AGILE SOFTWARE DEVELOPMENT METHODS FOR PROTOTYPING OF INNOVATIVE CONCEPTS IN MANUFACTURING AUTOMATION

NASSER JAZDI-MOTLAGH

Institute of Industrial Automation and Software Engineering – University of Stuttgart
Pfaffenwaldring 47, 70550 – Stuttgart – Germany
nasser.jazdi@ias.uni-stuttgart.de

VICENTE F. DE LUCENA JR.

CETELI / PPGEE / PPGI – Universidade Federal do Amazonas - UFAM
Av. General Rodrigo Octávio Jordão Ramos, 6200, Setor Norte, Coroado I -- Manaus, AM, Brasil
vicente@ufam.edu.br

Abstract— Many new challenges have arisen in manufacturing automation such as the integration of cyber-physical system technologies, distributed development and ever greater expectations of quality, flexibility, and customer-specific aspects requirements. There is a demand for new paradigms of development to meet the global competition. On the center of this discussion is the development of software for such applications and the existent development models. This article discusses innovative concepts for the development of software for advanced manufacturing systems, namely the use of agile development methods. Moreover, an experiment, where a prototype of a complete automated manufacture plant is constructed. This experiment was conducted by students in a simulation of a real-life demand. The obtained results, as well as the qualitative evaluation of the experiment, will be commented at the end of this work.

Keywords— Manufacturing Automation, Agile Methodologies, Scrum

Resumo— Muitos novos desafios surgiram na automação de manufatura, como a integração de tecnologias de sistemas ciberfísicos, desenvolvimento distribuído e expectativas cada vez maiores de requisitos de qualidade, flexibilidade e aspectos específicos do cliente. Há uma demanda por novos paradigmas de desenvolvimento para atender à concorrência global. No centro desta discussão está o desenvolvimento de software para tais aplicações e os modelos de desenvolvimento existentes. Este artigo discute conceitos inovadores para o desenvolvimento de software para sistemas avançados de manufatura, a saber, o uso de métodos de desenvolvimento ágeis. Além disso, um experimento, onde um protótipo de uma planta completa de fabricação automatizada é construído. Este experimento foi conduzido por estudantes em uma simulação de uma demanda da vida real. Os resultados obtidos, bem como a avaliação qualitativa do experimento, serão comentados ao final deste trabalho.

Palavras-chave— Manufacturing Automation, Agile Methodologies, Scrum

1 Introduction

There have been many innovations in manufacturing automation, in particular, cyber-physical systems, recently. Integrating cyber technologies makes products internet-enabled and enables innovative services: for example, cost-effective and efficient web-based diagnostics, maintenance, and operation (Scheuermann et al., 2015). This new standard leads to the possibility of implementing new business models, operating concepts and intelligent controls that give pride of place to the user and his individual needs (Givehchi et al., 2013), (Givehchi et al., 2014). Many new concepts for automation in manufacturing can be conceived but require a lot of system development efforts to prove their value (Santos et al., 2016).

In practice, software development of automated manufacturing systems can become intensive, complicated and costly (Carpanzano et al., 2004). However, innovative approaches to manufacturing automation need to be prototyped to check and refine new promising concepts (Vallee et al., 2009), (Amaral et al., 2004). Identifying a process and development methodology for the quick evaluation of new concepts and the utilization of the creativity and genius of the developers to a maximum is highly relevant.

This paper discusses how so-called agile development methods can be deployed to organize software production teams in automation projects. Those methods are among the most promising approaches in diverse application areas (Ringert et al., 2012). We believe that those methodologies can be engaged to motivate developers towards new ideas and quick development of complete systems or prototypes in the automation world.

This paper is organized as follows: section 2 presents the actual situation of the known methodologies for software development. In section 3 a brief discussion of the new challenges and requirements in manufacturing automation is presented. In section 4 the application of the Scrum methodology in the manufacturing automation is described. Section 5 presents a real-life experiment where the proposed approach could be tested, and finally, in section 6, this work is summarized, and the proposed future work will be presented.

