The Concept of 'Project': A Proposal for a Unifying Definition

Andreas Munk-Madsen
Dept. of Computer Science, Aalborg University

amm@cs.aau.dk

Abstract. "Project" is a key concept in IS management. The word is frequently used in textbooks and standards. Yet we seldom find a precise definition of the concept. This paper discusses how to define the concept of a project. The proposed definition covers both heavily formalized projects and informally organized, agile projects. Based on the proposed definition popular existing definitions are discussed.

Keywords: Project, definition, project management, software development, agile.

1. Introduction

1.1. Motivation

Many authors give us guidelines for how we should manage projects. Yet they often lack precision as to what they consider a project. When we read a guideline we want to know which phenomena it applies to. We want to know what the author considers to be a project and what is not considered a project. It would also be nice to know whether a guideline applies to a larger class of phenomena than just projects, or whether it only applies to a subclass of project, e.g. IS projects.

From a scientific point of view a precise definition is useful if we want to reproduce the reasoning, the experiments, or the observations that lead to the formulation of the guidelines.

Let us illustrate the problem we want to address with an example. A widespread model for software engineering, CMMI-SW (Carnegie Mellon Software Engineering Institute, 2002), makes heavily use of the word "project". The word is used to denote two of the fundamental process areas, "Project Planning" and "Project Monitoring and Control". CMMI-SW defines a project in this way:

> ... a "project" is a ... set of ... resources ... [that] typically operates according to a plan.

The complete definition is more complex, which in itself is a problem. We will deal with the complete definition later. Here we only focus on a few aspects.

The first problem is the choice of a general concept from which the concept *project* is specialized. *A set of resources* can be almost anything from money on a bank account to food in a refrigerator. A project has little in common with these sets of resources. So the general concept is not well chosen. It is too general.

The second problem is the word *typically*. It signifies that the ensuing feature is true for most, but not all, projects. Therefore this is not a distinguishing feature. We cannot use the feature to determine whether a given phenomenon is a project or not. And given a project, we cannot be sure that it possesses the feature.

The third problem is that the concept of a *plan* is linked to the definition of a project. It exempts CMMI-SW from arguing why a project must a have plan, as this by definition is true in most cases.

This paper will discuss how we can remedy such problems by putting more care into our definition of project.

1.2. Research Methodology

A definition is a part of a theory; actually a fundamental part. Creating and analyzing definitions is a theoretical activity. Thus, it is not possible in an empirical way to "prove" the "correctness" of a definition. Correctness is not an attribute that applies to a definition.

The qualities of a definition are pragmatic:

- a certain conformity to the intuitive informal use of the concept,
- the simplicity and the internal consistency of the definition,
- and the elegance of how the definition helps us structure and present existing knowledge.

The reasons for using a particular definition roughly sum up to "presentation power".

The methodology for creating definitions is not a deductive process where the definition is reached as a final conclusion. It is essentially an interaction between restating proposals for definitions and testing them against relevant parts of the existing body of knowledge. The discussion of other definitions uses a limited form

of textual analysis. Since these definition all claim generality, it is considered a reasonable approach to focus on the actual text of the definitions.

The simplest way to present a proposed definition is to regard it as a hypothesis: "This definition has high presentation power". The hypothesis can then be supported when the definition is used to present central parts of the relevant knowledge. The proof of the hypothesis basically resides with the reader.

That is the way in which the present paper is structured. Section 2 discusses definitions in general. Why should we define our concepts and how can we do this?

Section 3 presents two different definitions of project drawn from the field of organization theory. One definition focuses on the kind of tasks that is solved in a project, the other definition focuses on the way the work in the project is organized. Based on the theory of Mintzberg (1983) it is argued that these two definitions are equivalent.

Section 4 shows how these definitions can be used in presenting our understanding of software projects. It discusses the features that unite and the features that divide two popular schools of thought in IS management, the agile and the heavy methodologies.

Section 5 shows how the insight represented in our definitions can be used when we discuss other definitions of project found in literature.

2. Definitions

A definition is a statement explaining the meaning of a word (Collins Cobuild, 1987). It supports identification and understanding of a phenomenon. This section explains the purpose of definitions in science and discusses how we can construct definitions.

