Operationalizing Coordination of Mega-projects - a Workpractice Perspective

Lars Taxén and Joakim Lilliesköld

Linköping University Post Print

N.B.: When citing this work, cite the original article.

Original Publication:
Lars Taxén and Joakim Lilliesköld, Operationalizing Coordination of Mega-projects - a Workpractice Perspective, 2006, IRNOP VII Project Research Conference, Xi'an, China, October 11-13.
http://dx.doi.org/10.1016/j.ijproman.2008.05.009

Postprint available at: Linköping University Electronic Press
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-54024
Operationalizing Coordination of Mega-projects - a Workpractice Perspective

Abstract
In this paper we propose a workpractice approach towards operationalizing the coordination of mega-projects. The approach matured over several years in development practice at Ericsson, a major supplier of telecommunication products and services worldwide. Key points in the approach are the management of critical dependencies and the construction of a communal understanding on how to coordinate projects. Coordination is seen as a workpractice, where actors provide coordination services to the project. We discuss some results from the application of the approach in relation to an ideal-type classification of mega-projects such as Weber, Rambo and Gaia. The main conclusion is that the suggested approach enables the coordination of extraordinarily complex mega-projects.

Introduction
The topic of this paper is the coordination of large projects. The management of mega-projects is where modern project management began in the 1950s. However, even though the management of projects has received enormous attention and analysis throughout the years, the track record has been generally poor, especially for the larger and more difficult projects (Morris et al., 1987). Already in the early American military programs (like the Atlas and the Polaris programs), coordination was a big and challenging task. However, what makes it even more challenging today is that people in projects work differently (more team-oriented than in the past). Projects are distributed, several organizations may be included, the systems we build are different (more software than hardware), we are more often in a hurry (shorter and shorter time frames), customer demands and requirement specifications change constantly, and systems are built upon standard components instead of components specially developed for the system at hand. These circumstances increase the management complexity, and thus the coordination that must be done.

In this paper, coordination is conceived from a workpractice perspective. By a workpractice, we mean a socially organized setting where actors / producers produce an outcome for some clients (Schatzki, 2001, Goldkuhl and Röstlinger, 1999). The actors become proficient in their particular areas such as project management, system analysis, design of software or...
hardware, etc., thus creating his/her own workpractice world view based on their specific
tasks; specific tools and methods, rules, norms, etc. (Orlikowski, 2000). A project manager
will become familiar with instruments like work breakdown structure, Gantt diagrams, etc. A
software designer may be proficient in using use case diagrams, object oriented design. Sales
personnel are acquainted with customer needs. Production engineers know how to make a
product producible.

In general, most of the actions carried out in one workpractice tend to remain local to that
particular practice. However, some actions may influence other workpractices. Due to the
complexity of the system, it is difficult to anticipate the consequences of a certain action on
the development of a task as a whole. An action, which may appear perfectly feasible in a
certain workpractice, may result in unmanageable consequences for other workpractices.
Thus, in complex systems’ development projects, there is a need for instruments by which
actions in different workpractices can be coordinated. Such instruments must have at least
two qualities. First, there must be a shared or communal understanding about the meaning of
these instruments. However, this is not a straightforward task. Achieving communal
understanding has been identified as a major challenge (e.g. Davenport et al., 1992; Syed,
1998; Kraut & Streeter 1995; Marjanovic, 2005). Second, the instruments should enable
actors to recognize which actions are possible and anticipate the consequences of these
actions for other workpractices. Traditional tools and methods do not appear to have theses
qualities. To this end, Ericsson, a major provider of telecommunication systems around the
world, has introduced a construct called the “anatomy” in their system development practice.

Based on the workpractice and the anatomy, an approach towards managing mega-projects
has been chiselled out from experiences at Ericsson. The main point of the workpractice
instrument is that it enables the construction of a shared or communal understanding of how
coordination should be conceived. The main point of the anatomy instrument is that it enables
the planning and control of the project based on the basic functional dependencies in the
system. These two instruments are the core of an approach called Integration Centric
Development (ICD), which is described in detail in Taxén (2006). The purpose of this paper
is to discuss how the ICD approach meets the challenges of coordinating mega-projects. We
argue that ICD is one way of operationalizing or concretizing the often-vague conceptions of
coordination found in the literature (Larsson, 1990; Malone and Crowston, 1994).