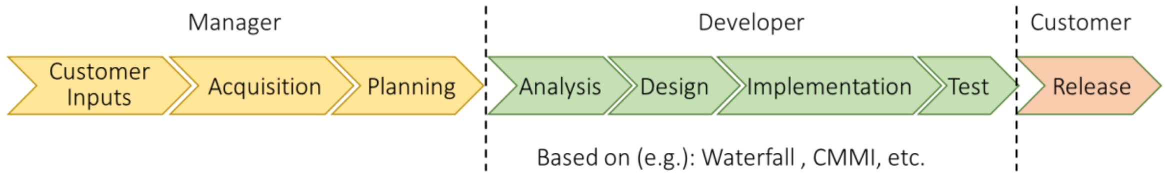

Fig. 1. Structured Software Development

2 State of the Art

Many software methodologies have been introduced in the software development field of manufacturing automation applications. Software developers are constantly trying to refine the old methodologies and to get more effective and new system development methods. The Waterfall Method is known as the most common and traditional system development model. It has had an enormous influence as a general approach to developing control and information systems (Lukman et al., 2010).

Figure 1 displays the typical phases in the software development cycle. The management and development take place after the customers have provided their inputs. Methodologies such as CMMi, Waterfall, V-Model, organization models of ISO 9001, or others are utilized for quality assurance purposes.

Those conventional methods emphasize a clear structure of the development process and its documentation. Despite the proven track record of conventional methodologies for the software engineering of manufacturing automation systems, there are also weaknesses. Several system development methods are already very hard to work with, since these methods need flexibility, but are rather rigid due to their systematic approach (Dong and Kun, 2010), (Börjesson and Mathiassen, 2004), (Phillips et al., 2013).

To address these challenges, we must focus on the objectives: e.g., flexible and lean software development processes, integration of customers into the process, and considering technical and social problems in software development. For this purpose, the agile methodologies which will be described in the following chapter were developed.

Figure 2 presents a study of a consulting company, where projects managed with agile methodologies are more successful and less problematic than those which are managed with the conventional waterfall methodology.

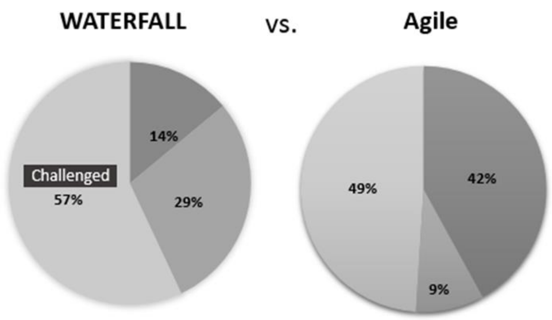

Fig. 2. Structured Software Development

2.1 Agile Methods for Software Engineering

The goal of agile software development is to make the software development process leaner and more flexible than with the traditional process models. Therefore, the agile approaches offer a welcome relief from traditional development processes; they also help to counteract the desire to make software development a mindless, repetitive process. As their creators and proponents would acknowledge, they are not a universal solution at either the mechanism or the organism levels, compared to the earlier attempts at codifying software. They do reassert the humanness of the business and have shed light on a neglected aspect of building systems: the way that people work together. Another aspect of agility that is refreshing is the degree to which proponents are willing to listen to each other and agree that yes, your idea might work, even though it is not my idea. The directing of energies toward bridging and building is a very positive sign for system development. According to Coldewey (Coldewey et al., 2000) “most lightweight processes” substitute interpersonal communication and teamwork for the extensive documentation and formal signoffs that are required by heavyweight processes. It is important to note, however, that agile does not mean chaos.

In fact, agile methods are becoming even more popular in the world of system development in many different areas. Other words for “agile” are lithe, mobile, flexible and dynamic. The Agile Alliance has a set of values and principles that describe the ultimate agile way of developing software. The description is not a method itself, but more of a guideline and a way of looking at software development.

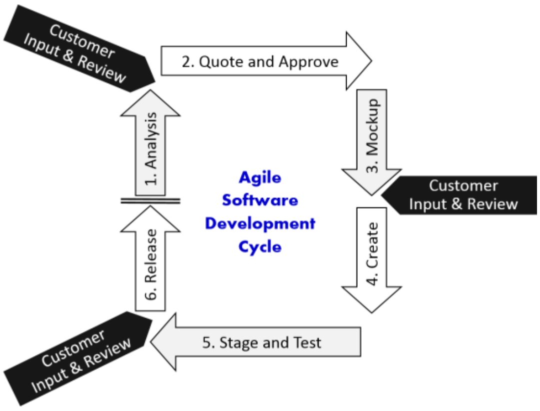

Fig. 3. Principle Agile Software Development

A few methods are more or less agile; therefore, you could say that the agile concept is like an umbrella. Methods like Extreme Programming (XP), Dynamic Systems Development Method (DSDM), and Crystal, for example, are likely to be found under this agile umbrella (Dong and Kun, 2010). Software development is, according to an agile modeler, a flexible project that can react to changes in a surrounding environment. A constantly changing world demands development processes which can handle changes as part of their daily routine (Cockburn, 2006). Figure 3 contains the fundamental principles of agile software development. In agile methodologies, the customer is integrated into the process more often, and the process is, by definition, more interactive. This new approach promises a more flexible and open development, especially in the event of vague requirements.

