2006

When to use what? – Selecting systems development method in a Bank

Jan Pries-Heje
The IT University of Copenhagen, jph@itu.dk

When to use what? – Selecting systems development method in a Bank

Jan Pries-Heje, Ph.D.
The IT University of Copenhagen, Denmark &
The IT University of Gothenburg, Sweden
Currently visiting at Curtin University of Technology,
Perth, Western Australia
Email: jph@itu.dk

Abstract

Since the very first systems development method was defined it has been discussed how and when to select which method(s). Several books and consultants have claimed to have found the philosophers stone, but never the less it seems that there is no single method that will ever work for (nearly) all development situations. The question then arises: When to use what? And how can one help a concrete organization decide on systems development method(s)? To answer this question we undertook an action research study in ScandiBank that employs 1700 IS developers. The action research took place in three learning cycles. The first cycle started in 2001, the latter ended in 2005. We report from the three learning cycles emphasizing and explaining the learning that took place in each cycle. The result – our answer to the research question on when to use what? – is a framework focusing on the final product, with a few well-chosen trails through the "maze" of possibilities, and some rules for election of method parts.

Keywords

Information systems development, method, methodology, action research, contingency model

INTRODUCTION

Information systems development (ISD) regards systems development processes and products. Systems development typically unfolds in a series of stages such as analysis, design, coding and testing. The stages do not have to be carried out sequentially but can be done more or less in parallel. Often each stage operates with a defined notation and will often result in a prescribed artefact, such as a requirements specification or a computer program.

An ISD methodology is a prescribed way of carrying out the development. The description typically includes activities to be performed; artefacts resulting form the activities; plus some principles for organizing the activities and attaching people to perform the activities. An ISD methodology can be aimed at a specific type of development, e.g. database-intensive applications with less than 10 people involved, or it can be specific to a company. However, many ISD methodologies claim to be of generic use.

Early ISD methodologies were based on practical experiences, i.e. when experienced practitioners simply described how one could develop. Newer ISD methodologies are often anchored in theory such as for example Andersen et al (1986, p. 47-60) that builds on a theory that says that ISD consists of nine distinctly different activities, or Highsmith (1999) that (partly) builds on the theory of lean thinking (Womack 1996).

Over the years ISD methodology has been a concept that has attracted a lot of discussion. One of the main topics has been the distinction between method and methodology. Originally the "-ology" originally implied a study of methods. But gradually, the distinction between "methodology" and its underlying term, "method" has become blurred by common usage. In this paper we have decided to follow the distinction that Checkland (1981) proposed; so a methodology is a set of principles of method, which in any particular situation has to be reduced to a method uniquely suited to that particular situation.

The important words in this definition are *reduced* and *particular situation*. Because the real difficult challenge lies in how this reduction to practice is done? In many situations where an ISD method was claimed to be used it could not be proven by researchers (cf. Bødker and Bansler (1993). Furthermore the practicality of ISD methods has been questioned altogether. A growing number of studies have suggested that the relationship of methodologies to the practice of information systems development is altogether tenuous (Fitzgerald, 1997, 1998, 2000) (Wynekoop & Russo, 1997). It seems that methodology has been so dominant in our thinking about systems development that it may have become somewhat imaginary: a self-confirming hypothesis. For example, one alternative viewpoint situates systems development as "amethodical:" the management and orchestration of systems development without the predefined sequence, control, rationality, or claims to universality implied by much of methodological thinking (Truex et al. 2000).

The conclusion of this line of thought, however, is intolerable. If any ISD methodology only has academic but no practical value then any ISD project is left on their own. It seems plausible to assume that we can describe something of some use for an ISD project, so the harsh critique of ISD methodology thinking may be a matter of not finding a useful way of *reducing* a methodology to a method for a *particular situation*.

So to answer this research question we set out to develop a framework. We did it in one specific organization – ScandiBank (pseudonym; for anonymity) – and we did it over several years (2001-2005) as an action research undertaking. We believe we have an answer to the question, in the form of a framework focusing on the final product, with a few well-chosen trails through the "maze" of possibilities, and some rules for election of method parts.

