METHODS & TOOLS

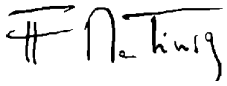
Practical knowledge for the software developer, tester and project manager **ISSN 1661-402X**

Spring 2007 (Volume 15 - number 1) www.methodsandtools.com

Too Young to Die, Too Old for Programming?

"Too Old to Rock 'n' Roll: Too Young to Die!" is a music album released in 1976 by Jethro Tull. Sometimes I feel disconnected with the current evolution of software development. I have the impression that part of the software development community advances ignoring or rejecting the past. Although the Agile Manifesto declare "while there is value in the items on the right, we value the items on the left more", it seems that people are denying value to previous approaches ("items on the right ") or trying to put an "agile" label on them, as if it was the only way to be accepted. Web material about the fact that modelling is useless or questions about the need for IDE caused the same impression. Ruby on Rails is a great idea, but it is a step beyond for me if the only development tool you can use is a text editor.

The problems of software development remain the same and the proposed solutions share similarities. I will use as an example the objectives of the Capability Maturity Model (CMM) explained in a 1987 document that provided an assessment questionnaire. The maturity levels contain the following goals: manage cost, schedule and requirements changes; review design and code; favour process metrics and improvement. I am not an agile expert, but it seems that these objectives share common concerns with the agile practices like Scrum project management, pair programming, focus on unit testing, retrospectives. You can discuss the relative value of pair programming versus code review, but both have a place in the software development toolbox. If you think that the CMM is only created for waterfall life-cyle model with heavy documentation, the 1991 document that presented the key practices stated that the CMM does not imply a particular life-cycle model, a specific set of documents or organisation structure. Measurement is the only major CMM practice not present in agile practices. There are many failures to adopt approaches judiciously in the software development world. If you do not understand the essence of an approach and adopt it without tailoring it to the context, you will face problems. As agile practices spread, they face the same challenges than previous approaches and generate their own "horror stories". This should not mask the positive aspects of agile approaches... like those provided by the CMM, Information Engineering, RAD or RUP.



Inside

Process Improvement – Is it a Lottery?

Paul Morgan
Technology Process Group, GTECH Corporation

Abstract

If process improvement is a journey then the experience of most organizations is that the path is a rocky one and the places encountered on the way are seldom the ones originally sought. A major contributing factor to this problem arises from the difficulties that organizations face when adapting their chosen process improvement model to their 'real world' situation. Models such as the Capability Maturity Model (CMM®) (1993) and Capability Maturity Model Integration (CMMI®) (2002) are generic by design, not prescriptive, and extensive work is required to ensure that during implementation they contribute positively towards an organizations business goals. Immature organizations are most at risk of allowing the model rather than their business needs to dictate the nature of their process improvement initiative.

GTECH is a global technology company operating in over 50 countries, providing software, networks, and professional services that power high-performance, transaction-processing solutions. This article provides an overview of the approach utilized to implement process improvement across its global organization without losing focus of its business drivers. It provides a practical overview of how over a four year period an organization moved from CMM® Level 1 to Level 3 and is currently transitioning to CMMI® Level 4. It will provide a candid insight including lessons learned and approaches adopted to achieve success. It will also provide examples of significant and measurable business benefits that can be accrued from adopting a documented and repeatable process improvement framework.

Introduction

"Not everything that counts can be counted; not everything that can be counted counts." Albert Einstein

In 1994 the Standish Group, undoubtedly with tongue in cheek, named their original research paper into the state of IT project management *The CHAOS Report* (1994). Whilst the dire situation described in that original report has undeniably improved over the intervening years there can be few people for whom the humour in that original title has not turned sour. A follow-up report published by the Standish Group entitled *Extreme CHAOS* (2001) states that still only 28% of IT projects are completed on time, on budget and with all the features/functions originally specified. Of the projects that were successful 97% had a project manager assigned as opposed to 79% for those that were unsuccessful. Whilst this statistic undoubtedly adds weight to the argument that assigning a project manager enhances the chance of success it does not answer the question as to why so many IT projects, even when run by a dedicated project manager, ultimately fail to meet their original objectives.

