

Metamodel of agile project management and the process of building with LEGO® bricks

Marko Velić, Ivan Padavić

Initium Futuri Ltd.

Kralja Zvonimira 61, 10000, Croatia

{mvelic, ipadavic}@ifnet.hr

Željko Dobrović

Faculty of Organization and Informatics

University of Zagreb

Pavlinska 2, 42000 Varaždin, Croatia

zeljko.dobrovic@foi.hr

Abstract. *Philosophy of agile approach in project management eliminates rigid structures and procedures, so learning and practicing the agile development are not simple tasks. In this paper we present a workflow for teaching and practicing the agile project management methods by using LEGO® bricks. Workflow is a result of the metamodel of the methods - agile project management and bricks building. Final goal is educational workshop creation. As a result of this experiment we give a new perspective on LEGO® as a toy and suggest new more creative ways of play that are in fact learned from agile software development.*

Keywords. Agile, Project management, Metamodelling, Education, Play

1 Introduction

Because of the lack of rigid documentation, framework and procedures [1], agile project management is challenging to teach and practice. As it is relatively young discipline, there are many problems and issues in its implementation. [2] In practice, implementing and using agile methods still needs a lot of experience to succeed [3] and just few organizations are able to adopt their principles in short time so their implementation can last as long as few years. [4]

Still, in the recent times, agile project management methodologies, especially in the domain of software engineering are becoming increasingly popular. [5], [6] Due to this, new methods of learning are welcome. In this paper we present a metamodel of agile project management and LEGO® bricks building concept. Goal of this metamodel is establishing the framework for educational workshop and a workflow for its implementation. Although agile project management principles can be used in various industries and hence workshop that is resulting from this study can be used

for teaching agility professionals from different fields, our main concern here was agile project management in the domain of software engineering. In the next chapter we will give a very short overview of key agile principals and artefacts. Elaborating the agile development in detail is not goal of this paper so reader is advised to follow the references for further information on that. Later we'll discuss models behind agile development and bricks building concepts and elaborate analogies. At the end metamodel, workflow for the workshop and a real-world experiment are presented.

2 Agile project management principles

Inspired by the unsatisfying success rate of IT projects, group of seventeen software engineering experts, advocates of agile project management met up in the hillside of Utah to try to solve existing problems in software engineering methodologies. Result of that meeting is a famous manifest - Agile Manifesto. [7]

Agile manifesto is a document that describes main characteristics of future agile methodologies.

Manifest can be sum up to these four key points:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

Since then, agile methods became widely adopted and eventually big software vendors and project management frameworks started adopting agile principles via implementing agile concepts in their frameworks or releasing special versions of their guidelines. So, we have Microsoft Solution Framework for agile development [8], Rational

Unified Process (RUP) has its own simplified version called Agile Unified Process (AUP) [9] and even CoBIT (Control objectives for information and related technology) found itself within the agile principles or has been implemented in somewhat agile way. [10-16]

Explaining of the all agile methods concepts is out of the scope for this paper so in this chapter we'll cover only some of them that are of special interest for presented metamodel. Those are covered briefly and a reader is encouraged to follow references for further explanations.

2.1 User stories and priorities

User stories represent main unit in planning, development and evaluation of the information system. User story is somewhat similar to use cases or features in traditional software engineering. [2] Basic point of user story is in its "agile philosophy". It is actually a short story describing customer needs. Who, what and why are main questions that are answered in one typical user story. Agile software development often connotes iterative development. As a result of that, user stories are classified by the priority where of course highest priorities are first served

2.2 Team and client

As one of the main postulates of agile development is communication, the most important characteristic of the development team is ability and willingness to communicate with the client. Size of the team depends on the project and different agile methodologies are created for different sized projects. Extreme programming (XP) is suited for small projects [17], for up to three man/months, SCRUM [18] is suitable for teams ranging from five to ten members and so on.

All agile methodologies advise strong influence of the client during the development process. Client himself must be aware of his responsibilities and importance of the whole process. Since client usually is not an IT person, it is hard for him to indicate all his needs at the start of the project and define all the requirements. Due to that, agile methodologies are trying to include client as much as possible to track down the progress and witness the development achievements during and especially after the each iteration.

