Cybernetic Aspects in the Agile Process Model Scrum

Michael Bogner, Maria Hronek, Andreas Hofer, Franz Wiesinger

University of Applied Sciences Upper Austria
Department of Embedded Systems Engineering
Hagenberg, Austria
Email: [michael.bogner, maria.hronek, andreas.hofer, franz.wiesinger]@fh-hagenberg.at

***Abstract*—Agile process models provide guidelines for modern software development. As one of their main purposes is to complete projects under external influences as successfully as possible, the question arises as to how reliably and routinely given project goals can be achieved by means of such process models. This is all the more relevant as today, unfinished software projects frequently lack certain functionality, or missed project deadlines are still on the daily agenda in software development. Therefore, research has been done to identify the coherences between agile process models and cybernetics. Cybernetics is a natural science based on biocybernetics which forms the basis for well-functioning processes. It was analysed how it helps to cope with complexity, thus allowing for viable complex systems or processes. Cybernetics, as a science of functioning, is also relevant for agile process models. Once the basic cybernetic aspects are applied, processes are kept under control and organized in ways that ensure long-term viability. This paper reports the results of the selected agile representative Scrum. It shows that although some major cybernetic aspects like communication, feedback and circularity are covered, other basic cybernetic principles are missing in Scrum. Yet, these shortcomings can be compensated in order to get essential reliability, especially in critical situations.**

Keywords–agile software development; Scrum; cybernetics; biocybernetics.

I. INTRODUCTION

A major challenge of today's project management is the increasing complexity and the dynamics of changing conditions. Project requirements are getting more and more complex and therefore often cause serious problems for project managers. To meet these requirements in modern software development and to cope with complexity we use process models like the agile model Scrum [1]. It is easily understandable but not absolutely easy to use. In some cases, complex problems can result in loss of control [2]. To counteract this, we typically try to eliminate one problem after the other arising during the development cycle. But this leads to higher costs and missed project deadlines. As a result, the product quality is strongly affected and the project goals will be missed in some cases.

We often overlook that there is a science that can help to cope with complexity. It helps to lead a project in the right way from the beginning to the end. It shows how to survey the complexity and how to deal with it without fighting against it. This science is called cybernetics. It is a natural science which is the basis of many well-functioning processes. Processes and procedures can be kept under control with this science [3]. Cybernetics is an integrative multidisciplinary meta-science. It comprises various theories, primarily the theories of information and communication, and the theory of regulation and control. Without the laws of cybernetics, almost nothing would work - no aircraft, no computer, no large city and no organism [4]. As one of the most fundamental and powerful sciences, cybernetics incorporates the essential mechanisms in order to cope with complexity: self-control, self-regulation and self-organization. In our world of increasing complexity, cybernetics provides the invariant laws of functioning. This holds true for biological, technical, physical, social and economic systems, but most people are not aware of that [5] [6].

As cybernetics is the powerful meta-science which helps to accomplish complex processes successfully, the question arises, whether agile software process models fulfill the basic requirements of cybernetics to reliably guide the entire development cycle. This question addresses not just certain business cases, but is effectively a fundamental question. Cybernetics defines the basic laws, which have to be fulfilled otherwise the development process could get out of control [3]. We have started our research activities with the agile process model Scrum. Scrum is not only well-known, but also widely used. Therefore, we have selected it as our first research candidate. The cybernetic aspects in Scrum have been worked out and they will be described and discussed below to show if they keep the development process under control. The results are presented in this paper.

In addition, we have analysed the aspects of biocybernetics to see if the process model meets the requirements to be long-term viable. Whether certain processes or systems will be viable and capable of "surviving" permanently depends on how far they obey certain basic principles of biocybernetics [7]. These principles and the results of our research have been summarized and are also included in this paper.

This paper is divided as follows: Section II briefly presents the main aspects of Scrum being relevant for this paper. Section III introduces cybernetics in order to make a comparison afterwards in Section IV. Biocybernetics with its basic rules is introduced in Section V. Each rule is discussed with regard to Scrum. Finally, Section VI summarizes the relationship of Scrum and cybernetics and concludes the paper.

