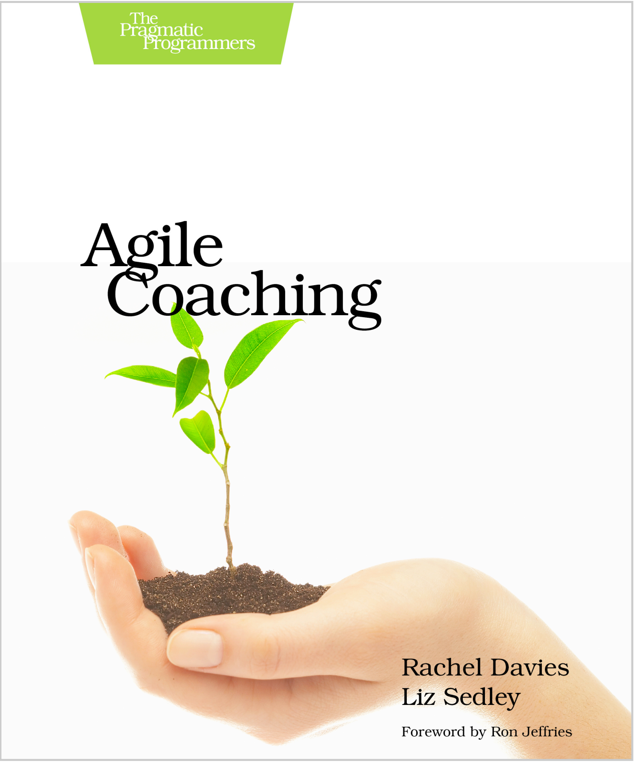
Non-Functional Requirements: Do user stories really help?

Rachel Davies

rachel@agilexp.com



My background

Past jobs:

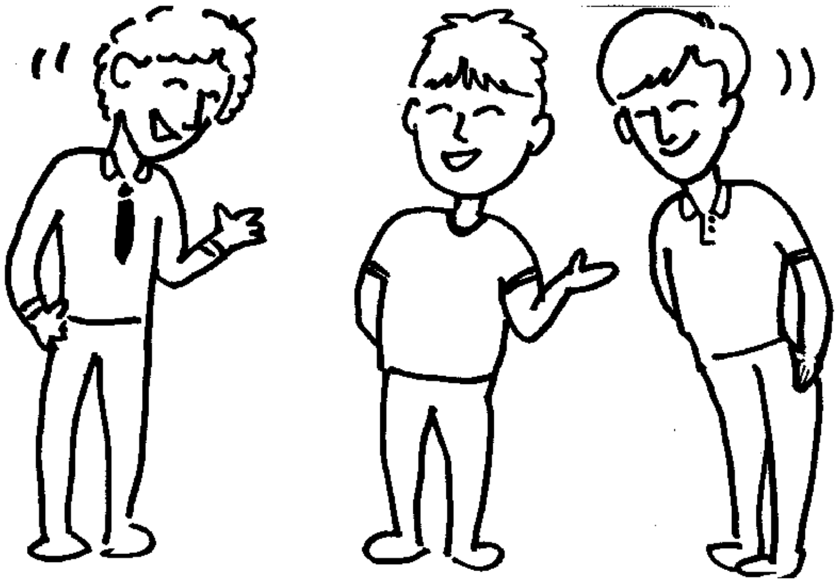
Software Engineer, Sys Admin, Systems Designer, Development Manager

Current jobs:

