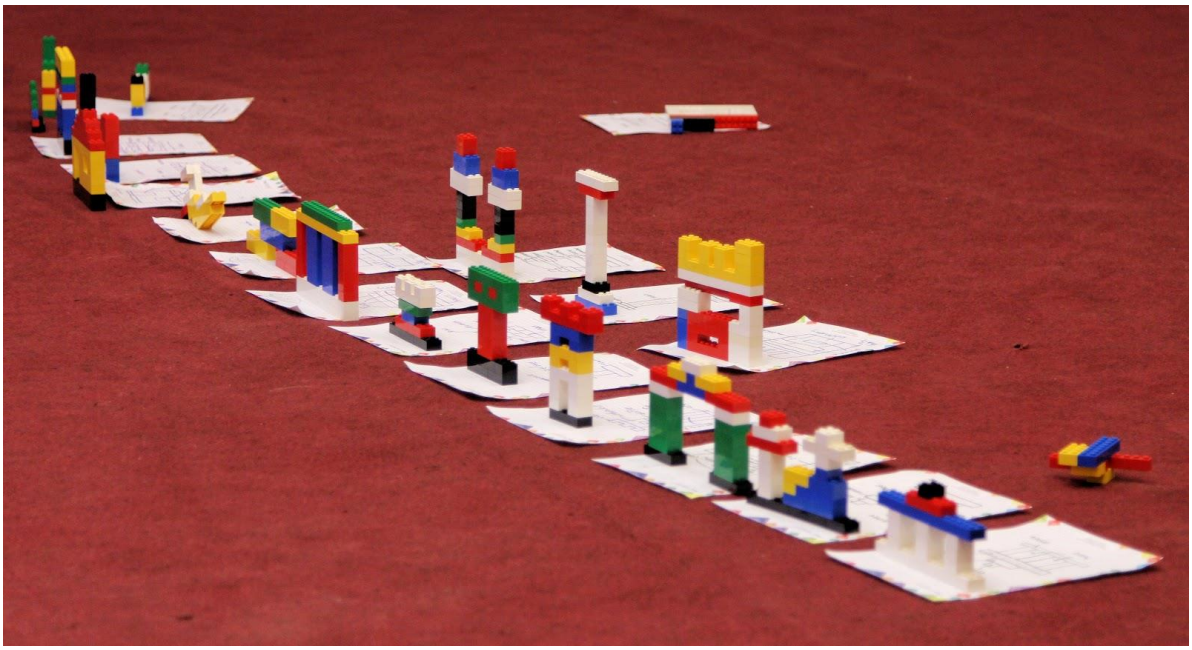
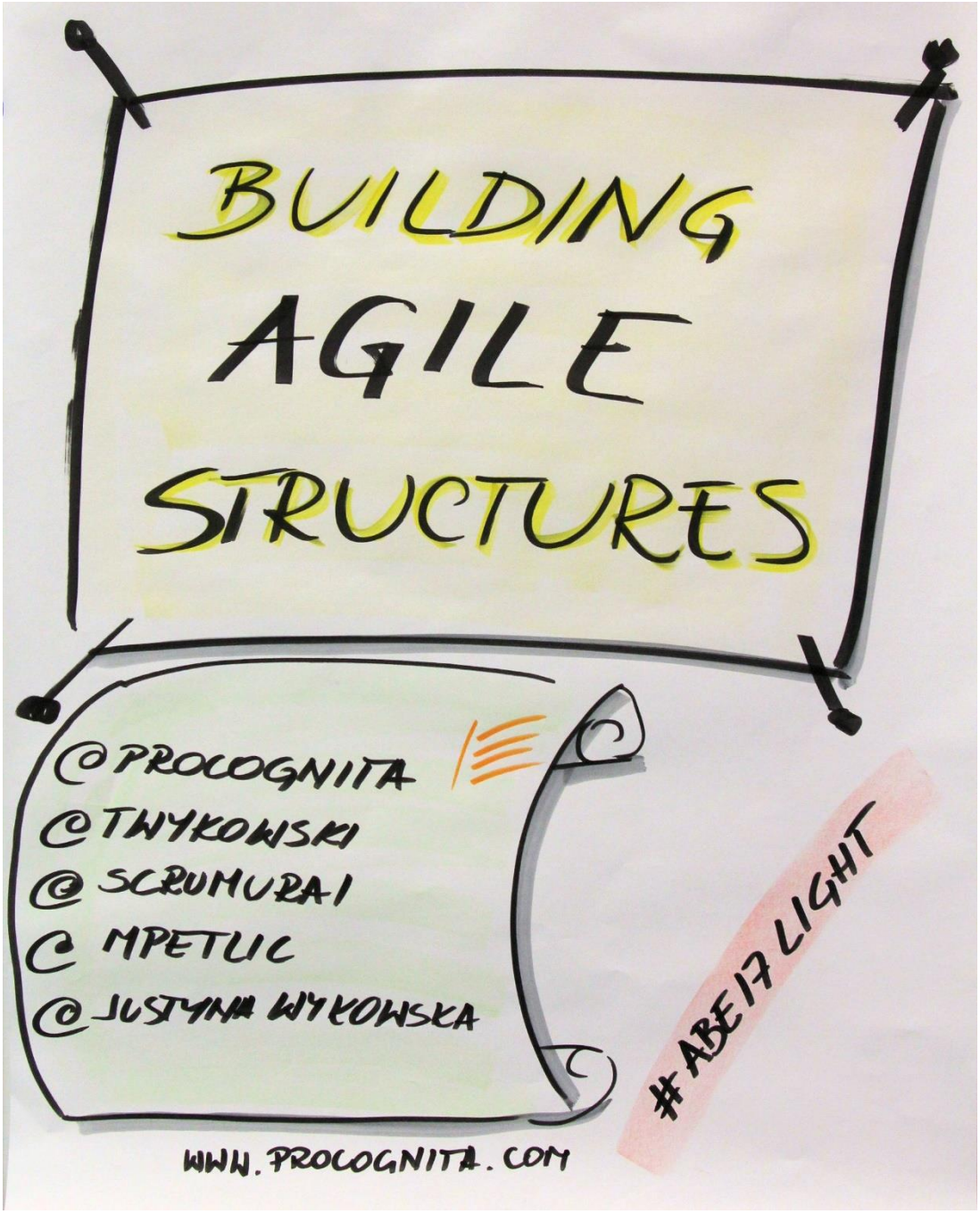
ProCognita
GET AGILE

