A Socio-Technical Perspective on Computerization[1]

Dr. Mark JG Govers and Pim Südmeier, Msc.

Strangely, information technology (IT) has never played a major role in (re-)designing organizations and work places from a socio-technical systems (STS) perspective. Mumford (2006) was the rule rather than the exception. Regardless, her valuable insights at the time never became a mainstream routine in STS thinking and practice. Three reasons 'force' STS thinking to change this ignorant routine urgently. First, IT systems profoundly determine organizational design choices. Especially enterprise IT systems, like ERP, are not a derivative of organizational design choices anymore. They have built-in organizational designs that are enforced on organizations and humans in workplaces. Second, IT creates the technical context in which many workplaces and organizations are operating. In many cases, IT is the context in which work takes place. Therefore, it is essential to take valuable, new business model opportunities but also potential social negatives coming from IT into consideration during organizational and workplace design processes. If not, the delicate balance between social and technical, pivot for STS, is interfered beyond repair. And third, information, especially the quality of information, is becoming vital in dynamic and turbulent settings in which more and more organizations and workplaces operate. In all, quality of information developed into a major design parameter along with quality of organization and quality of work. For this, IT requires attention from STS practitioners as it creates the architecture in which organizations and humans operate. Inspired by Ashby's Law on Requisite Variety – i.e., STS' core principle -, we are opening with this contribution the door to such a view for STS and IT practitioners.

1. All-in-One and One-for-All as dominant tendency of IT experts

Standardization of work processes has proven its value, also within the STS practice. With the introduction of enterprise IT systems, like ERP, standardization thinking is, however, overstretched with devastating effects on agility and, therefore, on productivity and the health of humans and organizations over time. It results in complex work processes that can be observed during the implementation process of enterprise systems. Heaping everything together is a time-consuming IT technical as well as a political process. It explains why ERP projects often run over time and budget, and maintaining enterprise systems are a pain. Why?

The popular ERP concept, for instance, grounds on integrating all business functions in such a way, that they meet in a work process. The ERP concept is grounded on integrating all business functions in such a way, that they meet in a work process Most operational work processes, like a client process, entails sales, logistical and financial aspects (process steps). With ERP an organization can run these process steps in one progress of a client process. But, in most organizations more similar processes take place. They differ in having various inputs and/or outputs, or in having various process steps. In other words, there is variety. In the enterprise IT practice, unlike the STS, the attempt is to heap all the variety together in one uniform process design and control model. It is done in denial of Ashby's Law – a core STS principles. Embracing variety is considered to go against the concept of standardization of work processes.

[1] Based on two articles published in Dutch: Govers and Sudmeier (2010) and Govers (2012).

Actually, the applied concept standardization is 'uniformization'. It combines two different forms of integration. The first can be called horizontal standardization. It is heaping process steps together in a coupled process flow. The second can be called vertical standardization. It is about optimizing process steps into one generic process step capable of dealing with many process varieties. Figure 2 shows how both work.

