

# PragPub

The First Iteration

IN THIS ISSUE

- \* Agile Flash Cards
- \* The Key to Better Software
- \* Building Trust
- \* The First Robot



## Contents

### FEATURES



#### **Agile Flash Cards** ..... 9

by Jeff Langr, Tim Ottinger

For the past year and a half, Jeff and Tim have been capturing the wisdom of the Agile community and their own combined 20+ years of Agile and XP experience on index cards. In anticipation of the release of these cards in decks, they share a few Agile in a Flash cards with you here.



#### **The Key to Better Software** ..... 15

by Brian Tarbox

I've become convinced that the approach to quality taken by most companies is misguided and based on a flawed model. This is the reason why so many companies are having such trouble increasing the quality of the software they deliver.



#### **Building Trust** ..... 17

by Rachel Davies

When you're coaching Agile teams, it's essential to work on building trust from the get-go. Without trust, Agile simply doesn't work. So how can we build trust on our teams so that Agile practices can take root?



#### **The First Robot** ..... 20

by Dan Wohlbruck

On September 20, 1928, Eric, the first human-looking robot, opened the Exhibition of the Society of Model Engineers in London. The inventors were inspired to build Eric when the Duke of York was unable to attend the ceremony opening the Exhibition. Eric proved to be a simple, successful replacement for the Duke of York.

**Up Front** ..... 1

by Michael Swaine

This month we introduce a new column by Pragmatic Programmer Andy Hunt, titled "Guru Meditations."

**Choice Bits** ..... 3

In Philadelphia, you must buy a \$300 license to blog. Meanwhile, Java is a restricted platform, Google is evil, Apple is a monopoly, Microsoft is the underdog, and jobs outsourced to India are being re-outsourced to unskilled Americans. We netted these and other amazing facts as we trolled the Twitterstream this month.

**Guru Meditation** ..... 6

by Andy Hunt

Context is everything, especially with agile methods.

**The Quiz** ..... 22

by Michael Swaine

I was inspired by Jeff Langr and Tim Ottinger's article on Agile in a Flash to come up with a card-based quiz.

**Calendar** ..... 26

Brendan McAdams, author of the forthcoming MongoDB: A Quick Start Guide, will talk about "From MySQL to MongoDB," "Scala + Mongo for Java Developers who don't know Scala," and "Python Development with MongoDB" at the MongoDB Conference in Boston this month. Also, the GNU Project turns 27. And other items for your calendar.

**Shady Illuminations** ..... 32

by John Shade

When I'm feeling more than usually insignificant, I remind myself of two things that give meaning to all our lives. I think of Schrödinger's cat and anthropogenic global warming. But lately I've been finding both less comforting than I used to.

---

Except where otherwise indicated, entire contents copyright © 2010 The Pragmatic Programmers.

Feel free to distribute this magazine (in whole, and for free) to anyone you want. However, you may not sell this magazine or its content, nor extract and use more than a paragraph of content in some other publication without our permission.

Published monthly in PDF, mobi, and epub formats by The Pragmatic Programmers, LLC, Dallas, TX, and Raleigh, NC. E-Mail [support@pragprog.com](mailto:support@pragprog.com), phone +1-800-699-7764. The editor is Michael Swaine ([michael@pragprog.com](mailto:michael@pragprog.com)). Visit us at <http://pragprog.com> for the lowdown on our books, screencasts, training, forums, and more.

ISSN: 1948-3562

# Up Front

## A Guru Meditation and Several Other Thoughts

by Michael Swaine



This month we introduce a new column by Pragmatic Programmer Andy Hunt, titled “Guru Meditations.” I’m pretty sure you’re going to like it a lot.

As the editor here, I have the tedious duty of making sure that nobody misses the in-jokes and obscure references. So let me give you this image



and [this link](#)<sup>[1]</sup>. That should suffice.