In the last few years, there have been a lot of discussions about the agile methods and how they could be used practically, and why it is beneficial to move towards them (Amaral et al., 2004). The agile practices are becoming more common and are used more often in industry. Many software academics and practitioners are carrying out research to adopt agile methods and apply in different domains (Graml et al., 2007), (Hametner et al., 2012). After starting the agile movement in 2001 and establishing the agile alliance, different methods were officially introduced to the software development society. All of them claim that they handle requirement changes even in late phases of the project while still implementing the project on time and within the budget without compromising the quality. In fact, that is what is needed to develop modern automation systems. The software development for a "smart factory" is very challenging and needs some special aspects that would be considered by agile approaches (Hernández-Reveles et al., 2016). To test this hypothesis, we chose a suitable agile approach, namely Scrum, and applied it practically to develop an automation prototype and evaluate the obtained results.

3 Challenges and Requirements in Innovation Development in Manufacturing Automation

The key challenge for manufacturing automation, in particular, includes the use and integration of new technologies, while maintaining or enhancing the competitiveness and the optimization of cost and profit (Boschi et al., 2016). The manufacturing industry will go through a profound change in the next few years: the transformation of software-intensive, increasingly networked products into services (Givehchi et al., 2017).

Figure 4 provides an overview of the key challenges of manufacturing which are pointed out by industry (Acatech, 2013). Many of those innovations are triggered by rather vague ideas such as the wish for more networking, flexibility or new business models due to the availability of affordable technologies. Many of those aspects are difficult to specify in scientific terminology, or even as part of an industrial requirement analysis and statement of work for developers. Nonetheless, many automation developers are inspired by novel approaches very much and are confident of achieving it (Spichkova et al., 2015). Fig 4 outlines some of these new industry requirements and challenges.

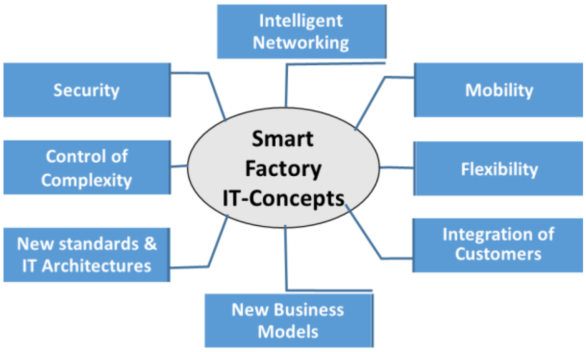

Fig. 4. Challenges and Requirements

The following challenges are particularly important to implement these trends:

A. Intelligent Networking

Industrial automation systems and equipment, internal logistics systems and resources are systematically linked by cyber technologies, such as wireless and Ethernet-based wired communications services, smart actuators, sensors and telecommunication technologies. Direct access to parent processes and services are thereby sustained. Completely new value creation and business models have emerged that support optimal resource utilization and intelligent control (Schlick, 2012).

B. Mobility

Mobile devices such as smartphones and tablets have been part of automation for a long time. They provide access to the processes and services of industrial automation systems independent of time and place. Its use has resulted in a new dimension in the diagnosis, maintenance, and operation of these systems (Babulak, 2009).

C. Flexibility

Manufacturing automation enables high flexibility in the development, diagnostics, maintenance, and operation of automated systems. You can select the best and most suitable solutions from a large pool of component, module and service suppliers when developing such systems. Diagnosis can be carried out in part by the user. Access to "Big Data" automation helps here. The information can be retrieved, used and linked on demand so that an automated diagnostic, or at least one that could be performed by the users themselves, can be implemented. Spare parts can be ordered automatically from the most favorable manufacturers, thereby counteracting skill shortage problems.

D. Integration of Customers

Manufacturing automation will make it possible to adapt products to the specific and individual needs of customers. Automated systems for the future will adapt to the needs and abilities of users of all ages. A modern ticket vending machine, for example, provides various operating procedures so that people with different disabilities can use it. Automated systems will support people in all situations and at various stages of their lives so that people are always supported, healthy and mobile.