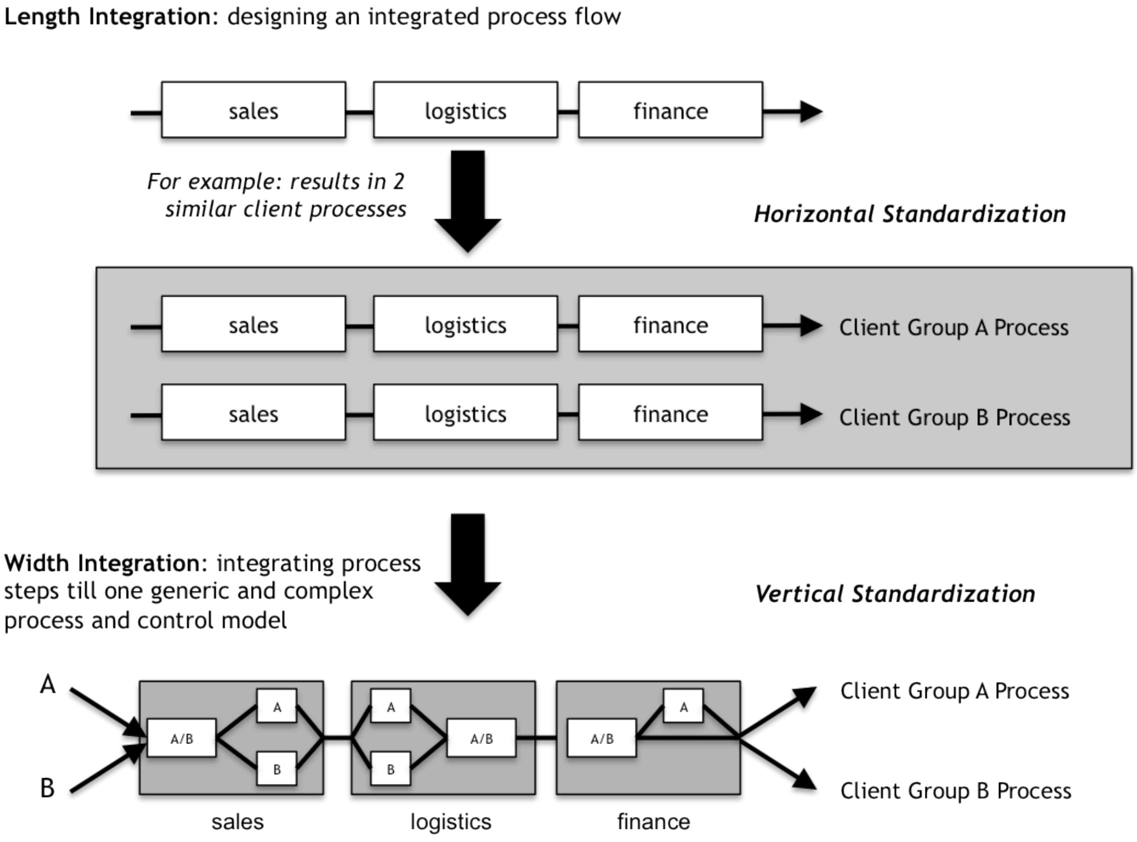
Length Integration: designing an integrated process flow
sales
logistics
finance
For example: results in 2 similar client processes
Horizontal Standardization
sales
logistics
finance
Client Group A Process
sales
logistics
finance
Client Group B Process
Width Integration: integrating process steps till one generic and complex process and control model
Vertical Standardization
A
B
A/B
A
B
A
B
A/B
A/B
A
Client Group A Process
Client Group B Process
sales
logistics
finance

Figure 1: horizontal and vertical standardization

The result of these two standardizations is twofold. Independent process flows (in the figure: different sale processes) are made dependent. The process design seems to be simpler as all variety is put into one standard. However, it seems to slip one's mind, that various more independent process flows are integrated. With this self-inflicted dependency the standardized process flow is more complex then is actually the case in reality. One uniformed process flow was created consisting of several independent process flow varieties. The assumption is that future changes affect these varieties in a similar manner over time. For organizations operating in predictable and stable contexts this is correct. Changes can be implemented in a gradual and planned manner. Efficiency advantages of the uniformed approach remain. If changes increase in numbers and frequencies, the uniformed approach works counterproductively. It takes more time and cost to change the way processes are computerized, as it creates domino effects. A required change in one variety leads directly to changes in the connected other varieties. Why? Due to vertical standardization the actually independent process flows are made interdependent, and this vertical standardization needs to be re-designed which impacts all connected process flows.

The persistency of vertical standardization does not limit itself to purely IT thinking and acting. It is also deeply rooted in control oriented management disciplines like accounting. ERP projects often start with the implementation of the financial module. These control disciplines get the opportunity to pour their control with concrete into the enterprise system. The primary, logistical processes have to find their way in the solidified framework. The variety of such primary processes get into a scrape and has to nestle itself into the set standard. The consequence is a width-way expanding standard: an overstretching standardization.