2.1. Why Define?

Must we create definitions? No, in many situations we may do well without precise definitions. Dahlbom and Mathiassen (1993) explain:

> A lot of our knowledge is tacit, unformulated. Our actions are to a great extent based on know-how, rather than on explicitly formulated rules and principles.

They make a distinction between Platonic and Aristotelian concepts:

> A lot of our knowledge is based on Platonic conceptions, on exemplary instances or paradigmatic cases, rather than on Aristotelian concepts, explicit rules and definitions.

And they explain where we need definitions:

> But if we want to develop our knowledge, to question and change our values, we must confront them by trying to make them explicit.

Alter (2000) argues for more precise definitions in the field of IS:

> …the lack of conscious attention to the meanings of basic terms and points of reference may be a significant impediment to effective communication and to our ability to make sense out of research findings and even journalistic anecdotes about what seemed to work or not work in particular situations.

Explicit definitions are important in science. Definitions improve communication and understanding. Precise definitions help us to ensure that we talk about the same phenomena. Precise definitions makes it easier to check that empirical evidence support theoretical theses. Precise definitions also help avoiding circular reasoning where what appears as a thesis is only a redundant restatement of basic assumptions.

Definitions are important prerequisites for the conceptual grounding that is a part of the multi-grounding of design theories proposed by Goldkuhl (2004).

There is a limit to the clarity we can achieve through definitions. We are using natural language to describe phenomena only partially understood. Some of these phenomena belong to the real world and can only be partially formalized. Still, this does not contradict the underlying assumption that some definitions are better than others for supporting identification and understanding.

2.2. The Format of Definitions

We normally define a concept by relating it to other concepts that we assume the reader is familiar with. This can be done is different ways.

We can *decompose* the concept and explain it as an aggregate of other concepts. E.g. “A chair consists of a horizontal plate, called the seat, to which is attached one or more legs…” Usually this kind of definition is hard to understand. It may help us identify or even build a chair, but it does not tell us why we need a chair, and it does not place the chair in any context.

If there is a small number of objects in the class denoted by the concept we may just *specify* them. E.g. “Scandinavia consists of Denmark, Norway, and Sweden”. This is a precise specification, but it does not say anything about the characteristics of the concept.

We may give *examples* of objects or subclasses in the class denoted by the concept, but that would only be a Platonic definition. It would illustrate the concept but not give any explicit explanation.

We can *associate* the concept to other concepts and explain the relations to these concepts. E.g. “Chairs are often used together with tables…" This will provide some understanding of the context of the defined phenomenon. However, this kind of definition may lose sharpness because of the introduction of unnecessary concepts.

The best format for a definition is the classic Aristotelian: *Definitio fit per genus proximum et differentiam specificam.* (Aristotle, 350 B.C.; Smith, 2004). Here we explain the concept by specifying a relevant superclass and some characteristic that distinguish the concept from neighboring classes. E.g. “A chair is a piece of

furniture for one person to sit on." The relevant superclass, *genus proximum*, classifies the concept and the concept inherits the properties of the superclass. Thus in our example the entire "theory" of furniture – including context and theses – now applies to chairs.

The distinguishing characteristic, *differentia*, should ideally tell us the features that only the objects in the concept possess. The choice of the dimension of the differentia is important. In the example with the chair the distinguishing dimension is the *use* of the furniture. We could have chosen another dimension, e.g. construction. This would give us a definition as: "A chair is a piece of furniture with a horizontal plate approximately 45 cm above the floor." The choice of distinguishing dimension in our definition depends on the kind of theory we want to present. Is it a theory of how to use chairs or how to build chairs? Of course we might want a combined theory of how to build useful chairs. In that case we need both definitions, and we must discuss whether they are equivalent.

The differentia should be both necessary and sufficient to distinguish the considered concept. *Sufficient* means that we will not permit irrelevant phenomena into the considered class. Insufficiency is fairly simple to demonstrate as it can be illustrated by an example. The inclusion of more than the *necessary* features in the differentia often involves redundancy. This leads to more subtle complications as it may confuse both argument and presentation. Elimination of redundancy from the differentia is basically an application of the principle of Occam's Razor.

Genus proximum et differentiam specificam is only a guideline for the format of a definition. Using the best format for a definition gives no guarantee that *genus* and *differentia* are well chosen. We still need to evaluate proposed definitions in relation to our other notions of the concept.