Below we will first discuss our research method and introduce our case organization ScandiBank. Then we give an overview of the three action learning cycles we have been through, and summarise the details of each learning cycle ending up with the framework that is now being used throughout ScandiBank for selecting (or reducing methodology to) method(s) for particular situations. Finally we conclude and discuss future research.

RESEARCH QUESTION AND METHOD

Several books and consultants have claimed to have found the philosophers stone, but never the less it seems that there is no single method that will ever work for (nearly) all development situations. The question then arises: When to use what? And how can one help a concrete organization decide on systems development method(s)?

In 2001 an interview study unveiled that the systems development methodology in ScandiBank was not being used in a satisfactory way. Some projects only used parts while other projects didn't use the development methodology at all. These observations were consistent with what has been called amethodical development where ISD is carried out without a predefined sequence, control, rationality, or claims to universality. Such development activities are unique and unpredictable for each information systems requirement such that even contingent methods fail (Truex et al. 2000)). Amethodical development has been compared to "growing" information systems as an alternative to "building" systems (Truex et al. 1999).

However, we were not willing to accept that nothing could be done. We thoroughly believe that it is possible to have organizational learning such that one project can learn from another projects success and failure. Thus we decided to undertake an action research study to cope with the problem identified - lack of use of the company systems development method.

Bob Galliers (1992) describes action research as an approach that allows us to create new theoretical knowledge in addition to something that has practical value for the organization under research. The approach that we adopted in our action research is based on the five phases recommended by Susman & Evered (1995): (1) Specification of infrastructure in project. (2) Diagnosis of problem. (3) Planning of actions. (4) Implementing actions. (5) Evaluation of results. Repeat phase (2) to (5), if necessary.
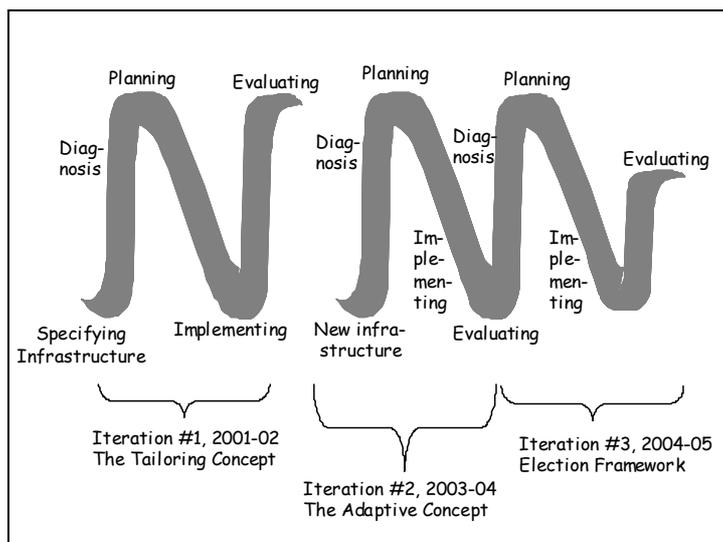

Figure 1: The three action research cycles in our project

In figure 1 we have shown the action research process used in this paper. Our first action research cycle took place in 2001-2002. Here we developed a "radar" where one could characterize a project on eight dimensions. However, it turned out to be extremely difficult to establish a link from the contingent "radar profile" to solid

recommendations for method use. The radar was useful for describing a particular situation but it did not work well in reducing the ScandiBank methodology to a method for a particular situation

After a period where nothing happened the project was revitalized in 2004. We now took our theoretical starting point in the thinking about patterns and families of patterns. However, the resulting framework became too complex, and we returned to diagnosis again.

Finally in 2004 we succeeded in developing a simple framework that a concrete project can use to elect method(s) from the ScandiBank methodology. The simplicity stemming from a few well-chosen "trails" that a project can take through the "maze" of possibilities.

Below we have used the action research phases to reflect on and describe our learning from the action research undertaking. Furthermore we have tried to give an account of the results in a way useful to other organizations facing the same problem.