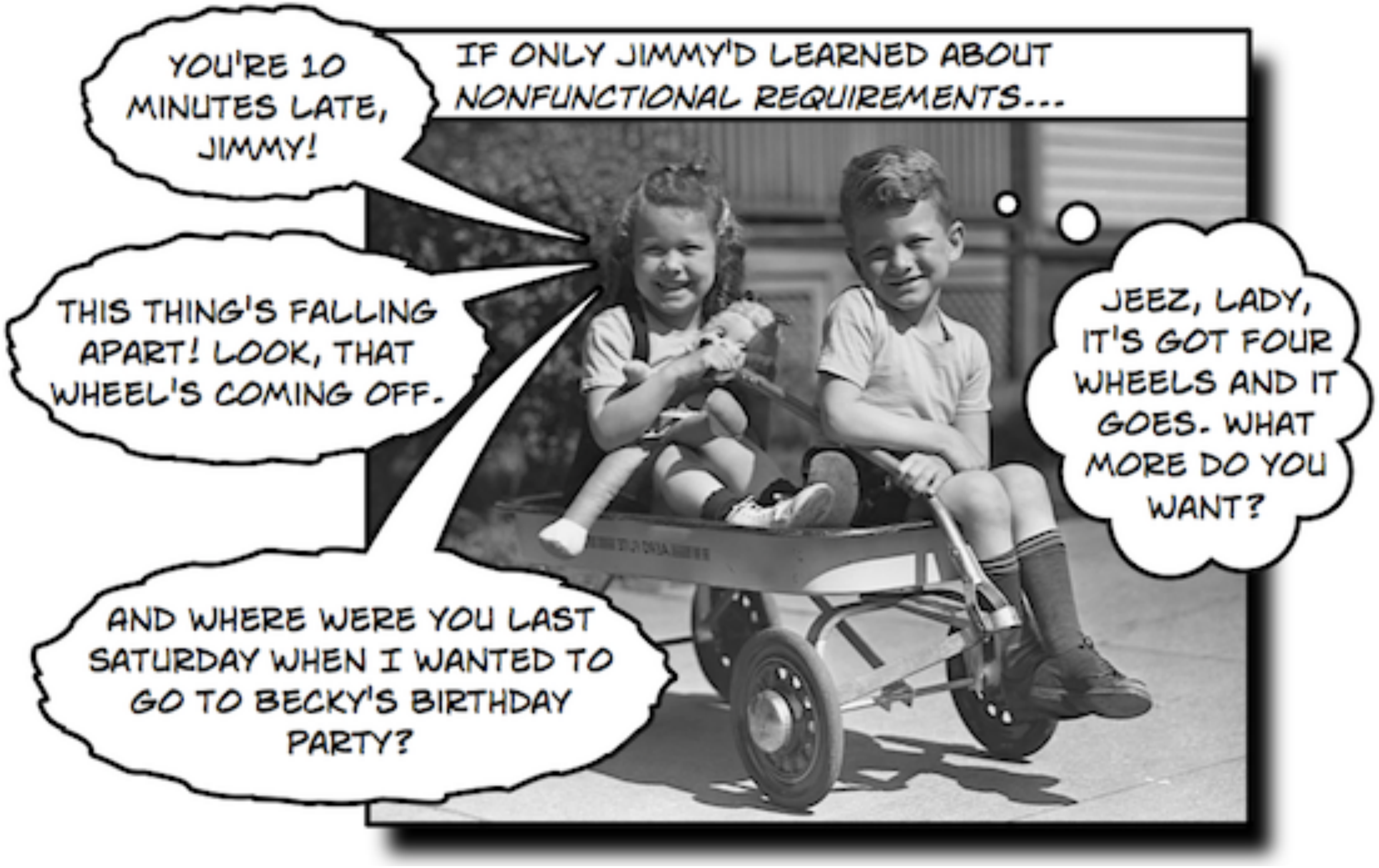
- Agile Coach, Agile Alliance Director, Book Author, Conference Organizer

Who's Here?