II. SCRUM

Scrum is a lightweight agile process model developed by Ken Schwaber and Jeff Sutherland [1]. It provides a framework to manage complex product development and it has clearly defined rules and regulations. The development

 ISBN: 978-1-61208-367-4

process remains flexible and transparent all the time. Scrum is based on iterations where each one usually lasts one to a maximum of four weeks. This iteration is called a Sprint. In the beginning, basic product requirements must be known and committed to the Product Backlog. The requirements are split into tasks and stored in the Sprint Backlog in order to start a Sprint. This process is shown in Figure 1. These tasks are the most important tasks which should be handled during the next iteration. The intention is to have a potentially executable product at the end of a Sprint, called the product increment. The functionality grows from Sprint to Sprint.

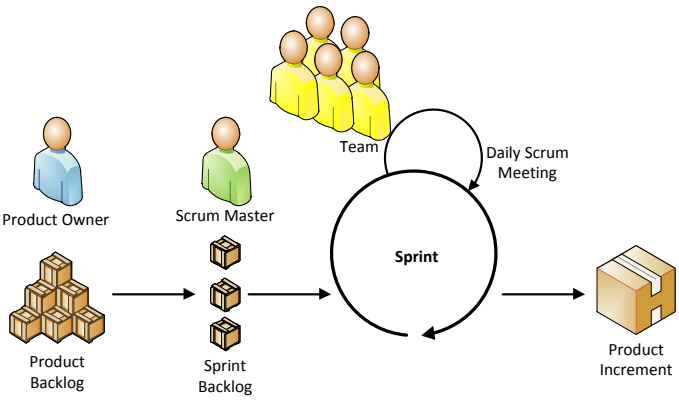

Figure 1. The Scrum Process

There are three essential roles that bring Scrum to life. The *Scrum Team* consists of a group of seven plus/minus two people. They execute the tasks from the Sprint Backlog. The team works in a self-organized fashion. Another important role is the *Scrum Master*. He or she has to see that impediments to the team's progress are removed. The Scrum Master also has to ensure that the process model is proceeded correctly. The *Product Owner* is the third defined role. He or she argues the project goal and defines and prioritizes the single work packages in order to maintain the Product Backlog.

Communication in Scrum is essential. Therefore, there are four significant meetings. During the *Planning Meeting* all three roles decide together which requirements from the Product Backlog should be processed during the next Sprint. They move all tasks to the Sprint Backlog. In the *Daily Scrum Meeting*, the team meets the Scrum Master for some short feedback. The Product Owner can participate. In the end of a Sprint there is the *Review Meeting* where the team shows the Product Owner what they have done. And there is also the *Retrospective Meeting* that acts as a feedback meeting for the team and the Scrum Master. In this meeting, they evaluate the last sprint and discuss what could be improved in the next sprint [8] [9] [10].

III. CYBERNETICS

According to Norbert Wiener, cybernetics is the science of control and the regulation of systems under real-time conditions. This also includes the automation and information processing of such systems. It is important to note regularities and to recognize the functional patterns of complex systems instead of specific details [3]. The real origin of cybernetics lies in nature and not - as often mistakenly assumed - in computer science. The most important factors of cybernetics are control, regulation and feedback. It can be referred to as a science of functioning [5].

This section provides a brief overview of the characteristics of complex systems and the fundamentals of cybernetics. At the end of this section, all the essential and relevant aspects for the comparison with the process model Scrum are summarized.

A. Characteristics of complex systems

In general, we have to distinguish between simple and complex systems. Simple systems are easily predictable. They are also easily applicable even without having knowledge of cybernetics. Complex systems can cause substantial problems if not held under control as they are much more interconnected and highly dynamic. It is not possible to intervene easily because this can result in unpredictable side effects [5].

The term "system" regarding cybernetics always means open systems, which interact with the environment and adapt to it over time. In contrast, closed systems do not interact and are self-contained. In such a given open and complex system, we cannot reduce complexity in order to simplify it, as it is often claimed. Complexity means variety. It is inevitable if the system has to accomplish all of its tasks reliably. If the complexity would be reduced, also variety would be reduced. Under certain conditions the system would fail. So we have to master complexity and to make use of it instead of eliminating it [2].

B. Importance of feedback

Due to cybernetics, "information" was recognized as third essential basic item supplementing the two basic elements "energy" and "matter", which both are not sufficient to explain how a system behaves. Information is the key which describes how things are organized in a dynamic system [2]. And to handle and manage such systems reliably, a special kind of information is needed, namely feedback. Without feedback it is simply impossible to hold a complex system under control [3].