2.3 Cycles

As is mentioned earlier, agile development process is organized into cycles, so called iterations. [2] Iteration is actually a group of activities performed by the development team that results in program code i.e.

application or system that can be used by client. Team member activities are usually represented by "tasks" which in turn form "user stories". Group of user stories can make up a "feature" or subsystem. User stories or features that form logical group and result in working software are developed during iteration. There are agile software development methods where these boundaries are not so rough and even one completed user story can be delivered to a client for test and usage. [2]

2.4 Retrospectives

Most agile development methodologies include meetings and discussions at the end of the each iteration. These are called retrospectives. During the retrospective, team members are trying to find the answers on following questions: "What have we done wrong?", "Where there any mistakes?", "How can we avoid similar problems in future" etc. [2]

Model of the key agile project management principles is shown in Table 1.

3 Education in agile methods

With the increase in popularity of agile methods, many books and educational programs arose on the market. Some agile methods have their own certifications programs [19], but as we mentioned earlier, experience is critical success factor in implementing agile principles [3], and adopting and teaching of the agile methods principles is a problem. [4]

In present literature there are several approaches and frameworks for education of the agile methods. Since switching of the development team from traditional to agile methods requires adaptation of not only programmers but also their managers, approach is needed that will be able to introduce those concepts to both groups. [20], [21]. If we consider recent trend that agile principles are being implemented throughout the whole organizations and not just development sector, this problem becomes even bigger. [22]

Table 1. Model of the agile project management

Elements of metadata (metadata model)	Objects: Entity class Attributes Rule	Objects: Entity class Attributes Rule	Objects: Entity class Attributes Role	Objects: Entity class Attributes	Objects: Artifact type Drawings
Data about a database (a data model)	Entity class: Module Attributes: Number Name User stories count Iterations	Entity class: User Story Attributes: Number Description Effort Estimate Priority Done Rule: If all tasks belonging to user story are done then the user story is done	Entity class: Task Attributes: Number Description Time Done Rule: Each Task is an element of exactly one User Story	Entity class: Client Attributes: Name Address Phone No. Contact Role: Each client must have at least one contact person/product owner	Artifact type: Burn down chart Drawings: X axis Y axis
Data about real world things	Module number: 6 Module name: Sales management module User stories count: 21 Iterations: 2 and 3	Number: 156 Description: Sales manager must... Effort: 20 points Estimate: 5 man/days Priority: 3 Done: No User Story #156: Sales manager must have a dashboard showing sales trends in selected period	Number: 478 Description: Task Entity class: Task Rule: Each Task is an element of exactly one User Story Time: 2 days Done: Yes Task #478: To create GUI for User Story #156	Client ID: 9 Client Name: Sales Angels Inc. Address: Cloud no. 9 Phone No: 0800-DSF-FORCE Contact: Mrs. Jane Doe	Iteration Number: 3 Duration: Four weeks Burn down chart: Burn down chart #3 X axis: One month Y axis: 500 points
Real world things	Module #6: Sales management module			Client #9: Sales Angels Inc.	Artifact: Burn down chart for third iteration

4 LEGO® - just a play?

Although the process of building things with LEGO® bricks is primarily just a play for kids, bricks are an interesting medium for transmission of the artistic or educational messages in the world of adults. A lot of adult fan clubs and communities exist and results of their activities are not just hobby-like fun. Idea of trying to learn adults about some concepts from business and management is not new. Even The LEGO Group Company supports its official “LEGO Education” [23] for schools and “LEGO Serious Play” educational program [24] for business managers.

This kind of play is selected here for three main reasons. First reason is the similarity between software development process and LEGO® building concept. That will be covered in following chapters. Second reason is the fact that many of the people actually understand that building process and due to that there won't be problems of adopting that technique as a working technique for an actual workshop on agile project management. Third reason is somewhat practical - if we gather some people on the workshop and those are all programmers with knowledge and experience of different programming languages or even not just technical but also business people like IT managers or totally non-IT people, building with bricks will be their “common programming language” they all know very well and initial background gaps will be overcome.

