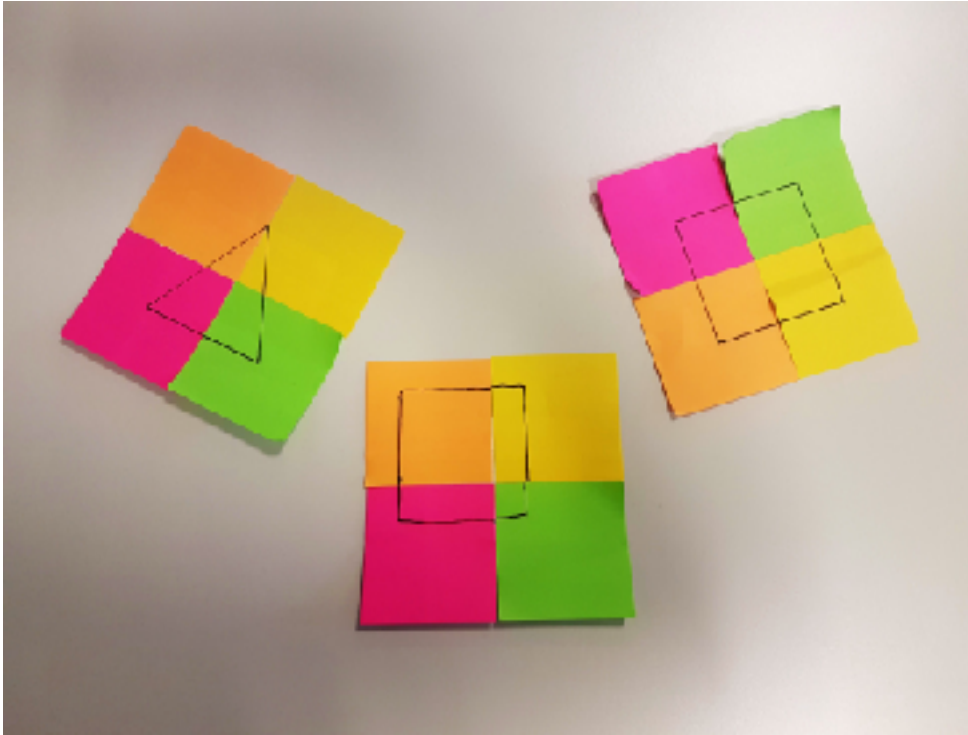


# Team Shapes

---

*Moving from component teams to feature teams*

Fredrik Lindgren  
Version: 2017-02-12



About this article .....	3
About the simulation .....	4
Origin of the Team Shapes simulation .....	4
Name of the simulation .....	4
Licensing .....	4
Time and materials needed .....	4
Executive summary .....	5
Execution Steps: .....	5
Detailed description .....	6
Step 1: Introduction .....	6
Step 2: Component teams .....	7
Step 3: Reflect on the challenges .....	8
Step 4: Cross-functional feature teams .....	8
Step 5: Reflection & Conclusions .....	9
Scaling this simulation .....	10
Variations .....	10

## About this article

This article is aimed at coaches, teachers, and anybody who wants to illustrate the benefits of organising into cross functional teams who can deliver a complete customer oriented feature compared to functional or component teams who specialise in delivering only parts of a feature.

The purpose of the simulation is to illustrate how moving from component teams to cross-functional feature teams improves several aspects of delivery such as speed and quality.

The article is primarily written for teachers and facilitators who want to know how the simulation works and how to facilitate it successfully.

## About the simulation

### Origin of the Team Shapes simulation

The Team Shapes simulation is inspired by an organisational dysfunction I had observed where a certain type of feature could only be delivered by combining the work of three different component teams, integrating them into the product by another team and eventually having them tested by a separate QA team before delivery.

I initially developed the simulation in 2014 as part of a training, introducing Agile Software Development to newly hired team members in a fast growing company. I wanted a quick experiential simulation to introduce the concept of cross-functional feature teams and to show why it is preferable compared to specialised component teams.

During the past year some of my colleagues at Crisp have picked up the game and used it successfully in other training contexts.

### Name of the simulation

The name of the simulation is “Team Shapes”. This is a play with words referring to both the tasks executed in the exercise and the different shapes of teams: specialised component teams vs cross-functional feature teams.

### Licensing

Feel free to use this simulation as you like, but I appreciate if you mention or write somewhere that the simulation came from me.

This guide and sample shapes for printing are available for download at: [blog.crisp.se](http://blog.crisp.se).

Technically the simulation is licensed under Creative Commons ([Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/)).