C. Basic rules to control complex systems

Cybernetics presents the laws of nature which are responsible for the reliable functioning of complex systems. Relevant rules are mentioned below:

1) Circularity: Circularity is gained by regular feedback. A cybernetic system works because of control loops, circularity and feedback. That means that the system gives itself feedback. Through that process, the systems can excuse errors and are very robust [7]. The technical representation of this aspect is the well-known feedback control loop in Figure 2. It uses the cybernetic terminology and shows that the regulator compensates the influences of any disturbance keeping the system under control. Such systems follow an evolutionary approach. They are self-adjusting. And this is why such systems are viable in the long-term.

2) Self-organization and self-regulation: The mentioned self-adjusting aspect includes also self-organization and self-regulation. A cybernetic system is not externally directed. Instead, it is autonomous in the context of the whole system. It directs and controls itself in order to cope with complexity.

 ISBN: 978-1-61208-367-4

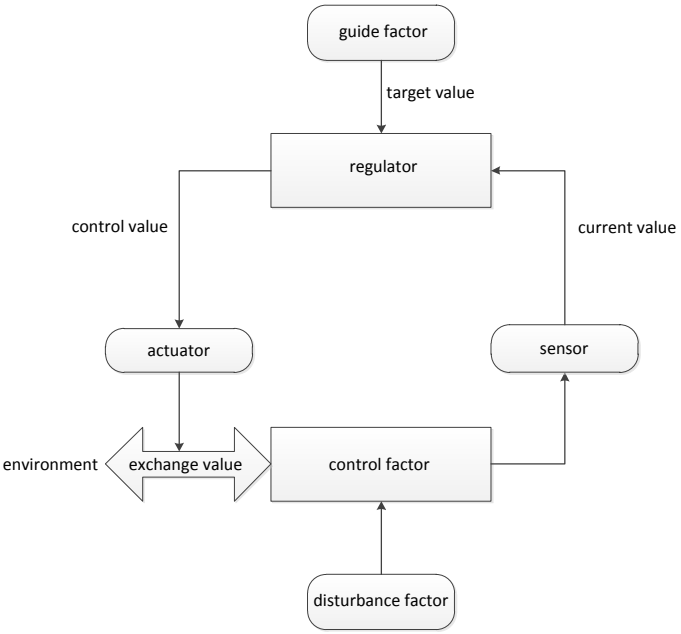

Figure 2. Feedback control loop according to Vester [7].

3) Theory of recursive systems: Cybernetic systems are long-term viable if they consist of interconnected self-adjusting subsystems building a recursive structure. The single systems interact with each other to represent the whole system.

4) Law of requisite variety: If we want to control a complex system, we need at least as much complexity and variety in the control mechanism as the controlled system itself has. If there is too less variety in the control system then the controlled system will sooner or later get out of control, as under certain conditions the control system cannot react properly to dominate the situation. The British cyberneticist and neurophysiologist W. Ross Ashby discovered the law of requisite variety, also called "Ashby's Law" [11]:"Only Variety can destroy Variety". It is the basic law of complexity. In simple terms, consider a switch with the two possible states "0" and "1". It is obvious that it is not able to control a system having three or more states. In order to control a system, we need as much variety (which means complexity) as the system itself [2].

D. Patterns and interconnections instead of details

To understand a complex system, it is not necessary to know as many details as possible. An abstract consideration and the recognition of patterns is the right way to understand a system. Interconnected thinking is extremely important. It is not the details that are important in a system but the coherences. The interconnections between the individual parts are important.

E. Cybernetic aspects put together

To summarize, complexity cannot be reduced but dominated. It is important that we work with it instead of fighting against it. To master complexity we need:

- self-regulation and self-organization,
- circularity for repeated and continuous operation,
- regular feedback in real-time for deterministic adjustments,
- communication and interconnectedness for a proper and continuous flow of information,
- autonomy allowing self-organization and self-control, and an
- evolutionary approach for possible adaptation due to changing conditions in the environment over time.

IV. CYBERNETICS AND SCRUM

Having shown the fundamental concepts, we will now compare Scrum with the essentials of cybernetics to verify whether it supports all the requirements of a sustainable and functional process. This includes mastering complexity, circularity, feedback, communication, real-time aspects, interconnectedness, autonomy, evolutionary approach, self-organizing and self-adjustment.