THE SCANDIBANK CASE

The ScandiBank group is a financial institution that provides all types of financial services such as banking, mortgaging and insurance in northern Europe. ScandiBank employs 17.000 employees and has more than 3 million private customers in Scandinavia. As part of the ScandiBank Group there is an IT department with 1700 employees located at four development centres; three in Denmark and one in Ireland.

ISD projects at ScandiBank vary widely in size; most are small and short-term, but there are also some major projects that have strategic implications for the entire ScandiBank group. Project teams of three to five people typically handle the smaller projects, which usually take from six to 12 months. Large projects, such as the Year 2000 compliance project, typically involve up to 150 people and last from 6 months to 3 years.

The four development centres are headed by a senior vice president. Each individual division is led by a vice president and organized into departments, with typically 20 to 50 people divided among five or so projects. Project managers oversee regular projects, while the vice president manages high-profile projects. IS developers in ScandiBank typically have a bachelor's degree in either an IT-related field or banking, insurance or real estate.

FIRST ACTION RESEARCH CYCLE: THE TAILORING CONCEPT

The infrastructure for the tailoring concept project was first established in the fall of 2001. A project group was established with five people from ScandiBank and a researcher from outside. Three of the six had worked together prior to this project in another quite successful action research undertaking (Reference to authors work removed for anonymity). This fact influenced the choice of an action research approach for this project.

Diagnosis of problem

Early in 2001 it was questioned inside ScandiBank whether the existing company ISD methodology was good and useful enough. An interview study was undertaken among IS project managers within ScandiBank. This study revealed that the existing company methodology was very hard to tailor to the specific needs in the project. The assumption that one size fits all projects had failed. Furthermore the projects demanded better help and tools for the tailoring process. Thus the tailoring concept project started.

Planning of actions

The first idea – or theory – in the tailoring project was to find a number of characteristics of different projects, and then use these characteristics to identify a subset of methods from the company methodology. We studied existing literature and identified eight dimensions that could be used to characterize a project. We decided to use a 1 to 5 scale to score each dimension. When a concrete project was scored, and the *project profile* thereby identified, we then wanted to have a set of guidelines on how to tailor the company methodology to the specific projects. In figure 2 we have shown the eight dimensions with an example scoring. We called this project profile the *radar*.

In the middle of the radar is the "sweet spot" meaning the characteristics that make the project as easy as possible. For example for the dimension called Team the scoring "1" would be given for a project carried out in a small group, with group member sharing the same background, have worked together before, and with a perfect mix of personality types and temperaments in the group. Likewise the scoring "5" on the Team dimension would be given for a large group; having different background (education and/or experience), an unbalanced mix of personality types and temperaments, and where the team members don´t know each other.
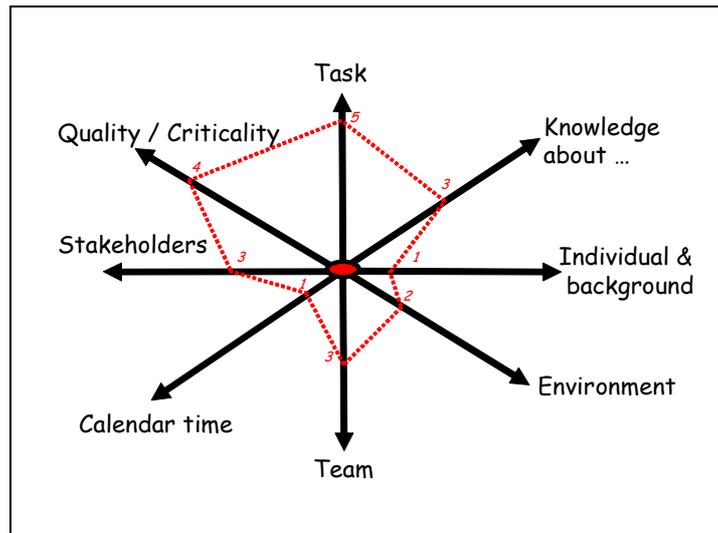
Figure 2: The "radar" developed to characterize a project along 8 dimensions

The radar gave a good insight into what kind of IS project one was facing. However, in practice it was very difficult for us to establish the causal relationship between the "radar picture" and recommendations for both the IS management process and the IS development process. In the literature we could find simple causal relationships, like "IF you have sparse calendar time AND on-time delivery is important THEN use time boxing" (inspired by McConnell 1996). However, we could not find complex relationships like "IF Team + Task is High AND Knowledge about is low THEN do this and that".