3. Projects

Project is a central phenomenon in the field of IS, as systems normally are developed and implemented in projects. Practically everybody who talks about system development methodology will also use the word project. However, as we shall see later, many authors do not give a precise definition of the concept.

In this section we explore two fundamentally different definitions of project and argue for the equivalence of these two definitions.

3.1. Two Definitions

The word project is derived from Latin where "pro" means "forward" and "jacere" means "throw". Thus the original meaning of *project* is something that in a figurative sense has been thrown forward, a proposal. The meaning has gradually

been extended to include the process of realizing the proposal and the people who perform the realization.

As a relevant *genus* for our definition of project we need a word that denotes people working together. For this we could use “organization”. However, some people understand *organization* purely as a legal entity. We want our definition to include *parts* of legal entities as well as people from different legal entities working together. For this reason the *genus* of project is chosen to be *organizational unit.* But colloquially we will use *organization* as a synonym.

We then need to specify the *differentia*, what separates a project from other kinds of organizations. One relevant dimension for the distinguishing characteristic is the kind of tasks solved by the organization. Inspired by Mintzberg (1983) we can suggest the following definition:

Definition 1: A project is an organizational unit that solves a unique and complex task.

By stating that the task is unique we exclude most organizations where task repetition is a prominent feature. This is not the case in IS development. In IS development the task is always unique, at least to the actual developers. If not so, they could solve the task once and just press “copy” for the rest.

The feature of uniqueness entails that the task must be delimited both in scope and time. This delimitation may not be entirely clear in the beginning of the project, and it may change during the course of the project. However, if we experienced a permanent stream of changing tasks we would say that this was no longer a single project.

The task must have some complexity before it belongs in a project. If the task is simple most people will know how to solve it, and the amount of organizational overhead normally associated with a project will not be needed.

We should note that both uniqueness and complexity are relative to the project participants. That somebody on the other side of the earth has great experience in solving the actual task and considers it simple is irrelevant if our participants are not aware of this.

Definition 1 looks at a project from the outside. It focuses on an important situational factor, namely what we use a project for. This raises the questions: What are the internal characteristics of a project? And which principles apply to managing a project? We shall address these questions shortly. But first we will consider another definition.

An important design parameter for an organization is the way in which the people coordinate their work, the prime coordinating mechanism. Mintzberg (1983) lists five different coordinating mechanisms:

- direct supervision,
- standardization of work processes,
- standardization of work outputs,
- standardization of worker skills,
- and mutual adjustment.

All of these coordinating mechanisms are used in all organizations, but in any organization some mechanism is the most important, and this can be used as a defining characteristic. Inspired by Mintzberg's concept of the adhocracy, we can define project the following way:

> Definition 2: A project is an organizational unit where the prime coordinating mechanism is mutual adjustment.

In this definition the *differentia* is an internal feature. This immediately raises the questions: What is the use of such an organization? What kind tasks is this type of organizations suited to solve? The answers to these and the previously raised questions follow when we argue for the equivalence of definitions 1 and 2.

3.2. The Equivalence of the Definitions

We shall argue for the equivalence of definition 1 and 2 in the sense that they in practice describe the same phenomena.

> Thesis 1: An organizational unit that solves a unique and complex task must use mutual adjustment as the prime coordinating mechanism.

The reasoning behind this thesis is that the other coordinating mechanisms cannot do the job. Standardization will be too expensive when we are dealing with a unique task. Direct supervision scales badly so any medium sized or larger task will overload the supervisor.

> Thesis 2: An organizational unit where the prime coordinating mechanism is mutual adjustment should only be used to solve tasks that are unique and complex.

The reasoning behind thesis 2 is that, albeit projects can solve other types of tasks, mutual adjustment compared to other mechanisms is dramatically inefficient for coordinating repetitive work or non-complex tasks.

The reasoning for thesis 1 and 2 depends on the assertion that Mintzberg's list of coordination mechanisms is exhaustive. This is an empirical fact that according to Mintzberg so far holds pretty much true.

Thesis 1 and 2 can be combined to thesis 3. This may be seen as an application of Mintzberg's Extended Configuration Thesis on the domain of project organization.

> Thesis 3: Definition 1 is equivalent to definition 2.