Circularity: The process model Scrum works in iterations. Scrum is already cyclically arranged since Sprints recur after a certain amount of time. During the Review Meeting the Product Backlog can be filled with new requirements from the customer as well. The result of the Retrospective Meeting is that the team can work better than in the Sprint before because they have reviewed the problems and obstacles. All the meetings improve the effectiveness of the team and the product quality. Therefore, the behaviour of the next Sprint is positively influenced. The iterations in Scrum represent the circularity required by cybernetics. It is the basis for continuous adaption and optimization and forms the control loop to keep the system under control.

Feedback in real-time: Feedback is established in various ways in Scrum. Considering the Sprint, which is determined by a duration of up to four weeks, there is the Sprint Backlog, which should be executed in order to get the product increment. At the same time, the Scrum Master gets the feedback of the team from a technical and a personal point of view out of the Review and Retrospective Meeting. Equally important is the feedback from the Product Owner and the customer which plays a minor role in Scrum. It is essential in order to detect problems early and to reach the goal in time. The Daily Scrum Meeting can also be seen as a feedback loop. This is real-time feedback. The Scrum Master can immediately react and appropriately guide the project if a team member is blocked by an obstacle and not able to work. Also, the team gets feedback from the outside world representing an open system.

Communication: Communication occurs between all the roles. In the Daily Scrum, regular communication takes place every single working day and also during all the other meetings. Through this constant communication, decision making processes improve. The purpose is that information flows between all stake holders. This is achieved in Scrum.

Interconnectedness: All the roles in a Scrum project are working together. All are interconnected, and this leads to a simplified coordination and a higher product quality as there are short communication paths. This is very important for direct information exchange.

 ISBN: 978-1-61208-367-4

Autonomy: The guidelines of Scrum let all team members work independently. They are free to act and decide during a sprint. This allows the team to do the right things right, as they are the experts in their specific domain. Autonomy also happens at the beginning of a Sprint when the development time of each work package has to be estimated. Every member estimates his task for the next sprint autonomously, which ensures much higher accuracy. Autonomy comes through the self-organizing teams, which means that they bear a great responsibility to work in a disciplined manner. Despite the autonomy, Scrum functions as a superior guideline.

Evolutionary approach: An evolutionary approach is inherent in every cybernetic system. It ensures a continuous development and adaption to changing conditions surrounding the system. So, the evolutionary approach keeps the cybernetic system viable. Looking at Scrum, it shows that changes of the agile process model are basically possible. Change proposals can be submitted. But one has to bear in mind that it is at the sole discretion of Jeff Sutherland and Ken Schwaber to apply any changes to the process model itself. Strictly speaking, this follows not the idea behind an evolutionary approach. Cybernetic systems are free to adapt to changing conditions as soon as they appear. Such systems do not have to wait for anything or anyone and are in an continuous process of adaption. In contrast, Scrum argues that it changes its framework infrequently.

From the strict cybernetic point of view, the process itself is not designed to change or adapt itself according new requirements appearing from the outside world. This is a missing aspect in scrum that cybernetic systems must inherently have. It means that the process model will support today's projects but unless adaptions it is uncertain if it will fit in future projects.

Self-organizing and self-adjusting: For the members of the Scrum Team, there is no precise formal rule or guideline how they have to do their job. Therefore, they can freely adapt to unknown complex project in the required way. They adjust independently to the task to be solved. That means that Scrum Teams regulate and organize themselves. So, they are working more efficiently, more motivated and more effectively because they have no precise rules to which they must adhere. Due to all these characteristics, the productivity can be sustainably increased.

So far we see that Scrum serves well as a lightweight agile process model, which helps to cope with the development of products with a complex scope. The structure of the agile process model combined with the distribution of the different roles overcomes the challenge of complexity and delivers a satisfactory result. If it comes to mastering complexity, everything collaborates including iterations, communication and regular feedback, self-organization and self-regulation. As Scrum supports todays projects well, we identified a lack in Scrum itself. It is not designed to adapt itself to new requirements.