I’ve allocated myself a little extra space this issue to talk with you about what’s coming up in *PragPub*. You won’t want to miss our December issue, when we’ll focus on Ruby. And if you’ve ever considered writing a book, or even an article, you really owe it to yourself to catch our November issue on becoming a writer. I have no idea what will be in the October issue. I’d explain that to you, but we’d be getting into the integral calculus of publishing schedules, and I’ve already bored you once.

Back to the present. To the issue that you have in your hands—so to speak. It’s a little disorienting to think that you may be reading this on our website or as a pdf file that looks something like a print magazine or in an appropriate format on your cellphone or Kindle. But however you’re reading it, it’s the content that matters. So...

In this issue you’ll find articles by Agile coaching guru Rachel Davies, Agile and XP experts Jeff Langr and Tim Ottinger, Motorola Distinguished Member of Technical Staff Brian Tarbox, and computer historian Dan Wohlbruck. They share their expertise in building trust in Agile teams, developing Agile flash cards, the need for continuing education in software companies, and the history of robotics.

Plus, of course, our usual events calendar, choice bits from the twitterstream, quiz, and off-center insights from John Shade. I think you’ll like this issue.

Okay, this next part is personal.

I’ve been an editor of tech magazines for about 30 years now, and I’ve never been more impressed with the caliber of writers I’ve been privileged to work with than I am with the programmer/writers who contribute to *PragPub*. And I say that as someone who’s edited tech writers from Margaret Wozniak’s son to Courtney Love’s father. Over 50 bright, witty, and generous authors—so

far—have contributed to making *PragPub* the worthwhile and entertaining resource that I like to think it is.

These writers meet the deadlines and put up with the edits I make to their words and come through with equanimity when I give them 24 hours to read the final proofs and then say, oh by the way, I need a 50-line bio and a picture immediately. And they do it for, as I phrase it, payment in fame and gratitude. They deserve more than they get of both.

So allow me to thank Chris Adamson, Jorge Aranda, Kent Beck, David Bock, Tony Bove, Daniel Brolund, Ed Burnette, Paul Butcher, Frederic Daoud, James Duncan Davidson, Rachel Davies, Scott Davis, Ian Dees, Bill Dudney, Ola Ellnsta, Chad Fowler, Eric T. Freeman, Adam Goucher, James Edward Gray II, Phil Haack, Stuart Halloway, Dan Haywood, Rich Hickey, Brian Hogan, Rob Holland, Dave Hoover, Mike Hostetler, Jason Huggins, Andy Hunt, John Jainschigg, David Koelle, Jeff Langr, Andy Lester, David McClintock, Jonathan McCracken, Staffan Nöteberg, Tim Ottinger, Alan Oppenheimer, Paolo Perrotta, Steve Peter, Noel Rappin, Jonathan Rasmusson, Craig Riecke, Johanna Rothman, Ben Scofield, Maik Schmidt, Liz Sedley, John Shade, Eric Smith, Daniel Steinberg, Brian Tarbox, Bruce Tate, Mike Taylor, Dave Thomas, Dan Wohlbruck, and Marcus S. Zarra.

Thank you for what you do.

**External resources referenced in this article:**

<sup>[1]</sup> [http://en.wikipedia.org/wiki/Guru\\_Meditation](http://en.wikipedia.org/wiki/Guru_Meditation)

# Agile Flash Cards

## Quite Possibly the Most Versatile Tool on the Planet

by Jeff Langr, Tim Ottinger

Your humble editor is a believer. When my significant other was managing editor of *MacUser* magazine, her most powerful management tool was a deck of index cards.



For the past year and a half, we have been creating *Agile in a Flash* index cards in blog form. These cards capture the wisdom of the Agile community plus our combined 20+ years of Agile and XP experience. The Agile community has generated many memorable mnemonics and lists, starting with the four values espoused in the Agile Manifesto, and including such stock phrases as “Red-Green-Refactor.” We’ve added a number of cards that represent our own inventions, such as Tim’s FIRST mnemonic that describes the qualities of a good unit test.

Our goal for the project was to produce a distributable deck of real, physical index cards. Many in the Agile community use index cards heavily to capture stories, later using these cards for planning and communication purposes. Capturing much of the essence of Agile thought in a handy card deck seemed so obvious to us that we’re surprised no one had done it before.