The paper is outlined as follows; first we describe the background of the research and the
research design. Then, we discuss different approaches on how to manage projects, and
especially how these approaches affect coordination. In the following sections we describe
the ICD approach and some of the coordination effects of using this approach in 3G
development projects. Next, we discuss the results. Our main conclusion is that the
suggested approach enables the coordination of extraordinarily complex mega-projects.

Research background and design
The anatomy has been used at Ericsson since the early 1990s. From approximately 1995 one
of the authors, Taxén, was involved in developing a method package, based on the anatomy,
for incremental development of large software projects. In 1998, a product data management
system; Matrix from Matrix-One (Matrix-One) was introduced at Ericsson to support
incremental development. During the period 1998 to 2002, Taxén carried out action research
on the implementation of the Matrix system at one site at Ericsson, both as a researcher and
active implementer. This work resulted in a framework that was subsequently elaborated into a theory; the Activity Domain Theory (Taxén, 2003; 2004; 2005).

Consequently, the research design can be seen from several perspectives: as a longitudinal case study, (since the research involves empirical observations from the Ericsson practice between the early 1990s until the end of 2002); as action research (Baskerville and Wood-Harper, 2003); as a single-case multiple-unit of analysis case study of the effects of the framework intervention in the 3G development practice (Yin, 1989). It is classified as single-case since it concerns only one organization and one case. Given that the effects are studied both from the framework as a whole as well as from its constituting elements, it is classified as a multiple-unit study. The case study database was built up from several different sources, including 14 interviews with participants in various roles such as project managers, method and tools coordinators, configuration managers, etc. Further, about 50 internal Ericsson documents, meeting notes, etc., were used.

Since the study was a result of an ongoing interaction between theory generation and application in practice, the research questions evolved during the study. This is not in line with the ideal design of a case study, where the questions are formulated in advance and remain unaltered during the study (Yin, 1989). However, the research questions and the empirical results were matched during the study. In this way validity, which concerns the relation between the results of the study and its research questions, was secured. The reliability of the study was grounded in several ways: triangulation, contributions to peer reviewed conferences and publications, observations over long time, and making bias explicit. Moreover, the results were grounded in industry by seminars, industrial fairs, demonstrations, etc. The relevance of the study is confirmed by the usage of the framework in the Ericsson practice. From 1998 until mid 2002 about 140 main and subprojects were impacted by the framework. The most notable result is that the framework was a key element in the coordination of the development of 3rd generation of mobile systems, which was the most important project at Ericsson at that moment (2002). A thorough account of these issues is given in Taxén (2003).

Managing mega-projects
Projects are unique by definition, and are of course managed and coordinated differently. One way to classify different approaches to managing projects is made by Marmgren and Ragnarsson (2001) who define three hypothetical approaches; Weber, Rambo and Gaia. A similar classification can be found in a thesis by Adler (1999) based on similar material.

In short, the traditional approach of managing projects is referred to as Weber. The basic idea is that complex projects can be managed and controlled by breaking down the task into smaller and smaller pieces, rigorously planning each piece. It is based on the assumption that the more detailed the planning, the more control it provides. This approach is found in most project management literature as the preferred approach to managing projects (Cleland and King 1988, Meredith & Mantel 1995, PMBoK 2000, Kerzner 2001, Milosevic 2003). In complex and ever-changing projects like many of today’s mega-projects, this approach often becomes insufficient. These projects might last a couple of years, and should be state of the art when they are delivered; making it basically impossible to create a rigorous plan during the start-up phase of the project due to rapidly evolving technological development (Marmgren and Ragnarsson, 2001).
The *Rambo* approach is fundamentally different from the traditional “Weber” process. The basic idea behind the approach is to have much of the development and design work driven by integration and verification in order to be delivered on time. The approach guides actors to focus on the most troublesome boundaries from the beginning in order to avoid surprises in later phases of the project. The insight that this approach provides is that advance rigorous planning and project optimization only provide the project with an illusion of being in control in complex system development projects. Weber does not support the actual work or provide any reduction of complexity or ambiguity in the project (Adler, 1999). The Rambo approach implies new perspectives on the development process, requiring more focus on the dynamics and the acquisition of new knowledge created in the project, rather than strictly adhering to the pre-optimized plan. In order to obtain this, two issues should be addressed: “What should be delivered to the customer?” (a simplified system architecture) and “How should the parts be integrated into a system that can be used by a customer at the end?” (an integration plan) The approach is thus based on the processes of product build-up and of integration and verification. In order to start all parts of the project early, all entry criteria are heavily downplayed in the Rambo approach in contrast to development methods following corporate stage-gate models. The Rambo approach presumes that everyone involved should consider the project and the system as whole, and not only focus on their own sub-part of the project. However, problem solving and coordinating the interfaces between subprojects and between the project and its surrounding is solely managed by a small team surrounding the project manager. Thus, if one or a few of these persons disappear from the project, it can cause serious problems.