Thus it becomes a matter of perspective which definition of project we choose. From a theoretical viewpoint the internal characteristics will perhaps best represent the essence of a project. From an application perspective the natural choice would be to start with the problem, definition 1, and then use thesis 3 to state that definition 2 is the solution.

4. Agile and Heavy Projects

A problem that has arisen in the last few years is how to explain the agile projects (Beck et al., 2001). They are definitely phenomena that we should call projects as they fulfil both definition 1 and 2 above. However, they fit badly into the CMMI-SW definition, as one of the core values of the agile manifesto explicitly downgrade the concept of plan (Beck et al., 2001):

> We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:…Responding to change over following a plan

It is a relevant exercise to explain, in a simple way, what agile and heavy projects have in common and where they differ. Definitions 1 and 2 is one way of explaining the communality between the two different types of projects. To explain the difference we must look deeper into the distinguishing property *mutual adjustment.*

There is a wide spectrum of ways in which mutual adjustment can take place. This spectrum is reflected in the great variation among different projects. In this section we first describe the different ways in which a project can be coordinated, and we relate some of these differences to the dimension spanned by the frequency with which the mutual adjustment is performed. Secondly we discuss which tasks agile and heavy methods are suited to solve. Finally we mention two other important dimensions that could be used to characterize the difference between agile and heavy projects.

4.1. Discrete or Continuous Adjustment

Mutual adjustment is not a very precise concept. Mintzberg also talks about liaison devices, and identifies the *meeting* as the prime vehicle used to facilitate mutual adjustment. Meetings span a whole range from ad hoc gatherings to the work of task forces and standing committees. Other liaison devices are integrating managers and liaison positions.

Using the ordinary vocabulary of project management we can list a number of liaison devices:

- People filling certain roles: Project manager, steering committee chairman, sponsor, customer representative, etc.
- Groups of people meeting to perform coordination: Project group, steering committee, user group, etc.
- Artifacts documenting agreements in a project: Requirement specification, project plan, product architecture, minutes from steering committee meeting, etc.

The extent to which the various liaison devices are used define a broad range of different ways to manage a project. Clearly the presence of some of these devices can make up for the absence of others. Thus it is problematic to focus on one of the devices, the project plan, and to include it in a definition of the concept of project.

"To plan or not to plan" seems to be a major distinction between heavy and agile projects. Boehm and Turner (2004) call the traditional methods or approaches, that are not agile, for "plan-driven". Abstracting a little further, we can see this distinction as a preference in liaison devices. We can also describe the distinction as a difference in the frequency of the mutual adjustment. This leads to suggesting the following definitions:

> Definition 3: An agile project is an organizational unit where the prime coordinating mechanism is continuous mutual adjustment.

> Definition 4: An heavy project is an organizational unit where the prime coordinating mechanism is discrete mutual adjustment.

4.2. Complexity and Ideology

When we tighten the *differentia* from definition 2 to definitions 3 and 4 we reduce the number of phenomena that fit the definitions. This leads to the question of what the corresponding restriction on definition 1 should be. This is the question of what kind of tasks agile and heavy projects respectively can be used to solve. Obviously the *differentia* to examine is *complexity.* Beck (2000) gives us a clue in the chapter where he discusses when you shouldn't try XP:

> Size clearly matters. You probably couldn't run an XP project with a hundred programmers. Nor fifty. Nor twenty, probably. Ten is definitely doable.

Highsmith (2004) is not happy with this restriction:

> One myth about agile approaches goes something like this: "APM (or pick any agile methodology) works well for smaller projects, but it doesn't scale to larger ones." [APM is Highsmith's abbreviation for Agile Project Management.]

Therefore Highsmith proposes a number of techniques to facilitate scaling. One of these is a "Commitment-Accountability Protocol Card". It describes

- an outcome,
- acceptance criteria,
- supplier team,
- consumer team(s),
- intermediate deliverables,
- and estimated work effort.

This is clearly a written documentation of an agreement. Once it is produced we would only expect it to be changed at discrete intervals. If we added a deadline this would be a reinvention of a project plan, albeit a decentralized one. So Highsmith has not contradicted Beck. He is proposing to scale the agile methods by including a key element from the heavy methods.