In short: the bureaucratic dangers of vertical standardization hits double hard. From one site from the technology and from the other site from the dominance from overly control focused management routines.

2. Ashby's Law as inspiration

Having an eye for variety and dynamics is essential to avoid bureaucratic effects of ERP and IT in general. Ashby's law of the requisite variety offers from an STS perspective a different direction to use IT, even ERP, in a non-bureaucratic manner. Inspired by Ashby (1956), two viewpoints are key. First, to determine which type of computerization is needed. And second, to determine how to computerize (primary) processes in enterprise IT systems.

Two types of variety play a role with computerization:

1. How dynamic are information needs: static (low) vs. dynamic (high)?
2. How dynamic is the information provisioning: static (low) vs. dynamic (high)?

This implies four types of computerization, as shown in figure 2.

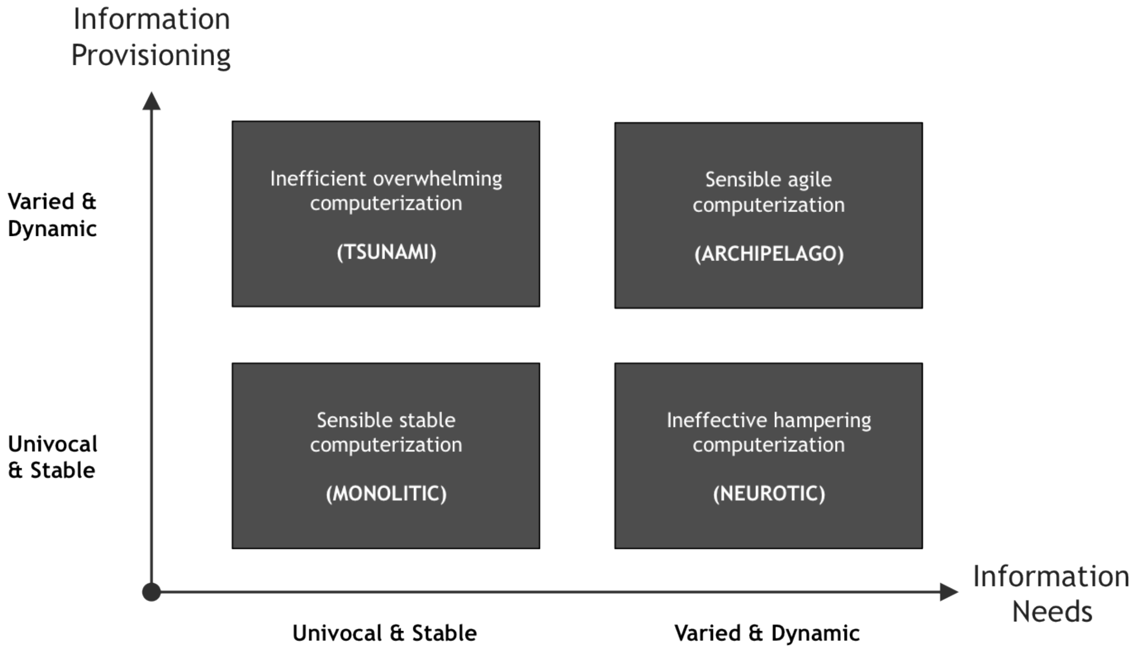

Figure 2: Types of Computerization

Monolithic computerization is efficient an effective if the information needs are stable and univocal. Consequently, the information provisioning can be stable and univocal as well. For example, an ERP solution is feasible for the context of the following producer of ship engines. Their product assortment consists of eight main engine types (cubic inches) with a few known variations per engine type (diesel, gas, ship, land, etc.). The production routings differ sporadic and sales are quite stable. A monolithic computerization makes sense here.