5 Analogies

Software development and bricks building processes have few analogies. If we consider software development as a process of interconnecting software components via specific interfaces than we can say that in brick building process various parts and brick types with their specific connecting elements are the first analogy. This is shown on Figure 1.

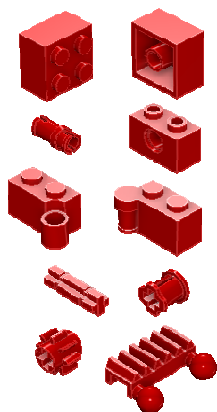


Figure 1. Different parts with different “interfaces”

In traditional software development e.g. waterfall approach, development process is divided into phases with covering documentation. Typical LEGO® building set comes with building instructions that depict the building process in details. Furthermore in larger sets building instructions are divided into few hard-copy books showing instructions for specific sub-models. In addition, as can be seen on Figure 2, bricks that are needed for a specific phase of building are divided into separate plastic bags, and each step of the building process shows which parts are needed and how many of the kind.

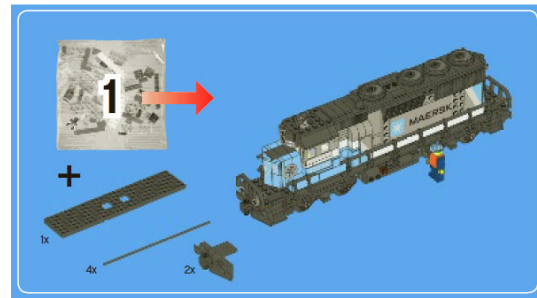


Figure 2. Fragment from the official LEGO® building set

Even errors are somewhat covered in bricks building process. When correcting an error or disassembling the build, one can use specially designed pieces made for easier parts separation - the brick separator, shown on Figure 3. That is actually a “debugging tool” of the bricks building process. We can say that classical bricks building process has much in common with traditional or waterfall software development process.

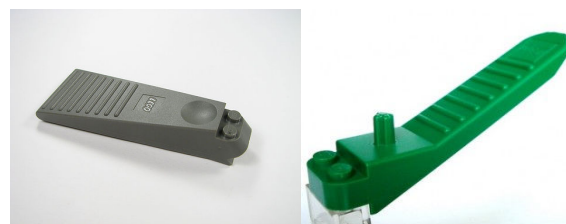


Figure 3. “Debugging tools” - Brick separators

6 Metamodel

After concluding that there are analogies between software development, software engineering principles and bricks building process, logical question that arises is whether similar analogies can be found with agile project management principles. If so, one could use them for establishing a framework for educational workshop on agile project management.

Suppose that one IT company is working for a client on one big corporate information system. Within the IT company there is a programming team consisting of programmers working in pairs. Whole project is, by the agile principles, divided into three

iterations and after each iteration team conducts retrospective and working software is released. During the iteration, team is working on different tasks with the goal to finish specific user stories that are selected for the iteration by their priority level. Given user stories form software features that are used by client's employees - users in different segments of the client's company. These features are de-facto subsystems and all those subsystems form one big information system. All time during the development process one employee from client's company works and communicates with the development team. This is product owner and his responsibility is to monitor and steer the progress of the development team.

7 Workflow

In this chapter we present workflow derived from our metamodel shown in previous chapter. Goal of this workflow is to serve as a guideline for designing the educational workshop for teaching agile development concepts through play with bricks. To simulate the situation on the market we will divide workshop participants into two groups. That correlates to two companies on the market. Inside both teams, participants are instructed to work in pairs. This is analogy with extreme programming and similar simple agile methodologies. Each team has dedicated workshop moderator who will play role of the product owner thus communicating with team members and helping them to understand and complete their tasks. Project for our teams will be building of the LEGO® city - the system.

