW 6th Ave, Suite 940  
Portland, OR 97204 USA  
1.888.5.danube  
T: +1.503.248.0800  
F: +1.503.248.0801

---

<sup>1</sup> [http://danube.com/blog/michaeljames/analysis\\_paralysis](http://danube.com/blog/michaeljames/analysis_paralysis)