V. BIOCYBERNETICS AND SCRUM

Cybernetics has its origin in biocybernetics. As a natural science, biocybernetics represents the fundamentals of the way living systems and organisms in the nature function successfully. Most natural systems have to deal with substantially more complexity than any technical system made by humans. Since billions of years, nature functions reliable and most efficient [12]. Frederic Vester, the founder of biocybernetics, defined eight basic rules every complex biological system has to fulfill in order to survive. Simply, these rules are principles of nature [7]. They can guarantee a successful evolutionary existence as they present possibilities for long-lasting development and survival of any living system - if it is a human being, an ecological system, a company or a city [13]. So biocybernetics represents the natural basis of cybernetics, which means that cybernetics comprises the corresponding science. Therefore, every technical system or process has to fulfill this eight rules of biocybernetics to be long-term viable.

As this is rather important, research has been done to analyse the biocybernetic aspects of Scrum. As an in-depth discussion of biocybernetics goes far beyond the scope of this paper, we give a short introduction to each of the eight rules and present our results afterwards. We recommend Vester's book, *The Art of Interconnected Thinking*, [7] for a comprehensive discussion of biocybernetics.

1st rule: Negative feedback cycles must dominate over positive feedback: In cybernetics, this means that for a system it is very important to be stable against interfering influences. To explain the principle briefly, an example for negative feedback is the control loop of the thermostat of a heating system. If a certain temperature is reached the energy input has to be decreased in order to not exceed the temperature value. If the temperature is too low, the energy input would be elevated. Negative feedback can also be found in nature. In general, there is almost exclusively negative feedback in order to keep the system stable. Positive feedback in nature appears in avalanches or steppe fire, for example. They build up continuously and achieve a new order at the end, which is exactly not desired in stable systems. That is why negative feedback has to dominate positive feedback, otherwise the system would collapse.

Negative feedback is therefore based on a control or feedback loop balancing the system. In terms of the Scrum model, this feedback control system means that the control factor would be the project and the regulator can be seen as Product Owner and Scrum Master. The guide factor leading the way would be the customer together with the market who will specify the product requirements. The team represents the actuator, which adjusts the actual and the desired condition. The current value is reported through direct communication and through test results of the product increment. Misconceptions can be critical for the process. They are reflected by the disturbance factor in the control loop. Misunderstandings can arise between the individual developers or between the team and the Product Owner or the Scrum Master. Also, longer absences due to illness or information deficits and wrong assumptions or misinterpretation of the requirements can lead to problems, which are solved through communication and correcting feedback during the sprint. And the feedback of the team at the end of a sprint leads to improvements that will be implemented in the next sprint.

In Scrum, the negative feedback mechanism is represented by this inherent control loop. It reliably prevents the process

 ISBN: 978-1-61208-367-4

to run out of control.

2nd rule: The function of the system has to be independent of quantitative growth: A system passes through metamorphoses while growing in order to survive. Based on the self-organization of cybernetic systems it does not have to be dependent on growth. Instead, there must be a restructuring step during the growth in order to move from one stable state to the next. After growth, the system is ready to get into the next phase. Therefore, the growth resembles a sustained "S"-shaped curve. But if a system suffers from unrestricted linear growth without proceeding to a stable state, it will lead to a collapse in the end. A butterfly would be the best example for growth and metamorphoses. The butterfly caterpillar pupates after a certain growth and envelopes itself in its cocoon to become a butterfly. At the hands of this transformation the linear growth is stopped, and continues as a butterfly in the next phase [7]. For complex systems, it can be deduced that reconstruction and metamorphoses are not replaceable by pure growth.

At this point the size of a Scrum Team matters. If a project gets bigger and more than five to nine people must work in a team, this team may not grow linearly. The project has to be split in order to get smaller teams that can work more effectively. This split is supported by the process model as Scrum is scalable for larger development teams. It is called Scrum-of-Scrums. It is important that the new sub-groups have superior coordination. There is a so-called Scrum-of-Scrums Meeting where all Scrum Masters of the single teams come together and can take over coordination [14]. If teams would grow linearly, communication would be very difficult because the communication channels rise exponentially with the number of people. The organizational overhead scales up and effectiveness and efficiency degrades. Finally, the system will run out of control and end in chaos. Therefore, linear growth should always be avoided. It needs "metamorphoses" for the purpose of a division in sub-projects.

3rd rule: The system must operate in a function-oriented, not product-oriented manner: The environment is constantly changing and that is why product requirements also vary. All products have a certain life cycle and will sooner or later disappear. On the other hand, the basic needs will not disappear and remain existent. The functional requirements of a product usually remain for a long time whereas products themselves change very often. For example, horse-drawn carriages are substituted by cars, telephones by modern smartphones, while mobility and communication as basic human needs further exist. Hence, it is always important to think function-oriented.