E. New Business Models

Production will be distributed and flexible in the future. New development processes, infrastructure, and services will be created. Products will be modular and configurable so that the product can be adapted to specific requirements.

F. New Standards and IT Architectures

Future manufacturing automation will require flexible and adaptable new IT standards such as wireless components and Fieldbus coupling modules. These standards will reduce the development effort for new applications significantly. Furthermore, these systems will require new IT architectures which support the integration of new modules.

G. Control of Complexity

Product individualization will be an essential aspect of future manufacturing automation systems. It will also be associated with risks and challenges that will need to be weighed carefully. Customized products and the related focus on customers and their specific needs will require new value-added concepts because the complexity of the requirements will increase disproportionately. Increasing efforts for configuration of automation systems due to the availability of ready-made systems components will also have to be addressed.

H. Security

Information security in manufacturing automation will be a particular challenge. Providers and operators of these systems must ensure that their products are secure. The data and information of the machine must be protected from unwanted accesses. For new products and systems, the security aspects must already begin in the design phase.

But smart manufacturing automation also brings many challenges that require extensive research. Many questions arise, such as: How can the reliability and safety of these distributed products be determined and how will they be certified? Another critical issue is the subject of privacy and security. Assurance must be given that expertise and privacy will remain protected and unaffected. To this end, new concepts and technologies are needed that enable many groups and units to work together with trust. Furthermore, ethical, legal, and social issues will have to be redefined.

These new challenges require new development paradigms and methods that can satisfy their requirements. It becomes more and more difficult to fully oversee all the aspects of the design and implementation in the initial phase of the process. An essential aspect of decentralized, global development is the high dynamic level of processes and continuous changes in requirements, as these cannot be defined beforehand. That is, on the one hand, due to different interpretation of the requirements in distinct cultures and, on the other hand, due to additional input. The development required paradigms need to meet these challenges. The agile methods presented below represent a very promising methodology in this respect (Li et al., 2013), (Harrison et al., 2006).

It becomes evident from the listed challenges and requirements that the implementation of new software to that effect needs to utilize the creativity of the developers to a maximum.

4 Scrum Methodology Towards the Engineering of Manufacturing Automation

Simple and efficient communication contributes to the success of project development not less than the technologies used. Therefore, the developers work in agile projects at the customers' site and are in constant contact with them. Details can be clarified quickly, and misunderstandings are rectified immediately. Agile development methodology, especially the so-called Scrum, have the potential to activate the spirit of developers promisingly. The Scrum process is briefly described in the following section.

The Scrum management process divides a project into short (30 day) iterations, or "sprints", where, during each sprint, there are short (15 minute) meetings ("Scrums") between the development team and the team management to track progress, note current and imminent obstacles to progress, and decide what work to focus on until the next meeting. Figure 5 resumes the entire process. The development goals are kept constant during the sprint. Each sprint's goals are negotiated just before the sprint. Thus, the project goals adapt iteratively. The Scrum approach has been developed for managing the system development process.

Scrum does not require or provide any specific software development methods or practices to be used. Instead, it requires certain management practices and tools in the various Scrum phases to avoid the chaos caused by unpredictability and complexity (Schwaber and Beedle, 2001). It is an empirical approach applying the ideas of industrial process control theory to system development, resulting in an approach that reintroduces the ideas of flexibility, adaptability, and productivity.

