adesso

TAMED AGILITY – THE BEST OF PLANNING AND FLEXIBILITY

New-School-of-IT.de

CONTENT

Dear Readers,

“Corporate IT is going to experience radical change” – that is the core message behind the concept of the New School of IT. You might wonder if there has ever been a time when IT development development stood still. Changes, radical transformations and paradigm shifts – triggered by new technologies - are a constant part of all our lives. Your current workspace has very little in common with a workspace from 2004 and virtually nothing in common with a workspace from 20 years ago. So what is so special about the current situation? Why do we describe the present developments with the drastic image of a “RevolulTon”?

Because this is the moment when three developments that were previously seen as separate have been united. Mobility, agility and elasticity have joined forces to change the rules for successful corporate IT. What does that mean for your company? Expensive software projects that take months to deliver results are not the right approach for your mobile applications, which require short development cycles. **Mobility drives agility.** The rapid and flexible development of software that is published weekly or daily cannot be held up by static IT infrastructure. **Agility drives elasticity.** Your systems must be able to handle 100,000 mobile users just as reliably as 100 users. **Elasticity allows mobility.**

These examples demonstrate that there are many connections; development is interdependent and drives further development. Such changes are like waves that meet and reinforce each other. The New School of IT shows you how trends look like, which interdependencies are typical and which instruments allow you to react perfectly.

However, we are not just facing a technical revolution. The New School of IT also means that the position of IT in companies is changing. Identifying connections, establishing new business processes and reaching new target groups: IT departments are being called on more often to provide the basis for such activities. Companies are becoming increasingly “digitised” and IT is freeing itself from the role of supporting specialist departments. It is going from being driven by new developments to driving them.

We want to guide you along this path, one we consider revolutionary. Please do not hesitate to contact me. Let’s discuss the effect that this developmental dynamic will have on your company and department.

Best regards, Prof Dr Volker Gruhn

P.S. I present the central themes of the New School of IT in four minutes at New-School-of-IT.de.

Tamed agility – the best of planning and flexibility

1. Introduction

For decades, a more or less fixed division of labour between the areas of software development and IT operations has been established: one side designs models, programs and tests solutions according to complete specifications as part of lengthy projects. The other introduces new software into product operations based on long-term planned release cycles and ensures that the appropriate applications function reliably.

This model has been very successful; otherwise, IT would not be where it is today. About ten years ago, however, requirements changed: IT increasingly started to make significant inroads into corporations. Greater digitisation of processes proved to be a competitive advantage. Such companies needed more new software and, above all, they needed it faster. At the same time, new competitors entered the market who managed to implement new programs in weekly or daily cycles, instead of the usual one or two-year cycles. Such companies, which tended to be start-ups in the digital industry, could react to the needs of users or customers quickly – flexibility which they could increase thanks to their market position. That is how traditional companies with traditional IT recognised the benefits of agile concepts.

But it is not purely about how IT departments function. It is also about how CIOs and their area of responsibility are perceived. Do they and their team merely meet the different requirements of the specialist departments? Or can they capitalise on the massive changes that companies are currently facing – such as the unchecked trend towards more mobile applications and business processes – and give IT the importance it deserves?

Several factors are presently working together, reinforcing each other and radically changing not only IT departments, but entire companies. The New School of IT analyses the relationship between the three main development drivers – mobility, elasticity and agility – and recommends action plans for decision-makers based on its findings. Several effects resulting from the trend towards agile project design and software development are presented in detail in the following.

2. Agile software development

The basic idea of agile software development is to develop software quickly and flexibly, to create executable versions within a short time frame, to reduce rules and bureaucracy to the bare minimum and to keep specifications and documentation within tight limits.

Agile development does not completely omit rules and documentation. One of the basic assumptions, however, is that not all requirements can be known at the start of a project and, as such, the entire development process cannot be planned completely. In agile projects, software development is understood as an ongoing learning process. Amendments are not avoided during the project, nor are they viewed as resulting from lacking specifications, but as a sign of progress.