The last approach, *Gaia*, is the least common. It has several similarities with the Rambo-approach. The main difference is that the project groups should organize and coordinate themselves, instead of being dependent on the small team around the project manager. The organization or the groups are viewed as a system. In this case, it is not only a few that have the knowledge of the project and the system as a whole, but a larger group. Changes that occur throughout the project can be coordinated at the proper level since more people have knowledge and input about the project. In order for this approach to work, the project has to be managed through easily understood pictures of the goal and the resulting system in order to create a communal understanding of the project.

None of the approaches described are sufficient in all cases, but neither are they useless. One major difference between the Weber approach and the two other is that Rambo and Gaia are based on the actual needs in the design process and are not derived from management needs such as decision support. This means that the approaches mirror the actual work processes and therefore become an integrated part and a shared map of the actual development work and its progress (Adler, 1999). Weber is useful when leading large projects where the prerequisites or the circumstances surrounding the project do not change or fluctuate. If changes occur, Weber creates a substantial additional amount of work due to re-planning; which is often described in the literature as the main problem with network plans; they become unmanageable when there are many changes. Rambo works adequately when management has the knowledge, skills and the capacities that are required for this approach. Other advantages and disadvantages of the approaches can be found in Table 1.

<table>
<thead>
<tr>
<th>Weber</th>
<th>Rambo</th>
<th>Gaia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advantages</td>
<td>Disadvantages</td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Indicates clear divisions of responsibility, planning and following up</td>
<td>Rigid bureaucracy, poor creativity and flexibility</td>
<td></td>
</tr>
<tr>
<td>Facilitates good control towards plan</td>
<td>Difficult to manage the unexpected</td>
<td></td>
</tr>
<tr>
<td>Provides a clear management All essential information is gathered amongst few persons Facilitates rapid and competent decisions</td>
<td>Wears out key persons, requires that the organization consult or confer with Rambo before acting. In turbulent situations Rambo will become a bottleneck for the project</td>
<td></td>
</tr>
<tr>
<td>Creates common goals and common rules of the project</td>
<td>Can lead to butter-fingered, bad control over the project status, risk of chaos and that things falls in between cracks</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 – Pro and cons of each approach (Marmgren & Ragnarsson)

**Coordination needs**

Each approach described above implies differences in the coordination effort. Coordination can be defined in different ways. This paper builds upon Malone and Crowston (1994) who define coordination as the “management of dependencies among activities”. The managers of a process often focus on how to manage activities. However, if a critical dependency of a process is not managed well, the process’s efficiency and effectiveness becomes low even if all activities in the process are performed well. Thus, “how to manage dependencies in a process” has an impact on the efficiency and effectiveness of the process (Hayashi and Herman 2002). In a changing climate, the project needs to attain some stability to build the project organization around it. Critical functional dependencies have proven to provide such stability (Adler 1999). However, all other kind of dependencies (product, deliveries, project, information etc.) need to be identified and managed as well, but the organization and planning doesn’t necessarily need to build upon them. One key element in Malone and Crowston’s approach is that, for each different kind of dependency, it is possible to identify a family of alternative coordination processes that can be used to manage dependencies of that type. Malone & Crowston (1994) also state that there are practical needs which should be solved such as designing tools that enable people to work together more efficiently and find more efficient, flexible and satisfying ways to organize human activity (ibid.).