Building Agile Structures

Lego Simulation



Fun and interactive simulation with a lot of Lego bricks showing consequences of different approaches organizational designs.



Content

Introduction

"Culture follows structure" is one of Larman's Laws of Organizational Behavior. It means that structure impacts employees' behaviours and beliefs and if you really want to change culture, you need to change structure first.

To show consequences of different organizational designs we developed a fun and interactive simulation with a lot of Lego bricks. Instead of focusing on specific approach to product development such as Scrum or Waterfall, we decided to show how building company structures around component versus feature teams impact communication and coordination. The simulation is flexible and can be efficiently run with group of 20-60 people and can be completed under 90 minutes. During the workshop, we learn by playing and hands-on, active participation make the whole experience more convincing.

This PDF contains detailed instruction for running the simulation, along with observations and reflections after our sessions, so that you could run this workshop by yourself. As we're planning another few sessions in the nearest future, so I hope to update this description with new findings.

Goal

This simulation mimics an organization with both business and development departments trying to work together in pre-defined structures. The goal of organization is to build a Lego city by delivering (1) as fast and (2) as many as possible Lego building according to specification delivered by facilitators. Three sessions let you feel first-hand how different structures are impacting Value Delivery and Time to Market. These imitate:

1) Component teams
2) Cross-component teams, single specialization members
3) Cross-component teams, multi-skills members.