- Raise your hand if …



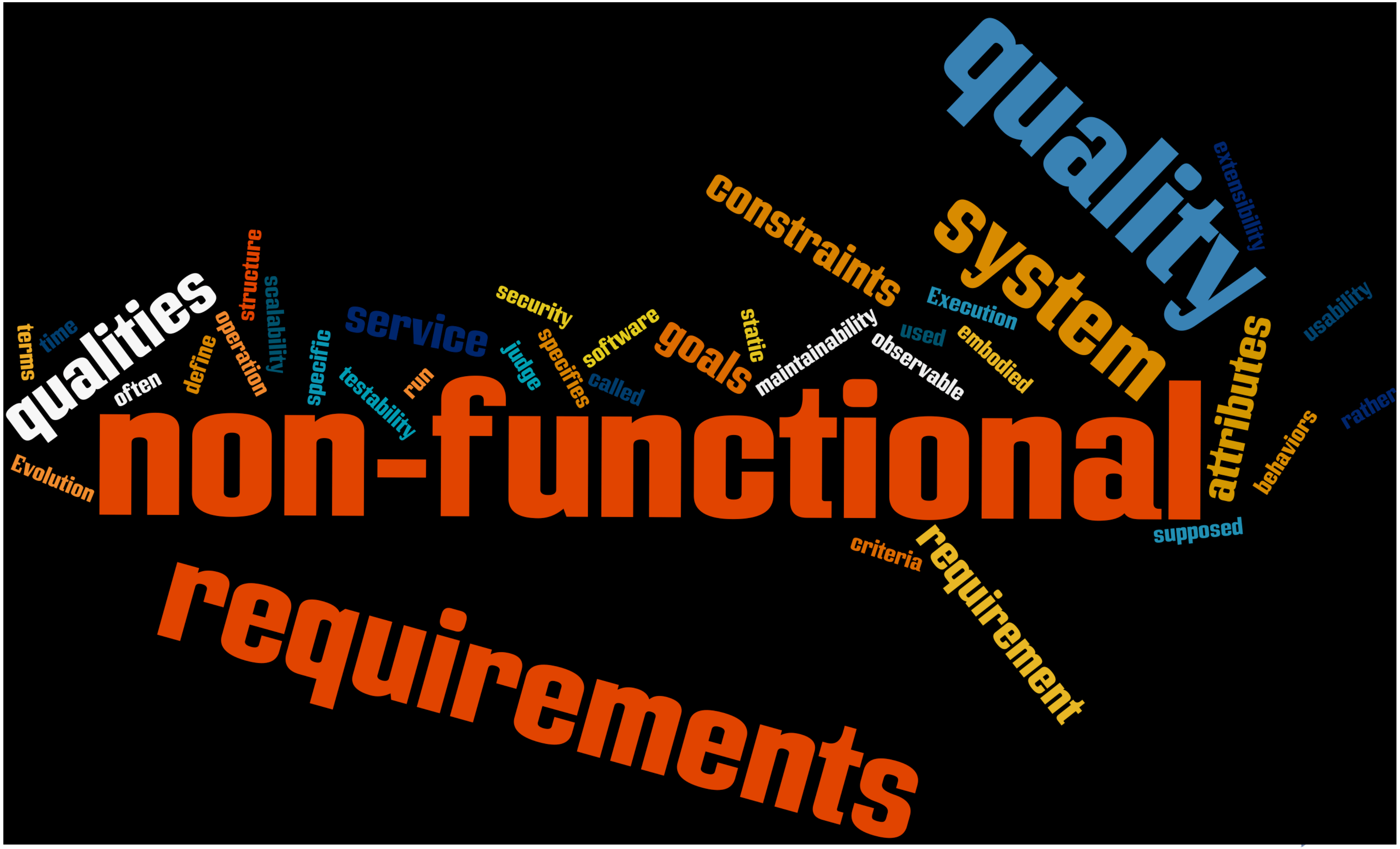
- new to agile?
- used agile?


IF ONLY JIMMY'D LEARNED ABOUT NONFUNCTIONAL REQUIREMENTS...
YOU'RE 10 MINUTES LATE, JIMMY!
THIS THING'S FALLING APART! LOOK, THAT WHEEL'S COMING OFF.
AND WHERE WERE YOU LAST SATURDAY WHEN I WANTED TO GO TO BECKY'S BIRTHDAY PARTY?
JEEZ, LADY, IT'S GOT FOUR WHEELS AND IT GOES. WHAT MORE DO YOU WANT?