### Time and materials needed

The core of the simulation takes about 10-15 minutes to run, plus another 5-10 minutes for debrief.

The materials needed are:

- Sticky notes (3 by 3 in / 76 by 76 mm), 4 different colours one full pad of each colour. The pictures in this facilitation guide assumes the colours: yellow, pink, orange and green.
- Small index cards (A7 or 3-by-5). A packet of 100 should be more than enough.
- Good markers, Sharpies, Pilot V-Sign or similar. At least 4, preferably more depending on number of participants.
- Printed pictures of the requested shapes. A sample of shapes sufficient to run the simulation is available together with this guide.

## Executive summary

In the Team Shapes simulation the participants will form an organisation which has four different roles: *component specialists*, *integration engineers*, *quality analysts* and *project managers*.

Component specialists have the advanced skill of drawing on sticky-notes of a particular colour. *Integration engineers* integrate the different stickies into complete shapes by sticking them onto index cards. The *quality analysts* make sure that defective features don't reach the customer. *Project managers* are responsible for accepting orders, coordinating the production and delivering the completed shapes to the *customer*.

The *customer role* is played by the simulation facilitator.

### Execution Steps:

1. Introduce the simulation including the different roles and explain what the organisation builds by show an example of a finished feature.
2. Component teams:
  - a. Organise into four component teams, one integration team, one QA team and some project managers.
  - b. Request small batches of features from the project managers and let the organisation build, integrate, check and deliver them to experience how the work and the communication flows.
3. Reflect on the challenges of conflicting priorities, quality, rework etc.
4. Cross-functional feature teams:
  - a. Reorganise into cross-functional feature teams with specialists from the previous teams. Let the previous project managers become product owners for one team each.
  - b. Again, request small batches of features and experience how the work and communication flows in the new organisation.
5. Debrief. Compare the two setups with regard to flow, quality, waste etc.

## Detailed description

### Step 1: Introduction

Depending on the time available for the simulation the introduction can be quite short or you can elaborate a bit to get the participants to get into character for the different roles.

I start by introducing the simulation as a way to experience some of the challenges of coordinating delivery of high quality fully integrated features to customers.

#### Definition of a feature

I tell the participants that in this simulation I will be playing the role of the customer. I explain that features in the simulation are represented by polygons drawn on coloured sticky-notes and put together on an index card. I also show them an example of a really simple completed feature that I have prepared in advance. For a delivered feature to be accepted it needs to match the colours and shape in the request and be a proper polygon with straight lines forming a closed shape. There is a quality checklist available for *quality analysts*.

#### Roles in the organisation

Next I present the different roles in the organisation. To get into the mindset of specialists, proud of the quality of their own component, I emphasise the deep expertise they have. How drawing on green stickies is nothing like drawing on pink ones etc. If this feels to silly, you can explain how the different colours represent typical components or areas in a software system, for example having the pink stickies represent the front-end, green the back-end etc.

The roles are:

- *Component Specialist:*  
A component specialist is highly skilled in drawing partial geometrical shapes on sticky-notes of a particular colour. It is the work of the component specialists that create the unique user experience of the features.
- *Integration Engineer:*  
The integration engineer is crucial to the organisation. While the visible features of the product are all made by the sticky-drawing component specialists it is the integration engineer who fits the pieces together by attaching them to the index card integration platform. Without the integration engineer the customers would get nothing, right?
- *Quality Analyst:*  
The quality analyst plays an important part in the process of delivering high quality features. Quality analysts goes through the quality checklist for each individual feature and assures that the feature corresponds to the specifications provided by the customer. Without the quality assurance the organisations runs the risk of delivering defective features to the customer, causing both disappointment and embarrassment.
- *Project Manager:*  
The project manager is tasked with the tough job of meeting with the customers to learn what features are requested and then getting the teams to build and deliver them sooner rather than later. Project managers realise how long it takes to get anything delivered and they feel the pain of the waiting customer.

## Step 2: Component teams

### Seating

When I have run the simulation, the participants have already been sitting at group tables so I have not had a reason to introduce the seating specifically for the simulation. Anyway, for the simulation you want to organise into six different groups, four of them will be *component teams*, the fifth will be the *integration team* and the sixth will be the *QA team*.

It is preferable if each team will have its own table. If you run the simulation in a smaller setting with a smaller group of participants, try to get at least three separate tables, two component teams can share a table while the integration team can share a table with the QA-team.

### Team setup

I introduce the component team organisation by suggesting that many organisations build teams around a particular skill or subsystem so that team members can easily work together and help each other. For that reason we start by organising into one team per sticky-note colour and one integration team which accepts completed components and integrates them into features by sticking them onto index cards according to the required configuration.

