This blog is about our Scrum Essential Training:

When designing my first Scrum training back in 2006, I came to the conclusion that Scrum cannot be taught, but the students have to experience the advantages and power of Scrum and agile within a case study. At that time, I had created the DOSBox programming example which is still in use by me and fellow trainers around the globe for Scrum.org’s Professional Scrum Developer trainings.

Joining Scrum.org as a trainer, I have been using the Zoo webpage example included in the Professional Scrum Foundation training from Scrum.org. The Zoo page does not require programming skills like the DOSBox, which opens the training for Product Owners, business people, and management. I had kept my fingers from Lego even though I grew up in a pile of Lego bricks and my son is stepping into daddy’s footsteps. I thought that Lego was for kids and was not appropriate to teach Scrum professionally.

I changed my mind completely last year: A company approached me for a training on short notice. There was no time to organize the laptops necessary for the Zoo webpage. So I grabbed my son’s Legos, a Lego city backlog from my workmate Peter Schmid and conducted
the training. With tremendous result:

- the people were very motivated working with Legos
- we had a lot of fun during the training (and fun supports learning)
- the attendees could see the advantages of Scrum within the first Sprint and discussions came to the point immediately
- all attendees referred to the Lego case study back at their work place, even months after the training
- the feedbacks were even better than from previous trainings using the Zoo webpage
- and the setup effort was much lower and there were no technical problems like unstable laptops or slow internet connection

Legos have changed my trainer life since then: I have improved the Lego based training and it has been used within our company more than 25 times (in 10 months). We are a growing trainer community of currently 6 trainers in three different countries.

Using Lego for Scrum trainings is not something new: The pages [www.lego4scrum.com](http://www.lego4scrum.com) or [Scrum simulation with Lego](http://www.lego4scrum.com) describe how to conduct brief exercises at the end of a training. But I’m speaking about three full fledged Sprints throughout the training, where the participants build up a complex city and all learning organized around the experience gained in these Sprints!

*This video was produced by a proud team. And shouldn’t they also be proud of the software they are developing?!*

I ran into resistance and doubts about using Lego by other trainers:

1. Lego is for kids, child play. There are many skeptical people sitting in the trainings. Not using a programming example like the Zoo webpage may lead to the impression that Scrum does not work for the complex world of computer programming.
2. Lego does not show the full complexity of Software development, integration and tools.
3. Building houses is the wrong metaphor for Software development.
4. Empiricism, technical debt, self-organization, and emergent design are best shown with a programming example. Other examples like those using Lego introduce an unnecessary level of indirection.

I hope to dispel some doubts with this blog. Let me describe how a training looks like that teaches professional Scrum with Lego:

**Preparation**

Training participants get a brief introduction into agile with the agile manifesto and some
brief exercises around it. They learn some rules around Lego city: To draw roads, rivers, etc. on paper, build houses with Lego, all things are of same scale, about the weather conditions, etc. After less than one hour, they find themselves within a randomly setup team in front of a table, laid out with paper, color pens and of course a pile of Lego bricks.

**Sprint 1**

The team receives a well refined and ordered Product Backlog. They usually create a mess by laying out the cards of the backlog on the table. They are not hanging them on the wall despite they have all equipment to do so. I have observed that especially seasoned Scrum teams start long discussions about the process. Other teams just start, pick up pens, draw the requested roads, the river, a park and some other nice goodies. They also create their first houses and place them in the city. In the second half of the Sprint, all teams are working concentrated on their city. But almost no team asks the instructor whether he likes the city or not. Almost no team tests against the acceptance criteria accurately noted on the backlog cards.

After 20 minutes, there is a harsh stop: All teams have to stop working since the CEO (acted by the instructor) enters the room and attends team 1’s Sprint Review. Now, they learn a lot about the intent of Lego city. Please understand that we do not publish the details here since they should remain a surprise for course attendees! But they learn e.g. that lines marking the river should not cross any road and that the quality of their drawings matter. They learn especially what a Sprint Review really is: The developers learn from business what is useful and what does not pay off. And they learn what Scrum really is: **Reducing time to Feedback!**
In-between Sprint 1 and 2, the attendees hear about the Scrum process, learn how to work with a Product Backlog (regular refinement!), get time to build up a Scrum wall, and elect Product Owner and Scrum Master.