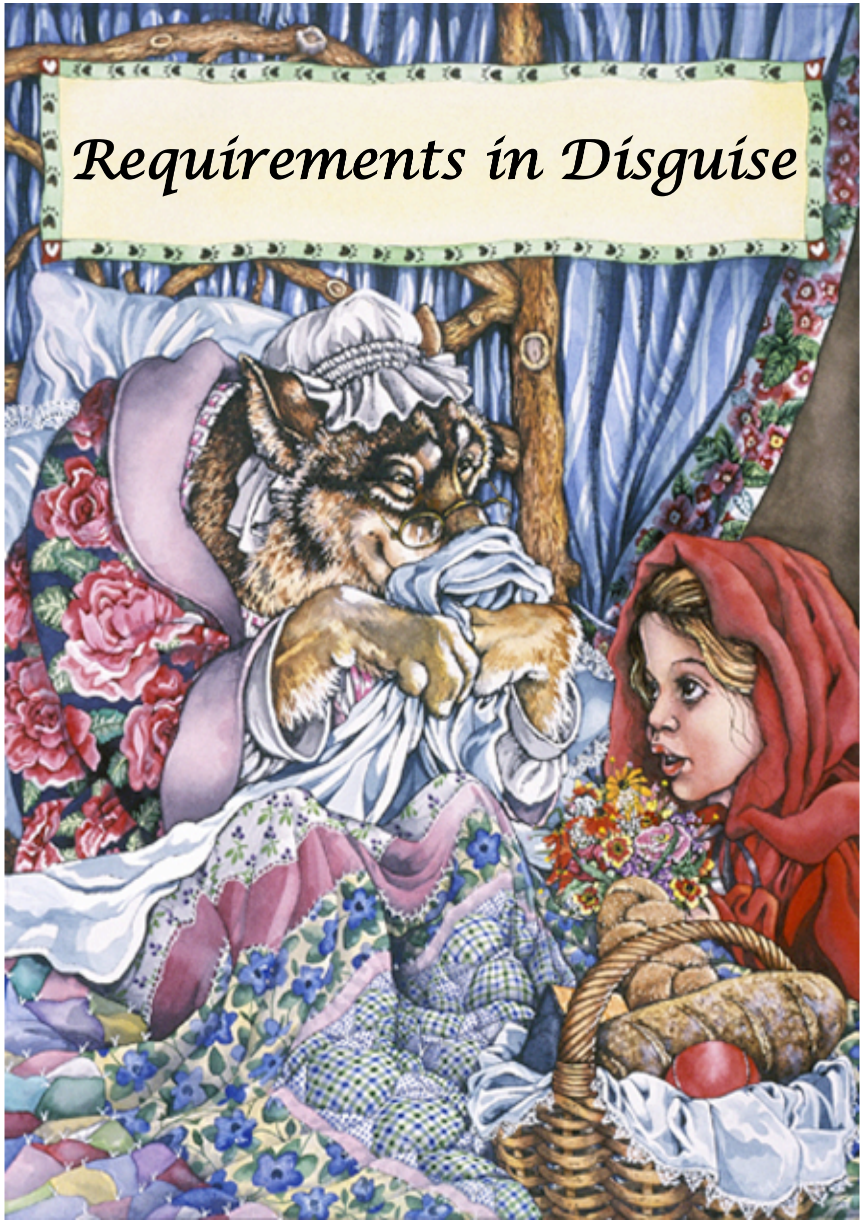
HOW not WHAT

Photo from Andrew Stellman's blog

Lots of terms



Requirements in Disguise



Agile Manifesto

Shared values and principles for better ways to develop software (2001)

www.agilemanifesto.org

Agile Values

Individuals & Interactions over Processes & Tools

Working software over Comprehensive documentation

Customer collaboration over Contract negotiation

Responding to change over Following a plan

While there is value in the items on
the right,
we value the items on the left more.

Agile Manifesto Principles (1)

- Our highest priority is to **satisfy the Customer** through early and continuous delivery of valuable software
- **Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.
- **Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must **work together** daily throughout the project.