I let the tables closest to me be the QA and integration teams. I give the QA team a copy of the quality checklist. I give the integration team the index cards. I then distribute the different colour sticky-note pads to the other teams. Once this is done I ask for 2 or three volunteer project managers. This completes the setup and the simulation can begin.

### Execution

Gather the project managers and explain to them that they are responsible for coordinating the work on the features. I then order a batch of features of a particular shape from each of them. This is done by handing them the printed picture of the specific feature and specifying a batch size. Batch sizes of 2 to 5 features are typically fine. I tell them that both quality and speed is important and that I trust them that they can get the work done. GO!

While the simulation is running, as the eager customer, every now and then, interrupt a project manager and ask for for a rough estimate for when their delivery will be done.

### When to break

I usually let the simulation run until the first batch is completed and then give it another minute until I break. This can take somewhere around 5 - 10 minutes depending on batch size and the level of quality accepted.

If there is a second batch that is just about to be delivered you can wait a little longer to allow it be completed.

If delivery is slow and you are stressed for time, you can break earlier. However, make sure that there are at least some integrated features delivered, to allow you to discuss quality.

### Step 3: Reflect on the challenges

Facilitate a quick group reflection of how this part of the simulation played out.

Some questions that may be relevant are:

- What was it like to be a project manager?
- What was it like to be in the QA/integration/component team?
- Was there any work that had to be redone?
- Who is responsible for the quality of a feature?
- What else did you observe?
- What could you do to improve the flow of features from request to delivery

### Step 4: Cross-functional feature teams

At this point I introduce the concept of cross-functional feature teams. I usually refer to that Scrum and other agile methods suggest organising into cross-functional teams with all the skills necessary to deliver complete features.

#### Reorganisation

I ask the participants what skills we need to gather in such a team for this organisation. It is important that the project managers are not forgotten and that specialists for each of the components are included too.

I then ask them to reorganise into feature teams. Usually the number of previous project managers will be the limiting factor for the number of feature teams. However, when running the simulation in small groups you will end up with a single feature team. It is not necessary for the simulation that all participants get to join a feature team.

Once the teams have formed I suggest to the teams that we make a small adjustment to the project manager role. We will change the title from *project manager* to *product owner*. The product owner is still responsible for handling the customer orders and communicate them to the team. However, the product owner no longer coordinates the work. This is left to the rest of the team to decide.

#### Execution

For the feature orders I usually use slightly more complex shapes compared to the first execution to reduce the effect of just getting faster by repetition and practice. Apart from that, the execution is similar to the previous one.

Gather the product owners and hand them orders just like in the previous setup. Again, batch sizes of 2 to 5 features should be fine. While work is going on you may check in with the product owners to get an estimate for when the batch will be delivered.

Once a few teams have delivered their orders you can stop the simulation.



## Step 5: Reflection & Conclusions

Facilitate a group reflection about the simulation.

Usually both the lead time and the quality will improve significantly with the feature team organisation. The level of stress and confusion is also lower, especially for the product owners who were project managers in the previous setup.

Here are some questions that I have found useful for the debrief:

- How did that feel compared to the component team organisation?
- What was it like to be a product owner compared to a project manager?
- As a tester, what changed by being a member of the feature team?
- What were the reasons for faster delivery?
- What were the reasons behind better quality?
- What else stood out as a difference?
- Is there some work in your real organisation that is structured according to the component team setup? What are the consequences?
- If the customer would ask for a new innovative shape without a clear specification, how would the two different structures deal with it?

There are a number of Lean & Agile principles that can be illustrated by this simulation. Adapt the questions to what you think the group may need to learn.

## Scaling this simulation

The simulation works well with a medium size group (15-35 people or so) grouped around small tables. However, it also works with smaller groups down to as few as 6 participants.

If you need to scale it up to really large groups I would recommend to split the organisation in step 2 into multiple parallel organisations with 4 component teams and one integration team each. This is something I haven't tried myself yet though.

## Variations

There are many ways the simulation can be tweaked to emphasise particular points or cater for the group of participants. Here are a few that I have tried or intend to try in the future.

### **Skip the quality analyst role and QA team**

One variation for smaller groups is to skip the quality analyst role and the QA team. In this case the customer will need to pay more attention to quality and possibly reject some deliveries.

### **Measure lead time**

To emphasise the flow of value and lean metrics you can ask the project managers/product owners to track the time it takes from order to delivery.

### **Local optimisation**

In the middle of the component team execution, break and ask the teams to do a one minute retrospective to improve how they work in the team. The teams do not have authority to change the organisation structure.