

Five Days of Software Development with Style and Grace

Ron Jeffries / Chet Hendrickson

Denver, June 25-29

Is “Agile” tasting a bit watered down to you these days? Would you like a stiff drink from the bottle under the bar? Then we’re the guys to set you up.

Let’s go and hang out at the Tech Center Marriott in Denver. You’ll join a roomful of people with experience and interest in Agile, where we’ll run through a series of exercises that will show you what we mean when we talk about Agile and XP and what software development should really be like. We’ll work together so that we can all become more successful and can feel when we are stepping off the path that leads to success.

Agile projects rely on iterative development of the software, with continual improvement of the design from small and simple to larger and more robust. To do this, we need a growing network of automated tests, ensuring that the system can continue to improve from its simple beginning to the desired end product.

Stress and pressure reduce our ability – or will – to keep the code clean and well tested. In the rush to add features, we create “legacy code” every day, rather than an improving product. When software development feels smooth and fluid, things go well. When it feels tense or rushed, things go poorly.

Sometimes teams have difficulty getting stories completely done by the end of the iteration. What does “Done means done” really imply, and how can we get there?

What is our definition of success in software development? Where are we feeling gaps between our current position and success? What might we do about that?

How can we get software done every month, every week, every hour, without undue pressure, without slowing down, with satisfaction as a daily outcome?

Let’s try different things at all levels of our projects, to find out how they work, and how to sense when things are improving.

June 25th through 29th at the Denver Tech Center Marriott, Ron Jeffries and Chet Hendrickson will be hosting Five Days of Software Development with Style and Grace. If you are a developer with a working knowledge of XP /Agile Software Development, at least enough JAVA to get into trouble, and a desire to grow a bit, then we want you!

We’d like you to bring a laptop, with Eclipse 3.2, Java 5, JUnit 4, FitNesse, and Subclipse, ready to go. We’ll be working in pairs, so if you can’t bring a laptop, let us know.

The tuition for the five day session is only \$2500. Class size will be limited to 18. For more information and group discounts contact us at: training@hendricksonxp.com.

See below for a description of some planned topics.

Core Topics

Planning Triage

Most projects need a rough estimate of how long they'll take to determine whether they are feasible at all. The bulk of the project is probably capable of being estimated quickly, with a few aspects needing more work. We'll practice dividing up stories into those we can estimate, those that need more exploration, and those that are dead on arrival.

Blitz Design

In each iteration, we need to move quickly from a group of stories to a plan of implementation. We need to envision the current design, determine how we'll change it, estimate and divide the work, and go. In a series of exercises, we'll practice finding the design in a matter of minutes.

Spiking and Estimation

Given a clear understanding of our design, we can usually estimate the difficulty of most stories with ease and accuracy. However, some stories may require a new algorithm, or use of a bit of technology that we're not familiar with. We'll practice creating time-boxed spikes and using them to arrive at an estimate of the real story.

Story Splitting

The only good story is a small story. We'll practice creating smaller stories, to get finished sooner, give us more flexibility, and help separate the frills from the essentials.

Progress Reporting

We need to know what's going on. Our customer needs to see progress, so that she'll be encouraged to manage scope. Management wants insight into our project. We'd like to meet these needs with as much transparency, and as little work, as possible. We'll practice producing useful project information frequently and quickly.

Continuous Release

From the individual programmer up to the corporation, frequent releases offer great benefit. Code that's held back is not earning its pay. We'll practice keeping our system ready to go at every moment.

Special Topics

You'll probably have some ideas come up that you'll want the group's help with. We'll discuss things that come up and devise exercises to work on them, where time and the group's ability are up to it.