The main components of agile software development include:

- User story: Software system requirements are formulated in a readily understandable sentence, for example: "The department head wants to access regional sales figures, broken down by customer groups."
- Product backlog: All system requirements are collected here and prioritised by the product owner. The backlog is not a static filing system, but rather is continuously updated during the project.
- Sprints: Functional releases are produced in short cycles. The requirements from the product backlog are processed sprint by sprint.
- Test-driven development: Matching the step-by-step approach to software development in agile processes, this is also tested in small units. Test-driven work means that first the test code is written and then the actual program code that meets the test requirements.

Agility should not be understood as an alternative set of tools to pick a few concepts from in order to use them in otherwise conventional software development processes. Agile software development is a whole new perspective on the development process. That is why it initially presents a challenge to corporate culture and the understanding of the work to be done by those involved in the project. A company has to review its tried-and-tested concepts of project organisation and responsibility and, if necessary, abandon them.

3. Tamed agility

A discussion about plan-driven approaches or agile software development can quickly take on a missionary tone among professionals – which is not surprising, as plan-driven and agile models seem to be worlds apart. On the one hand, plan-driven models rely on the assumption that specifications are largely complete and subsequent adjustments should be avoided. Agile models, on other hand, often have the reputation of waiving many project standards, like proper documentation. In IT departments, however, it is not ideology that counts but facts and results, and here it is useful to combine both approaches.

Both concepts have benefits when developing applications:

- It is simply unrealistic to expect that all requirements can already be fully specified when starting to develop software. Adjustments that become apparent during the development process cannot just be ignored.
- Complete flexibility is not compatible with delivery deadlines and planning budgets. Fundamental requirements for the planning of projects have to be met to make it possible at all to draw them up and implement them in a corporate environment.

3.1. Agility and planning – the right balance

It is necessary to combine the advantages of agile software development with the security of being able to plan a project. Those in charge need to find the optimum balance between planning and flexibility. And they should draft an organisational framework that allows for agile projects. Companies have to tame agility to ensure that it can achieve its full potential.

Pure agility is more of a theoretical construct than a realistic approach for companies. In practice, a balance between the benefits of agile software development and the requirements of planning will always have to be found. The right ratio depends on a number of factors and can vary even between companies in the same sector that have similar application environments, distribution channels and products. Finally, the personal preferences and experiences of those involved in dealing with agility are also important. The more inclined important project members are towards agility, the greater the role this concept can play.

Other important factors are the size and importance of a project, the dynamics of its environment, the particular corporate culture and the development team's industry expertise.

Finding the optimum balance between planning and agility can be a difficult process, but it can be facilitated with the appropriate concepts.

3.2. Concepts to tame agility

Agility acts in two different directions within a company: firstly towards users – customers or contracting entities – and secondly towards IT operations.

- The agile process requires that IT professionals and specialist departments cooperate closely when developing new software systems. The specialist and IT departments often lack a general understanding of each other, which affects the process and project results negatively.

This calls for concepts and tools that can build a bridge between the different groups involved in the process. The aim is to ensure that agile concepts can also be developed in interdepartmental cooperation. One such tool is the Interaction Room (IR), which helps to create solutions based on a common understanding of the problem by providing a simple set of rules and the space to visualise a project.

- On the other hand, the agile process impacts IT operations. Traditionally, such work has been characterised by a few major software releases that occurred at longer intervals – developers thought in months rather than in weeks or days. Agile software development sets a new pace and produces new releases every day or even every few hours in extreme cases. The frequency of the releases requires new processes, both in software development and IT operations. The existing boundaries between the departments have to be overcome in this case as well. One proven means of achieving this is to combine development and operations in one organisational unit – DevOps.

In addition to organisational issues, the keywords "continuous integration" and "continuous delivery" also speak to certain approaches in terms of software that can be used to tame agility. Installation routines that continually check whether modified software can be put into operation provide support to IT departments publishing new releases. The use of these routines makes it possible to translate the boost of momentum resulting from agile development into the operational processes.