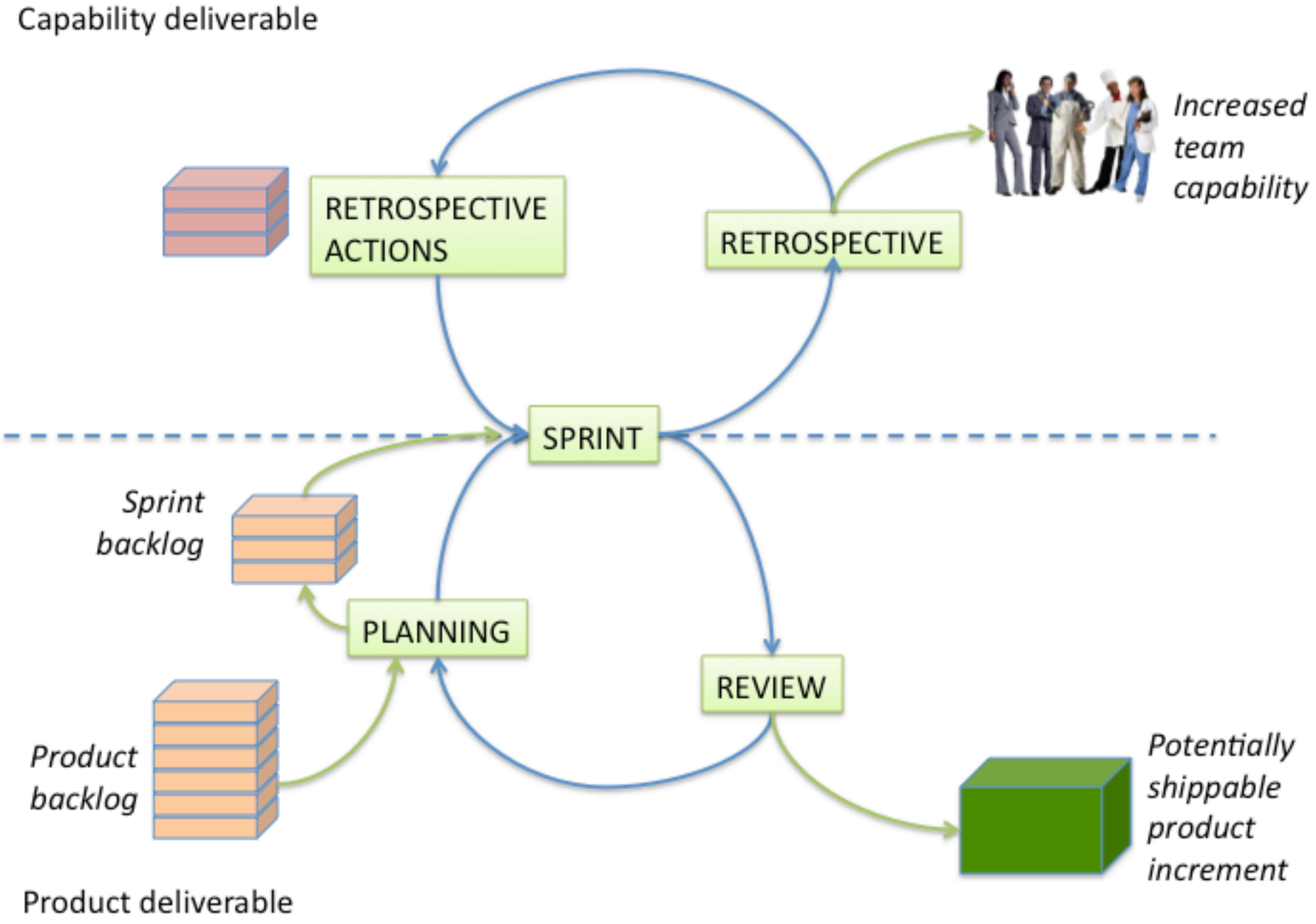
Agile Manifesto Principles (2)

- Build projects around **motivated individuals**. Give them the environment and support they need, and trust them to get the job done
- The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.
- **Working software** is the primary measure of progress
- Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

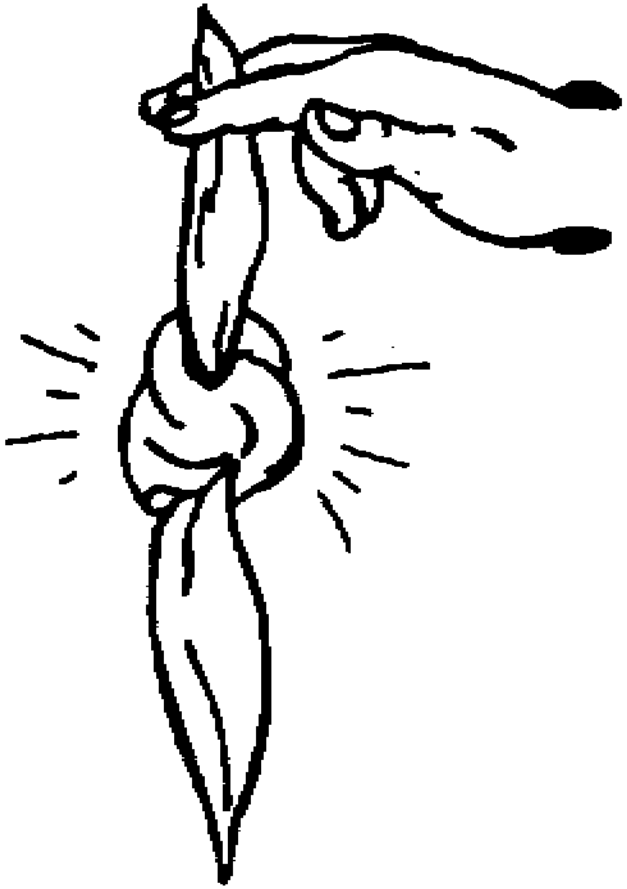
Agile Manifesto Principles (3)