**Sprint 2**

The newly elected Product Owner gets a job: increasing tax income! He or she feels almost instantly what a Product Owner has to do. There are many options to increase tax income and it is up to the Product Owner to use his team to reach the goal. The Development Team practices Scrum at its best. It is challenged by unclear or surprisingly large items in the Product Backlog, by stakeholders stepping in with surprising requests, and it learns the importance of refining the Product Backlog. Almost all teams perform much better than in the first Sprint: They practice development and learn how important repetition is.

Attend a training and learn what went wrong here!
Topics in-between Sprint 2 and 3 are estimation, forecasts, quality, Definition of Done, vision and story mapping. Not necessary to say that they apply all learned techniques immediately to Lego city.

**Sprint 3**
The last Sprint is almost always very smooth and deepens the gained knowledge. Sprint Planning is based on past velocity and with refined and estimated items. The Development Team works according a Definition of Done and the Product Owner creates a vision with a story map. A release burnup shows how many Sprints would be necessary to complete a first working city.

After completing the Sprint, the training ends with a course retrospective, conducted by the trainer and using methods and process promoted by retrospective goddesses [Esther Derby and Diana Larsen](https://www.retrormat.com) and the [Retromat webpage](https://www.retrormat.com). There is additional time to talk about scaling, Kanban, distributed teams, how to introduce Scrum in a company, etc. It is up to the course participants to select the topic they like to talk about.

**Conclusion**
Lego is not only for kids; Lego Serious Play and Lego Mindstorms have paved the way into the professional world. Almost all participants of the training were positively surprised and even thrilled that they can leave their computer-daywork behind them and can create a city with Lego bricks. Only one participant (out of 250) was not amused to play Lego. However, he has found ways to contribute to the team and told me after the training that the example has helped him a lot to understand what Scrum really is. Other attendees have almost setup a live stream home to their kids and many of those kids now want to become developers too.

Building houses has nothing to do with Software development. But we are creating a city, not building houses. There are no detailed building instructions for houses given to the students. It is up to the students to contribute building blocks to the growing city. They have to adapt existing houses and roads to the emerging city. This is exactly what Software development is about!

There are for sure some drawbacks: The concept of technical debt cannot be shown easily. People new to Software development do not learn about the importance of tools and the complexity of development. However, in my opinion, they are outweighed by factors:

- One team had sent two team members to another table. They wanted to integrate their work seconds before the end of the Sprint. And you know what? They did not fit to the scale of the city.
One company suffered from a product management asking too many new features every Sprint. The product management Scrum team attending the training produced a large city with many houses, but the quality was not sufficient. A little kick at the table side let all their building have toppled over. They invested the entire third Sprint to fix that. And they learned the importance of following a Definition of Done!

A team did not accomplish anything in the first Sprint. As I learned, they are not allowed to make mistakes back at their workplace and therefore do everything to not need to write Software.

Almost all teams have a huge stress during their development time in the first Sprints (20 minutes). They learn that finding their own velocity needs time and they usually have no stress in the last Sprint despite they accomplish even more than expected.

A team has had a truck on their roads despite it was not planned to construct any car during this Sprint. They answered the CEO that this is their test equipment: They wanted to check with the truck whether the roads were wide enough.

A fellow trainer and I started an argument at Sprint 1’s Review whether the roads were here for cyclists (my hobby) or for the trucks going to and from my friends gravel pit outside the city. We kept arguing for quite some time until one brave developer stopped our discussion: “Sorry, but this discussion does not help us”. He became Product Owner in Sprint 2!

A manager attending the training together with his team told me afterwards that he has learned more about the dynamics in his team during these two days than all the months before.
I could go on with that list forever. I think that teaching Scrum with Lego is reasonable for introductionary training. The learning experience sticks for a longer time because it is tangible and participants often refer back to the experience gained during the Lego Sprints. I still use the DOSBox or the Zoo webpage when appropriate. But I definitively prefer Lego.