

УДК 378

LIPOWSKY STEFAN

Lecturer at University of Applied Management, Erding, Germany
Senior Manager at it-economics GmbH, Munich, Germany

ADAPTING SELF-ORGANISATION METHODS FROM PRODUCTION INDUSTRY'S LEAN MANAGEMENT TO SCIENTIFIC COLLABORATION AT UNIVERSITIES

Aim of the study

Aim of the study is to point out how to adapt methods of self-organizing of production and software engineering to internal organization of an university.

Hypothesis

Hypothesis: self-organizing agile structures are transferrable to and valuable for scientific collaboration at universities.

History and motivation of lean and agile management

During the first years of 21st century Henry Ford organised, based on Frederick W. Taylor's theories of so called *Scientific Management* his staff by optimisation of exactly defined production processes. He divided processes into steps, and so he could deploy people knowing only how to do this step for having cheap employees and full control. Finally, Ford was very successful in significantly decreasing the car's unit price – maybe more because of the invention of assembly line than because of Taylor's ideas. But on the other hand, there are disadvantages of this proceeding, such as inflexibility and especially dividing physical from brain work. Workers were neither asked nor allowed to think about what and how to do their tasks. After the Second World War, when automotive industry was growing fast and heavily competing, Toyota implemented its own *Toyota Production System*, also called *Lean Manufacturing*. Toyota tried to reach their main goals high productivity, best quality and on-time-delivery by continuous improvement and decentralised competency for decisions. Therefore they built teams of employees which were working together, and implemented regularly conducted methods to improve their working process step by step in small iterations. So Toyota used the power of their employee's brains, guided by the idea that the one who's doing a job knows the best how to make it better. The Toyota Production System contains a lot of tools for organizing work, improving collaboration, setting goals and controlling success, of reporting and finding solutions [2, p. 117]. One of the likely most common tools is Kanban, which will be described later in this article.

Production industry stepped on towards improvements of these ideas with Six-Sigma, Lean Six-Sigma, Total Quality Management and other methods, but for this article it's more interesting to follow software development methods. There's a huge difference between industrial production and software development: Industrial production's goal is high efficiency by having minimal errors. Software development's goal is creativity by finding errors early. To organise software development, IT-management was invented, and project management methods were more and more developed while the projects became bigger and more complex. During the 1990ies, it came more and more out, that the more complex the software project is, the more management decreases efficiency and speed, and reduces creativity. Borrowing the ideas of Toyota, Ken Schwaber and Keith Sutherland created a way for running software development by *empirical process control*¹, what means experimentation,

¹wrongly to German translated by Boris Gloger to ,empirischer Kontrollprozess'

observation and continuous improvement. The base of the change was involvement and empowerment of the staff, and changing management to observation and supporting [3, p. 20]. As the *Lean Management* was established to organize a production process, which means highly efficient recurrence by achieving high quality, Sutherland and Schwaber adopted these methods to a creative development process, which is based on uncertainty and experiment – calling their attitude *agile* and their method *Scrum*. *Agile* means organisational forms following the four agile principles: “Individuals and interactions over processes and tools, Working software over comprehensive documentation, Customer collaboration over contract negotiation, Responding to change over following a plan”[8]. Scrum is based on lean and agile ideas and implements this to software development by defining roles, ceremonies and artefacts [3]. While a production process should be managed to be more and more predictable, a creative process should be managed to reach goals by keeping room for ideas and creativity.

One of the fundamental theories for lean methods is about constant flow: The maximisation of a system’s throughput is reached at less load than the maximum: a load of 80% for all parts of the chain typically leads to maximum throughput. Rising utilisation of that system would increase output progressively smaller, and rising utilization above a certain critical level, output would radically drop to zero. This theory leads to work on cadence and to work with pull principle. Working on cadence means using regular rhythm within a process. To work on cadence allows periodical resynchronisation, makes waiting times predictable, and enables smaller queues [7, p. 169]. Therefore, work packages should not be too big, and have to be completable within a cadence. To work with pull principle means the responsibility of the team members to pick up a new open task when the old is finished. So there’s no manager, who determines somebody should do something during a special time period – it’s the team member who is actively asking for the next task.