Thus we went looking for a simpler way to characterize projects and found a framework developed by Mathiassen and Stage (1990) that simply had two dimensions. The two dimensions are complexity and uncertainty. The degree of complexity represents the amount of relevant information that is available in a given situation. In contrast, the degree of uncertainty represents the availability and reliability of information that is relevant in a given situation. Complexity can be measured on a 2-point scale from simple to complex. Likewise uncertainty can be measured on a 2-point scale from stable to dynamic.

Using the resulting 2-by-2 matrix we succeeded in establishing the relationship between project characteristics and recommendations (Mathiassen & Stage, 1990). We have shown this in figure 3.
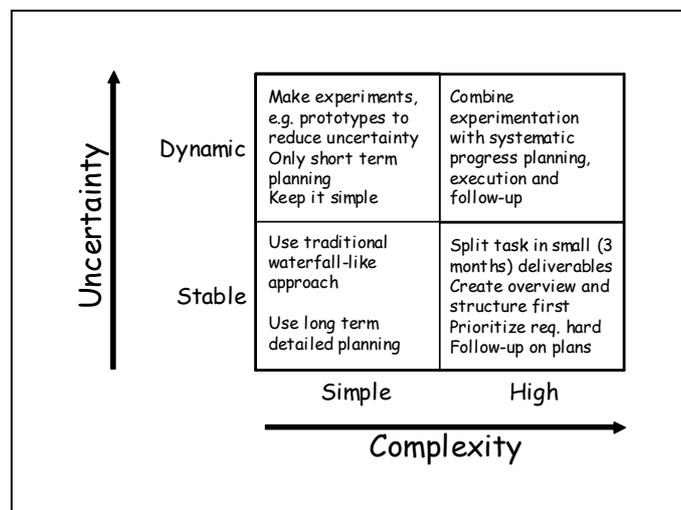

Figure 3: The simple 2-by-2 matrix we ended up with for tailoring

Implementing actions.

While developing the theory we also tested the concept with projects in ScandiBank. In the concrete we did three pilot tests each lasting 4-6 hours and carried out as a kind of facilitated workshop (with the author of this paper as one of the facilitators).

After each pilot test workshop we asked the participants to evaluate their subjective satisfaction. On a scale from 1 to 5 we had an average of 4.11 with 5 being the best score. So we know that the three projects felt that the tailoring concept as we have presented it above was valuable and useful.

Evaluation of results.

Even though the projects were happy we could not convince management in the organization that the concept was useful. At the same time ScandiBank implemented an organizational restructuring and the tailoring project was stopped.

Our learning from this first phase of the action research undertaking can be summed up in two lessons:

Lesson #1 – Tailoring the IS (project) management process is easier than tailoring the IS development process

We define the management process as everything project management related like estimation, planning, phases, milestones, follow up, and iterations. Likewise we define the development process as the actual building of a product, including requirements, design artefacts such as diagrams, charts, drawings and other formalisms, programming and testing.

Tailoring the development process is important many say. But when they exemplify this statement of importance they typically only talk about tailoring the management process. If we look at the development process then Cockburn (2000) and Rational Unified Process mentions that adaptation of the methodology or model is important but they don't give concrete advice on how it is done. CMM (CMM 1.1, 1995) also have tailoring as one of the key process areas, and they mention some practices to use when doing it. But none of the practices are development process practices as we define it.

A possible reason why it is more difficult to tailor the development process may be that it requires both domain specific knowledge and IS specific knowledge for example on object orientation. And in many industries managers don't have both kinds of knowledge. And if they have it is typically outdated in these times of Java, .NET, Components, and Web Services. If we on the other hand look at the management process then a plan, a stakeholder analysis, a risk analysis an many other artefacts look exactly as they did 10 years ago, and they look the same for an IS project as for any other kind of project.