The number of developers is not a sufficient *differentia* when we wish to determine the kind of tasks where agile and heavy projects are useful. Boehm and

Turner (2004) proposes 5 dimensions to describe the situational factors distinguishing agile from heavy projects:

> As a "summary of summaries," we have concluded that there are five critical factors involved in determining the relative suitability of agile or plan-driven methods in a particular project situation. These factors ... are the project's size, criticality, dynamism, personnel, and culture factors.

Some of these factors may be abstracted into the *differentia* of *complexity*. But the readiness of the IS people and the surroundings to accept agile or heavy methods clearly matters. This is what Boehm and Turner call *culture*. We might also talk about *ideology*. So we may conclude that there is not a simple extension of definition 1 that can define agile and heavy projects based on the difference of tasks.

4.3. Technology Cost and Strategy

Two other dimension that in general should be involved in characterizing IS projects are *strategy* and *technology*, in particular the costs of using various technologies. Indeed these are the defining distinctions for Highsmith (2004):

> When we reduce the cost of experimentation enough, the entire economics of how we do product development changes - it switches from a process based on anticipation (define, design, and build) to one based on adaption (envision, explore, and adapt).

The available technology, in this case the technology for experimentation, and we could add the technology for rework, is a major characteristic of the task. The strategy, in this case anticipation or adaption, is a major internal feature of a project.

5. Definitions in Literature

In this section we will discuss various definitions of project found in literature. We take a look at some dictionary definitions, a textbook definition, three definitions from management and IS standards, and a definition from general project management theory. The overall impression is that - although it is hard to find two identical definitions - all definitions revolve around a common center, and that this not too far from the definitions in this paper.

5.1. Dictionary Definitions

Collins Cobuild (1987) defines project this way:

> A **project** is **1.1.** an idea or plan that you intend to carry out in the future or that is being carried out at present. **1.2.** a detailed study of a particular subject.

Webster (1989) defines project this way:

> **project. 1.** something that is contemplated, devised, or planned; plan; scheme. **2.** a large or major undertaking, esp. one involving considerable money, personnel, and equipment. **3.** a

> specific task of investigation, esp. in scolarship. **4.** *Educ.* an educational assignment necessitating personal initiative on the part of a student.

In most of the definitions there is an absence of a proper *genus proximum*. Some of the definitions indicate complexity as a distinguishing characteristic. It is hard to find uniqueness as a property. These definitions are probably typical for the popular perception of the concept of a project. In their vagueness they are not incorrect, but they are not a sound basis for building a theory about projects. It is difficult to understand that many authors of textbooks on projects do not take the effort to discuss their own definition.

5.2. A Textbook Definition

Many textbooks and standards make heavily use of the word *project* without defining the concept explicitly. Among them are Highsmith (2004), McConnell (1998), Briner et al. (1996), and Page-Jones (1985). Two authors that do define project are Weiss and Wysocki (1992):

> A *project* is defined as having the following characteristics:
>
> - Complex and numerous activities
> - Unique - a one-time set of events
> - Finite - with a begin and end date
> - Limited resources and budget
> - Many people involved, usually across several functional areas in the organizations
> - Sequenced activities
> - Goal-oriented
> - End product or service must result

If there is any priority in this sequence we will notice that the first two characteristics are the same as in our definition 1. This definition illustrates that authors of textbooks cannot depend on the popular definitions. This definition is much more narrow and precise than the dictionary definitions.

There is an abundance of characteristics in this definition. Some of them could be derived from the others. That would reduce the redundancy in the definition.

The *genus* is not stated explicitly, but the following reveals that it is *task:*

> ... it is evident that a task becomes a project when the above factors begin to dominate ...

This is a typical way of using the words *task* and *project*. It makes it difficult to distinguish between the task and the organization set up to solve it. And we need to do that when we talk project management.

5.3. A Standard Definition: PMBOK

"A Guide to the Project Management Body of Knowledge" (Project Management Institute, 2000) has the following definition:

> …a project is a temporary endeavor undertaken to create a unique product or service.

Here the *genus* is *endeavor*. This facilitates the distinction between the *task* and the *process* of solving the task. However, the identity of the people who perform this process is weakened by this definition.

Uniqueness is a distinguishing characteristic along with the *time limitation.* However, the *uniqueness* is associated to the result and not the task. This is too narrow a definition. Reproducing an existing product under quite different circumstances could be a very challenging task that would justify a project.

Complexity is absent from the definition. This makes it too broad.