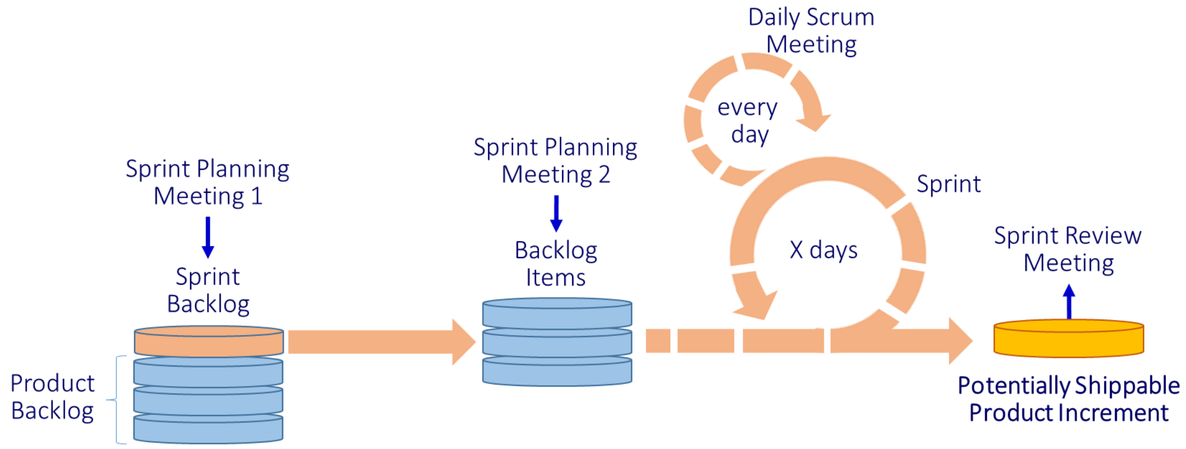

Fig. 5. Scrum Cycle for Sprints

The main idea of Scrum is that system development involves several environmental and technical variables (e.g., requirements, time frame, resources, and technology) that are likely to change during the process. That makes the development process unpredictable and complex, requiring flexibility of the system development process for it to be able to respond to the changes. Because of the development process, a system is produced which is useful when delivered (Schwaber and Beedle, 2001). It helps to improve the existing engineering practices (e.g., testing practices) in an organization, for it involves frequent management activities aimed at consistently identifying any deficiencies or impediments.

Scrum is a method that aims to help teams to focus on their objectives. It tries to minimize the amount of work people have to do tackling less important concerns. Scrum keeps things simple in the highly complicated and intellectually challenging software business environment (Schwaber, 2004). It is based on two pillars: team empowerment and adaptability. Team empowerment refers to teams being relatively autonomous. Scrum may empower teams to become completely self-organizing and self-regulating.

Scrum admits that every team member is expected to understand every problem and all the steps in developing a system to solve them that is a significant limitation of this method since it is hard to believe that all team members will have such broad knowledge. That is why Scrum is most suitable for small teams of 5 to 10 members, so if more people are available, multiple teams should be formed. Scrum provides the agile management mechanisms and is a suitable mechanism for developing a dynamic system.

5 Experimental Validation of the Methodology for Manufacturing Automation

It is not easy to give an overall evaluation of Scrum. In this project we considered a set of data from a consolidate process models (V Model), using the same KPIs (Key performance indicators) and the same application. In our case study, we compared Scrum with this established process model.

We compared two teams, each of them had 8 members. One team developed the application after V-Modell, the other one used Scrum. As we mentioned, the scrum team developed the application in 10 weeks, also 3 weeks earlier. We also mentioned that the scrum team had a better intercommunication and worked whit more efficiency. A project for a student group was defined to investigate the possibility of using the Scrum method in developing one application in the context of manufacturing automation. The group consisted of 8 developers. At the beginning of the project, the requirements were formulated very roughly. Specifically, they were:

- A smart factory application must be implemented.
- Two LEGO Mindstorms robots are available and must be used.
- The application must be implemented in 10 weeks.
- The Scrum method must be used as the development method.
- The group should decide all other demands.

One of the most important challenges in this project was the fact that the requirements were not defined completely in advance. The developer team got the order to simulate the structures and functionalities of the smart factory. The programming language to be used and the exact implementation had been left to them. In agreement with the "customer," who represents the client, they were to clarify the feasible parameters and functions. The entire process of development had to be done with the management ideas and structures of Scrum.

The team carried out the project, with all its requirements, in 10 weeks. They developed control software for a smart factory consisting of a server application, a client application, an SQL database, and an Android app. The software included more than 25 classes.

The smart factory was simulated by a map (Figure 6). It consists of multiple workstations and tracks. The robots took the pieces and brought them to the workstations. The route depended on the production planning. Based on the production planning, the robot knew which task steps had to be executed. The robots negotiated to define an optimized and efficient workflow.