For both environments, production and development, establishing an open culture is important. Open culture can only be established by the management. Empowering the people to decide by themselves means to give room for decisions, to trust in and to accept their decisions. Furthermore, another central component is an open error culture[4, p. 56]. This error culture is necessary to find errors early, to reduce error costs, and to create an open-minded frame for creativity[9, p. 166]. This cultural change cannot be successful without commitment of the management for this organisational change. Finally, enabling groups for self-organizing means establish self-regulation[6, p. 62].

So how does it work? For this article there should be given two examples of working methods. A very famous method is Kanban, what is first of all a tool for visualisation and evolutionary changing the work of an organisation, mostly a working team. Eye-catcher of that method is a big board with at least three or even more columns (e.g. ready, in work, blocked, done) and cards with tasks on it. Tasks are described shortly, but understandable for every team member, and assigned to one of the team members. Cards ready to be started are placed on the ‘ready’ column (until then not yet assigned to a team member), also called *backlog*. If a team member has free capacity, he² takes one card to the ‘in work’ column and marks it as assigned to himself. In case that work on that card cannot be finished, e.g. because some information is missing, this card will be moved to ‘blocked’ column - when the work is continued back to ‘in work’, when the work is finished, to ‘done’ column. Highly

² for a better reading, only male form will be used

recommend at this point is to define when a task is ready, in the meaning of sufficiently described that work can be started, and when a task is done, in the meaning of no further work to do. Up to now this method is applicable to every team, it's cheap and easy, and it's surprising that it will immediately transparent, if there are problems in the work of that team. Very often there are many 'blocked' tasks, that means that there are impediments to finish, maybe some missing information, tool or material. Very often many cards are assigned to one team member, but no task will be finished, maybe because of too many 'most important' tasks. But having too many 'most important' tasks will not rise the capacity of that team member. As pointed out before, goal for high team efficiency is a permanent flow, so in these two examples it's immediately visible at what points could be room for improvement. Both described problems lead to task switching, which lowers efficiency, because the team member comes back again and again to the unfinished task – and finally it's demotivating for the team member. So that means: Kanban is a change tool by visualising what happens within the team[see 1, p. 65].

First of all, the team should follow the Kaizen idea to implement continuous improvement with small steps. So there should be no complete analysis which would take maybe months, followed by some huge master plan which will never come to life or, if it would unexpectedly be realised, it would come so late that problems are already different. For continuous improvement it's necessary to have some regular team meetings which can be done on two levels. First level, a kind of daily working mode, can be borrowed from an agile method Scrum[3, p. 172]: very often (e.g. daily, at least weekly, therefore in this article named *daily*) and very short. The Team comes together and each member answers three questions within one minute at maximum – so the meeting is no longer than 10 minutes. The three questions are: what I did yesterday, what I will do today, what hinders me. Task size and meeting cadence should be adjusted in a way, that an average task can be done during one or two meeting cycles. In that meaning, every meeting as many cards as team members are moving. The answers on the first two questions are replacing the status report and this part replaces these long and boring weekly team meetings where usually work is organised. If the third answer brings an impediment, a new card is written, and someone from the team takes it immediately and solves that problem. If there is some deeper discussion necessary, the two affected team members can just arrange some short meeting after this daily. This daily should make sure that cards start to move continuously, that everybody is working effectively, and that communication within the team is improved by this initiating point.

A second level of meeting could be done every two or four weeks: the retrospective as a part of the team's self-regulation. The retrospective meeting needs more time than the daily, it can be done within one hour and it's about collaboration. The problems, visible on the Kanban board (see my two examples above), are analysed, as well as other issues coming out from the team. After collecting the problems and prioritizing them, tools like Ishikawa diagram, could be used for getting into detail and localise the root cause. This Ishikawa diagram looks like a fishbone where into the head of that fish is written the problem. There are six areas to analyse, represented by 'bones' of the fish, which are Equipment, Process, People, Materials, Environment and Management. All these areas are to be discussed by the team according to the considered problem, to get deeper and deeper towards the real cause. Again, the findings will be tasks and placed to the backlog in order to solve them until the next meeting – solving just the next most important problem, not saving the whole world. This meeting, as well as the daily, has team-building and self-organizing character, and can be seen as a meta-cognition on a team level. This approach helps to

involve team members personally and to take over responsibility. These both meetings are working only if the cultural environment of the team is supporting decentralised decisions and error-friendly attitude. Without enabling for decisions, team members and teams will not develop self-efficacy, and will not try to influence their environment. Without error-friendly culture team members and teams will not be able to speak about root causes, as long as they are involved in the problem[see 1, p. 82].