Coordination is thus a multi-faceted phenomenon. In this paper we will focus on the following aspects, which we claim are imperative for coordinating complex development task in turbulent environments:

- Agility – the ability to manage changes during the project.
- Informality - coordination cannot be confined to a rational planning and controlling process only. Informal, spontaneous coordination of actions “on the spot” are equally important. The gap between the design and use of coordination mechanisms must be narrowed.
- Distribution - Coordination decisions should be taken where there is a need for coordination. Unnecessary propagation of such decisions to the total project management level should be avoided.
- Understanding – there should be a communal understanding among all project members of the goals of the project and how everyone’s contribution fits into the overall project. In particular, a communal understanding of dependencies is important.
- Operational – methods and tools for coordination should be useful in practice.
These needs are somewhat addressed by the Weber, Rambo and Gaia approaches as follows.

**Weber**
In the Weber approach, coordination is perceived as a non-value-adding activity that takes time from the actual development work; additional time spent on coordination is seen as an added cost. The approach strives for independent organizational units specialized on a specific function (Adler 1999).

Larsson (1990) describes the coordination required in Weber projects as “formal”. Formal coordination mechanisms focus primarily on pre-planned coordination. This leads to a large separation in time between the design of coordination mechanisms and the application of these mechanisms, which may be inadequate in situations characterized by uncertainty, instability and complexity. According to Larsson (ibid.), the traditional conceptualization of coordination focuses on concrete material dependencies in terms of work input-output flows. However, there are also abstract dependencies that need to be coordinated, like information flows, perceptions, and meaning relations, which are only implicitly captured in the formal coordination, and thus often left out. Typically the formal coordination mechanisms used in these projects are: project meetings, status reports, sub-system meetings and project plans.

**Rambo**
The coordination need is rather high, but it is controlled by a number of key actors. These key actors spend a large amount of time on coordination. This is a disadvantage according to Malone and Crowston (1994) who state that coordination should be decentralized and not confined to a top-down approach, which is what occurs when applying this approach. The actors are however free to apply a proactive approach to coordination and use earlier project experiences to focus on troublesome integration, which is an advantage compared to the Weber approach (Adler 1999).

The simplified system architecture and integration plan are vital tools in coordinating, visualizing, and monitoring the system progress. These tools imply that the coordination is highly controlled with known dependencies (Adler 1999). The problem with this approach, however, can be that no or poor coordination is performed between the different functional specifications that are created, if not the management team initiate it. Further, integration is not an issue until full stability has been reached at the different subsystems, which can imply late integration. Typically, the coordination mechanisms used in these projects are both formal and informal. Formal processes are: project meetings, status reports, overall pictures (providing a view of the system as whole). Informal coordination processes are for instance meetings helping with integration, meetings to update and make sure every key person involved in the project still have the same view of the project as whole, etc.

**Gaia**
The coordination needs in this approach are extremely high due to the use of project configuration based on building dependencies. The involved actors spend much of their time coordinating their actions (Adler 1999). Intense coordination is considered necessary for project success. Coordination is thus made as often and as early as possible, using a large number of both formal and informal coordination mechanism. The formal mechanisms are typically: project meetings, status reports, system specification work, overall pictures (providing a view of the system as whole), etc. The formal coordination mechanism sets up the rules for the development effort. In this approach, a number of informal coordination mechanisms must be used as well including; system emergency board, temporary forums,
corridor talk, sense-making. In the case of Gaia, formal and informal coordination of actors not involved in the project are needed as well, including system specification work, fault report board, and continuous customer contacts (Adler, 1999)

Larsson (1990) describes informal coordination as taking place between individuals, and thus difficult to take down on paper or plan. Often, a project with a large amount of informal coordination is referred to as a self-coordinating team. The difference between Rambo and Gaia is that the Rambo approach focuses on top-down coordination and leaves out a much of self-coordination by the actors. Self-coordination is useful since it can create well-adjusted on-the-spot coordination in unforeseen situations (i.e. frequent changes in the system specification) where pre-planned coordination fails (Larsson, ibid.). Both approaches however use methods to build dependencies in order to facilitate coordination.