- Continuous attention to **technical excellence** and good design enhances agility The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.
- **Simplicity**--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from **self-organizing teams**
- At regular intervals, the team **reflects** on how to become more effective, then tunes and adjusts its behavior accordingly

Agile Lifecycle



How does an Agile team remember them?



Is there a new Agile practice?

Whole Team

Business people and developers must **work together** daily throughout the project.

Talk about the lifetime of the software

YAGNI

Mantra from XP - “You aren’t gonna need it”

-- Ron Jeffries

- Developing for future NFRs slows us down! Keep options open
- Now vs Later trade-off
- When is “Last Responsible Moment”?
- Educate the customer about the choice being made
- Can use Risk impact to prioritize NFR work.

The Customer / Product Owner

- Don't rely on the Customer to tell you about non-functional requirements!
- Include technical stakeholders in prioritization meetings.
- Make sure you have people with ops experience in planning meetings

The Team

- Agile relies on cross-functional teams with relevant skills to deliver/maintain
- Include sys admin experience on team
- Share the knowledge (by pairing)

User Stories

- User stories explain what the software needs to do from a user perspective.
- Knowing who the user is and what problems they are trying to solve helps us develop better software.

Three Cs to a user story

Card: user goal written on an index card

Conversation: team gets to ask questions

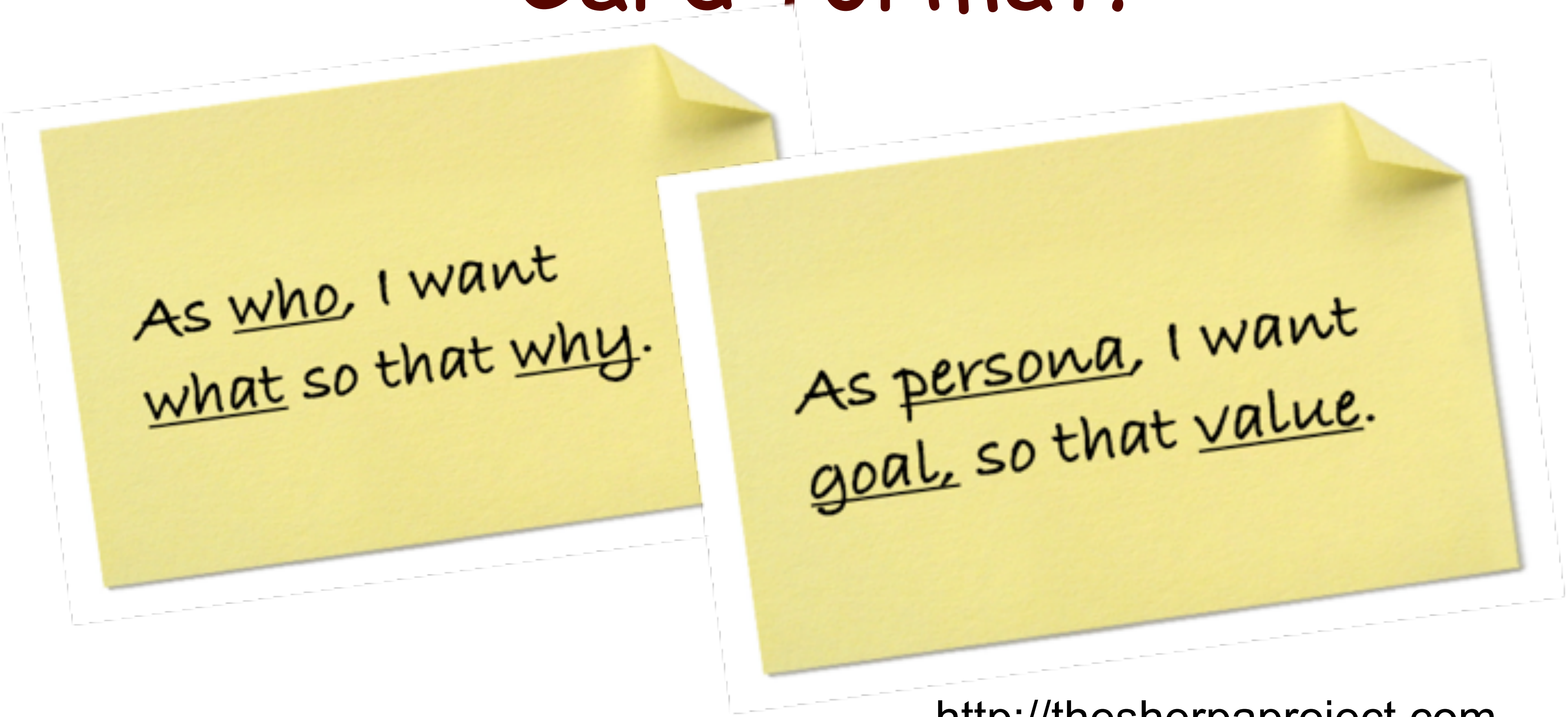
Confirmation: acceptance criteria