City is divided into themes - subsystems. Themes are consisting of models - applications. Next step is trying to solve one of the main points of being agile and that is minimizing the documentation. For that purpose we suggest that initial documentation for workshop tasks remain simple by giving just a few photos of possible solution - model built from bricks without detailed instructions. Further, we can suggest few short requirements that describe the model i.e. its features. Here, task card represent user requirement. Picture of the possible model realization represents

prototype or e.g. GUI (Graphical User Interface) sketch and short requirements represent user stories. Sample of this initial documentation is shown in Table 2.

Each requirement has estimated time that is needed for building it and due to that complexity points can be given. Going in more detail, user stories i.e. requirements are naturally breaking down into tasks at execution time - finding the right bricks, connecting the bricks etc. Regarding the client's employees who will eventually use the software, analogy in building game is LEGO® mini-figure, so each model must be built in mini-fig scale. After preparation of all materials, for successful workshop we need the right process and that is where agile principles can be used in its full potential. At the start of the workshop moderator draws project backlog and models represented with post-it papers are to be put on the board. Whole building process is run in iterations.

Depending on the model size, number and complexity, time required for building them can span across one or more iterations. At the end of the each iteration retrospective is done with the help of the team moderator. During retrospectives, moderator checks which requirements are satisfied - testing. Based on the building progress, moderator calculates collected points and draws progress on the burn down chart. Also necessary changes on the project backlog are made - to move post-its that represent finished models to "done" section, and new ones that are planned for building in next iteration to "doing" section etc.

Depending on the number of models and given time, a few iterations can be organized. At the end of the process, all points are calculated and winning team is announced. In that way participants can be motivated and competing spirit can be established.

Various scoring systems can be utilized for further motivation of the participant e.g. every completed theme (subsystem) can bring extra points etc.

Metamodel on the data model level that represents our vision of this and analogy with brick building process is shown on Figure 4. Pointing arrows on the figure actually define the workflow.