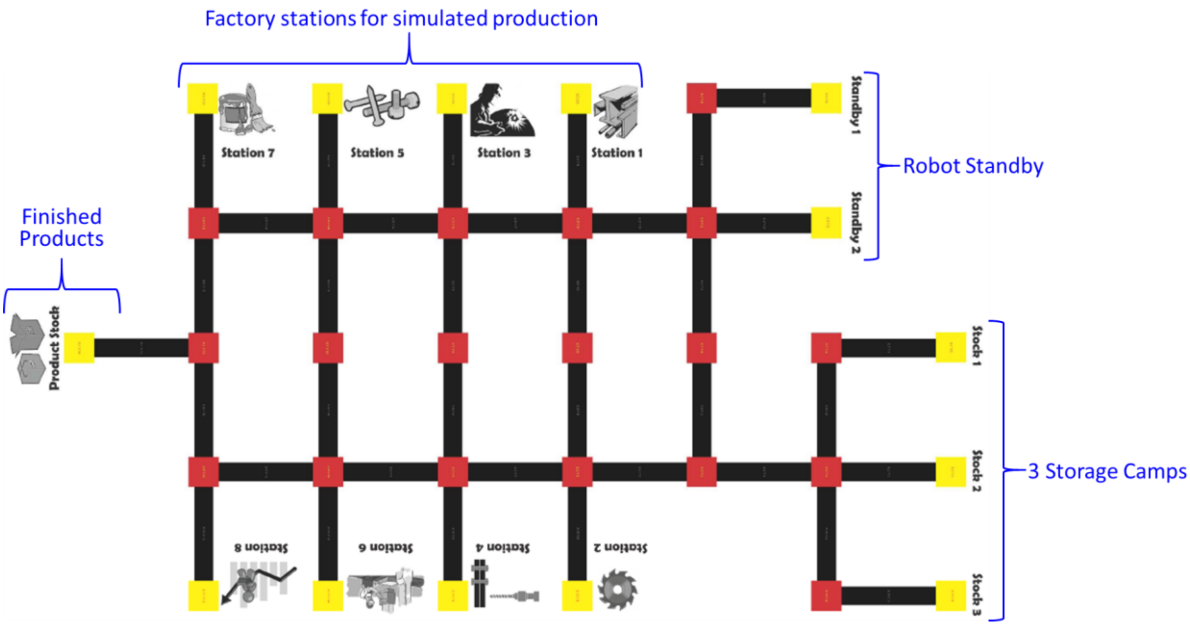

Fig. 6. Factory structure

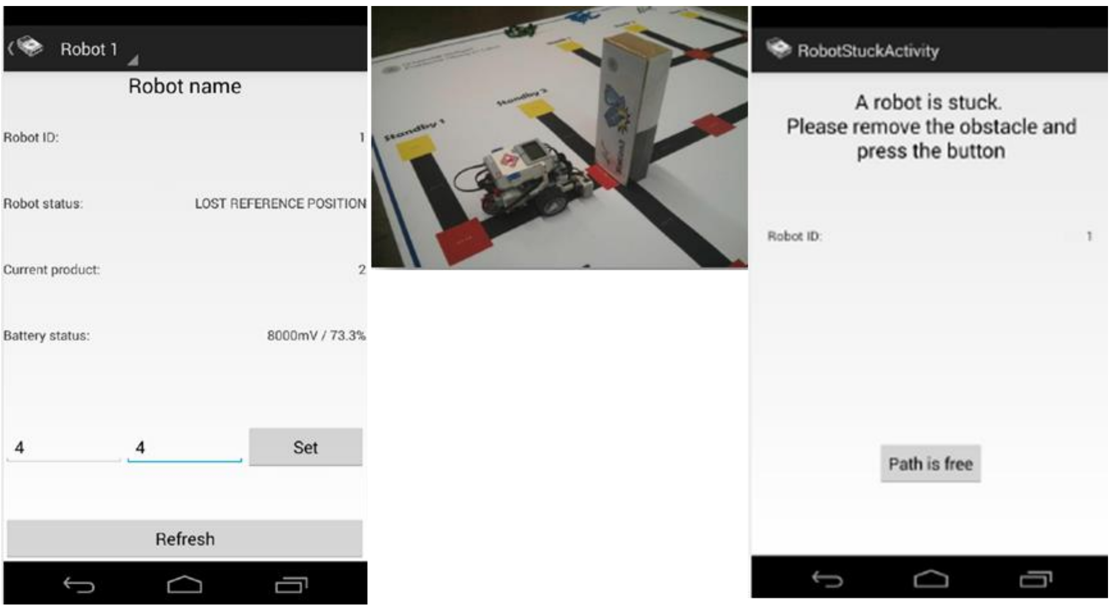

Fig. 7. Screenshot Setting a robot status and blocked Robot activity

Each robot had a Wi-Fi module to communicate with other devices. The client could control the demands on the production; for example, adjust the priorities or the production speed. The robots took the orders and negotiated a new optimized workflow and executed the production autonomously. The road detection was done by using color sensors. Figure 7 shows the control software.