Neurotic computerization occurs when a monolithic approach is used in a situation where information needs vary and are dynamic. In a way the computerization is overwhelmed by the users' ever-changing information needs. It is tough, often impossible, to fulfill these information needs on time. To illustrate such information stress an example from our own practice. As a company we where looking forward to shifting to iPhones. One colleague, though, wanted to stick you his Nokia. No problem you would say to extent our phone contracts with our telecom provider. After half an hour calling to headquarters and clicking through all kind of menus, the sale person became stresses. The sales person could not offer varied contract forms, as he could not get in into his computerization. Luckily another provider could, as their systems allowed sales person to differ contract forms and to draw-up varied agreements.

Tsunami computerization occurs when a monolithic approach is used in a situation where users need have stable and simple information needs. Users are overwhelmed users by the computerization offering too many options (variety) without information needs asking for it. As a consequent, certain users cannot see the wood for the trees anymore. Hospitals are exemplary. The number of IT systems and digital protocols are so large that many users have difficulties with patient information requests. Quite often doctors and nurses have no idea how to find the simplest information need. This creates a different type of information stress: in need of information (provisioning) but having difficulties to find it in the systems.

Without knowing, many organizations work with tsunami and neurotic computerizations and are in a situation of structural information stress. Information stress implies a misbalance between the used and offered information provisioning and the actual information needs in an organization and at workplaces. In IT jargon it is called: misalignment (Henderson & Venkatraman, 1993).

Like work stress (Karasek, 1979), information stress affects quality of work and organization (De Sitter, 1994; see also: Vriens & Achterbergh, 2011) negatively. Knowing that in the current information era IT affects organizations and humans, the necessity grows of having information provisioning capable of dealing with various and dynamic information needs. Govers (2003) calls this ***archipelago*** computerization.

3. Archipelago thinking as an STS alternative

The socio-technical ordering principle of De Sitter (1994; see also: Vriens & Achterbergh, 2011)) grounds the outlook for archipelago computerization that suits STS designed organizations and workplaces. This design principle for designing organizations and work places offers guidance for designing computerization as well. Translated to computerization, it boils down to the following design order:

1. Reduce information needs via complexity reduction by creating independent primary process flows.
2. Increase information provisioning by creating the requisite information variety for each primary process flow.

Information needs can structurally be reduced by complexity reduction. For this, De Sitter (1994; see also: Vriens & Achterbergh, 2011)) offers an effective design framework for the diversification of primary processes. It reduces the complexity of relations with the environment and reduces the internal interdependencies. Looking for independent parallel market or production flows (streams) is the first step. Within these streams looking for segments of strongly coherent activity, is step two. Both steps, applied by designing computerization, implies that each stream gets, ideal typically, its own computerization to deal with the variety and dynamics of that stream. Basic data, like customer info, are computerized and connected 'under the water line' to provide overall management information; a data warehouse architecture can be used for this. 'Above the water line' each process stream has its own options and progress of primary and supporting processes. Like an archipelago, islands are connected under the water line, and are disconnected above the water line. In practice an archipelago computerization (see figure 3) can consist of various (here: three) parallel, independent enterprise systems instead of having an all-embracing one.