Lesson #2 – A project can be characterized by the degree of uncertainty and complexity it is facing, but how that characterization is used in practice is not covered well in the literature

By trying the framework presented by Mathiassen and Stage (1990) we found that it could be used in that concrete projects (in our three pilot tests) could understand the dimensions and characterize themselves. Mathiassen and Stage (1990) gives some advice on what to do to cope with respectively uncertainty and complexity, but in practice we found that the advice given was insufficient to cover the need for tailoring expressed by the projects.

SECOND ACTION RESEARCH CYCLE: THE ADAPTIVE CONCEPT

A little more than a year after the first iteration three of the six original action researchers were put together in a project again. We decided to build on the learning from the first action research cycle. This was an easy decision to make in that the organization at this point in time (second half of 2003) had decided to emphasise the separation of the management from development activities and were developing a new company methodology in accordance with this decision. Furthermore it had been decided to implement a way of working along the lines of Service Oriented Architecture - SOA (Datz 2004; Erl 2004) and Computing (Papazoglou & Georgakopoulos 2003).

Diagnosis of Problem.

In 2002 and 2003 the Method Department in ScandiBank worked hard to implement the company decision on using SOA in the Methodology, and put emphasise on splitting management from development activities. A management process with a fixed number of check points was established. At each check point Management in ScandiBank can make a go or no go decision.

It was decided that the new company methodology should be a generic model meaning that it should be useful in any kind of IS project in ScandiBank, but that the projects should be capable of adapting their project process from the general company methodology. Furthermore the company methodology should cover the whole project meaning both IT and business oriented activities. As it happened the new methodology ended up being a huge model including numerous methods. It wasn't long before projects in ScandiBank indicated that they needed help in adapting the general methodology to their projects. And so we set out to develop an *adaptive concept*.

Planning of Actions.

The reorganization of the IS organization in ScandiBank that came to signal the end of our action research learning cycle 1 established a number of Competence Centres. There was for example a centre for Architecture and Design, another for Business Processes, a third for Testing and so on. At one point there was more than 20 Centres, a number that was quickly reduced, however.

Each Competence Centre was asked to formulate methods within their area of competence. Methods were described as activities that should be carried out under specific circumstances. It could for example say "If the project is large in size or scope, complex or of strategic importance to the organisation the Method A should be carried out in the project".

We decided to take all the rules from all the Competence Centres, put them together to see whether we could find some patterns or families of patterns across them. After a lot of hard work we ended up with a huge decisiontree including 205 methods and nearly as many questions.

Implementing Actions.

We never reached the point where we could pilot test the decision tree. It was obvious that it was too huge. More than 100 questions to reveal which of more than 200 methods to use in the project could not exactly be seen as a help to a project. It was obvious to us and to the sponsors of our project in ScandiBank that another perspective was needed.

Evaluation of Results

Our learning from this second phase of the action research undertaking can be summed up in two lessons:

Lesson #3 – The idea of having a number of methods grouped into a pattern seemed a promising idea.

A pattern is an idea that has demonstrated its value in practice. Unfortunately we did not have an opportunity to study practice in projects – not in this learning cycle. We had to assume that the Competence Centres had a good grasp of practice, and building on that assumption we created the huge decision tree where each branch on the tree could be seen as a family of patterns. However, an interview study with nine project managers carried out in this action learning cycle showed that the use of the company methodology was so diverse at this early point that it probably would be hopeless to identify good practice or patterns that would be of use.

Lesson #4 – We aimed at identifying some commonalities that could be use to group the decision on when to use what method – but it ended up being to complex.

With more than 100 questions as our result of trying to identify commonalities among the rules for when-to-use that each of the Competence Centres had formulated we had to realize that we had reached a level of complexity that was too high for anything useful and helpful for the projects.

THIRD ACTION RESEARCH CYCLE: THE ELECTION FRAMEWORK

The third action research learning cycle naturally followed the second cycle. No changes to the project infrastructure.

Diagnosis of Problem

At this point we needed a new approach. Our "bottom up" approach where we tried to group the rules for when-to-use by commonalities had failed. We were looking into a maze of possible choices of 100s of methods all included in the company methodology.