You can use the cards in many ways—as references, reminders, discussion points, posters on the wall, or even paper airplanes acting as air mail. You might even consider using them as the musician Brian Eno did with his [Oblique Strategies card decks](#) [U1] in the 1970s. The idea behind Oblique Strategies was to inject random ideas into the process of creating music, such as “Discover the recipes you are using and abandon them,” or “Go to an extreme, move back to a more comfortable place.” It’s amazing how many of the Oblique Strategies cards can provide useful insights for an Agile team. One card simply states, “Courage!,” an important XP value that mirrors one of our *Agile in a Flash* cards.

The cards in the *Agile in a Flash* deck can similarly work as triggers for new ideas. Struggling with what to tackle during retrospectives? Pull out a random *Agile in a Flash* card. Use the card to trigger discussions around how well your team is adhering to the advice. Then figure out what’s lacking.

## The Deck

In our eighteen months of blogging, we’ve produced almost 90 different cards. In preparation for publication, we carefully chose a deck of the best cards. The deck contains four categories:

- The Idea—cards relating to Agile itself, including its values and principles.
- The Plan—cards around stories, planning, and estimation.
- The Team—cards discussing team practice, such as retrospectives, stand-ups, and pairing.
- The Code—cards covering technical, code-oriented topics, such as TDD, continuous integration, and refactoring.

Each card provides a summary list or diagram on its obverse side and more detailed information—including nuggets of our experience-based wisdom—on the reverse. The nuggets are condensed, mind you: There's not a lot of space on an index card to be wordy!

The Pragmatic Programmers are helping us realize our goal of distribution. We'll share a few *Agile in a Flash* cards with you here. These are meta-cards—cards about the use of cards themselves.

## Why Cards?

### The Value of Index Cards

- Low-tech and high touch
- Dynamic reorganization
- Can hold schema-free markup
- Reminders for much bigger ideas
- Extreme portability
- Inexpensive and replaceable

We love index cards for their versatility as a tool.

Cards are *low-tech and high-touch*. You can put your hands on them, hand them to other people, stick them on the wall, tape them together, sort them on the desktop, etc. Having a tactile device such as an index card can make various workspace rituals possible (like moving a card from “in progress” to “done”).

You can *dynamically reorganize cards* to order them by point cost, functional area, assigned teams, or originators. You can use cards to discover and create connections that no one intended originally. This puts index cards a step above most software programs—you can routinely repurpose them on the fly.

Cards can hold *schema-free markup*. For instance, the security or documentation team can write notes on them. Developers and product owners can mark them up with issues or orderings. You can annotate them with priorities, pictures, or notes with arrows. Or they might be so stark as to contain only a two-word

name. If you have a card and a marker, your only constraints are space and penmanship.

Cards are *reminders for much bigger ideas*. Being small, cards hold very little content. Yet they can have vast evocative powers. Once we discuss a story in depth, a two-word description written on a card is memorable enough to recall the discussions for the next couple weeks as we build the story. Our *Agile in a Flash* deck can also provide this power: Once you know a principle or practice, a card bearing as few as three words can help you do your job better.

As small physical items, *cards are extremely portable*. You don't have to copy them to a USB drive or mail them to your teammates. You can use them on the bus, train, or hiking trail. The cards don't mind being without web access and are immune to OS and browser incompatibilities. They move through airports without setting off metal detectors. Wherever you are, the cards are the same.

*Cards are inexpensive and replaceable*. Of course, you never want to be without your *Agile in a Flash* cards, but you can lose most other cards without any real cost. Tucking cards into jean or shirt pockets can expose the cards to the danger of a washing machine. But if they are important, you can easily reproduce your cards. If not, maybe you didn't really need them (Mom always said, "It must not have been that important!"). Every day we can tear up a story card or two is a good day. Cards are not heirlooms to be maintained for the life of a project, and that makes them even better.