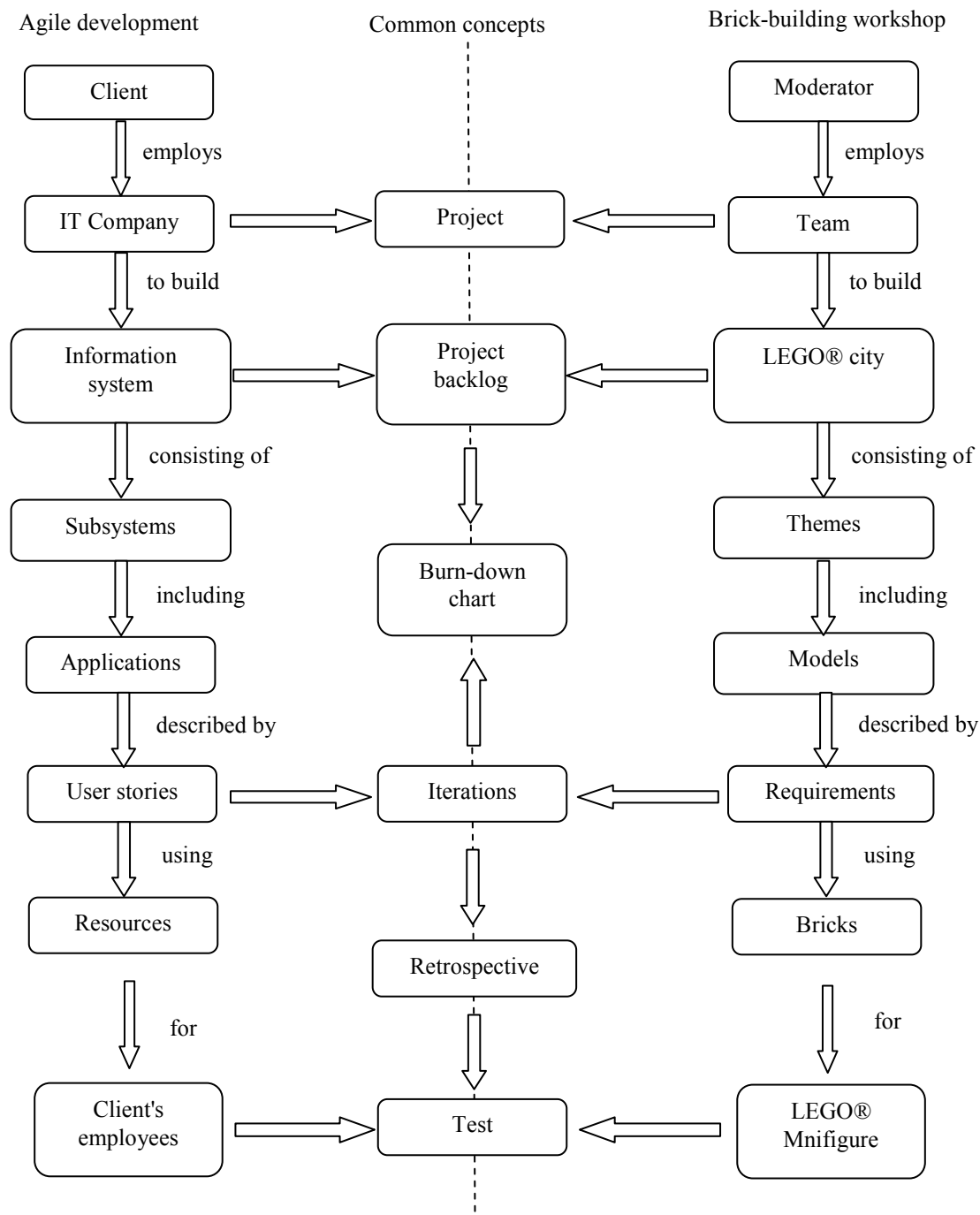


Figure 4. Metamodel and the workflow

8 Test

To test this model for educational workshop, one workshop was designed and held at the 23rd CASE conference, Zagreb, Croatia, June, 6th, 2011.



LEGO models were prepared in advance and brief "user stories" were made targeting main functionalities models must have, with two or three pictures of the models. Then models were

disassembled and bricks were assorted in boxes. Participants were divided into two competing teams and inside teams attendants had to build in pairs (analogy to XP - extreme programming). They had to build a LEGO city (information system). Themes (subsystems) were given at the beginning - an airport, farm, building yard and a gas station. Every theme had corresponding models, eg. airport had an airplane, helicopter, tower etc. Building process was run in

iterations (4 * 10 min) and at the end of the each iteration retrospective was ran. During retrospectives, development process was followed on a project backlog and burn-down chart. Participants were both technical and business people. They learned basic

concepts of agile project management like retrospectives, iterations, burn-down charts, backlogs. Also some general "programming" problems were covered like gold plating.

Table 2. Examples of the initial documentation/instructions for the workshop

Model	Airplane	Crane
Points	15	5
Estimated time	3 iterations	1 iteration
Prototype		
Theme	Airport	Construction Yard
Requirements	<ol style="list-style-type: none"> 1. Plane must have wheels and motors so that it can take-off and land (2 points) 2. Plane must have big wings to fly and small wings for stability (3 points) 3. Pilot cabin is essential - pilot figure must fit inside (5 points) 4. Plane must have space for passengers (5 points) 	<ol style="list-style-type: none"> 1. Crane must be standing on a high platform (1 point) 2. Cabin of the crane must rotate and fit for construction worker figure (2 points) 3. Crane must be functional with working mechanism to carry cargo (2 points)

There were 24 participants from various Croatian companies and after the workshop simple survey is conducted and results were the highest marks given by the participants. Example photo from the actual workshop is presented on Figure 5.



Figure 5. Actual workshop on agile development

9 Conclusion

In this paper we presented a metamodel of the agile project management principles and LEGO® bricks

building process. The goal of the metamodel is a workflow that can be used for designing of the educational workshop for teaching and practicing the agile principles. Bricks are used as a medium to transmit agile principles practices to participants with various background knowledge and experience. Model of the workshop is designed and tested on one ICT conference in Zagreb, Croatia including participants that had either technical or business background. Workshop was honored with the highest marks given by the participants.

Considering the success we had with this workshop we can think of new use of the LEGO® bricks in a way that official LEGO® set do not need to be "waterfall like" thus introducing more creative way of play and communication "between the client and developer" can lead to funnier way of playing for parents and kids.