Planning of Actions

At this point we came up with idea of "cutting" through the problem by defining a few well chose trails through the "maze". We defined a *project trail* as a logical subset of a project; a subset that results in a product to the target group. The target group can either be users within or outside the IS-organisation in ScandiBank. Thus project trails focus on the product that has to be developed by the project. We defined seven project trails:

1. No IT – Creation or change of product with no IT content

2. Maintenance – Correction or change of existing IS

3. Further development – Enlargement of a existing IS with new processes, functionality or data

4. New development – Development of completely new IS, even though it substitutes existing systems

5. Company group-wide business infrastructure – meaning something that other projects can build upon

6. Software technology – Roll out of, development of, or change of existing software

7. Hard- and middleware technology – Purchase and customization of IT systems (not software)

The project trails 2, 3 and 4 can all be seen as development of commercial IS, it is just the scale that varies.

A project may need to develop more than one thing (product) in parallel to achieve the goal; each of the things can belong to a specific trail. So one project can have more than one trail.

On top of the trails we then developed a small set of questions for each trail that could be used for electing methods. Thus the trail decides a subset of methods from the company methodology, and on top of that a few well chosen questions elect or reject a few more methods.

To make it easy for projects to make the decision we build the election questions into a spreadsheet. When a project then answers a question the relevant methods to elect are coloured grey.

In developing the questions we came up with a number of principles:

- Any question should only be asked one time, and should lead to an indication for all methods implied by the question

- Any method should only be placed once in the spreadsheet. Thus a grey colouring indicating election is unique.

- It is ok that several questions can lead to colouring of a specific method.

It was not a trivial task to decide which trail a project belongs to. So we made a decision tree with 4 questions, relatively easy to answer. Again, the questions are focusing on the task the project is to solve, and the point is, that the project does not have to know the definition of the trails to answer the questions. By answering those 4 questions, the project at the same time identifies all the trails they have to follow.

To ensure that any question only should be answered once, we made a grouping of all the methods along five dimensions: (1) Business, (2) Process, (3) Function, (4) Data, and (5) Technology.

To each dimension we then formulated questions. This turned out to be a challenging process because it was difficult to find words for the questions with the right and precise meaning.

In figure 5 is shown an adapted excerpt from the spreadsheet where an answer to one question has led to the grey colouring of three methods.

Our framework is meant to be used at any point in time in a project. We recommend however, that the framework is used as early as possible. No matter at what time the framework is used the decisions on what methods from the company methodology to use will cover the whole project.
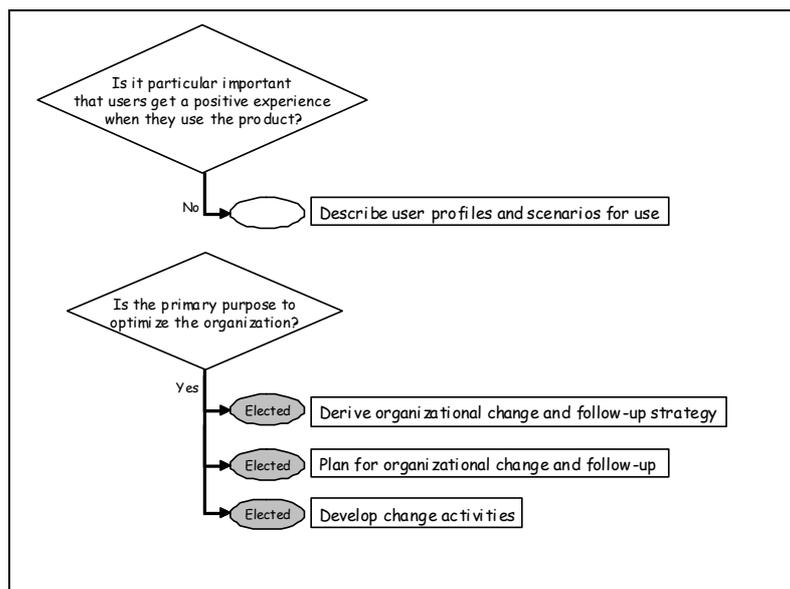

Figure 5: One answer leading to the election of three methods

In the spreadsheet we have five matrices for development activities and only one for management activities. In this one matrix we have kept alive the questions on uncertainty and complexity (that we developed in the first action research cycle –see also figure 3). However, now we have related it to content, quality, time, resources and the project environment. We have no project trail for management activities. The management matrix is about activities in the process of developing the product.

