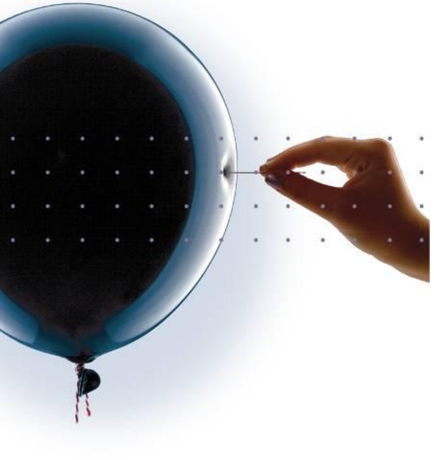
Agile Testing At The Speed of Light

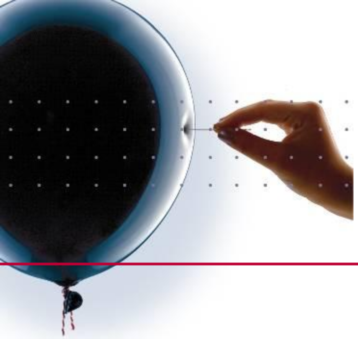
Geert Lemmens,
ps_testware,
Belgium



ps_testware
Your devil's advocate

Agile Testing at the speed of light

An XP story

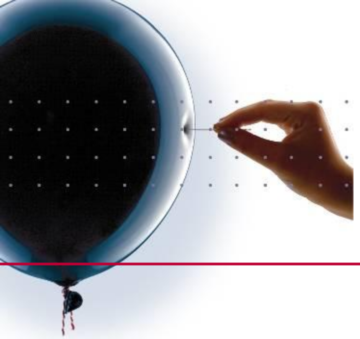


Who’s that guy?

- Geert Lemmens
- Already 11 years as consultant at ps_testw
- Projects at AXA, De Post, VMW, Fortis AG, Fortis Bank, ...
- First agile project was in 2005
- Now: Test Manager for Securities handling – about 15 applications that need to be followed-up together with 3 to 5 projects during each release.

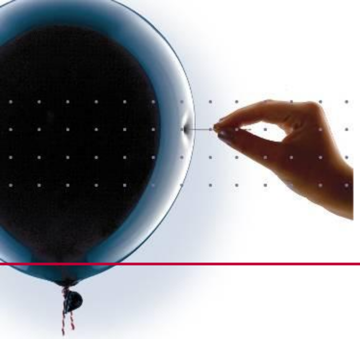
Why this presentation

- ✓ Eurostar examples
- ✓ Agile is not always a beautiful story
- ✓ People encounter problems but are reluctant to share their knowledge about it
- ✓ Share experience



What will we see and hear

- 6 problems we encountered during the project
- Each problem will be discussed
- What should we have done in order to prevent the problem?
- How did we tackle the issue

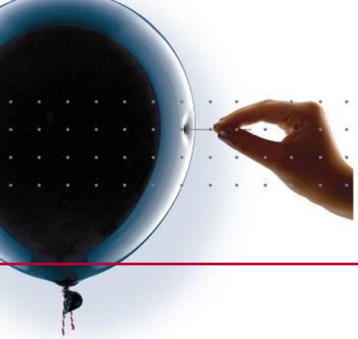


What we will NOT see and hear

- Theoretical mambo jumbo
- Agile is the solution for all problems
- Agile always works
- Anybody can start working with agile techniques
- I have the solution for all problems

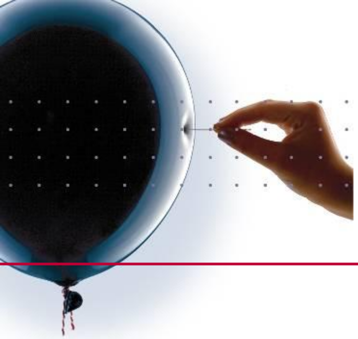
1 general remark

- Agile is sometimes not a good idea whe
  - You are involved in enormous projects (especially when you are going to use agile techniques for the first time. Literature recommends 12 developers as the maximum).
  - If you only have junior people in your project
  - If people working in the company or on the project are reluctant or showing significant resistance against constant changes.
  - If the client is not willing to participate



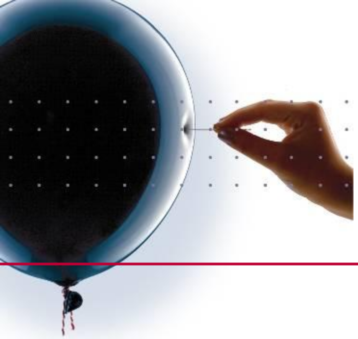
Welcome to the project

- Social security
- Replacing the old mainframe systeem
- 20 developers + 3 development leads
- Original 3 testers + 1 full time test manager
- 4 analists
- 1 program manager
- XP & Scrum
- Java
- JUnit, Eclipse, Selenium, Clearquest, RequisitePro, Ant, Cruisecontrol, Struts



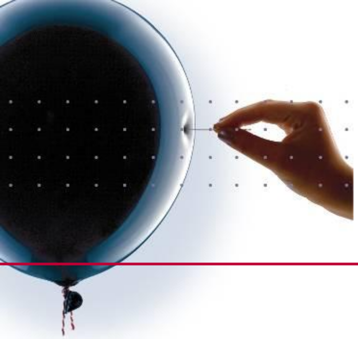
Problem 1: senior management involvement

- Very commercial program manager
- Convinces by personality, not by content or knowledge
- Even worse: little knowledge on senior mgt level, but they created the contracts with the clients
- ‘Tell me what’ AND ‘tell me when’
- Agile is ‘cool’ and development team made the decision on their own
- Testers were obliged to work agile on one side and using the classical waterfall model on the other side.



