

The Embedded Muse 64

Editor: Jack Ganssle (jack@ganssle.com)

August 25, 2001

An Interesting Book

Software engineering is a field that seems to proceed in fits and starts. Most of us write code the same way we did back in college, though occasionally a new approach does come along. I'd count Fagin Inspections as one, OOP, another.

In the last couple of years, though, Kent Beck's Extreme Programming (XP) has surfaced as another interesting approach to writing code. And **code** is the operative word. XP starts with the requirements in the form of user stories. The customers deliver and prioritize the user stories. The developers analyze the stories and write tests for them.

Everything ends with code. The code is developed in pairs of developers to increase quality. Quality code is the goal, and that's obtained by constantly rewriting it (refactoring in XP lingo), pair programming so two pairs of eyes look at it all, and constant testing/integration. The output is clean, clear code that fulfills the customer's wishes, with no extra frills or hooks for extensibility.

One book that does a great job of describing XP is Kent Beck's Extreme Programming Explained (ISBN 201616416), a slender but complete \$29.95 volume.

I sometimes find these sorts of books tiresome. An evangelist pushes what some might see as a wild-eyed new way to create software, while the evening wears on and my interest flags. This one is different. Between the writing style and the quite fascinating ideas behind XP I found the book compelling.

XP requires a customer who lives on-site, constantly providing feedback to the development team. A very cool idea. Practical? I have doubts, especially in the embedded world where so many of us build products used by thousands of disparate customers. But a cool idea nonetheless.

XP demands conformance to a coding standard. Excellent! The pair programming I'd find a little too "in your face", but is an interesting concept that builds on the often-proven benefits of code inspections, though in my experience two pairs of eyes are not enough.

Copyright 2001 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

XP teams focus on validation of the software at all times. Programmers develop software by writing tests first, then software that fulfills the requirements reflected in the tests. Customers provide acceptance tests that enable them to be certain that the features they need are provided. There's no such thing as a big integration phase. This is the XP practice I find most brilliant. Even if you're not going to pursue XP, study it and take the testing ideas to heart.

Constant testing plays into the "frequent releases" XP requirement. Don't build a big thing and then dump it on a customer. Release portions of the project often to get feedback. This is much like the spiral development model, which seems to offer the only practical hope to meet schedules. Of course, neither spiral nor XP development promises that we'll know a real delivery time at the project's outset; instead, we evolve the product and schedule together. Most managers can't accept that.

Another interesting book is *Extreme Programming Examined* by Giancarlo Succi, Michele Marchesi (ISBN 0201710404). It's much less readable, being an uneven collection of talks presented at a conference. The most interesting papers talk about failures implementing XP.

Finally, I'd imagine most of us would quickly buy in to XP's belief in 40 hour work weeks. Tired programmers make mistakes. Go home!

Thought for the Week

To balance the previous words about writing great code, do check out "How to Write Unmaintainable Code", by Roedy Green with contributions from many others, at <http://www.mindprod.com/unmain.html>.

One of my favorites: "Since the computer ignores comments and documentation, you can lie outrageously and do everything in your power to befuddle the poor maintenance programmer."

About The Embedded Muse

The Embedded Muse is an occasional newsletter sent via email by Jack Ganssle. Send complaints, comments, and contributions to him at jack@ganssle.com.

To subscribe, send a message to majordomo@ganssle.com, with the words "subscribe embedded *your-email-address*" in the body. To unsubscribe, change the message to "unsubscribe embedded *your-email-address*".

Copyright 2001 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

The Embedded Muse is supported by The Ganssle Group, whose mission is to help embedded folks get better products to market faster. We offer seminars at your site offering hard-hitting ideas - and action - you can take now to ***improve firmware quality and decrease development time***. Contact us at info@ganssle.com for more information.

Copyright 2001 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

The Ganssle Group, www.ganssle.com