The ICD approach towards coordination
In this section, we will briefly recapitulate the two foundations of the ICD approach: the Anatomy-Based Engineering Process (ABEP) and the Domain Construction Process (DCP). A detailed description of ICD is found in Taxén (2006). The ABEP appeared at Ericsson in the early 1990s as an attempt to address increasing demands on project management (Adler, 1999). The DCP, including the information system support for coordination, emerged around 1997. The ICD approach was applied in full during the development of the 3rd generation of mobile systems at Ericsson between approximately 1999 and 2003 (Taxén, 2003). In a sense, this evolution reflects a gradual move form the Weber type project over Rambo to Gaia.

The Anatomy-based Engineering process
The basic motivation for the ABEP is the need to understand and control critical dependencies when developing a complex system. Such dependencies might reside in the system itself, in the development project, in the market, etc. Experiences from many projects at Ericsson indicate that traditional engineering processes and project management techniques do not address this problem sufficiently.

The ABEP is executed in three steps: anatomy definition, increment planning and integration planning. The purpose of the anatomy definition is to achieve a communal understanding about how the system works. The anatomy is an illustration – preferably on one page – that shows the functional dependencies in the system from start-up to an operational system. Here, the term “functional” should be interpreted as the capability a certain system element provides to other system elements. For example, a power source and a bootstrapping procedure are needed in order to enable the operating system of a computer.

The gist of ABEP is to design and test the system in the same order as the functions are invoked. In a metaphorical sense, this can be seen as the order in which the system “comes alive”, hence the term “anatomy”. In Figure 1, an example of an anatomy is shown. In order to illustrate the principles, an example of building a house is used. The boxes indicate functions and the lines dependencies. The anatomy should be read from the bottom of the figure to the top.
The purpose of the second step: increment planning, is to outline the implementation of the system. The functions are grouped into development and integration steps — increments — in such a way that the functionality of each increment’s addition is verifiable. The intention is to parallelize design and testing as much as possible. In Figure 2, a possible increment plan of building the house is shown.

In the third step: integration planning, the purpose is to divide the work between subprojects and establish a communal understanding over what is delivered, from whom, and when.

Resources are assigned and dates for deliveries of the increments to system integration are
negotiated. For each increment, traditional time and resource plans are made. The dependencies in focus are those between sub-projects.

During the project, the plan is used as an instrument for communicating the progress of the project. The state of each increment is visualized by traffic-light cues such as Green – On Time, Yellow – Warning, Red – Off Track, etc. Impacts of delays are clearly shown, which gives project management the time to take corrective actions. In Figure 3, an integration plan of the house is shown. As can be seen, the integration plan is tilted to signify a time line. Moreover, the internal functions in each increment are subdued.

At a quick glance, the integration plan appears similar to traditional network plans in the sense that it shows the dependencies between tasks. However, the way the integration plan is constructed is very different from the construction of network plan. Moreover, the focus of the integration plan is on expressiveness and ease of use, although the ultimate purpose is to manage complexity. A comparison of various project-planning techniques from this point of view is given in Ekstedt et al. (2003).

The Domain Construction Process
The purpose of the Domain Construction Process (DCP) is to construct the coordination of the ABEP. In stating this, we indicate that coordination is something that needs to be worked out in an organization in order to become operational. The definition of coordination given by Malone & Crowston (1994) emphasizes the dependencies between activities. However, the experiences from Ericsson indicate that this is insufficient. There may be critical dependencies between functions in the system, between sub-projects, between requirements, etc., that must be managed.