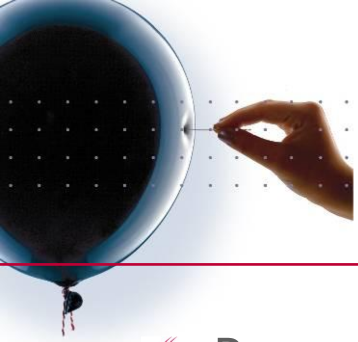
Problem 1: senior management involvement

- ✓ Prevent:
  - Make sure that Sr. Mgt. at least understands the basics of agile testing, especially if they do the negotiations with the client
  - Always make sure someone with agile expertise joins these meetings
  - Involve the testers!
- ✓ Solution: The only solution at hand was the renegotiation of the contracts but at that time, already a lot of time and budget was lost.



Problem 2: Planning game and follow-up

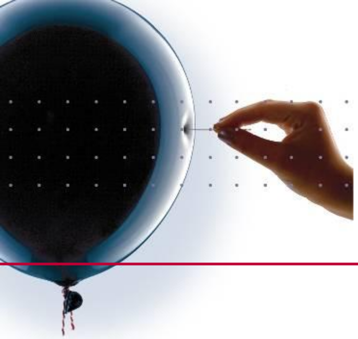
- They had no experience with agile techniques so it was very hard for them to create and estimate stories
- Constant adaptation of user stories, but not because business was changing
- Scope changed constantly but only in one sense: new stories were added frequently
- Stories were seen as ready while this was not the case
- User stories often had no business value so testers and end users found it hard to validate them → progress of the project was not transparent



Problem 2: Planning game and follow-up

- Prevent:
  - Start using agile techniques in a small project with a clear and limited scope.
  - Involve business people during the planning game, not only proxy users
  - Business has to sign the scope with ‘blood’ and understand the impact of changes. ‘You pay for what you ask’.
  - Scope can be changed but users must validate the stories so that they have business value.

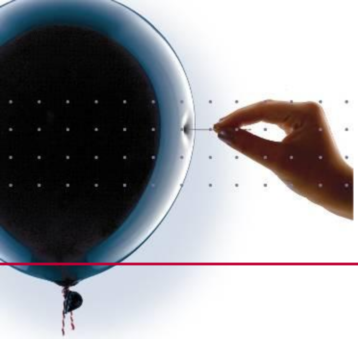
- Solution:
  - Redo planning game after redefinition of the scope
  - Although late, involve the client and let him sign the scope/stories
  - We agreed that stories are only finished when the tester indicates them as ‘passed’



Problem 3: maturity project members

- ✓ Agile method was supported by 1 person who had read 1 book concerning XP
- ✓ Other team members had no experience with agile techniques
- ✓ Client didn’t know anything about agile so their expectations were wrong.
- ✓ The project leader felt that something was going wrong so he changed the methodology to his likings.

- ✓ Prevent:
  - Healthy mix of knowledge (experienced and less experienced)
  - Give info sessions to the clients
  - Let a very experienced colleague build clear guidelines (and not the person who has the most theoretical knowledge)



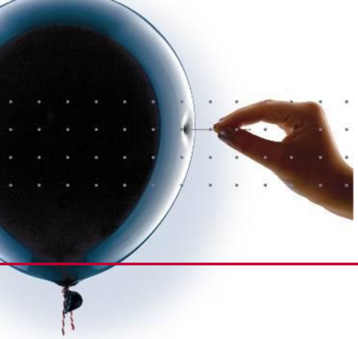
Problem 3: maturity project members

- Solution:
  - Books and articles are very handy but it is not sufficient. Organize info sessions for agile co-workers and other people involved in the project.
  - Add extra agile experts
  - Do an audit
  - Do not change the methodology but clarify the procedures. If you really need to change the methodology, make sure this is only done by highly experienced people.

Problem 4: Documentation

- Way too much documentation:
  - Use cases
  - Stories
  - Flow diagrams
  - Procedures and strategies
  - User interface documents
  - Test cases
  - Requirements
  - …

- QA required several formal reviews of all documents before they were approved

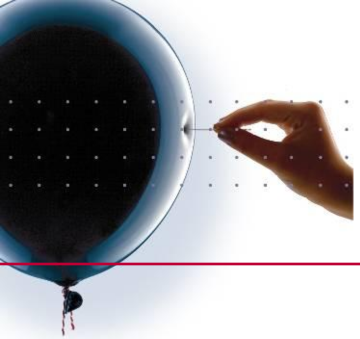


Probleem 4: Documentation

- ✓ Prevent:
  - This problem was triggered due to a lack of knowledge of agile development within the company (not only within the project).
  - Company processes need to be adapted when implementing this methodology.