3.2.1. The Interaction Room

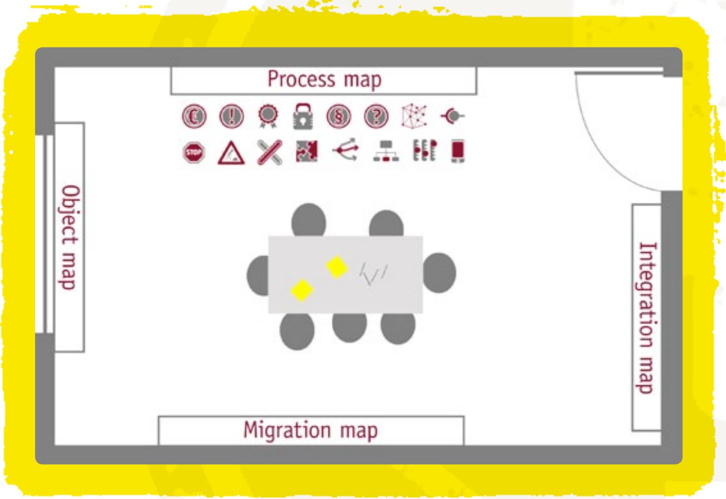
The concept of the Interaction Room was developed primarily to improve communication and collaboration between specialist and IT departments at work on developing solutions for critical business processes – a problem that frequently comes up during software development projects. The one side possesses detailed knowledge of work processes and structures, while the other has the IT expertise necessary for developing and operating the systems. It is imperative that both sides understand each other. Different objectives, work methods, conceptions of the project and jargon all serve to complicate a mutual understanding.

The Interaction Room is a medium that allows specialists and IT experts to communicate more effectively with each other. Improved communication is the result of portraying processes in a way that is transparent and not dominated by the IT perspective. The transparent depiction allows representatives from the specialist departments to participate in the project discussions. Without a neutral medium such as the Interaction Room, relevant topics quickly become subject to a great degree of technical talk that non-IT experts have great difficulty following.

The Interaction Room is a real – not a virtual – space. The walls in the room are equipped with whiteboards, which help users to visualise processes and to illustrate project details. Under the guidance of a moderator, the members of the Interaction Room work through the processes on the walls, record information and visualise project progress. Each of the four walls represents a central aspect of the project.

The Interaction Room is perfectly suited to agile models. It is important to discuss all relevant aspects of a project, especially if those involved with the project forgo an extended phase of formal specification. Updating cost estimates and continuous budget tracking – such as with the help of value annotations – tame agility. The Interaction Room provides an organisational communications structure whose design is both flexible and transparent. The Interaction Room has already proven itself for use in projects. Barmenia insurance, for instance, implemented an extensive SEPA project with the help of the Interaction Room.

More about the Interaction Room: New-School-of-IT-.de



The schematic representation of the Interaction Room

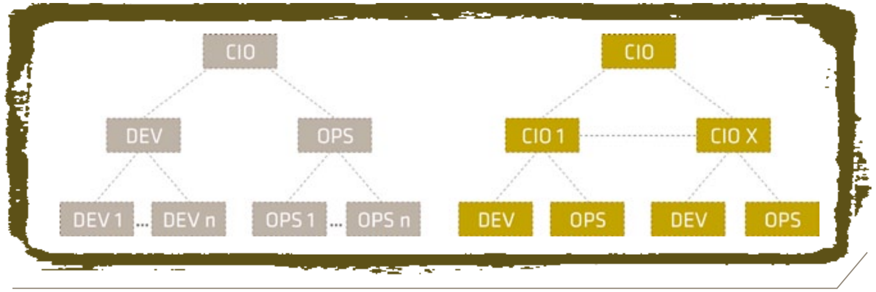
3.2.2. More flexible organisation with DevOps

Traditional IT has typically been organised as follows: There is someone responsible for the entire development of the application, and he or she has a counterpart responsible for IT operations. The two reporting lines normally come together with the person responsible for overall IT or the CIO. This system has several advantages. One key advantage is that overlapping expertise in development and operations only has to be represented once. This ensures optimum use of resources, especially in light of expensive hardware and specialised operational processes.

This type of IT organisation proves not to be flexible enough for ever-shorter release cycles, however. If problems arise during the process, multiple hierarchy levels have to be consulted to determine if the software or its operation is faulty, which impedes the error analysis. Meanwhile, systems are taken offline for an unnecessarily long period of time, which does not live up to today's demands for high availability of a steady stream of new releases.

The problem is caused by a structural basis formed by the different perspectives and aims of those involved in the project. Software development aims to – indeed, must – develop and release software with the functions specified by the customer as swiftly as possible. The goal of IT operations is to ensure stability and continuous availability.