Card

Conversation

Confirmation

Ron Jeffries refers to story cards as tokens. The cards do not provide every last detail to be implemented regarding a customer's needs. Instead, when a development team has availability to work on something, the customer tells them a story about business needs. The team and customer discuss the story in further detail, until the team understands the story well enough to understand its scope and when they might be able to deliver it. The card captures a terse summary of the story, a simple placeholder or token.

The customer and team will need to continue conversations around the story as the iteration or sprint progresses. Often, the customer will omit at-the-time unnecessary detail when they first present the story to the team. The card is a reminder that both parties will have to continue discussions around exactly what should be built.

The specifics of what we're building must ultimately be clear to the customer and the team who will deliver. The customer captures these criteria in acceptance tests, designed to exercise the system in enough ways to demonstrate that the feature really works. When all members of this set of acceptance tests for a given card pass, it confirms to the customer that the story is truly done—the software does what they asked. The nebulously stated card can disappear—the remaining documentation is the system plus its acceptance tests.

# A Story Format

As a ...

(person in a certain role)

I want to ...

(accomplish some goal)

So that ...

(we obtain some biz value)

Mike Cohn popularized this story format in his book *Agile Estimating and Planning*. It provides a simple template that you might consider for how you word the story summaries on your cards. “As an actor, I want to accomplish some goal, for some reason.” For example, “As an administrator, I want to see a list of multiple login failures, so that we can identify potential security breaches.” A story summary written in this form is almost like a condensed use case.

Using the story card template can provide some value. On a simple level, a standard template can help prevent wasteful arguments over formatting and wording. Thinking about the actors involved (the “as a” people) can help trigger the introduction of important stories that you might otherwise miss. The phrase “I want to” reinforces that stories are goals customers want users of the system to be able to accomplish, and not just technical pipe dreams. “So that?” As you write out dozens of stories in release planning, you’ll want to remember the rationale behind certain stories (but not all of them!) And sometimes understanding the “why” can trigger other useful considerations.

But before you insist that every card follow this rigid format, remember that the story card ain’t the thing, it’s a reminder that should stimulate more communication. It could be that the card has just one word on it, and that’s sufficient. During the course of your iteration or sprint, you’re not going to forget who the story is for, nor why it exists. Remember: The story *card* should

disappear once implemented. The story details about what, how, who, and why are best moved to the acceptance tests.

We prefer a more spartan story summary on our card, viewing the boilerplate parts as oh-so-much-duplication that we would immediately stamp out if we found it in our code. But if you like the format, use it. Just make sure you re-assess the worth of your choices from time to time.

## Last Thoughts

The cards we present as part of the *Agile in a Flash* project are tools, not gospel. They should help by prodding you if you get stuck and by giving you ideas providing guidance. In most cases, you should follow their advice unless you have darned good reasons not to. Even then, you should only take your own path once you understand why the advice exists.



### About Jeff

Jeff Langr has been building software for over a quarter century. He is the author of *Agile Java* and *Essential Java Style*, plus more than 80 articles on software development and a couple chapters in Uncle Bob's *Clean Code*. He runs [Langr Software Solutions](http://langrsoft.com)<sup>[U2]</sup> from Colorado Springs and happily builds software as an employee of GeoLearning.



### About Tim

In addition to developing the [Agile in a Flash](#)<sup>[U3]</sup> card deck with Jeff, Tim Ottinger has over 30 years of software development experience including time as an Agile coach, OO trainer, contractor, in-house developer, and even a little team leadership and management. He is also a contributing author to *Clean Code*. He writes code. He likes it.

Send the authors your [feedback](#)<sup>[U4]</sup> or discuss the article in the [magazine forum](#)<sup>[U5]</sup>.

#### External resources referenced in this article:

- [U1] <http://www.rtqe.net/ObliqueStrategies/Edition1-3.html>
- [U2] <http://langrsoft.com>
- [U3] <http://www.pragprog.com/refer/pragpub15/titles/olag/agile-in-a-flash>
- [U4] <mailto:michael@pragprog.com?subject=Agile-cards>
- [U5] <http://forums.pragprog.com/forums/134>