ARE WE STILL TALKING?

Has the agile world seen any improvements in its application of 'continuous communication', or is poor communication still the number one reason for projects failing?

KASPAR VAN DAM
CONSULTANT
IMPROVE QUALITY SERVICES BV,
THE NETHERLANDS

With over 12 years' of experience in IT, Kaspar advises colleagues and clients on matters concerning testing and/or collaboration and communication within agile teams, projects and organisations.

Back in 2015, I introduced the term 'continuous communication.' It was a way to create awareness that, despite the introduction of methodologies like agile (2001), specification by example (2004) and behaviour driven development (2006), communication (or the lack thereof) is still the one thing that causes many IT projects to fail. Not because we don't communicate at all, but because we tend to forget to keep communicating. A stand-up or retrospective meeting isn't the only moment at which we are allowed to talk. We should be communicating all the time, hence 'continuous communication.'

Over three years' have passed. Time for a retrospective on where we're standing now with continuous communication. Have we learned from our mistakes? Have we learned to effectively communicate all the time, or is poor communication still the number one reason for failing projects?

In this article I'll be talking about my own experiences with continuous communication and also the experiences and anecdotes of others working in IT related projects. Their thoughts are included in this article as communication definitely isn't just a one-man thing!

ENDLESS DIGITAL POSSIBILITIES

We live in a time of 'communication.' There are more than enough means to communicate with each other: email, chat, social media, video conferencing and telephones, etc. Digital communication is everywhere and it certainly makes life a whole lot easier. We can work from home and still stay in touch with our co-workers. With the push of a button we can find anyone within our global company and ask them for the information we require.

However, as my colleague Benjamin Timmermans pointed out, all these means of digital communication seem to have more focus on sending information and less on receiving it. Thus, it's all about talking – but not much actual conversation. Because of this, the use of communication tools poses a huge risk, especially since people working in IT tend to quite easily put all their trust in digital 'communication.'

This was clearly demonstrated in this anecdote from another colleague, Gerbert Gloudemans: "I once got this totally serious question about what I would do if a Business Analyst, seated 10 metres away, provided specs that weren't entirely clear. My quite shocking response was that I would walk over to them and ask them to clarify the specs!"

When it comes to communicating nothing works better than an old-fashioned face-to-face conversation but, more often than not, digital communication is used instead of real-life conversations, when it should really be supplementary to it. That's bad news for effective continuous communication.

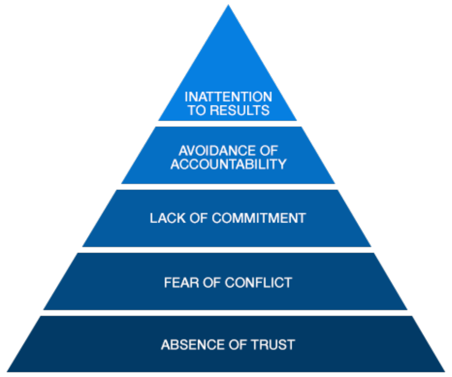
Does this mean we should abandon digital communication? Absolutely not! However, as discussed by Agile Coach, Ralph van Roosmalen, it requires the right mindset: non-face-to-face communication should be more explicit. Sharing information and talking about things should be a conscious activity. People should be aware that communication and collaboration is part of their work and that it will not happen by itself. Especially when you're not in the same room and you need to make a phone call or start up a video conference. It's very easy to forget all about communication when working at a distance. There are, of course, tools to keep communication lines open and help teams when people are not always able to meet face-to-face.

Communication is all about conversation, not about talking. And, as said before, social media and most other digital communication channels tend to be more about talking: sending information instead of having a conversation about something. The importance of (face-to-face) communication through conversation is even part of the twelve agile principles: 'The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

We can certainly adapt tools that can help us communicate better, faster and/or easier, but they should never replace face-to-face conversation, which does happen nowadays, all too often

TRUST AND CONTINUOUS FEEDBACK

Something else that almost everyone I spoke to about Continuous Communication for this article came up with, was this one very important thing: the importance of trust. This is also demonstrated in Lencioni's pyramid of team dysfunctions:



Effective communication within an agile team isn't possible without trust. As also mentioned by senior agile consultant at Xebia Group, Bart Bouwers: "Within a team you need to be able to speak out about what's on your mind. You should also be allowed to make mistakes and learn from them!" If you don't trust team members enough to criticise them out of fear of conflict, or if team members are afraid to criticise you, then the team will (most probably) develop a culture of keeping quiet and just focusing on their own work. Frustrations will grow, collaboration will be low and conversations will be kept to a strict minimum.

One of the keystones of agile work is feedback: 'At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.' However, there's one issue with this agile principle: what is meant by 'regular intervals'? People tend to translate this to that one meeting (retrospective, review, stand-up) where we are allowed to give feedback – and when that meeting has finished you'd better mind your own business.

"Within a team you need to be able to speak out about what's on your mind. You should also be allowed to make mistakes and learn from them!"

However, if a team wants to be the most efficient with effective communication there should be a culture in which there's room for continuous feedback: speak out about what's on your mind and don't hold it back until that one specific meeting (all the while growing increasingly frustrated). This should not only concern personal matters, but also the software being built and the way in which it is being built.