What's outside our focus:

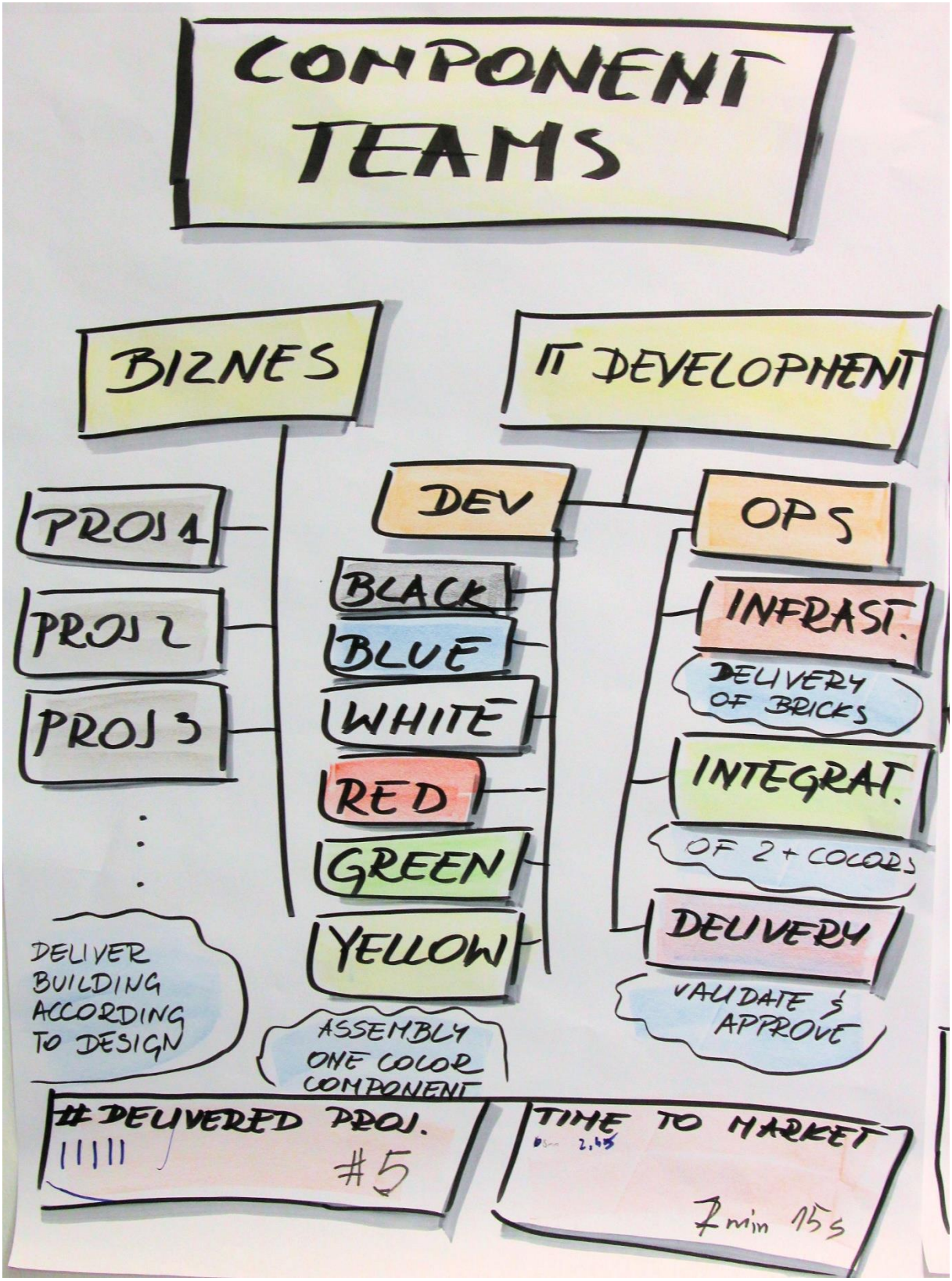
To make simulation simpler and focus on structures only we've deliberately not taken into consideration:

