

# CONTINUOUS COMMUNICATION

## PART 1

'Let's talk!' urges **Kaspar van Dam**,  
Consultant, Improve Quality Services, in this  
two-part article on communication.



**H**istory was made on 11 February 2001, when 17 experts in a ski resort in Utah put their heads together. Two days later they emerge to the outside world again carrying the stone tablets currently known as the Agile Manifesto. Within the IT-business this turned out to be a revolution, changing the way we work and especially changing the way we communicate at work. Fifteen years have passed. Where are we standing now? What did the agile revolution bring? Are we actually working more efficiently? Are we communicating better? Or did we lose the principles of the Agile Manifesto somewhere on the way?

It seems a simple question: is agile actually as successful as we often think it is? But answering this question seems to be mere impossible. For instance, at GotoCon, industry veteran Linda Rising asked the audience the striking question: “Who of you uses agile because of the thoroughly executed double blind tests and scientific research?” Needless to say, no one raised a finger. There is no hard research showing that the agile way of work is actually more successful than the good old waterfall methodologies. Simply, because it’s all about people working together. About communicating, about understanding what the team needs and wants. It’s no exact science.

However, in recent years we do see more and more doubt rising around the question if agile is actually the Holy Grail of the digital world. Take well-known expert, Jez Humble, who told the same GotoCon audience about continuous delivery actually working in the real world, in contradiction to agile which he hadn’t actually even witnessed working as intended in reality. With this he was pointing to the term ‘semantic diffusion’ as Martin Fowler calls it. Fowler is one of the original 17 people who came up with the Agile Manifesto, but he saw that the principles they came up with were interpreted differently everywhere. Now, the fact that people interpret the Manifesto in a way that suits them best is actually how a Manifesto should be used. It shouldn’t tell people what they should do and what they shouldn’t do, it just states which things one should value more over certain other things.

However, Fowler seems to have some doubts if agile hasn’t watered down too much. Others, like Dave Thomas (who was also one of the original 17), take it a step further and claim that agile is actually dead in the water and state that it’s about time to come up with a new term that suits the original values of the Manifesto better.

## HOW TO MAKE THE MOST OF AGILE

While it’s good to philosophise about things like this, reality is we are using agile every day. We are on a moving train and we simply can’t just put it to a halt. So, this article series won’t be about this new terminology for agile. It’s not about creating a brand new way of work or creating a second revolution to change the way we think about software development. It’s about what you should, or could, do to make the original values of the Agile Manifesto work while keeping in mind we’re not living in some software developing Utopia where we are practicing agile in its most purest form (does that even exist?).

It’s a simple fact that many organisations are in some sort of split between traditional line management and the relatively new agile way of work. There’s a contradiction between the year planning, budget estimations and documentation on one side and the ‘agile’ development team on the other side. In the real world this is happening that much it even earned itself a name: ‘WaterScrumFall’.

This article won’t offer you some sort of Silver Bullet to dissolve this contradiction, simply because this silver bullet doesn’t exist. We’ll have to live with the fact that we can only bring agile to practice in a suboptimal world. So, what can I offer you to make agile work better in the real world? It’s actually not something I came up with. It’s something the Agile Manifesto already says by itself and the main reason why Kent Beck, also one of the founders of the Manifesto, originally suggested to give the methodology the name ‘conversational’ instead of agile.

Some points from the Manifesto itself:

- Individuals and interactions over processes and tools.
- Customer collaboration over contract negotiations.
- Business people and developers must work together daily throughout the project.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

It’s really that simple: Just talk! Talking to each other is the way to successfully work together and to take software development to the next level, as was intended with the creation of the Agile Manifesto.

However, it may sound simple to just talk to each other and it may sound like yet another open door. Reality has proven that it’s actually really hard within ►

There’s a contradiction between the year planning, budget estimations and documentation on one side and the ‘agile’ development team on the other side. In the real world this is happening that much it even earned itself a name: ‘WaterScrumFall’



**KASPAR VAN DAM**  
CONSULTANT  
IMPROVE QUALITY SERVICES

*With over 10 years of experience in IT, Kaspar advises colleagues and clients on matters concerning testing and/or collaboration and communication within (agile) teams. He has published a number of articles on test automation, agile ways of work and continuous communication and is a speaker on these matters at events.*

It's really that simple: Just talk! Talking to each other is the way to successfully work together and to take software development to the next level, as was intended with the creation of the Agile Manifesto

a software development project. Business and IT don't speak the same language, there are politics going on, people have different agendas. With other words: there is no safe environment where we feel free to start an open and honest conversation, and thus we tend not to talk. For example, we are often afraid to show people there are things we don't (yet) know and we make assumptions, by the dozen.

Communication, conversation, interaction, collaboration. That's what agile is all about. This shouldn't be limited to just a few people or to a select few pre-planned meetings (e.g. stand-up). It's a continuing process in which all stakeholders should be involved. Not every once in a while, but always: continuous communication.

### APPLICATION IN REAL LIFE

Now, back to reality. An average scrum team within an average project. The team is fed with stories which are written by a business analyst. The business analyst has based the stories on his talks with a product owner. The product owner obviously has no time to be a full member of the team, since (s)he is working on different projects and/or is part of different teams within the project. The stories are part of an epic, the epic has a certain predetermined deadline since line-management has promised this to business managers to get funding for the project in the first place. There is no room for improvement based on feedback except for the incidental bugfix. It must be right the first time to have everything ready before the next release.

Recognisable? Now, what can we do to get the most value for money for the customer in an environment like this where we claim to be agile, but really are more working WaterScrumFall? How can we be more agile whilst ignoring some of the basics described in the Agile Manifesto? How can we create an atmosphere in which continuous communication plays a central part? The answer might be found in the Three Amigos concept.

### THREE AMIGOS

The term 'Three Amigos' is getting more and more popular. It's basically nothing more than putting three people in a room: the developer, the tester and the business (product owner/business analyst). However, it turns out the name was rather poorly chosen.



The principle behind

Three Amigos has nothing to do with these three people alone.

It's about discussing what you want before you start building and testing it and doing this with all

relevant stakeholders at once. Not just the product

owner and the business analyst, but also developer(s), tester(s), operations, DBA, etc. This could be three people, or it could also be a lot more. However, don't overdo it. Too many people in a room tend to lead to endless debates about topics that don't really matter (much).

So what's the benefit of discussing things in advance with all stakeholders inside and outside the team? It's an effective way to decide where the low hanging fruit is and thus creating a minimum viable product (MVB) with the least effort. For instance, the product owner can tell what the business really needs, developers can tell how complicated it will be to build it and testers can give an indication on how much work is needed to guarantee a certain quality. Together you decide if the created business value outweighs the efforts needed to create it.

And this meeting doesn't need to be a huge thing planned ahead for weeks. You could even start using this principle by introducing recurring Three Amigo sessions of at most 15 minutes right after the teams stand-up. It is very important though to ask the right questions. Things like: 'Why should we build this?', 'How can we build and test this?', 'What will go wrong if we don't build it?', and probably the most important one: 'Give me an example of how this should or should not work?'

This principle works well by itself, but there are some preconditions to make it work and to make it into a continuing process, a way of work, maybe even a way of life! ☺

*Part two of Kaspar's article, which will outline guidelines on how to achieve continuous communication will be published in the next issue of TEST Magazine.*