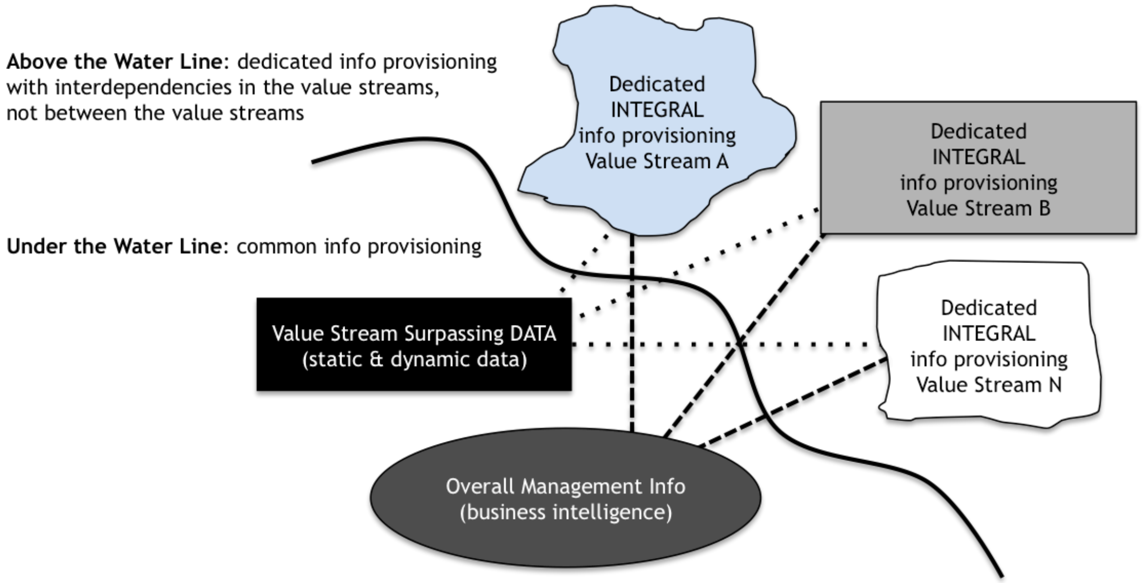
Above the Water Line: dedicated info provisioning with interdependencies in the value streams, not between the value streams
Under the Water Line: common info provisioning
Dedicated INTEGRAL info provisioning Value Stream A
Dedicated INTEGRAL info provisioning Value Stream B
Dedicated INTEGRAL info provisioning Value Stream N
Value Stream Surpassing DATA (static & dynamic data)
Overall Management Info (business intelligence)

Figure 3: Archipelago Computerization

A light version of archipelago can be a menu card structure. Like in a cafeteria, a menu of an enterprise system is built around clear-cut, varied processes. Figure 4 shows the difference of having everything in one purchase process design or having the different varieties in a purchase menu card design. Related to the previously discussed horizontal and vertical standardization, it recommends avoiding vertical standardization. We need to avoid it, as creating interdependencies of process steps increases the probability of information stress. Besides that, it causes time and money consuming implementation and maintenance as explained before.