In order to elaborate on the definition of Malone & Crowston, the notion of “workpractice” is taken as a point of departure (Taxén, 2003). In the workpractice, actors work on some object such as, for example, a software module, in order to produce an outcome. Coordination can be regarded as a specific type of activity within a workpractice, the purpose of which is to coordinate the actions in the workpractice. When coordination is in focus, only coordination aspects are relevant. In ICD, this perspective of the workpractice is called the coordination domain. The items subject to coordination are the same as those used in workpractice, although seen from a coordination point of view. For example, the text describing a requirement item is relevant when developing a software module. However, in the coordination context, this text is less relevant. Other things like identity, revision, state, attributes, etc. are in focus as well as how requirement items are related to other items like products, customers, test cases, deliveries, etc.
The construction of coordination in DCP has mainly three targets. The first one is a model that describes which items in the workpractise are subject to coordination and how these are characterized and related to each other. This model is called the context model, indicating that it signifies the coordination context in a workpractise. In Figure 4 an example of a context model is shown. The boxes signify items that have been found relevant for coordination. Most of these items have icons associated with them in order to increase the visibility of the model. The lines indicate relationships between the items. In addition, a number of other properties are defined in the model such as attributes, state sets on items and revision control rules (not shown in Figure 4). The item in focus, the anatomy, is shown in the centre of the model (ANATOMY ITEM). The gist of the context model is that it signified the communal understanding at Ericsson in 2002 about what needs to be managed in order to coordinate the anatomy-based engineering process.

The second target is information system support for coordination. This is achieved by implementing the context model in the system. The final target is communal understanding about the context model and its implementation constructed. Taken together, the outcome of the DCP is a socially constructed reality, which aims at coordinating complex system development tasks.

The DCP consists of the mathetic, consolidation and pragmatic phases (see Figure 5). The terms “mathetic” and “pragmatic” are adopted from Halliday (1975), who distinguishes between pragmatic and mathetic functions of language. Pragmatic functions involve the coordination of actions when a communal understanding is established, while mathetic functions concerns the construction of communal understandings.
The purpose of each phase is as follows:

- **Mathetic**: In this phase, the initial construction of the coordination domain is carried out. The main purpose is to achieve a tentative domain structure in terms of a communal understanding and corresponding objectified artefacts such as domain models and their implementation in an information system. The work is carried out in a “daily build” manner by a small “task force”. Provisionary domain models, rules, etc., are suggested and implemented in the information system. The results are discussed and evaluated with respect to usefulness. Changes are suggested to the domain models and implemented anew. This iterative process is continued until a working consensus is achieved. The focus in this phase is on the mathetic function of language.

- **Consolidation**: The purpose of this phase is to boost the trust over the feasibility of the domain as constructed in the mathetic phase. Key issues are getting all actors to trust the data in the information system. This may be done in an on-going development project, that is, a project that develops a product for a customer. The task force is still driving the construction. Additional user roles around the project are involved and immediate, personalized support is provided. The construction of the domain in the consolidation phase progresses by controlled changes. No major reconstruction of the domain is allowed at this stage.

- **Pragmatic**: In this phase, actors in several projects are included in the domain. As in the consolidation phase, the construction is done by controlled changes, however now in a formalized way. The domain may also be expanded to include new types of coordination entities. The focus in this phase is on the pragmatic function of language.

The domain construction process is based on an experiential learning approach (Kolb, 1984). This gist of the process is an ongoing iteration between reflection and action where a communal understanding is gradually established among the actors. In this process, a shift is made from the mathetic to the pragmatic functions of language. Thus, the context model in Figure 4 and its implementation are only the tangible results of the DCP. The intangible communal understanding about them is equally important to achieve. This means, for example, that the construction of the context model must be done in the workpractice where it is useful.
**Some coordination effects from ICD**

The development of the 3rd generation (3G) of mobile systems was a major challenge for Ericsson. The requirements were unclear, the 3G standard was not settled, the project was very large and globally distributed, the technology was at the cutting edge, etc. This situation was expressed by the total project manager as follows in 1999:

“The total technical changes being implemented in this project are enormous. Such changes are needed in order for Ericsson to get a world-leading product first to market. Using traditional methods then the scope of change implemented in single steps will be too large and can not be managed.”

From earlier experiences it was clear that the ABEP was indispensable to use. Moreover, the increased demands on coordinating increments and integrations resulted in a decision to use the DCP. In addition, a specific information system was chosen to support the coordination: the Matrix Product Data Management system (Matrix-One Inc.). Thus, after running projects according to the Rambo project model for several years, the 3G project can be characterized as one of the first Gaia-type project at Ericsson, supported by dedicated information system.