EAT YOUR OWN DOG FOOD

Another theme that gets a lot of attention these days is accountability. A good agile team should feel responsible for the software they develop. They should be proud when they build something that really helps the business and they should be ashamed when it turns out they actually developed rubbish. Therefore, fast feedback on the product is essential. This is where, as stated by Agile Consultant at Avisi, Berry Kersten, such things as information radiators can really help: visual displays on the working floor that show the current status of the software in development. They can show results from the nightly run, but also results on how the software is being used in production. How many customers used the software without issues, but also how many errors were thrown. The team should be aware that their product is actually being used and that it's functioning correctly – or if it is not!

This is one of the reasons why DevOps can help in creating a sense of responsibility and accountability. If you build the right software, and build it well, you won't end up with much work on your plate to keep it running. However, if there are issues with the software the team itself will end up with a ton of work trying to fix every single issue encountered on production. It's even better if teams actually use their own software – better known by the principle of 'eat your own dog food'. It's a great motivator for any team to build the best software they can when they will also benefit from it themselves.

This almost certainly leads to open conversations about the software when team members discover the software they so proudly built doesn't work as intended and/or they can't use it in real-life situations the way they thought. The availability of information gives people something to talk about. Information radiators and DevOps can improve this mindset in which people continuously share conversations on how the software being built is functioning and how it can be improved, both within the agile team itself and with stakeholders, end-users, etcetera.

COMMUNICATION IS ALWAYS KEY

It's 17 years after the introduction of agile, which was all about communication within teams, fast feedback and (face-to-face) conversation. In 2009 the book Bridging the Communication Gap by Gojko Adzic starts with, "I am getting more and more convinced every day that communication is, in fact, what makes or breaks software projects". It's 2018 now, and I think this is still the case today.

I have asked many people in the industry if they actually know what the end-user really wants. Almost everyone thinks they do. However, when asking the one single question 'why does the end-user want that?' it turns out most people have no clue what drives the end-user and what they therefore need from the software being built.

Have you ever walked over to a real end-user and asked them why they're doing things the way they are doing them? And why they would want things (the software being built) to change?

It turns out that 'why?' is a really powerful question to start a conversation about what a partner really wants or needs. In IT we tend to immediately jump to conclusions and solutions. Quite often the business even asks for a specific IT solution.

A good example I encountered myself was when the business asked the IT team I was working in if we could build two buttons on a screen for generating a report. One button should say 'Excel' and result in a report being generated in Excel, the second button should say 'PDF' and generate an error message stating that a PDF report isn't available.

My obvious response was 'why on earth would you want a PDF-button just to show an error message?' Of course, they didn't really want that button, they just wanted the report in Excel. However, there had always been these two buttons for any reporting functionality so they had assumed that was the only possible way to build that screen. In the end we didn't even build a button – it turned out users much preferred to just press the 'Enter' key on the keyboard to download the Excel report they needed

THE THREE AMIGOS

When we talk about communication through conversation it's also relevant with who you should talk. This is often overlooked and people tend to stick to their usual circle of people. The people they know best and learned to trust. This is also the focus of the 'three amigo's principle' which describes the power of conversation between three 'friends': business, tester and developer. Within methodologies like behaviour driven development these three amigo meetings are really important and a really efficient way to quickly come to a shared understanding about what the business really wants and on how to get this realised as quickly as possible (think minimum viable product).

However, three amigo's doesn't mean you're not allowed to talk to other people besides your 'friends'. Get out of your comfort zone and start conversations with other stakeholders as well: possibly the real end-user, or other agile teams working on software that provides a service to your software (or vice versa).

As pointed out by Gerbert Gloudemans, agile and SCRUM have a strong focus on the team itself, but they don't say much about collaboration and communication between different teams and the outside world. Quite often this is where communication fails and things go sideways. This leads to frustration and instead of speaking out about what's on our mind we often keep things within our own team and start complaining about 'that other team that always gets things wrong.'

When this is happening there's still no constant communication mindset. And communication, conversation and collaboration should be part of the mindset and culture of the entire organisation, not just within one single team. And, as mentioned by my colleague Jochem Gross, this mindset is all about people. For people, communication should be part of everyday work – and not just within the comfort of their own teams – but with everyone involved. And not through just one single email or during one single review session, but all the time. continuous communication.

CONTINUOUS COMMUNICATION

It's safe to say communication is still a huge bottleneck within any IT related project and, even though digital means have added many different ways of communication that can help us improve collaboration, it's still not easy to keep talking about the things that matter. It's all about fostering a mindset of communication and realising that this is part of our daily work. Even though pitfalls like making assumptions, jumping to conclusions/solutions still exist, in the end it's all about people, face-to-face conversation and trust. It's about giving continuous feedback, asking the right questions and about leaving the comfort of your own circle. It is also about starting a conversation with those stakeholders that you don't yet know.

So, where are we standing now when it comes to continuous communication? When listening to what many people in the IT field have to say, combined with my own experiences, I think things have certainly improved during the last few years. However, there is a tendency to still keep communication within our own circle of trust and we also have a tendency to limit communication through conversation to just those specific review meetings – or even worse, that one single email! However, speaking out about what's on your mind, whenever it's on your mind, can be a hard discipline to develop and also requires a lot of trust. No matter if communication is face-to-face, or over a distance through communication tools, it requires an effort to keep communicating all the time.

So, while I feel we haven't mastered continuous communication yet, increasingly, people are becoming aware that we're not quite there yet and are, in themselves, looking to work on it. Who knows where we'll be in another few years from now.