Scrum works without dedicated products. It works regardless of whether a database application, a smartphone application, a server application or a desktop application is the product. Scrum is detached from products and represents the "function".

4th rule: Exploiting existing forces (Jiu-Jitsu-Principle): Normally, Jiu-Jitsu is a Japanese martial art, which is used for a self-defence. The main principle in Jiu-Jitsu is that the force of the opponent should be utilized instead of defending against it. In systems this energy serves as control-energy. Applied here it means that the existing force and energy should be used instead of rejecting it.

This rule can be reflected in Scrum. Scrum uses the customer and the market with its requirements and wishes as external energy. They call the shots and decide what will happen next, so they are the driving force for the project. All changes that they bring along are very important for the quality of the final product and their competitiveness, as already the Agile Manifesto states [15]: "Welcome changing requirements".

5th rule: Multiple use of products, functions, and organizational structures: Viable systems put emphasis on reusability. If every product would be designed, produced, sold and used on its own, the costs and energy input would increase considerably. The efforts of energy, matter and information can be reduced substantially gaining a synergy effect.

In every project guided by Scrum, the organizational structure is intentionally left identical. Meetings are at the same time, at the same place, and have the same structure. This leads to more efficiency as stable conditions makes it a routine work. Another aspect is the specific knowledge of the Scrum Team. This knowledge and also existing software products, like libraries or frameworks, can be used in other projects. This makes multiple use of products and functions very easy and comfortable in Scrum.

6th rule: Recycling: Nature never produces garbage. Due to its cyclic and interconnected processes, waste does not exist and is used elsewhere as important resource. Recycling is one of the most important rules humans should revert to in order to keep a system alive.

It is hard to incorporate this rule into the process model Scrum because a Scrum Team will not produce waste in terms of material waste. Maybe, functions which are implemented in the actual Sprint and will not be used in the end can be seen as trash. But they do not have to be recycled. They can, for example, be provided for other teams or taken as features. The development is usually consumer-market-controlled so there is no real overproduction.

7th rule: Symbiosis: This usually means the cohabitation of two or more species in a common environment, which benefit from each other. In order to enable symbiosis, diversity in a small space is required. That means that many different elements within a system can share resources and functions in order to help other elements accomplishing their work in a more effective way.

The members of a Scrum Team should be located in close proximity to each other. This enables better communication and cohabitation through symbiosis. They benefit from each other if there are any problems. Here, especially the Scrum Master takes action and eliminates any obstacles or issues so that the team can work efficiently. And arising requirements not discussed so far can be cleared by the Product Owner, which itself benefits from much higher product quality. They complement each other.

8th rule: Biological Design: All systems, products, functions or organizations should be developed in respect of the nature. Building anything against nature is plain unnatural. So, nature always matters as it defines what is right or wrong.

 ISBN: 978-1-61208-367-4

Scrum has to correspond to the structure of a viable system and may not be unnatural in its structure and process. This sounds abstract but means nothing else than Scrum has to follow natural processes if it wants to act viably. Scrum fulfills this requirement of rule eight as it follows biological design and not an artificial one. It brings along all these preconditions of biological design with circularity, feedback, autonomy, self-organization, recursive structure and all other mentioned aspects satisfying this last rule of biocybernetics.

In summary, the fundamental eight biocybernetic rules are met in different degrees. From the perspective of our research activity, basically most of them can be seen as fulfilled. Looking at the third rule where functions dominate products, shows that the point of view is essential: as Scrum is not focused on special products and therefore flexible for software projects and applications of different kinds, it fulfills this rule quite good. In contrast, if we look at Scrum itself, it shows some shortcomings in continuous adaption to changing conditions. The sixth rule, which means recycling to avoid waste, is applicable only partly due to the immaterial nature of computer science.

VI. Conclusion and Future Work

Even if Scrum fulfills most of the requirements cybernetic systems must have, we come to the conclusion that it is not a true cybernetic process. Not only that Scrum does not claim itself to be cybernetic. The history of Scrum begins in lean management strategies of Japanese companies. It incorporates a lot of best practices and has not been designed with cybernetics in mind.