- **Different business value** of each Lego structures, assuming that every building has exactly the same value, which is certainly not true for product development. Therefore the only dimension that can be taken into account when prioritizing is building complexity. However, if you want, you can specify stating Business Value for every building.
- **Interdependencies** between Lego structures, which is a problem for many teams working on the same codebase is not to be perceived during this simulation. We understand that's one of the major challenges of large organizations, but for simplicity we avoid it on purpose.
- **Product discovery** by having development team working closely with business side on re-iterating the product toward higher business value as they discover their product better. In other words, business is given specification upfront and cannot change it, which in real life is one of the biggest asset of agile approach.
- **Development approach** would it be Scrum, Kanban or Waterfall is also ignored in this simulation.

What you'll need:

- **Time:** If planned well then 1,5h is just enough for running this simulation. There is more value for participants if you spent some time before or after the simulation going more into details about what are consequences of each of structures and what happened during each section.
- **Lego:** lot of it, simple, large bricks in several colors, so participants can assembly several buildings in parallel.
- **Specifications:** for every building you'll need an up-front specification. For our simulation we've created around 20 specs, showing building front wall shape and colors.
- **Timer:** Big and visible. I personally use online-stopwatch.
- **Mic:** in case your voice is not strong enough. I prefer to have two microphones, so I'm holding one all the time and second can work as a talking stick for participants during debrief.
- **Posters with organizational structures:** Not mandatory, but useful if you want to speed up introduction and clarification of rules. See each section for details
- **People:** Over 20 seems to be minimal number. Over 60 might be problematic. The rules are flexible so that you can accommodate more or less people if needed or even changes to the group that sometimes happen in the middle of simulation.
- **One or two facilitators:** Second one is very useful in both observing the dynamic and acting as a scribe during debrief.
- **Room:** large enough, so that the whole group can move easily. For every team you'll need a separate table. Chairs are useful. Some place where the city will be build (buildings will be gathered). Flipchart or whiteboard so you can collect observations.

Round 1 - component teams



1) Introduction:

How you're going to start the workshop certainly depends on your style. Don't forget to explain the goal and what's not the goal and why, as people might be confused by that. Answer any question if necessary, but keep it short - people will understand simulation better when it will finally starts.

The first round simulates organization build around component teams which is common for traditional organizations (or these who didn't get agile at all). There are two departments:

- business trying to deliver several 'projects' (buildings) at the same time (as "everything is important" is a major theme for these organizations) and
- IT with several small single-specialization teams focusing on specific tasks (based on "we fully utilize our people" concept).

2) Define Roles

As quickly as possible move to roles assignment. First select Business and IT Managers. I used volunteers, but you can pre-assign them if you want. These people role is to make sure their departments work as efficiently as possible and everyone has something to do. You can give them some hats, whips or any other signs of their status.

Then, explain that (around) half of the company is in business department, second in IT.

IT Department is divided into teams of specialists to make better use of people (or resources if that's how they call it in your organization). Each team can do only one specific task such as:

- **Development** - a team can assembly bricks of specific color (Black, Blue, White, Red, Green, Yellow)
- **Operations** - a team is having a supporting role such as:
    - **Infrastructure** - delivers Lego bricks to development teams. In other words, they're expected to take bricks of specific color and bring them to dev teams
    - **Integration** - integrates (assembly) output from two (or more) different development teams, as each dev team can only work on their color and a building usually consist of several colors.
    - **Delivery** - validates if the building is done according to specification. They can reject a building in case of errors.

Notes:

You'll need at least two people in each team. Try to make them equal in size, as it will be useful in next rounds. Operations' teams can be bit larger. So far we've run this simulation with 2 and 3 people per team.

It's useful to create 'role tags' so people can select their roles. This way you can also limit number of people in IT and make the assignment quick. The other way you can do it by limiting a number of seats per table, so people left standing are in Business Dep.

Each team needs to have its own table. Infrastructure team should be given the bricks (unless you want them to find Lego boxes themselves). Don't worry much about layout. The more chaotic the more interesting the exercise will be.