References

- [1] Jim Highsmith **Agile Project Management: Creating Innovative Products**, Addison Wesley, USA 2004.

- [2] Padavic, Ivan; Velic, Marko; Ljubobratovic, Dejan. **Agile development of software products**, Proceedings of the 10th International Scientific – Professional Symposium INFOTEH@-JAHORINA 2011, Jahorina, Bosnia and Herzegovina, 2011.
- [3] McAvoy J., Sammon D., **Agile Methodology Adoption Decisions: An Innovative Approach to Teaching and Learning**, Journal of Information Systems Education, Vol 16, 2005.
- [4] Qumer A., Henderson-Sellers B., **A framework to support the evaluation, adoption and improvement of agile methods in practice**, The Journal of Systems and Software 81, 2008, pp. 1899–1919.
- [5] Dyba T., Dingsøyr T., **Empirical studies of agile software development: A systematic review**, Information and Software Technology, 2008, Vol 50, pp.833–859.
- [6] Laanti M., Salo O, Abrahamsson P, **Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation**, Information and Software Technology, 2011, Vol. 53, pp.276–290.
- [7] **Agile Manifesto**, available at <http://agilemanifesto.org/>, Accessed: 01th April 2012.
- [8] **Microsoft, MSF for Agile Software Development Process Guidance**, available at <http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=5365>, Accessed: 01th April 2012.
- [9] **The Agile Unified Process**, available at <http://www.ambysoft.com/unifiedprocess/agileUP.html>, Accessed: 01th April 2012.
- [10] Ambler S.W., Kroll P, **Applying agile and lean principles to the governance of software and systems development**, Rational Software, IBM Software Group, IBM Corporation 2011.
- [11] Hobbs G., **Forming a Practice Perspective for Agility in Information Systems**, 19th Australasian Conference on Information Systems, 3-5 December, Christchurch, Australia, 2008.
- [12] IT Governance Institute, **COBIT 4.1: Framework, Control Objectives, Management Guidelines, Maturity Models**, IT Governance Institute, Rolling Meadows IL, 2007.
- [13] Hobbs G., Scheepers, R, **Identifying Capabilities For The IT Function to Create Agility In Information System**, Pacific Asia Conference on Information Systems (PACIS), PACIS 2009 Proceedings, Association for Information Systems, 2009.
- [14] Mahnic V, Zabkar N, **Using COBIT Indicators for Measuring Scrum-based Software Development**, WSEAS Transactions on Computers, Issue 10, Volume 7, October 2008.
- [15] **IS Auditing Guideline System Development Life Cycle (SDLC) Reviews Document G23**, available at www.isaca.org, Accessed: 01th March 2012.
- [16] Hardy G, **Using IT governance and COBIT to deliver value with IT and respond to legal, regulatory and compliance challenges**, Information Security Technical Report 11, 2006, pp. 55 – 61.
- [17] Beck K, **Extreme Programming Explained: Embrace Change**, Addison-Wesley, Massachusetts, USA, 1999.
- [18] Sutherland, J., Viktorov, A., & Blount, J, **Distributed Scrum: Agile Project Management with Outsourced Development**, Agile 2006, international Conference. Mineapolis, 2006.
- [19] **Scrum Alliance**, available at <http://www.scrumalliance.org/>, Accessed: 01th April 2012.
- [20] Moe N.B, Dingsøyr T., Dybå T, **A teamwork model for understanding an agile team: A case study of a Scrum project**, Information and Software Technology, 2009.
- [21] Chan F.K.Y., Thong J.Y.L., **Acceptance of agile methodologies: A critical review and conceptual framework**, Decision Support Systems 46, 2009, pp. 803–814.
- [22] Kettunen, P, **Adopting key lessons from agile manufacturing to agile software product development - A comparative study**, Technovation, 2008.
- [23] **Lego education**, available at <http://education.lego.com/>, Accessed: 01th April 2012.
- [24] **Lego Serious Play**, available at <http://www.seriousplay.com/>, Accessed: 01th April 2012.