Nevertheless, Scrum is of course very suitable for today's software development projects. As responding to change is an important aspect in Scrum, it addresses a fundamental concept of cybernetics to hold a system under control. And there are quite more major principles in Scrum beside circularity and feedback, namely autonomy, self-organization and self-adjustment within the context of the overall process structure.

Scrum guides the project management process in the right way and successful projects are no accident. Also the biocybernetic requirements are largely fulfilled, which leads to the same conclusion. Although, some of the rules are not directly applicable due to the immaterial speciality of Computer Science, we consider them as satisfied, as we have not discovered major inconsistencies or conflicts.

If we do not look at today's projects but on projects in the distant, or maybe, not so distant future, the missing evolutionary approach must be mentioned. Submitting change proposals differs from a cybernetic way. But at this point, Kanban can be deployed [16]. Kanban is a management technique for software development incorporating continuous improvement of the process itself in small steps. So with Scrum and Kanban combined, this essential cybernetic aspect can also be satisfied, which keeps the system long-term viable.

Beside the overall process, which has been analysed here, shortcomings can be discovered in some other areas. For example, Scrum defines "roles" although cybernetics requires "functions". The process model uses this term in order to determine key tasks and to define responsibilities. So, every role has a certain focus as already mentioned, but Scrum does not explicitly forbid additional tasks arising during the development process. In practice, it is often seen that additional tasks are carried out in order to get a product with the required quality. Therefore, autonomy and self-organization are the key aspects to get this done right, although this is not noted in Scrum.

Another issue concerning autonomy and self-organization is the Scrum Team. Scrum does not define any cybernetic approach the team has to follow. Therefore, it can be completely ignored meaning that the recursive cybernetic structure is broken. As before, it is the responsibility of the autonomous team to do the things right. The prerequisites are met as both, Scrum Master and Product Owner, can be present during the Daily Scrum Meeting in order to support a cybernetic approach.

After this analysis of the coherences between Scrum and cybernetics it can be seen that many cybernetic aspects are already covered in Scrum. So far, our recommendation is to additionally apply Kanban and basic cybernetic principles in order to overcome the mentioned shortcomings. In future work, we will analyse this promising combination in more detail.

References

[1] J. Sutherland and K. Schwaber. The Scrum Guide. [Online]. Available: https://www.scrum.org/Portals/0/Documents/Scrum Guides/2013/Scrum-Guide.pdf (2014.05.23)

[2] F. Malik, Komplexität - was ist das?(Complexity - What is this?). Cwarel Isaf Institute, 1998.

[3] N. Wiener, Cybernetics: or Control and Communication in the Animal and the Machine. New York: Wiley, 1961.

[4] F. Malik, Strategy: Navigating the Complexity of the New World. Campus, 2013.

[5] ——, Strategie des Managements komplexer Systeme: Ein Beitrag zur Management-Kybernetik evolutionärer Systeme (Management Strategy of Complex Systems: A Contribution of Management Cybernetics of evolutionary Systems). Haupt, 2008.

[6] ——, "The six cultural values for the reliable functioning of an organization," Malik on Management, Volume 22, 2014.

[7] F. Vester, The Art of interconnected thinking. Mcb, 2013.

[8] A. Cockburn, Agile Software Development. Addison Wesley, 2006.

[9] H. Wolf and A. Rock, Agile Softwareentwicklung: Ein Überblick (Agile Software Development: An Overview). dpunkt.verlag, 2011.

[10] K. Schwaber, Agile Project Management with Scrum. Microsoft Press, 2004.

[11] R. Ashby, An introduction to cybernetics. Chapman Hall, 1956.

[12] F. Vester, "The biocybernetic approach as a basis for planning our environment," System Practice, Volume 1, No. 4, 1988.

[13] G. R. M. Harrer, "The biocybernetic approach as a basis for planning and governance," Proceedings of the 54th Meeting of the International Society for Systems Sciences (ISSS), Waterloo, Canada, 2010.

[14] U. Biberger, Gestaltungshinweise für agile Software-Entwicklungsprojekte unter dem Blickwinkel der Kybernetik (Design Guidelines for agile Software-Development Projects from the Viewpoint of Cybernetics), 2009.

[15] Agile Alliance. Manifesto for agile software development. [Online]. Available: http://agilemanifesto.org/principles.html (2014.05.23)

[16] D. Anderson, Kanban. Blue Hole Press, 2010.

 ISBN: 978-1-61208-367-4