5.4. A Standard Definition: CMMI-SW

"Project" is a central concept in the Capability Maturity Model Integration for Software Engineering (CMMI-SW) (Carnegie Mellon Software Engineering Institute, 2002). Three key process areas carry the word "project" in their names: "Project planning", "Project monitoring and control", and "Integrated project management". In CMMI-SW we find the following definition:

> "In CMMI models, a “project” is a managed set of interrelated resources that delivers one or more products to a customer or end user. This set of resources has a definite beginning and end and typically operates according to a plan. Such a plan is frequently documented and specifies the product to be delivered or implemented, the resources and funds used, the work to be done, and a schedule for doing the work. A project can be composed of projects."

"A ...set of...resources" could be interpreted as a *genus*, but not as a *genus proximum*. It is not a close superclass. "Organization" is defined in CMMI-SW. However this concept is defined as an aggregate of projects, so it cannot be used as a superclass of "project":

> "An organization is typically an administrative structure in which people collectively manage one or more projects as a whole,..."

The CMMI-SW definition of "project" is in reality a definition by decomposition. The concept of a "project" is defined as an aggregation constructed from mainly "resources", "products", and one "customer". Still the definition could be correct, albeit hard to understand. But the distinguishing quality of "complexity" is missing, which makes the definition too broad. A newsboy who temporarily delivers a paper to a summer address is also a project according to this definition.

On other aspects CMMI-SW's definition is too narrow. There is only one customer or end user. The situation where two or more users disagree is thus excluded. And the software developers are abstracted into "resources". As a consequence we should not expect to see politics or motivation as key process areas in the CMMI-SW.

Half the text in CMMI-SW's definition of a "project" is used to define the concept of a "(project) plan". Again definition by decomposition is used. A plan is an aggregation of specifications of "products", "resources", "work", and a "schedule". Without *genus* the essence of the "plan" is lost. Is it a unilateral

directive or a multilateral agreement? Humphrey (1997) made a very strong case that a project plan must be a negotiated agreement in order to sustain commitment. That thesis is not supported very well by the CMMI-SW definition of *plan.*

There is a reservation in the word "typically", but we are not told what happens in the non-typical situations. Apart from that, the definition implies that a project must have a plan. This is not just a narrowing of the definition. It is an inclusion of a non-trivial thesis. The necessity of the key process area "Project Planning" does not need to be proved anymore. That is a pity. The proof could provide us with conditions for when a project must be planned, and arguments for why a project must be planned.

5.5. A Standard Definition: Sysperanto

Sysperanto (Alter, 2005) is an attempt to define core concepts of the IS field. It is denoted an ontology. It defines project this way:

> A project is a work system designed to go out of existence after producing a particular product.

Work system is the central concept is this ontology. It is defined this way:

> **Work system.** A view of work as occurring through a purposeful system.

Work is defined this way:

> **Work.** Effort applied to accomplish something within an organization or across organizations.

Work systems are aggregates of nine elements:

- work practices,
- participants,
- information,
- technologies,
- customers,
- products & services,
- environment,
- infrastructure,
- and strategies.

In this definition we have approximately the same *genus* as in our definitions. Alter reserves the word *organization* for an aggregation of *work systems*. The defining characteristics for a project, *a particular product* and *time limitation*, are close to our *differentia, uniqueness.*

But *complexity* is missing from the definition. Without this characteristic *work systems* solving trivially simple tasks are including in the class of *project.*

It is interesting to note that *technology* and *strategy* are among the nine types of components in a *work system* and hence in a *project.*

5.6. The ‘Temporary Organization’ School

In the field of management theory a contemporary school views projects as primarily *temporary organizations* (Lundin and Söderholm, 1995). We immediately notice the agreement on the *genus.* As *differentia, temporary* is clearly necessary, but in itself insufficient.

In definitions from this school we sometimes miss Mintzberg’s clear distinction between situational factors and design parameters. One definition is given by Turner and Müller (2003):

> A project is a temporary organization to which resources are assigned to undertake a unique, novel and transient endeavour managing the inherent uncertainty and need for integration in order to deliver beneficial objectives of change.

If we accept *novel* and *complex* as overlapping concepts, then the first half of this definition alligns pretty much with our definition 1. The second half of the definition points to *uncertainty, integration* and *change* as elements of a project. In their paper Turner and Müller argues that these elements are necessary consequences of the features mentioned in the first half of the definition. That is why the latter elements should be excluded from the definition and placed in a subsequent thesis.