The organisational concept "DevOps" – Dev from application development, and Ops from IT operations – is based instead on joint responsibility for the results as regards IT applications across the value chain. The aim is to overcome the points at which breakdown occurs between application development and IT operations on a lasting basis. The new organisational structure is able to produce and provide software more quickly and with fewer bugs.



DevOps organisation

On the other hand, however, it is necessary to double up on several areas of expertise and overlapping services in DevOps organisational units. This in turn impacts the cost structure of IT. And a DevOps team will not act fully independently: it is necessary to continue providing the team with commodity services such as databases, standard hardware and other infrastructure systems. The form of organisation – or the mix of organisational forms – suited to a specific company must be determined on a case-by-case basis.

3.2.3. Continuous integration and continuous delivery

Software developed as part of an agile process can only be put into operation in keeping with the frequency of releases if the systems have been modified accordingly. The software, in addition to the development, test and production environments, has to function automatically to a significant degree. For the most part, such a high degree of automation has not been necessarily until now, as the associated costs would have proved to be uneconomical for companies with only a handful of releases every year. In addition, the structure of the objective is more than likely to change if there is a longer interval between two software updates.

The rather static conditions for traditional software development require manual work. The labour-intensive processes depend greatly on individual employees, however. Unforeseen absences, the loss of key experts or specialists with varying degrees of qualification can all create problems, even to the point of bringing the process to a halt.

In an agile landscape where the time between releases amounts to mere days or even hours, operational IT processes have to be organised differently. In such cases, installation routines are called for that continually check whether modified software can be put into operation. This approach is at the heart of the concepts known as “continuous integration” and “continuous delivery”.

Current software solutions offer a wide range of possibilities, from addressing heterogeneous infrastructures through to the automated provision of environments and their integration into existing IT infrastructure, such as Puppet or Chef. These solutions also include tools for implementing monitoring and logging mechanisms, for automated testing, for managing software components, such as Nexus, and for orchestrating the entire process, such as Jenkins.

Providing the infrastructure is turned into a single process which can be supported by the same methods and, in part, even by the same tools as software development. Examples of this include versioning code for the infrastructure or developing design patterns for certain infrastructure tasks, such as setting up a web server.

THE AUTHOR

Prof. Dr. Volker Gruhn co-founded adesso AG in 1997 and is now the Chairman of the Supervisory Board. He is the Head of the Software Engineering Department at the University of Duisburg-Essen.

4. The role of the CIO in an agile environment

The way that software is developed, operated and used at companies is currently undergoing fundamental changes. Ideas that just a few years ago were only used by start-ups have now made their way into established large-scale enterprises. And this trend is changing more than how IT departments function; it is also impacting what IT topics and ultimately the CIO mean to a company.

At young companies with new business models, IT is the most important means of production in most cases. They define themselves through digital processes and take the requirements of IT as their point of orientation during organisational set-up. In contrast, established companies still often see IT as nothing more than a service provider. At such companies, the specialist departments are responsible for determining the features of a software application; IT is still frequently referred to as EDP, and the department itself is limited to creating or procuring the solution.

This model worked as long as it was not necessary to implement IT-based disruptive innovations. But new digital products, new distribution channels, newly fragmented business processes and new ways of cooperating with partners require precisely this: IT-supported innovations which replace or at least supplement established business models. In the process, IT takes on the role of business enabler, which means that IT departments have to cooperate closely with specialist divisions to demonstrate the potential of the technology before the requirements are specified. They also have to know how the respective markets function.

The old model of slow and continuous improvement of products and processes is reaching the limits of what it can do. Providing IT has to become a more flexible and dynamic process. This is why the influence of the CIO and his or her team can no longer stop at the boundaries of the IT department. Technology and infrastructure will increasingly become a decisive factor in the success or failure of an entire company. The role of the CIO will likewise continue to change from that of a pure provider to the designer of business processes.

JOIN

We prepared further topics related to the New School of IT here:

www.New-School-of-IT.de/downloads

adesso
adesso AG
Adessoplatz 1
44269 Dortmund, Germany
T +49 231 7000-7000
F +49 231 7000-1000
info@adesso.de
www.adesso.de

New-School-of-IT.de