If you don't have enough people for two people in each team then remove some teams in advance. Delivery team is first to go. Then infrastructure. You can either remove few colors or make two colors teams (e.g. Green & Yellow team).

Business Department consists of several small Business Teams. Each team is responsible for one building. As in most organization everything is important, so everyone will be trying to deliver everything in the same time. Provide each business team with one specification of the building.

Notes:

The more teams you have the more chaotic first iteration will be. That represents organizations' inability to decide on priorities. I prefer to have around 10 business teams, but Business Department is the place where you can easily accommodate more people. We ran it with 2 people per team, but it can be one or three as well.

I've never identified sub-roles in Business Teams, but you can easily do it with Product Manager and Project Manager.

You can share specifications (1) randomly, (2) according to some pre-defined concept, or (3) just put them on the table and let teams select theirs.

1) Make final preparations:

Set up the timer for 10 minutes. Ask Business Manager to collect the number of delivered buildings and time to market of first building (when it was delivered). Don't spend too much time on questions. As quickly as you can start the timer.

2) Run simulation:

Have fun. Watch what's happening. Do not interfere unless you notice that they're not following the rules.

Notes:

Expect a lot of chaos in first round. First, because of organizational design, second, because it's first round. That's ok.

People will be coming to you with a lot of questions. It's up to you if you want to answer them, I usually don't, leting them figure it out themselves.

Business Teams need to go to development teams and ask for parts. As development teams are missing brick they need to get them from Infrastructure Team. In case people from Infrastructure are still sitting then it's either business or IT Manager role to get them moving.

You'll be surprised but some buildings eventually will be delivered. The first one most likely after 6 or 7 minutes.

You can validate if buildings are according to specification. It's up to you how detailed specs are and how much building need to according to it (e.g. do you care about exact length of wall).

If one building is ready you can give new specification to the business team.

When timer is out stop simulation. That might be harder than you think, because people will still be trying to "just finish something". Don't let that happen as they'll miss debriefing and might make other focus on what they're doing as well.

3) Debrief:

Gather all participants in one place so that they can see and hear each other. Use mic if needed. First, start with Business Department. Check how many building were delivered and when the first one was ready. Ask for feedback and observations.

Then ask IT manager about feedback. Move to volunteers from Business departments then from IT.

Notes:

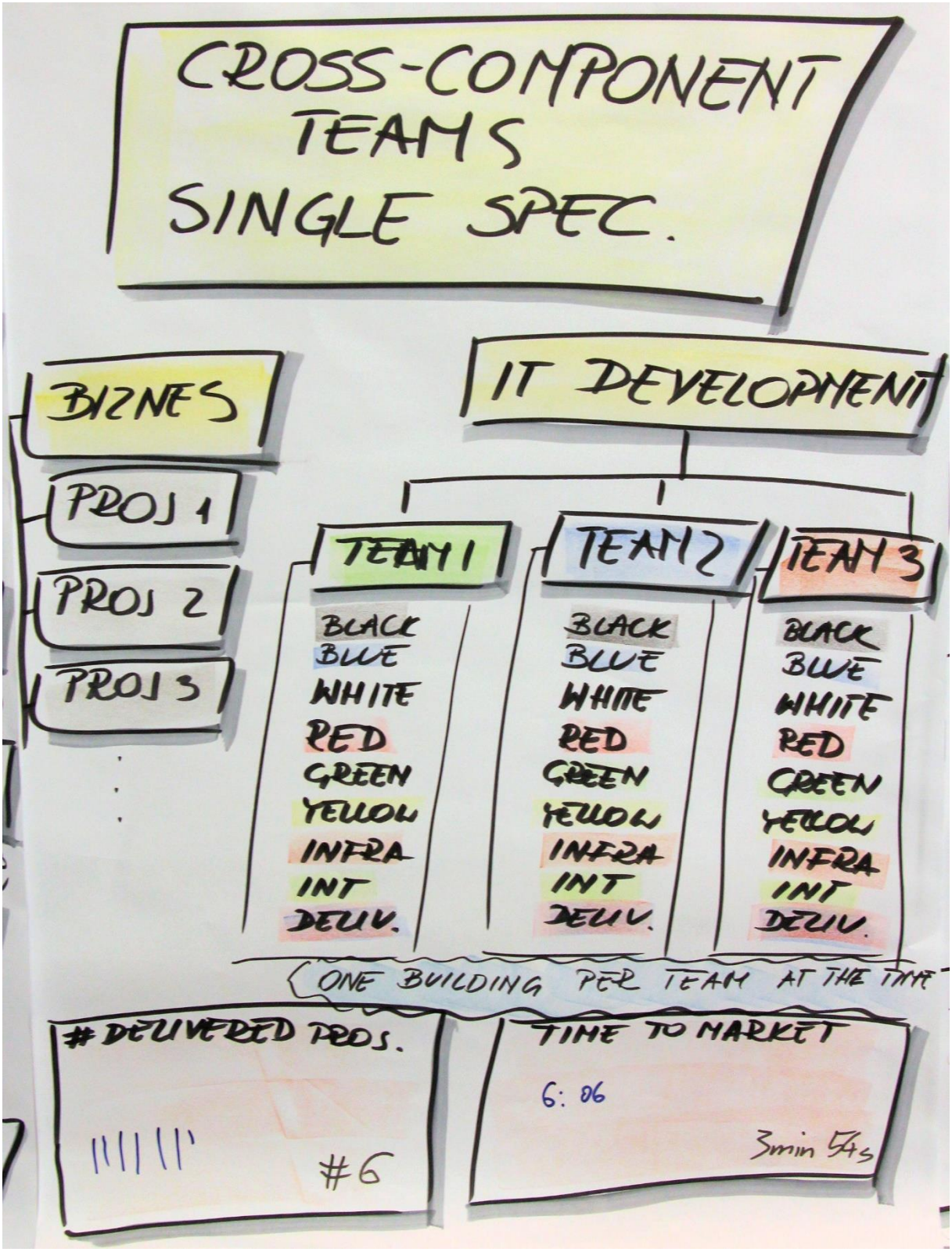
Keep it short and focused. With 50 people the discussion can last forever, so your facilitation skills are crucial.

Underline key takeaways:

- Everyone is busy working on something important from their perspective (well, maybe not the Integration and Delivery team at the beginning of simulation)
- There's lot of chaos
- Business teams are running all the time, being "customers" to IT teams.
- IT teams are working on multiple projects at the same time often changing priorities depending on who's standing over them.
- There's no alignment between Teams, more than often they disturb each other.
- Everything seems to be important. No clear prioritization.
- Huge Work in Progress during and at the end of simulation.
- It's really hard to get anything done and act of doing it is seen as sign of heroism.
- It might happen that business decide to start integrating or testing themselves to speed up development, taking that responsibility from IT
- Some Development teams might be missing bricks. Some might have too much "just in case".
- Managers might switch to micro-managing mode to get things done.
- Someone can suggest that it would be easier if we create a plan up-front. That's exactly how we end up with long-term planning, coordination roles (such as Project Managers) and lack of agility.
- There might be a blame-game. Look for it even in jokes. Ask if that would be so funny if their career depends on the success or failure.
- Integration will be the most painful process.
- You can discuss how does it relate with component teams often seen in organizations and project requiring work in several components to be delivered.

Before you move to next round disassembly buildings and any work in progress from first round to start with the same environment.

Round 2 - Cross component teams, single specialization members



Second round simulates organizations new to agile cross-functional teams, where individuals still keep their specialization ("we've analyst, front-end, back-end and tester"). There's no change to business department.

1) Create cross-functional teams

Explain that we're now creating in IT department cross-component teams. Each team should consist of (at least) one person from each component teams. That's why you needed at least two or more people in your component teams in first round. Give people few minutes (2-3) to figure that out. You can also mention Self-Designing Workshop now.