For implementing these lean and agile methods, management has to ensure that not only the team is working on new methods, but the company has to create a culture of openness, fearlessness and trust. There's a simple manager's self test existing to check if employees trust: just take a piece of paper and write down the situations during last three months, when a employee came and told having really messed up something – very often this paper remains blank. But the idea of Kaizen, that every situation can be improved, and it best can be done with small steps, is valid for managers and company culture as well. Managers have work on their improvement continuously and ask constantly for feedback, and try to be more a 'servant leader' who is supporting his teams than a commander[see 1, p. 49]. Observation the team's throughput ensures the work to be done, therefore several tools have been created like the Kanban board itself or measuring throughput of a team.

After having implemented Kanban to several software development and operation teams, we realized that our internal human resources department seemed to have the same problems like these teams: The employees felt overloaded, many tasks could not be closed, people who were waiting for some delivery were complaining about slow completion, leaders were asking what they are doing all day and much more. When we recommended trying Kanban, we again heard the same objections like we have heard from our software teams: it's not possible because of too different tasks and task sizes, they were afraid of transparency, too much additional work and others. Finally, after bringing Kanban to that team with three workshops and some coaching and retrospective sessions, we could see that it works perfect in this environment, and team members liked it. They liked it, because they saw that it's much less work than expected, that it's easy to cut down tasks to small pieces if you know how to do, that they could concentrate on actual tasks and finish them, they could work more effectively, because they had better descriptions what to do and less context switches, and finally, that transparency protected them from urgent, high prioritized tasks: they just showed their board and asked: "you can see what I'm doing at the moment, who requested it and how urgent it is – if I'm doing your task now, on which card should I stop working?" We've been impressed that it worked so well, and as a next step we adapted Kanban to part time working teams: They are working just 10% of their working time in these teams, and they are distributed across different cities in Germany. So we used a virtual, electronic Kanban board, and organized virtual weekly – instead of daily – meetings. Even if it's harder in this case, to bring that team together, and requires a lot of discipline for all involved, it finally worked, and currently we start our second step of implementing continuous improvement.

Conclusion

Adaptation to collaboration at University is possible, and here should be given some examples. First example could be organizing internal services as are full-time co-located teams like human resources, administration and property maintenance. By organising chair's work, there are many dependent tasks, which require collaboration of the staff like research projects, coordinating student's class schedule, and organizing exams and conferences. And even improvement of teaching and adaptation of novice teachers is a task, which requires open collaboration [5]. For all these tasks, lean and

agile methods can improve efficiency, and increase involvement, responsibility, fun and satisfaction what is highly necessary in times when protection and support of general government is lowered.

Outlook

Implementing continuous improvement of teams means to introduce a kind of self-regulation and meta-cognition. Even if these terms are used in pedagogical science mainly for individuals, it's possible to bring them to team level– by combining with experiences from the psychological concept of Kolb's *experiential learning theory*. This is the topic of the Author's dissertation, which will be finished at the end of 2016.

LITERATURE

1. Anderson D.J. Kanban: Successful Evolutionary Change for your Technology Business / D.J. Anderson, Washington: Sequim, 2010.
2. Chiarini A. Lean Organization: from the Tools of the Toyota Production System to Lean Office / A. Chiarini, Milano: Springer, 2013.
3. Gloger B. Scrum: Produkte zuverl?ssig und schnell entwickeln / B. Gloger, M?nchen: Hanser, 2011. 342 p.
4. Lipowsky S. Error Management Culture in Software Development Teams: Ways to measure Riga: RPIVA, 2015. p. 54–60.
5. Mon A. Features of Novice Teachers' professional Adaptation According to O.G. Moroz (1940 - 2007) Riga: RPIVA, 2015. p. 60–66.
6. Parker C. The Open Corporation: Effective Self-regulation and Democracy / C. Parker, Cambridge: Cambridge University Press, 2002. 362 p.
7. Reinertsen D.G. The Principles of Product Development Flow. Second Generation Lean Product Development / D.G. Reinertsen, Redondo Beach, CA: Celeritas, 2009.
8. Schwaber K., Sutherland J. Manifesto for Agile Software Development [online source]. URL: <http://agilemanifesto.org> (accessed: 03.10.2016).
9. Weingardt M. Fehler zeichnen uns aus. Transdisziplin?re Grundlagen zur Theorie und Produktivit?t des Fehlers in Schule und Arbeitswelt. / M. Weingardt, Bad Heilbrunn: Klinkhardt, 2004.