5.1 Lessons learned

The team carried out the task completely in 10 weeks. The developers themselves decided to realize the controller as an app so that a high degree of flexibility was given. The use of Scrum provided us with the following conclusions.

Modification of Scrum: Since our time and workforce resources were strongly restricted, the developer team had to modify the common Scrum development to fit it to their needs. The developers did not have much time, so they had no time to plan everyday meetings. They met only once per week and mostly communicated via E-mail or WhatsApp. Furthermore, the team elected a product owner and a Scrum master, but due to the small team size and the nonexistent experience of all team members with Scrum, everyone had to be a little of everything. In usual Scrum-Teams, the Scrum-master and product owner aren't elected by the team; their position in the specific company chooses them. Nevertheless, the results of the project were quite satisfying. All requirements were implemented and fulfilled. In our experience, Scrum is pretty good at adapting project-specific boundaries.

Dynamic adaptation and modification of the requirements: At the beginning of the project, the requirements were not very clear, being only vaguely formulated. Over the course of the project, the requirements became more detailed. The requirements were constantly discussed with the customer, so a mutual understanding of the project was created. The customer was completely satisfied with the results.

Transparency was a very important aspect of this project. All successes and disappointments were transparent to all involved at any time. Thus, the team was able to respond quickly to undesirable situations, and the chance of success of the project was increased. During the project, a common language was developed so that everybody could communicate well.

6 Conclusion

The project results have shown that agile methods, particularly the Scrum method, are suitable for the development of applications for smart factories. The developed prototype covered all phases of a real project and gave us enough information to evaluate the complete process. During this application development at the University the following observations were made:

- The Scrum team could cope with very vague requirements given to them and intensively activated the individual generosity of the group member to come up with proposals and discuss those with the user.
- The methods also allowed a dynamic adaptation of the requirements while the developing process was happening.
- The team was very independent and self-motivating, not requiring any form of hierarchical supervision.
- The project was developed just-in-time.
- The team identified some additional requirements which had not been proposed by the customer.

We have concluded that this methodology is particularly useful for cutting edge development in which the requirements cannot be defined in fine detail at the very beginning of the project. Scrum demonstrated that works well for small, rather fast-moving projects.

However, team members need to be motivated and autonomous. Otherwise, the project can lose focus because quality management is difficult to implement.

In future works, we plan to use the Scrum method further to demonstrate its transferability and adaptability to larger projects.

Acknowledgments

We would like to thank you very much FAPEAM, DAAD, and CAPES that supported the cooperation between UFAM and IAS-University of Stuttgart.

References

Acatech (2013). Acatech-National Academy of Science and Engineering: “Recommendations for implementing the strategic initiative INDUSTRY 4.0”, Final report of the Industry 4.0 Working Group, http://www.acatech.de/uk, 2013.

Amaral, V.; Barroca, B.; Carreira, P. (2012). Towards a Robust Solution in Building Automation Systems: Supporting Rapid Prototyping and Analysis. 8th Int. Conference on the Quality of Information and Communications Technology, 2012, pp. 261 – 264.

Babulak, E. (2009). Next Generation of Applied Internet Technologies in E-manufacturing. Computer Modelling and Simulation, 2009. UKSIM '09. 11th International Conference on, pp. 386-390.

Börjesson, A. and Mathiassen, L. (2004). Successful Process Implementation, IEEE Software, Vol.21, Issue 4, July-Aug. 2004, pp. 36-44.

Boschi, F.; Zanetti, C.; Tavola, G.; Taisch, M. (2016) Functional requirements for reconfigurable and flexible cyber-physical system. 42nd Annual Conference of the IEEE Industrial Electronics Society - IECON 2016, pp. 5717 – 5722.

Carpanzano, E.; Cataldo, A.; Lacasella, A. (2004). Development of manufacturing automation systems through object-oriented concepts and international standards. Industrial Informatics, 2004. INDIN '04. 2004 2nd IEEE Int. Conference on, 2004, pp. 607 – 611.

Cockburn, A. (2006). Agile Software Development: The Cooperative Game (2nd Edition), ISBN-13: 978-0321482754, 2006.

Coldewey, J.; McBreen, P., Eckstein, J., Schwanninger, C. (2000). Deploying Lightweight Processes, Workshop at OOPSLA 2000, Minneapolis, USA, pp. 131-132.

Dong, W. and Kun, J. (2010). Resilient industrial control system (RICS): Concepts, formulation, metrics, and insights. Resilient Control Systems (ISRCS), 3rd International Symposium on, 2010, pp. 15-22.