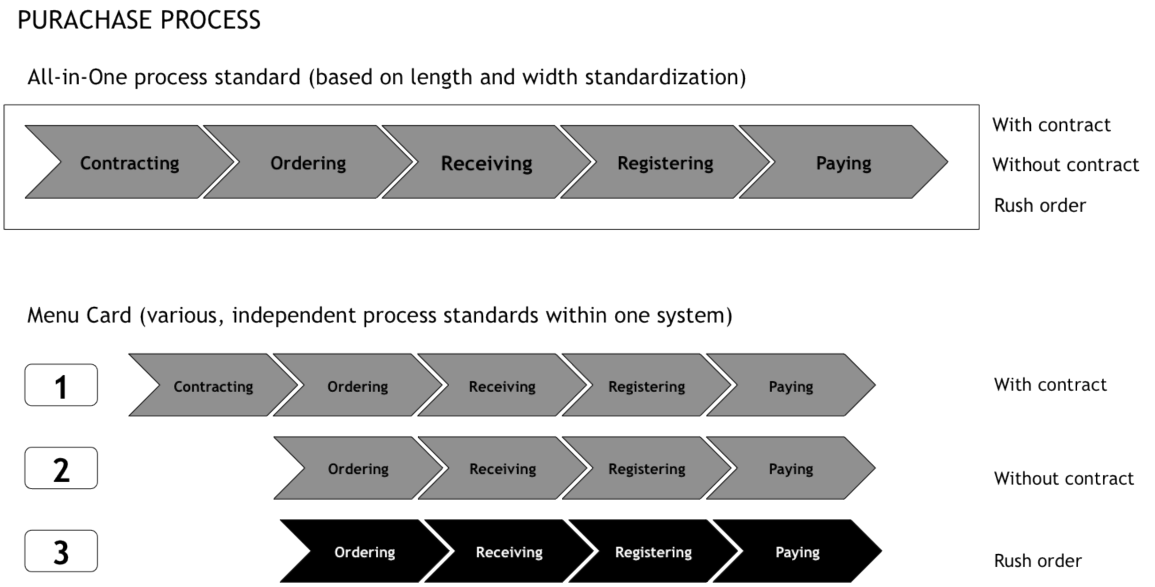

Figure 4: Menu Card concept

4. Unlearning old and learning new routines

Archipelago computerization is more then just architecture. It requires from IT and organization design experts to let go of routines. IT design experts who distance themselves from the routine "avoid redundancy". IT experts are shy of taking similar processes in parallel if parts (steps) look alike. They prefer to apply vertical standardization. Giving each parallel process stream its own data is even more taboo. With modern IT, this is not an issue anymore; even with ERP systems. It almost dogmatic that holding on to 'avoid redundancy' and vertical standardization is not necessary anymore. With letting go of both routines, IT design experts help to prevent IT driven bureaucracy. **For organizational design experts it implies that they have to let go of the routine that IT has to support the organizational architecture they designed. They have to understand that production of product and (especially) services happens more and more within the context of IT**. Computerization is not a derivative of an organizational design anymore. As IT and computerization create a technical framework in which work processes and workplaces are taking place, the **information (computerization) design has to go in parallel with the organizational design**. Changing both routines also implies that **both experts need to start working with each other as a team** instead of 'against' each other.

Sole monolithic IT routines are the wrong answer to facilitate healthy and productive organizations and workplaces. At least two new, archipelago, IT routines need to be learned that build on STS thinking. Instead of focusing with a functional view on business processes (finance, purchase, HR etc.), we have to start focusing with an integral view on primary processes from input to output. And instead of searching for the greatest common deviator, we have to start searching for the smallest deviating variety when designing work processes. As shown in figure 4, we should not look to bring the three various purchase processes together in one if they have some steps in common. Instead, they should embrace variety and design three processes independent from each other, and as such computerize the three. If not, the varieties are made interdependent which has undesired effects in terms of time and cost when confronted with dynamics, as explained before. It is not self-evident to apply these new routines. It requires leadership to dissociate IT and business professionals from their common functional differentiated and 'uniformization' thinking and acting.

5. Upcoming new IT approaches and technologies

In the meantime, the IT world is not sitting still. In terms of project management, so-called 'scrum' and 'agile' are new ways of developing software that are focusing on teamwork (Agile Manifesto, 2001) – similar to STS. Emphasizing teamwork to develop software, 'scrum and agile' have no outspoken view on organizational and workplaces design. In a scrum and agile way of working, the mentioned routines of having a functional differentiated and vertical standardization (standardization) are not brought upon discussion. Scrum and agile help to speed-up the design processes of IT, but this does not imply new – like archipelago – design routines for organizations and workplaces. STS has to be aware of this, as scrum and agile are getting that popular in the IT world that what really needs to be changed remains untouched. This is the outcome of the IT design process: an archipelago architecture, if information needs are various and dynamic (see figure 2). STS can help the IT world not only to speed-up IT processes, but also help them to deliver aligned information provisioning.

In addition to project management, also new IT technologies are of interest like 'always-on' connectivity, cloud computing, and apps. By means of 'always on', users can be online and have access to systems and information any place and any time. With cloud computing data is stored in databases accessible with internet anytime and anyplace, and from any device. Apps offer a user the opportunity to assemble (construct) their own information provisioning out of small applications. We assume that apps focusing on enterprise related information needs, so-called enterprise apps (Kerschberg 2015), are particular of interest for STS. Enterprise apps provide the tools to design specific and dedicated information needs for value streams. Even more, we predict that an information architecture built with enterprise apps will be easier to maintain and to renew.

For STS, the upcoming development of 'enterprise apps' is especially interesting. More organizations, or parts of them, work in turbulent fields. In such fields, the life cycle of value streams is low. It means that value streams decay more frequently and rapidly, and consequently, new ones have to be created and designed. A trend is to **create such value streams in co-creation with other organizations in temporary network settings**. Such value streams require dedicated, agile and information provisioning crossing organizational boundaries. We believe that enterprise apps positioned in an archipelago architecture offers the framework for this.

