

Teach As You Develop

Emmanuel CHENU

emmanuel.chenu@fr.thalesgroup.com

<http://emmanuelchenu.blogspot.com>

This year, I've started to teach software engineering to a class of last-year engineering students.

As Chad Fowler puts it in "My Job Went To India":

When you teach, you have to answer questions that may never have occurred to you. Through teaching, we clean the dusty corners of our knowledge as they are exposed to us.

I find that a interesting way to teach software development to students is to teach as you develop: Plan, Do, Check, Act (thanks Mr.Demming). Therefore, the course is led as a software development project.

PLAN!

Each lesson starts by a short and focused meeting. During this 15-minute project meeting, we

- recall the important matters taught in the **former session**, and
- identify what we commit to deal with in the **current session**.

Each feature to teach is then **estimated** to check it will fit in the day's lesson.

DO!

I never delay an answer to a student's question - even if it forces me to change and re-plan the sequence of the course. The course is **driven by the students needs**. It must be **adaptative**.

CHECK!

At the end of each lesson, the students must fill-in a short anonymous evaluation. They are asked provide the following **feedback**:

- What is really important in today's course?
- What did you not understand clearly (and did not dare to ask orally ;o)?
- Is there something we forgot to talk of?
- Do you have any remarks to improve the lesson's animation?

ACT!

With this feedback, I try to **continuously improve** my teaching and insist on the important matters the students have misunderstood (*because I was'nt clear enough*).

CHECK! (again)

Then, at the end of the full course, I give the students a full anonymous evaluation.

- Rate the contents from 1 to 4.
- Rate the animation from 1 to 4.
- What did you prefer in this course?
- What did you not enjoy in this course?
- Do you have any suggestion to improve the course?
- Would you recommend this course to another student?

ACT! (again)

With this **feedback** I try to **improve** next year's session.

I believe this practice of teaching software development enables students to **feel what a development project is like**. Indeed:

- The customer (*the students*) are involved in the project (*the lesson*).
- Each lesson is an **increment** of knowledge.
- Each lesson is an **iteration**.
- Each iteration starts with a little **planning**, a little **estimating** and some short-term **commitment**.
- The planning is not predictive. It is **adaptative** as the contents of an increment (*or lesson*) may change according to the customers (*the students*) needs (*questions and remarks*).
- Each iteration ends with a **retrospective** to provide **feedback** to **improve** the next iteration.

I haven't yet managed to practice **continuous testing**, as the students would feel the pressure too hard ... one day maybe ;o)