
Meeting Minutes / Issue

Project/Issue XP Day 2008
Place Church House, Westminster
Date 12 December 2008
Attendees
Copy to
Author Immo Hüneke

Zuhlke Engineering Ltd
Marble Arch Tower
55 Bryanston Street
London W1H 7AA
United Kingdom

Phone +44 (0)870 777 2337
Fax +44 (0)870 777 2366

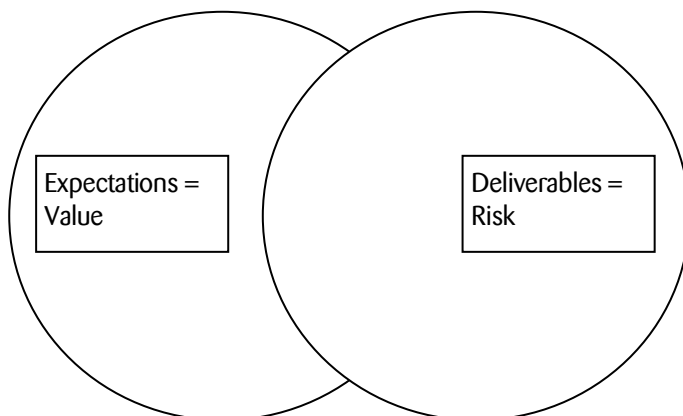
info@zuhlke.com
www.zuhlke.com

Lightning Talks

1. Software Testing

James Lyndsay

The scope of testing:



Testers are being used wrongly – they are the ones focused on the right side of the diagram

2. Can the use of language make us more agile?

Simon Woolf

Java is far from the most agile language

- Tends towards verbosity (cf. Python has only some 30% of the bulk for the same thing)
- Tends towards complex architectures (EJB2...)
- Heavy investment in IDEs
- XML sit-ups – many configuration files, cf. Django's "configuration in code" concept
- Static typing → bondage, discipline, wait for compilations, worry about interfaces
- MVC frameworks encourage DRY, better modularity, makes refactoring quicker & easier

Spend less time writing code, spend more planning, designing and testing! Some languages can be more agile than others. 3:45 today – share your experience of alternative MVC frameworks.

3. Challenging Requirements

Gojko Adzic

Very often the customer would accept a different solution if it could be done faster or more cheaply – often “solutions” are presented as requirements.

Customers explain what they want by use of examples – from their own perspective.

- Example: Client wants “real-time reporting” on a transactional system that was designed for batch processing. On closer inspection, “real time” meant more than once per day!
- Another (Martin Fowler): Order-tracking system being replaced. Printing could not be implemented correctly. Client: “Printing is embedded in our business process”. On investigation, it turned out that printouts were only being used to re-enter information into another screen!

Proposition: ask what the problem is, not what the suggested solution is.

4. Reengineering Teams

Wim

One team of four was treated as the scapegoat for everything that went wrong. Wim cancelled all work for an afternoon and put all four teams into a room. Main complaint: responsibility of teams wasn't well defined. Small groups: suggest a better split. Answer: need 3 teams, not 4. Wim: game of “musical chairs” to force teams to self-organise. After two or three goes, the group had settled into three teams. Still not quite happy, but attempts to move individuals across teams were already being resisted.

Break, repeat exercise. After 3 hours, there were three groups – it was a compromise, but the team had taken responsibility for its own composition and duties.

5. Funny Hats

Julien G

Response to “check-in chicken challenge” (mentioned yesterday). In Julien's development team, anyone who breaks the build has to wear a stupid hat (at this time of year, the Christmas hat). Also stop the production line so that everyone can contribute to fixing the problem.

6. Real Options

Portia Tung, Pascal Couwenberge et al

Space game: Galactic Council has asked you to defuse hostilities and save the Galaxy. Transport a ring to a distant planet.

7. Agile Values

Joseph Pelrine

Software development is about people. Yet a lot of the people drawn to it are not keen on talking.

Ben Fuchs and Joe have devised an open space session called “shrinks' session”.

8. Designing the Agile Company

Allan Kelly, Giovanni Asproni

Dialogue sheets used for facilitating conversations. Go to David's session beforehand

9. Agile 2009 in Toronto

Karl Scotland

Stages called “Breaking Agile” and something else have been removed from this year's conference – see David Anderson's open letter on agilemanagement.net and sign the petition.

10. Agile Mentoring

Simon Woolf and Douglas Squirrel

Based on Portia's *Yellow Brick Road* exercise. These two guys actually formed a mentoring pair (so did Portia and Pascal). They found many differences between them.

Useful practices:

- Business lunches (which could be expensed!)
- Share useful information without fear of compromising competition
- Recruitment resource
- Shoulder to cry on
- Selling iterative development
- Similar solutions to similar problems despite different territories (P&P):
 - Winning the trust of the team being coached
 - Icebreaker exercises etc.
- Another idea from the floor: regular get-togethers of coaches (once a month) e.g. Pair Programming coaching sessions starting up soon in Belgium; coding dojos.

11. Requirements by Example

Keith Braithwaite

Been doing some requirements work recently with a steam turbine manufacturer. Interesting exercise: analyse the sub-systems within the turbine. Used a cutaway diagram on the wall to reveal disagreements between the domain experts about where the boundaries were.

Plato, Aristotle etc. tried to analyse people's perceptions of the world. They found that:

- People identify objects in the real world
- People assign objects to categories
- Categories are identified by a set of predicates
- Categories have no structure and they are hard-edged: objects are either in them or not
- The Greeks tried hard to categorise "man" – came up with creatures that have two legs and no feathers. But it was easy to come up with counter-examples – Diogenes simply plucked a chicken!

These ideas stayed with us until at least the 1970s and have informed much of the development of computer science.

Once researchers began to investigate how people really perceive the world, it turned out that people identify the categories of objects by their resemblance to known exemplars. These categories are fuzzy-edged. You can even measure the "distance" of some object from exemplars.

In the steam turbine example, there are extremely complex examples of rules e.g. to limit the amount of blade flutter permissible. By talking about the different cases, the requirements become much clearer. That's why Keith is so excited by tools like FIT – they describe a system by example.

12. Obvious is not necessarily true!

Gojko Adzic

Draw a five-pointed star shape – how many points are in the picture? Some say 5, some say 10, some say infinity. Just learn that you can't take "obvious" things for granted. You have to verify everything by reference to the customer.