Ліповські Стефан. Адаптація методів самоорганізації: lean-менеджмент від промисловості до наукового співробітництва в університетах.

Анотація. Lean- та agile-методи організації роботи розроблено у галузі промисловості та вдосконалено у розробці програмного забезпечення.

Зазначені методи ґрунтуються на самоорганізації та саморегулюванні та сприяють підвищенню ефективності праці, збільшенню залучення працівників до діяльності, формуванню більшої відповідальності та задоволення від виконуваної роботи, а також допомагають компаніям вижити в умовах високої конкуренції та підвищення складності.

Стаття на прикладах висвітлює теоретичну базу зазначених методів, як вони працюють, і яким чином вони можуть бути використані в такому контексті як співробітництво в університетах.

Ключові слова: самоорганізація, lean-менеджмент, співробітництво в університетах, залучення, метапізнання.

Липовски Стефан. Адаптация методов самоорганизации: lean-менеджмент от промышленности к научному сотрудничеству в университетах.

Аннотация. Lean- и agile-методы организации работы разработаны в области промышленности и усовершенствованы в разработке программного обеспечения.

Обозначенные методы основаны на самоорганизации и саморегулировании и способствуют повышению эффективности труда, увеличению привлечения работников к деятельности, формированию большей ответственности и удовлетворения от выполняемой работы, а также помогают компаниям выжить в условиях высокой конкуренции и повышения сложности.

Статья на примерах освещает теоретическую базу указанных методов, как они работают, и каким образом они могут быть использованы в таком контексте как сотрудничество в университетах.

Ключевые слова: самоорганизация, lean-менеджмент, сотрудничество в университетах, привлечение, метапознание.

Lipowsky Stefan. Adapting self-organisation methods from production industry's lean management to scientific collaboration at Universities.

Summary. Lean and agile methods of organising work are developed in the production industry and improved in software development. These methods are based on self-organisation and self-regulation and help to improve efficiency, and increase involvement, responsibility, fun and satisfaction, and help companies to survive in faster competition while having raising complexity. This article points out by some examples on what theories these methods are based, how these methods work, and how they can be used in other contexts like collaboration within Universities

Keywords: self-organisation, lean management, collaboration at Universities, responsibility, involvement, meta-cognition

УДК 378.371:53

ЛІТКОВА О. І.,
Державна морська академія, м. Херсон, Україна

ЯКІСТЬ МОРСЬКОЇ ОСВІТИ У ВИМІРІ АВТОНОМІЇ ВНЗ

Проблема якості вищої освіти стала ключовою у послідовно ухвалених документах Європейського простору, починаючи з Болонської декларації (1999 р.), закінчуючи Бухарестським комюніке «Використання освітнього потенціалу з найбільшою користю: консолідація Європейського простору вищої освіти» (2012 р.), і посіла п'яту позицію у формулюванні: «Сприяння Європейському співтовариству в забезпеченні якості освіти». [1, С.5]

Вперше національну систему забезпечення якості вищої освіти було унормовано Законом України «Про освіту» (2014 р.) на основі побажань і рекомендацій Європейського простору вищої освіти. Цим документом було передбачено запровадження принципово нової моделі якості вищої освіти, створеної на основі компетентнісного підходу, яка б мала властивості вимірюваності, порівнянності, вірогідності та конкурентоспроможності.

Відповідно до Закону України «Про вищу освіту» (2014) серед інших концептуальних засад забезпечення якості вищої освіти особливе значення належить поширенню автономності і самоврядності на внз з метою перетворення їх з «реактивно діючих на проактивні», тобто такі, що не страхаються ставити перед собою довгострокові цілі і реалізовувати власну стратегію розвитку, є зацікавленими у максимальній ефективності і відповідальними за результат перед споживачем освітніх послуг, перед громадою і перед державою.[1, С.43]

Аналіз останніх досліджень і публікацій свідчить про наявність різних точок зору у науковців відносно трактування вищезгаданих понять «якість вищої освіти» і «автономність внз».