



## Stop "Going Agile"!

The three conversations you need to have before you start



BY MARTIN SUNTINGER

### BROWSE TOPICS

- Agile manifesto
- Scrum
- Kanban
- Agile project management
- Product Management
- Agile at scale
- Software development
- Design
- The agile advantage**
- Overview
- About the Agile Coach

All articles

All too often, companies set out with the mission to "go agile" before truly understanding what that means. Cracks begin to show and expectations are missed, leaving everyone questioning the value of "going agile" altogether – and hurting your chances of ever getting there.

The truth is that going agile will result in more productive teams and faster delivery of projects, but only if everyone can agree on the rules of the game. Join Heather Fleming and Justin Riservato, from e-commerce giant Gilt, as they discuss why gaining consensus on the principles of agile is more important than implementing a process.

Specifically, Heather and Justin explore the answers to three vital questions that any team needs to be prepared to answer prior to embarking on the agile journey:

- **"But when will you be done?"** Why getting rid of the concept of deadlines is the most important (and most difficult) conversation when going agile.
- **"This is my top priority, but I can't meet with you until next week."** What to do when your business partner can't (or won't) be a full member of the team.
- **"I just want to code. Why do I have to be in all these meetings?"** Why implementing Scrum is not the first step to going agile.

## Watch and learn



## Q & A

Here, Heather and Justin selected some of the top questions from the Q&A session of this presentation. Read more to dive deeper into tactics on how to apply agile methodologies.

### Q1: Digital agile board vs. physical agile board? What's your take on them?

**A1:** It really depends on your team setup. Are you all in the same place? How big is your team? Do you have space for a big physical board? We have done both at Gilt, but have found that as we grow and expand to dozens of teams that the agile boards in JIRA Software are more practical than physical boards. They are easier to set up and make changes to, and easier to share with remote team members. What we love about physical boards is that you just can't ignore them, they are so in your face. And they are a great place to hold impromptu discussions about current work, or to have stand-ups if you do those.

### Q2: How can you work with a manager or client who does not follow or understand the agile process? I feel like an unsuccessful workflow coach sometimes.

**A2:** It's important to think about your order of operations. If you are trying to work in an agile process with people who don't believe in it, then you are jumping the gun a bit. The most important part of making this work is to gain

### RELATED TUTORIAL

Learn scrum with Jira Software

[Try this tutorial →](#)

### SUBSCRIBE

Sign up for more articles

Email

[Subscribe](#)

consensus in the philosophy before you execute a process. We have done this in the past by looking at specific problems that a team is having with the current process and solving them in an agile way. Can you walk your manager or client through real problems they're trying to solve and how you would approach it in an agile framework? Can you take them bit by bit towards becoming agile, rather than in a big changeover of the total process? You can then start to show real results this way, incrementally, as the team is working better. In short, take an Agile approach to going Agile. ;)

**Q3: How can you implement an Agile process when the project is fixed bid and/or fixed schedule with a set list of requirements to implement?**

**A3:** First off, it is impossible to complete a project successfully given a fixed schedule AND a fixed list of requirements to implement, so is there a way to get everyone to agree that this is Fantasy Land? Most constraints around deadlines and requirements are not true constraints: they are wishes. Start discussing why you are doing the work, or what is the problem that you are trying to solve. If you truly understand the goals of the project and the reasons for the constraints, you can make sure that the team is doing the right work at the right time. Writing all the requirements down with dates next to them doesn't magically make everything happen on time.

**Q4: Most projects have a release date that usually is communicated to partners and customers. In this scenario, the only negotiable thing is the feature set (although some compromise on quality). How do you work within the constraints of a firm deadline?**

**A4:** We think you answered this yourself – you do this by negotiating scope. If you don't, you are right that quality will suffer. Thinking that you can just jam in the scope regardless is a dream – you need to make sure your teams are working with reality, even if it's not what people want to hear. Heather wrote a short blog post on this which you can read here.

**Q5: How do you think the way scrum is implemented should be changed?**

**A5:** The rigidity of scrum is our biggest problem with it. To think that a single highly-prescriptive process will work for all teams, regardless of what they are working on, and who they are is presumptuous. We have seen it work for teams, but scrum is not the only way to be agile, and a lot of teams fail with agile because they think they have to implement scrum in a specific way with all of the prescribed job roles, user stories, acceptance criteria, meetings, and artifacts. Heather also has an issue with the title "Scrum Master". ;)

**Q6: How can you prevent stakeholders from directly influencing the team members?**

**A6:** Well, a good stakeholder IS a team member. So ideally bring the key stakeholder into your team so you can all work together! If you have the kind of stakeholders that just throw things over the wall to your team, or swoop in during the project and try to change everything, it's important that the whole team understands what they are doing and why. So no matter who stakeholders talk to they will get the same answer. That's what makes you a team rather than a collection of individuals. You need to communicate – a lot – and ensure everyone is on the same page and moving in the same direction.