Implementing Actions

The result of using our framework is two lists of activities, one for management activities (methods) and one for development. To help projects use the framework we have developed a facilitated workshop. The idea is that the first time a project manager shall use the frameworks he can get help from experienced facilitators, but later – second, third, fourth time – he can do it himself.

Evaluation of Results

At writing time we have used the framework in 50 projects. All projects have been very satisfied with the result and have told us that they were happy that finally they could get some help in deciding what part of the company methodology to use in their project. We cannot claim the concept to be a complete success yet. We have to wait for some of the 50 projects to have ended, at which point we have the possibility of comparing the results with similar projects that didn't use the framework. And we can ask the project manager and the project participants whether they still believe that the methods identified were the best possible set.

Our learning from this third phase of the action research undertaking can be summed up in two lessons:

Lesson #5 – Interestingly enough there seems to be a difference between management oriented methods and development oriented methods

ScandiBank has decided that most management methods (activities) are mandatory. I.e. all projects shall have six milestones, all projects shall make a network plan in MS project, all projects shall make a stakeholder analysis. Thus there is not much election in this part of our framework. However, as we worked with it we found that management methods, as opposed to development methods that you either do or don't, can be done to various extent. In small projects you may only have to use 2 hours to make a stakeholder analysis of three key players where as in a large project a stakeholder analysis can be a week's work. Therefore our election framework for management came to be about the extent to do a given method.

Lesson #6 – Focusing on the end product instead of on the process proved to be extremely valuable

Before a lot of discussions in projects on what to use from the company methodology came down to nagging over what the individual liked or not. Very *amethodological* one could say. But by using a framework that begins with a focus on the product to be developed, decide a trail there from, and then goes into details about the process we have managed to create fruitful discussion in all 50 workshops. In the discussions the project members contribute on equal terms with the facilitators. They don't have to know the different methods in advance. And by answering the questions they themselves deliver the reason for using the methods elected.

CONCLUSION

We believe we have found a useful way of *reducing* a methodology to a method for a *particular situation*. We have done it in one company ScandiBank. But we don't think we have done anything so special or characteristic of the bank that our framework cannot be used in other places.

The resulting framework support what in ScandiBank has been called "The Whole Project" meaning that an artificial split between an IS part of the project and a business part is being avoided. By using seven well chosen trails based on characteristics of the product to be developed we have managed to create a framework useful in all the projects we have tried it in. And we have managed to avoid a lot of frustrating discussion on whether he or she likes or dislikes a certain method.

It is obvious that the discussions in relation to the questions for electing or rejecting methods will work differently the first time and in later use. When the participants come to know our framework we cannot avoid strategic behaviour or sub-optimisation where one person answers a question in a certain way to make sure his favourite method is elected. However, we have chosen to see this as a feature and not a defect. As long as the individual takes responsibility for his or her own elections and can explain why what methods were chosen.

ScandiBank's IS organisation has existed for many years and projects are carried out all the time. However, before the choice of methods from the company methodology was totally up to the individual and was often based on unfounded prejudices. From a company point of view the result was recurrence, lack of forward drive, inferior architecture or quality, and thereby high-maintenance solutions. Now projects achieve an overview of their

process in the form of a flow (the trail) with activities (the chosen methods). And this overview is achieved in a workshop that on average have lasted 6 hours.

Looking back on a three year process we can ask the question whether action research were the best approach to answer our research question on how to find a useful way of *reducing* a methodology to a method for a *particular situation*.

Our answer to this question is positive. We believe it has meant a lot for us that we had a method that naturally incorporated learning, and where we always were forced to make explicit the theory or model that we at the given point in time trusted in. The pilot tests have elicited extremely valuable information. And the projects have happily worked as champions for the framework so it has now been mandatory for all projects to use.

THE FUTURE OF THE FRAMEWORK

The election framework is now popular in ScandiBank and projects are actually using it. The use addresses one concrete problem namely what to use in the specific project from the methodology. But now the projects want more. In the concrete we have identified three ideas for further development of our framework.