6. Summary

It has been proposed to define a *project* as *an organizational unit that solves a unique and complex task.* It has been demonstrated that this definition can embrace the traditional, heavy project management methodologies and the extreme, agile methodologies. It has been illustrated that the equivalent definition of a *project* as *an organizational unit where the prime coordinating mechanism is mutual adjustment* in a simple way can be extended to explain the difference between heavy and agile project management methodologies. The idea is to use the frequency of the mutual adjustment as a distinguishing characteristic between agile and heavy projects.

It has been illustrated that some existing standards and textbooks have a concept of project that revolve around a common center, and that this not too far from the definitions in this paper. The proposed definition has been used to identify shortcomings of some of the existing definitions.

An obvious extension of this paper – if space had allowed it – would be an examination of the distinguishing characteristics between different IS projects. This would require more dimensions than those discussed in this paper. Foremost we would need a concept of the technology involved to characterize the project’s task, and we would need a concept of the project strategy to characterize the management method.

References

Alter, S. (2000). “Same Words, Different Meanings: Are Basic IS/IT Concepts Our Self-Imposed Tower of Babel?”, *Communications of the Association for Information Systems*, vol. 3, art. 10, 1-89.

Alter, S. (2005). “Architecture of Sysperanto: A Model-Based Ontology of the IS Field,” *Communications of the Association for Information Systems*, vol. 15, 1-40.

Aristotle (350 B.C.). *Organon, Topics*. Translated by W.A. Pickard-Cambridge, URL= http://classic.mit.edu/Aristotle/topics.html

Beck, K. (2000). *Extreme Programming Explained.* Addison-Wesley, Boston, MA.

Beck, K. et al. (2001). *The Agile Manifesto.* URL = http://agilemanifesto.org/

Boehm, B. and Turner, R. (2004). *Balancing Agility and Discipline.* Addison-Wesley, Boston, MA.

Briner, W., Hastings, C., and Geddes, M. (1996). *Project Leadership.* Gower, Aldershot, England.

Carnegie Mellon Software Engineering Institute (2002). *CMMI for Software Engineering (CMMI-SW, V1.1)*, Staged Representation, URL = http://www.sei.cmu.edu/cmmi/models/model-components-word.html

Collins Cobuild (1987). *English Language Dictionary.* Collins, London.

Dahlbom, B. and Mathiassen, L. (1993). *Computers in Context.* NCC Blackwell, Cambridge, MA.

Goldkuhl, G. (2004). “Design Theories in Information Systems – A Need for Multi-Grounding,” *Journal of Information Technology Theory and Application*, 6:2, 59-72.

Highsmith, J. (2004). *Agile Project Management.* Addison-Wesley, Boston, MA.

Humphrey, W.S. (1997). *Managing Technical People.* Addison-Wesley, Reading, MA.

Lundin R.A. and Söderholm A. (1995). “A Theory of the Temporary Organization,” *Scandinavian Journal of Management*, vol. 11 (4), 437-455.

McConnell, S. (1998). *Software Project Survival Guide.* Microsoft Press, Redmond, WA.

Mintzberg, H. (1983). *Structure in Fives: Designing Effective Organizations.* Prentice-Hall, Englewood Cliffs, NJ.

Page-Jones, M. (1985). *Practical Project Management.* Dorset House Publ., New York.

Project Management Institute (2000). *A Guide to the Project Management Body of Knowledge.* Newton Square, PA.

Smith, R. (2004). "Aristotle's Logic", *The Stanford Encyclopedia of Philosophy* (Fall 2004 Edition), Edward N. Zalta (ed.), URL= <http://plato.stanford.edu/archives/fall2004/entries/aristotle-logic/>.

Turner J.R. and Müller R. (2003). “On the Nature of a Project as a Temporary Organization”, *International Journal of Project Management*, vol. 21, 1-8.

Webster (1989). *Webster's Encyclopedic Unabridged Dictionary of the English Language.* Gramercy Books, New York.

Weiss J.W. and Wysocki, R.K. (1992). *5-Phase Project Management.* Addison-Wesley, Reading, MA.

= = = = = end of text = = = =