Some of the projects in the 3G development were very demanding. For example, one of the most complex nodes in the 3G system, the so called Mobile Switching Centre node, was developed during very turbulent circumstances by 27 sub-projects distributed over 22 development units in altogether 18 countries. Two main coordination domains were constructed: one in Stockholm, Sweden and one in Aachen, Germany. In this section, we will summarize some coordination effects of the ICD approach at Ericsson (see Figure 6). These effects are all critical in Gaia-type projects:

![Figure 6: Coordination effects from the ICD approach](image)

**Agile planning**

In situations characterized by uncertainty, instability and complexity, coordination mechanisms must be designed for change. In ICD, two types of changes are managed. The first one concerns changes due to customer preferences, development faults, system improvements, etc. Such changes are often expressed as engineering change orders and affect the system to be developed in various ways. In ICD, the impact of this type of changes can be traced to functions in the anatomy, increments in the integration plan and impacts on resource and time planning in the integration plan. Moreover, these impacts are reflected in the information system supporting the coordination. The positive effects of ICD in this respect
are illustrated by the following statement (CR, Change Request, is the Ericsson-internal naming of engineering change orders):

Interviewer (I): Did you also suffer from like changes in the project? Respondent (R): We have a record from R8 \[a pervious release of the 3G system\] who had to deal with 400 CRs on functionality… I: You need some kind of re-planning? R: Yes. And then you need to have a quick process of re-planning, stabilizing again and communicating a clear picture that this is what is going to be on the build \[the SW load module in the system\]….. I: So in some sense the tool and the procedure have been valuable? R: Yes, I think so, very valuable (Project manager, 3G development)

Another type of change is related to the evolution of the coordination domain. This domain may change due to new coordination demands, new knowledge acquired in the domain construction process, etc. This is also supported by the ICD:

“If a trouble report has been found in this block, you have to make sure that in previous versions or later releases, that you correct the same fault. Then, ok how the hell are we going to follow up on this? And then, we entered an extra measure attribute to the block type. We really used that, and yes, that was put in at a later stage. It is implemented within 5 minutes and also rolled out in the same speed almost.” (Methods and tools coordinator, 3G development)

**Informal coordination**
Informal coordination emerges from repetitive interactions between members creating uniform states of mind which crystallize into mores, norms, institutions, etc. This is also supported by the ICD as is evident from the following statement:

“I think that the tool allows us to have a better understanding of impacts of own part of the impacts it could have on other parts and the other way around. That they could see OK, how the other areas influence their design. They could really see that there was a need for coordination and so on. The need for coordination was more or less identified by the tool. They could equally see, OK I’m working on this work package \[the term used for increment in the 3G development\], which are the other work packages involved and how does that relate for them to other work packages and so on. ” (Project manager, 3G development)

**Distributed coordination**
The anatomy and the increment and integration plans can facilitate more distributed coordination by allowing and encouraging actors to coordinate their actions directly without time-consuming propagation of decisions to higher levels. This is further supported by having all coordination items stored in one information system:

“Yes, what is the great benefit is that you have one common place where all the project area stored the information. It means that a lot of the coordination, which previously went via the main project, now can go directly. A lot of coordination is now happening on the level it should be.” (Project manager, 3G development)
Communal understanding of dependencies

The ICD supports the management of all types of dependencies needed for coordination. These dependencies are expressed in the context model and implemented in the information system support. New, abstract constructs and their dependencies will gradually acquire a communal understanding in the DCP if they are found useful in the development practice. For example at Ericsson, the concept of “work package” was coined to signify increments. This was systematically manifested into the context model, implemented in the information system and tried out in actual usage in ongoing projects.

“Before, every role maintained a piece of information it was responsible for. But in the end […] they should build an overall picture and what Matrix enables us to get, this full picture, also to cross the border and see “aha this is information somebody else in another role thinks is connected to this one” that is a complete picture of the overall view and not just the limited view the person is responsible for. That is the main benefit I think.” (Methods and tools coordinator, 3G development)

Operational

The main effect of the ICD approach is that extraordinary complex development projects of the Gaia type can indeed be coordinated by the ABEP and the DCP. This is amply expressed in the following quotation:

“Especially for the execution part I think we would not have been able to run this project without the tool. I think if you simply look at the number of work packages, the number of products that we have delivered, the number of deliveries that we have had, if we would have to maintain that manually, that would have been a sheer disaster.” (Project manager, 3G development)

Other effects from ICD on coordination are reported in Taxén (2003; 2006). In summary, the experiences indicate that the ICD approach is capable of addressing the coordination needs of Gaia-type projects.