2) Forbid multitasking

When your IT department is re-designed, clarify to business that every IT team can work only on one project, that's no multitasking rule. Therefore they'll need to decide which business teams are going to get their work done first. You can have a short discussion about how to decide, but at the end Business Manager need to be responsible for making that decision. This is how we came to having single Product Owner for our Product (Lego City).

3) Prepare

Share specifications. You can use the same as before, although I usually mix them up, so the business teams do not work on the same building.

Business manager is still responsible for collecting number of building delivered and time to market for first building. Timebox is 10 minutes again.

4) Run simulation

As in first round. Make sure rules are followed, as there will be pressure to start multitasking.

5) Debrief

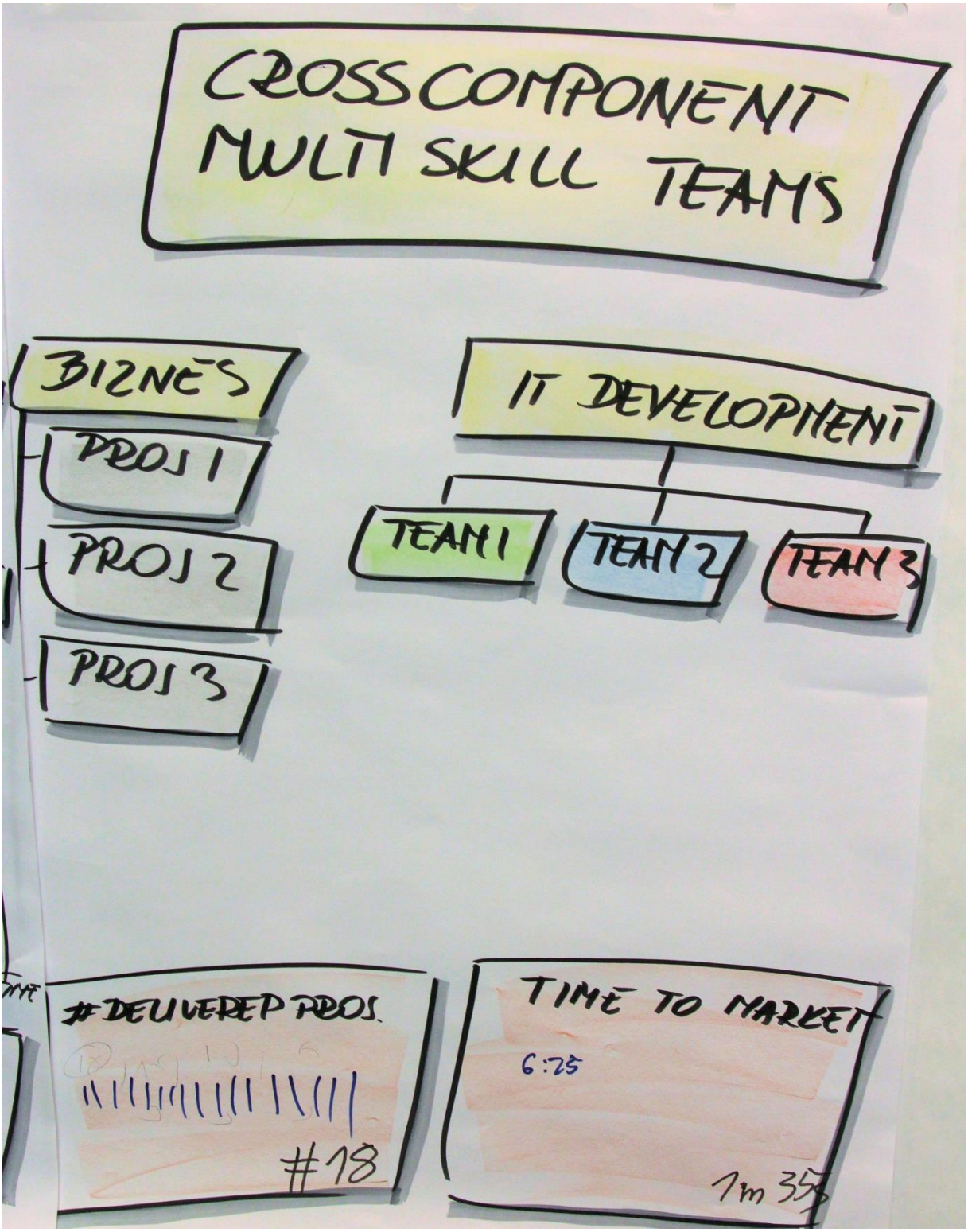
As in first round. **Key observations:**