Mentioned upcoming IT approaches and technologies offer new opportunities for designing organizations and workplaces. The archipelago architecture offers an integral framework to position these in conjunction with enterprise IT systems for value streams.

6. Impact for STS Design

The impact of new IT opportunities and computerization is far reaching. Besides – the traditional – impact of technology on work and coordination relations, information becomes an imperative aspect for designing work in teams, value streams, organizations and even organizational networks to stay tuned with the changing eco-systems organizations interact with. Information can no longer be approached as a derivative of the design of work and organizations. Information and information technology have evolved into a key design issue for work and organizations.
"

The viability of monolithic computerization designs are becoming absolute, and are migrating to varied and dynamic information archipelago architectures. This shift offers new perspectives for STS experts to apply STS design principles to help organizations and IT experts to design such information architectures. Besides executive and regulation tasks, **STS experts have to start embracing information tasks in their designs** as well. It boils down to design questions like: **what information is needed in teams to perform efficiently and effectively, and how to design such information tasks effectively in agile information architectures**?

7. Conclusion

It may sound paradoxical: STS design principles offer a toolkit for designing non-bureaucratic computerization without being aware of it. Based on one of its core grounds, Ashby's Law, the notion and affect of variety and dynamics are made clear for designing information provisioning aligned with changing information needs. Key is to understand that information needs are not univocal and stable. They are becoming more varied and dynamic, and, therefore, ask for varied and dynamic information provisioning as well. In this contribution and for that purpose, archipelago computerization was introduced and developed. Archipelago thinking asks to depart from old routines and to embrace new routines. The old, bureaucratic determined, routines like taking a functional-differentiated view on business processes and over-standardization processes are fatal for rapidly designing flexible computerizations. For this, STS determined new routines are required like focusing on the primary process (again), and on looking for the smallest deviating variety. This shift in routines is in urgent need of STS experts with an eye for information technology, or IT experts with an eye for socio-technical thinking and acting. Upcoming IT trends offer new options to engage IT and STS into a mutual strengthening efforts to design healthy and productive organizations and workplaces.

References

Agile Manifesto (2001). Manifesto for Agile Software Development. See: http://www.agilemanifesto.org, d.d. 25 January 2015.
Ashby, W.R. (1956), An Introduction to Cybernetics, Chapman & Hall, London.

Kerschberg, B.(2015), *"Four Critical Reasons to Build Enterprise Apps"*. Forbes TECH, 15 Jan. 2015.

Govers , M.J.G. (2003), Met ERP-systemen naar moderne bureaucratieën?, PhD Radboud University Nijmegen. (Setting our for modern bureaucies with ERP systems)

Govers, M.J.G. and P. Südmeier (2011), *“De Sitter in the informatietijdperk”,* M en O : tijdschrift voor organisatiekunde en sociaal beleid, Vol. 65, No. 2, pp. 31-45. (De Sitter in the information era)

Govers, M.J.G. (2012). *“Gevraagd: nieuwe agile IT-principles”,* Automatiseringsgids, 5 dec 2012. (Wanted: new agile IT principles)

Henderson, J. C., and N. Venkatraman (1993), *“Strategic Alignment: Leveraging Information Technology for Transforming Organizations*”, IBM Systems Journal, Vol. 32, no. 1, pp. 4-16.

Karasek, R.A. (1979), *Job demands, job decision latitude, and mental strain: implications for job design”*, Administrative Science Quarterly, Vol. 24, pp. 285-308.

Mumford. E. (2006), “*The story of socio-technical design: reflections on its successes, failures and potential”*, Info Systems J, Vol. 16, 317–342.

Sitter, L.U. de (1994). Syngertisch Produceren, Van Gorcum , Assen. (Synergetic producing)

Vriens, D. and J. Achterbergh (2011*),”Cybernetically sound organizational structures I: de Sitter's design theory”*, Kybernetes, Vol. 40 Iss: 3 pp. 405 – 424.