Discussion

The essence of the ICD approach is the focus on managing all kinds of dependencies and to arrive at a communal understanding concerning these. As one of the originator of the ABEP stated:

“The most important issue when working with complex things is to work from how things depend on each other.” (Jack Järkvik, around 1990)

The ICD approach has proven its capability of supporting the management of extraordinary complex mega-projects. Thus, Ericsson has moved away from the traditional Weber type of projects, that was the standard approach at the company, to creating a self-organizing project environment of the Gaia type. This approach is now standardized at Ericsson.

The ICD approach implies some disadvantages or risks that must be controlled. The functional focus in the ABEP may result in insufficient attention to performance and other characteristics of the system. More persons from various disciplines need to be involved from the beginning in developing the anatomy and its associated plans. Too much parallelism in
developing and verifying the different increments cannot be managed, and there are examples of increment plans that had to be revised for this reason. Experienced designers may resist to work according to the ABEP. Sometimes they are inclined to start working on the most problematic functions instead of following the inherent “bring alive” order of developing functions in the ABEP. The enhanced visibility, clearly indicating who is responsible for what, is not always appreciated.

The DCP implies that construction of coordination evolves during the project. The positive side of this is that new needs can be implemented more or less on the spot. The drawback is that the implementation in the information system might turn into a patch work design. In one coordination domain at Ericsson, at one point in time, it was necessary to do a complete reconstruction of the implementation. Moreover, the “daily build” way of working increases the stress on the users of information system, since the interface and functionality of the system changes frequently.

The Rambo approach requires ABEP in order to create a communal understanding of the project, and to ensure everyone is focusing on the project as a whole. This engineering approach was introduced at Ericsson in the early 1990s and has been and further developed ever since (Adler 1999, Lilliesköld et al. 2005). It provides the projects with tools, so that changes could be managed and discussed within the project. Moreover, it helps the project managers to keep focused on what is important in the project and promote communication. However, the team around the project manager in the Rambo approach can replace DCP, which however is a necessity in order to create a Gaia based environment in the project.

The main limitation of the ICD approach is that it has, to the best of our knowledge, been in use only at Ericsson. This raises questions about the transferability of the results to other settings. However, the development of telecom systems is a paradigmatic example of the difficulties most complex system developments face today. This would indicate that the ICD approach is indeed transferable. Another issue concerns the information system used in the DCP. Again, the experiences so far come from one particular information system, the Matrix system. It is desirable to identify general criteria by which other information system could be evaluated for use in the ICD approach.

Topics for further research include grounding the ICD approach theoretically. The many successful projects executed at Ericsson according to the ABEP indicate that this process has captured essential aspects of how humans deal with complex coordination problems. For example, Taxén & Lilliesköld (2005) have suggested that the anatomy, the increment plan and the integration plan can be seen as manifestations of fundamental dimensions of human activity coordination. However, this line of research needs to be further developed.

Conclusions
In this paper we have discussed the workpractice approach towards the coordination of mega-projects. The main conclusion is that the approach is capable of coordinating extraordinarily complex development tasks in turbulent environments. The different parts of the Integration Centric Development; the Anatomy Based Engineering Process and the Domain Construction Process are important instruments used to manage and gain control in self-organizing types of projects. The Anatomy Based Engineering Process provides the project with the right focus and the means for communication. The Domain Construction Process provides the projects with the tools to keep track of dependencies, requirements, work-packages, changes, etc., that
are necessary in order to avoid the chaos that can occur in self-organizing projects. Thus, the workpractice approach, and the tools described are useful for system developers who try to create an environment that can cope with constantly changing customer requirements and rapid technological developments.

References
Halliday, M., 1975, Learning How To Mean, London: Edward Arnold.