The first one is the extent or depth of activities. In lesson #5 above we found that management methods that were mandatory could be done in more or less depth, and in our framework we included help to make that decision. But now projects want the same difference of depth for development activities. It has been mentioned that all methods could be found in a 'light' and in a 'heavy' edition. But how to make the decision on when to use what is a matter for future research.

At a point in time we expect to have carried out so many workshops that we can begin to see patterns in our data. We already sort our findings along the seven trails we have identified, but it is possible that we within each trail can identify typical patterns of answers to our election questions.

ScandiBank has many other projects than IT projects. We already have a trail called "Non IT" but we are sure this could be developed further.

REFERENCES

Bødker, K. & J. Bansler (1993). A reappraisal of structured analysis: design in an organizational context. *ACM Transactions on Information Systems (TOIS)*, Vol. 11, No.2.

CMM 1.1 (1995). *The Capability Maturity Model: Guidelines for Improving the Software process*. SEI Series in Software Engineering. Addison-Wesley.

Cockburn, Alistair (2000). *Agile Software Development*. Addison-Wesley. ISBN: 0-201-69969-9.

Datz, T. (2004). Architecture – What You Need to Know About Service-Oriented architecture". *CIO Magazine*, 17 (7) , pp. 78-88.

Erl, Thomas (2004). *Service-Oriented Architecture – A Field Guide to Integrating XML and Web Services*, Prentice Hall PTR, New Jersey.

Fitzgerald, B. (1997). The use of system development methodologies in practice: A field study. *Information Systems Journal*, 7(3), 201-212.

Fitzgerald, B. (1998). An empirical investigation into the adoption of systems development methodologies. *Information and Management*, 34, 317-328.

Fitzgerald, B. (2000). Systems development methodologies: The problem of tenses. *Information Technology and People*, Vol. 13, No. 2, pp. 13-22.

Highsmith, J. (1999). *Adaptive Software Development*. New York, NY: Dorset House.

Ian, S. (1996). Software process models. *ACM Computing Surveys*, 28(1), 269-271.

Jørgensen, N. (2001). Putting it all in the trunk: incremental software development in the FreeBSD open source project. *Information Systems Journal*, 11(4), 321-336.

Mathiassen, L., & Munk-Madsen, A. (1986). Formalizations in systems development. *Behaviour and Information Technology*, 5(2), 145-155.

Mathiassen, L & J. Stage (1990). *Complexity and uncertainty in software design*. In CompEuro '90 - Proceedings of the 1990 IEEE International Conference on Computer Systems and Software Engineering, May 1990, Tel Aviv, Israel, pp 482-489.

McConnell, Steve (1996). *Rapid Development*. Microsoft Press.

Papazoglou, M.P., and Georgakopoulos, D. (2003), Service Oriented Computing: Introduction, *Communications of the ACM*, 46(10), pp. 25-28.

Sawyer, S., & Guinan, P. (1998). Software Development: Processes and Performance. *IBM Systems Journal*, 37(4), 552-569.

Stamper, R. (1988). Analysing the Cultural Impact of a System. *International Journal of Information Management*, 8.

Truex, D., Baskerville, R., & Travis, J. (2000). Amethodical Systems Development: The Deferred Meaning of Systems Development Methods. *Accounting, Management and Information Technology*, 10, 53-79.

Truex, D. P., Baskerville, R., & Klein, H. K. (1999). Growing systems in an emergent organization. *Communications of the ACM*, 42(8), 117-123.

Womack, J.P. (1996). *Lean Thinking*. New York, NY: Rawson Associates.

Wynekoop, J., & Russo, N. (1993). *System Development Methodologies: Unanswered Questions and the Research-Practice Gap*. In J. Degross & R. Bostrom & D. Robey (Eds.), Proceedings of the 14th International Conference Information Systems (pp. 181-190). Orlando, FL: The Society for Information Management.

Wynekoop, J. L., & Russo, N. L. (1997). Studying system development methodologies: An examination of research methods. *Information Systems Journal*, 7(1), 47-65.

COPYRIGHT