Ron Jeffries, Xprogramming.com

Planning with User Stories

The cards aren't mini-requirements specs!

Card format?



As who, I want what so that why.

As persona, I want goal, so that value.

http://thesherpaproject.com

Can describe NFRs like this but is it worth it?

BBC Example story cards

As an operations engineer, I want to be able to reconfigure the timeout of a specific service request without needing to restart the backend service process

As an operations engineer, I want to know how many people are currently watching Eastenders on iPlayer

Examples from
Kerry Jones, BBC

Notice they are not
 "As a system”!

Titles + Tests

Keep it simple:

- Story Title on the front of the card
- Acceptance Criteria on the back of the card
- NFRs are listed as Acceptance Criteria

Definition of Done

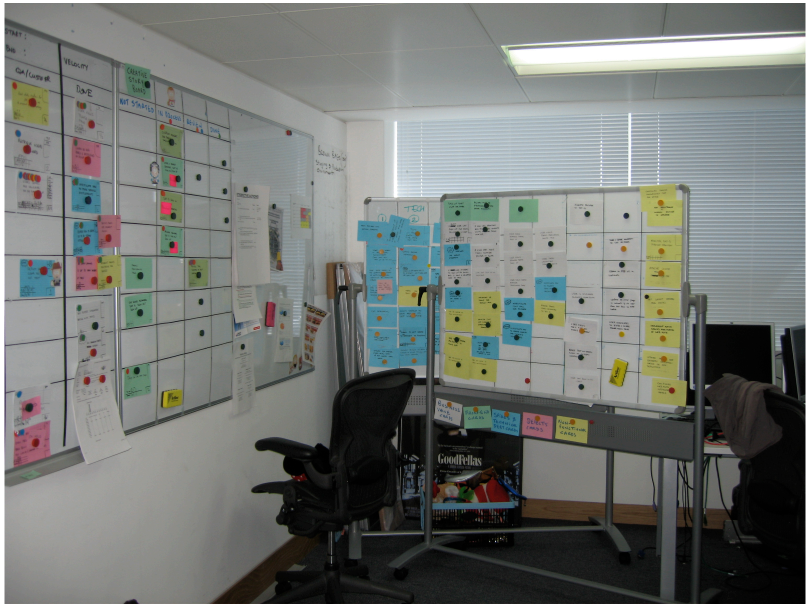
Agree a “Definition of Done” with team

May need per story checks

Also per release checks - especially if some kinds of testing is expensive

Class of Service

- Clean code not a customer choice
- Agree quotas of different types of work like a “quality tax”
- Use different color cards
- Eg, Yellow cards for operational stuff, Blue for technical debt/ spikes



Questions?

Who
Can
When
Why
What
Questioning
Do
Where
Do
Can
If
When
How
Why

Thanks to:

- Simon Baker
- Jason Gorman
- Kerry Jones
- Duncan McGregor
- Ivan Moore
- Dave Nicolette
- Duncan Pierce
- Gus Power

For sharing their experiences.

Thank you

Rachel Davies
rachel@agilexp.com
www.agilexp.com

Agile Infrastructure

Team takes responsibility for work on infrastructure and apply Agile techniques

- Visible
- Testable
- Automated, etc