- Less Chaos. In fact, second round looks boring compared to the first one.
- Prioritization is done in business department in a form of negotiation between PO (Business Manager) and Business teams. Usually based on building complexity (unless there's a better way to define building value).
- Most of business teams are doing nothing for most of the time. There might be situation that some teams don't get their buildings started in that iteration at all. If that's the case point it out and ask what does it means from business perspective.
- Some IT team members will be doing nothing as well (e.g. when current building doesn't need color someone is specialist in). Again, ask for consequences from their organization's' perspective.
- Despite that more building are delivered and time to market is significantly improved. Which means that being busy doesn't equal being effective. Who'd have thought?
- Discuss the consequences of multitasking how they'd impact number of buildings and time to market. (If you're still not sure that multitasking is a bad idea then you could run 2nd round with multitasking and 3rd without to compare results).
- Work in Progress is significantly lower, compared to first round.
- IT teams became self-organizing. Therefore IT Manager's role change from micro-managing to supportive one. (notice, if IT Manager is still trying to micro-manage teams. That can happen sometimes).
- IT teams finally know what they're doing. They can provide feedback to business and propose better solution.
- Integration is just way easier.

- Co-ordinators roles (e.g. Project Managers) are not needed as there's not much artificial dependencies. Ask what organizations can do with these people. If someone suggest "fire them" then inquiry on how people would react to a change that could lead to potential layovers. Most likely someone mention "move them to development" which we will do in third round.
- Problem of team size can be raised. That's a big issue if you've a very narrow specializations and lot of different components and technologies in your product.

Again, disassembly all buildings before starting third round.

Round 3 - Cross component teams, multi-skills members.



Last round will simulate truly agile organization, where team members can help each other in their assignment. Of course learning takes time so we will treat it as a long-time investment. In this workshop we simulate it by shortening last round by two minutes (so the last round last 8 minutes only).

1) Learning new skills

In other words, during last round everyone in IT Department can do everything. People can suggest to create additional team from free members of Business Department and / or splitting current IT teams into smaller ones. That's ok as long as new teams aren't too small (less than 4-5 people), as this is another form of multitasking.

2) Prepare

As in previous round, the only differences are that the timebox is 8 minutes and there might be more development teams and less business teams by now.

3) Run simulation

Don't be surprised if they manage to deliver all buildings in this round

4) Debrief

Key takeaways:

- Huge improvement in number of buildings and time to market despite shorter time of iteration.
- Better engagement. Working is fun (again)
- Team member need to talk to each other (that's when daily standups would be useful)
- Learning takes time. You will not invest into it if you think about employees as contractors and organize work around short term projects.
- Building an effective team takes time, so it's reasonable to keep them long-lived.
- Better cooperation between business and teams.
- Delivery builds trust

5) Celebrate

You're done. Congratulate your participants. It's time to celebrate.

Summary

Congratulations! In 1,5 hour you were able to show major consequences of different organization structures. It's up to you how you conclude the session, however one of good idea is to ask about one thing people could do when they're back to their offices.

This workshop was inspired by Lego4Scrum simulation by Alexey Krivitsky, which we highly recommend. So far, we've run the workshop few times on Agile conferences and local group meetups. We hope to improve it basing on next session and feedback from you. Have you tried the simulation. Did you like it?

About us

From left to right: Tomasz Wykowski, Justyna Wykowska, Krzysztof Niewinski, Mariusz Petlic.

We're a team of Agile enthusiast working together in ProCognita. You can meet us at different conferences and local communities' events where we share our passions with others. Let us know if you would like us to join your event! You can contact us at office@procognita.com