- ✓ Solution:
  - Stop writing unnecessary document (flow diagrams, use cases, user interface documentation,...)
  - Work with mock objects (interface) which can be used directly within development (you can gain a lot of time)

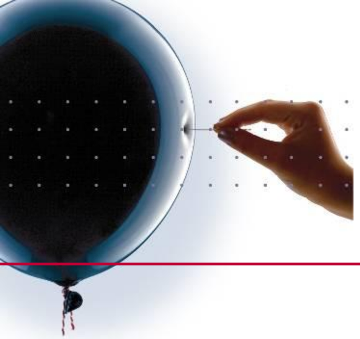
ps_testware
Your devil's advocate



Problem 5: Testers & Users

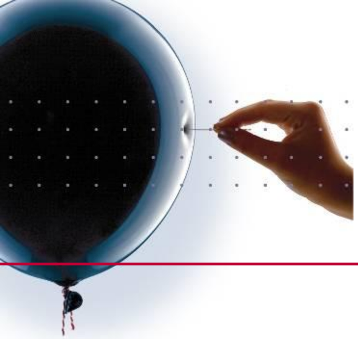
- Testers only had functional experience
- No technical knowledge
- Only limited business knowledge
- No users involved, only proxy users who had no experience within the business

- Prevent:
  - Client MUST be involved and present on the work floor
  - Testers preferably have technical knowledge since most tests need to be automated



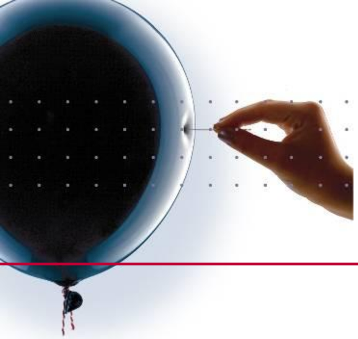
Problem 5: Testers & Users

- Solution:
  - Conversion script that transferred test cases in excel to html tables which could be read by selenium.
  - Added 1 technical person who could write VBA. He could build and maintain the conversion script and add specific checkpoints (for example using Xpath) on demand of the testers
  - Once stories were written, the analysts were added to the test team in order to augment the business knowledge within the team.
  - Client added 1 person (2 to 3 days a week) to the project and communication channels were improved.



Problem 6: Testing knowledge developers

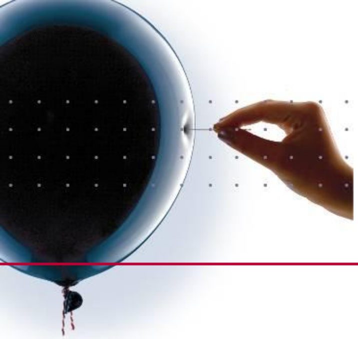
- Unit tests were too limited
- Because of a switch of application servers, the unit tests didn’t work anymore so they were put ‘in comment’
- There was a strict separation between the tasks of the testers and the tasks of the developers
- No new unit tests were made when defects were found.



Probleem 6: Testing knowledge developers

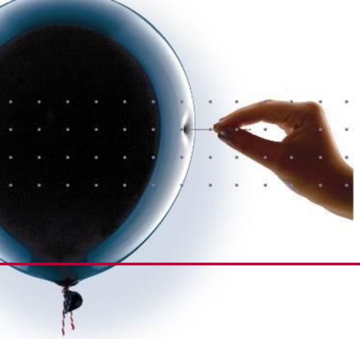
- Prevent:
  - Developers need to know how to test
  - According to theory, we shouldn’t talk about testers and developers but team members should be interchangeable. There is truth in these words!
  - For each defect found, a unit or at least an automated test should be foreseen in order to prevent regression.

- Solution:
  - ‘Testers’ helped ‘developers’ with their unit tests. Since these people were not technical and could write the tests themselves, they helped by defining the motivations of possible unit tests after which development wrote them.
  - Testers joined the developers and performed some rudimental tests right after the story was delivered (while the developer was watching). If a defect occurred, he adapted the code and wrote a unit test



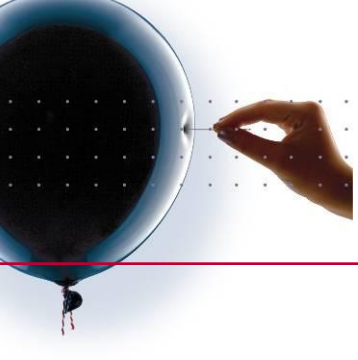
Conclusion

- Agile works!
- First step of solving problems is acknowledging them
- During your first agile project: stick to the methodology!
- Reading created knowledge but sharing experience is creating wisdom.



More information

- Pete McBreen - Questioning Extreme Programming
- Gary Chin - Agile Project Management
- Lisa Crispin & Tip House - Testing Extreme Programming.



QUESTIONS??

- You can always contact me: geert.lemmens@pstestware.com

ps_testware
Your devil's advocate