**Q7: Do you estimate stories based on rough order of magnitude estimates (in hours) or based on points?**

**A7:** We do both, and some teams don't estimate at all. Points are great because they are more abstract, and are not tied to any specific calendar time. Hours can be helpful as a transition if your team is resisting estimation since it is more tangible. The point of estimating is to be able to tell whether your sprint is too heavy or too light and adjust, and they really serve no purpose once the sprint starts. We find that once you have worked with a team for a while, the estimation process becomes unnecessary. We can all look at the work and tell pretty easily whether it is the right amount for the sprint.

**Q8: How much value do you put in project leads/managers having deep analysis skills and product knowledge vs. just coordinating meetings between tech and biz stakeholders to gather requirements?**

**A8:** Pretty much all the value :) Coordinating meetings, taking notes, etc... are not specialized skills. Anyone can do them. While it is important that they happen, it is not really the biggest value-add you can provide for the team. If all you are doing is administrative work, then the team is right to question why you are a part of it. Everyone in the PMO at Gilt has a deep understanding of the relevant subject matter and the tools and techniques to get work done and they bring that with them to every team they work on. Many of us were engineers previously or worked with other departments at Gilt and bring with us unique subject matter expertise.

**Q9: Generally how large are the teams and what background do people have at Gilt?**

**A9:** Ideally we want our teams to be lean but large enough to be self-sufficient, allowing them to move forward on projects without dependencies on other teams. We follow the 'two pizza' rule: teams should be able to be fed by two pizzas. We also feel strongly that each individual on the team brings with them a set of unique talents, and they should be able to bring those talents to the team regardless of what their job title is. So if, traditionally, the product owner is responsible for all presentations, but they stink at it and if there's an engineer who is amazing at storytelling and winning over an audience, we're going to let the engineer bring that talent to the team. You are more than your job title!

**Q10: How do you manage a separate QA team, especially where testing may occur in a different sprint from development?**

**A10:** This is one of the more controversial positions that we take, but we don't have a separate QA team at Gilt. We believe in automation testing throughout the development and deploy process. Teams are responsible for the quality of their code. If you have the time and expertise to write code, you have the time and expertise to write tests for it. Throwing this over the wall to a QA team has never shown good results for us, and requires a lot of additional documentation and time to bring QA teams up to speed.

**Q11: If you have teams that work on several 'products' simultaneously, would it work to have all product managers in the room during sprint planning and have them determine relative priorities across products? Any other ideas?**

**A11:** Stop! Clearly this isn't going to work. The team should have a product manager of its own and not be working on multiple products for multiple product managers who are outside of the team. Whomever is acting as the team lead should step-up here and clearly outline the team's methodology for prioritization, and in what order they are going to tackle the execution of that work based on this methodology. It ties back into our point that, "you have to be aligned on your methodology before you can put a process into place."

**Q12: I am trying to get a marketing creative services team into agile. We have some deliverables that HAVE to be delivered on a certain date (designing an ad to print in a magazine). How do we fit these projects into agile framework?**

**A12:** Agile is able to handle constraints like this. It is up to the team to identify what has to be done and by when and to plan sprints accordingly. Agile should help you hit these deadlines since it gives you the ability to adjust your priorities and your planned work (scope) every sprint. If you start tracking your velocity, you'll soon be able to tell whether or not you're going to hit those deadlines sooner rather than later. It's then up to a good team leader to be able to negotiate what the team needs to be successful.

**Q13: Doesn't changing goals seem like scope creep?**

**A13:** Yes it does, but we don't call it 'scope creep' because we want to encourage change during a project. One of the

biggest advantages of an agile philosophy is that it allows you to adapt to things that are outside of your control. If the competitive landscape changes, or the needs of your business change, or there is new technology available, do you really want to plod on through the requirements matrix that was made months ago? If you want to deliver the best product for your customer, embrace change and use it to your advantage. There is no 'scope creep' in agile (insert Jedi mind-trick here).

SHARE THIS ARTICLE



MARTIN SUNTINGER

Martin currently leads Atlassian's ecosystem team. Having successfully founded a startup in the Atlassian Marketplace before, he's incredibly passionate about making sure our customers get access to a great choice of high quality apps in our Marketplace, as well as provide a great experience for all developers building on our platform. Outside the office, you'll likely meet him mountain biking, kayaking, or chasing down the best coffee in town.



TUTORIAL

### Learn scrum with Jira Software

A step-by-step guide on how to drive a scrum project, prioritize and organize your backlog into sprints, run the scrum ceremonies and more, all in Jira.

[Try this tutorial →](#)



ARTICLE

### Scrum - An introduction

Scrum is a structured framework for product development that is frequently used by agile software development teams. Read this beginner's guide to scrum.

[Read this article →](#)

#### Agile Topics

Agile project management  
Scrum  
Kanban  
Design

Software development  
Product management  
Teams  
Agile at scale  
DevOps

Sign up for more agile articles and tutorials.

Email

[Subscribe](#)



Up Next  
[DevOps →](#)