Givehchi, O.; Trsek, .; Jasperneite, J. (2013). Cloud computing for industrial automation systems — A comprehensive overview. IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA), 2013, pp. 1 – 4.

Givehchi, O.; Imtiaz, J.; Trsek, H.; Jasperneite, J. (2014). Control-as-a-service from the cloud: A case study for using virtualized PLCs. 10th IEEE Workshop on Factory Communication Systems (WFCS 2014), 2014, pp. 1 – 4.

Givehchi, O.; Landsdorf, K.; Simoens, P.; Colombo, A.W. (2017). Interoperability for Industrial Cyber-Physical Systems: An Approach for Legacy Systems. IEEE Transactions on Industrial Informatics, 2017, Volume: 13, Issue: 6, pp. 3370 – 3378.

Graml, T.; Bracht, R.; Spies, M. (2007). Patterns of Business Rules to Enable Agile Business Processes. 11th IEEE Int. Enterprise Distributed Object Computing Conference (EDOC 2007), 2007, pp. 365 – 365.

Hametner, R.; Winkler, D.; Zoitl, A. (2012). Agile testing concepts based on keyword-driven testing for industrial automation systems. 38th Annual Conf. on IEEE Industrial Electronics Society, IECON 2012, pp. 3727 – 3732

Harrison, R.; West, A.A.; Lee, L.J. (2006). Lifecycle Engineering of Future Automation Systems in the Automotive Powertrain Sector, Industrial Informatics, 2006 IEEE International Conference on, 2006, pp. 305-310.

Hernández-Reveles, J.; Sobrevilla-Dominguez, G.; Velasco-Elizondo, P.; Soriano-Grande, S. (2016). Adding agile architecture practices to a Cyber-Physical System development. IEEE Int. Conf. on Software Process Improvement (CIMPS), 2016, pp. 1 – 6.

Li, J.; Morrison, J.R.; Zhang, M.T.; Nakano, M.; Biller, S.; Lennartson, B. (2013). Automation in green manufacturing. Automation Science and Engineering, IEEE Transactions on, Volume 10, Issue: 1, 2013, pp. 1-4.

Lukman, T.; Godena, G.; Gray, J.; Strmcnik, S. (2010). Model-driven engineering of industrial process control applications. Emerging Technologies and Factory Automation (ETFA), IEEE Conference on, 2010, pp. 1-8.

Phillips, K.; Gruszka, B.; Carroll, J.; Bauer, M.; Maharaj, O. (2013). Connecting industrial automation software to a discrete manufacturing plant model for research and education, AFRICON, 2013, pp. 1-5.

Ringert, J.O.; Rumpe, B.; Schulze, C.; Wortmann, A. (2017). Teaching agile model-driven engineering for cyber-physical systems. IEEE/ACM 39th Int. Conf. on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET), 2017, pp. 127 – 136.

Santos, T.; Ribeiro, L.; Rocha, A.D.; Barata, J. (2016). A system reconfiguration architecture for hybrid automation systems based in agents and programmable logic controllers. IEEE 14th International Conference on Industrial Informatics (INDIN), 2016, pp. 98 – 103.

Schlick, J. (2012). Cyber-physical systems in factory automation - Towards the 4th industrial revolution. Factory Communication Systems (WFCS), 2012, 9th IEEE International Workshop on, pp. 55.

Scheuermann, C.; Verclas, S.; Bruegge, B. (2015). Agile Factory - An Example of an Industry 4.0 Manufacturing Process. IEEE 3rd Int. Conf. on Cyber-Physical Systems, Networks, and Applications, 2015, pp. 43 – 47.

Schwaber, K. and Beedle, M. (2001). Agile Software Development with Scrum (Series in Agile Software Development), 2001, ISBN-13: 978-0130676344.

Schwaber, K. (2004). Agile Project Management With Scrum, published by Microsoft Pr., Library of Congress 2003065178, 2004.

Spichkova, M.; Zamansky, A.; Farchi, E. (2015). Towards a Human-Centred Approach in Modelling and Testing of Cyber-Physical Systems. IEEE 21st Int. Conf. on Parallel and Distributed Systems (ICPADS), 2015, pp. 847 – 851.

Vallee, M.; Kaindl, H.; Merdan, M.; Lepuschitz, W.; Arnautovic, E.; Vrba P. (2009). An automation agent architecture with a reflective world model in manufacturing systems. Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on, 2009, pp. 305-310.