This article will describe the steps taken to avoid such failures by institutionalizing project management good practices across its global software development organization. It will explain how a process improvement framework was utilized to translate generic project management practices into specific processes and procedures more relevant to the development of software. In other industries, such as construction, the wellbeing of the project can be assessed using all of ones senses from the moment the first foundation is laid. Due to the intangible nature of a software product, and more specifically its component parts as they are transformed through the development cycle, such faculties are either not available or considerably dulled for those

assigned responsibility for managing a software development project. Our experience indicates that this constraint, inherent in all software development, can be successfully mitigated by adopting a software process improvement framework wherein management and engineering activities are integrated to establish process controls that provide the prerequisite visibility. Further, by aligning this approach with the business goals of the company the benefits realized by individual projects can be leveraged across the organization.

Process Definition

The Model

The CMMI® was developed by the Software Engineering Institute (SEI) based upon best practices observed within successful development organizations. (Exhibit 1) We adopted the staged version of the model with each successive level demonstrating an evolutionary plateau that establishes a new capability for performing business functions. This capability realised in terms of organisational maturity, results in improved productivity and quality, and reduced project risk. Each level is broken down into a number of process areas that identify a cluster of related activities considered important for establishing process capability at that maturity level.

Maturity Level	Focus	Process Areas
5 Optimizing	Continuous Process Improvement	Organizational Innovation and Deployment Causal Analysis and Resolution
4 Quantitatively Managed	Quantitative Management	Organizational Process Performance Quantitative Project Management
3 Defined	Process Standardization	Requirements Development Technical Solution Product Integration Verification Validation Organizational Process Focus Organizational Process Definition Organizational Training Integrated Project Management Risk Management Integrated Supplier Management Integrated Teaming Decision Analysis and Resolution Organizational Environment for Integration
2 Managed	Basic Project Management	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management Measurement and Analysis Process and Product Quality Assurance Configuration Management
1 Initial		

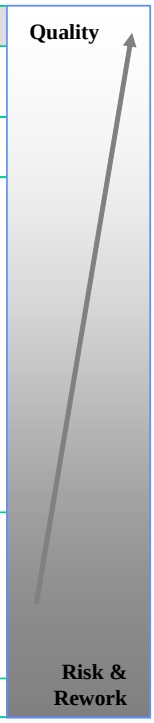


Exhibit 1 – Capability Maturity Model Integration (CMMI®)

Level 1 of the model symbolizes those organizations that operate within an unstable environment that lacks sound management practices. Project commitments are typically not controlled and success rides on individual talent and heroic effort. Practitioners argue against engineering discipline under the guise of “individual creativity”. Those standards and practices that do exist are often sacrificed to management priorities. Process capability is unpredictable with schedule, cost, and quality targets often missed. A level 1 organization might unconsciously do some of the practices expected at higher levels of maturity but they do not get repeatable results, even if they are perceived to be effective.

Level 2 of the model is referred to as “Managed” wherein policies for managing a software project and procedures to implement those policies are established. Realistic project commitments are made based upon previous project results and the unique requirements of the current project. Management tracks software costs, schedules, and functionality against plans. When problems are identified appropriate corrective actions are taken. Project artefacts are base-lined and their integrity controlled by exercising disciplined configuration management. Strong customer/supplier relationships are developed and managed. From an organizational perspective, projects are allowed to have their own unique processes. Level 2 recognizes that project management practices provide an essential foundation for effective software engineering.

Level 3 of the model is referred to as “Defined”. A standard set of processes for developing and maintaining software is documented and used across the organization. This standard process includes both software engineering and management processes and integrates them into a coherent whole. Projects tailor the organisations standard process to develop their own defined software process, which accounts for the unique characteristics of the project. A group within the organization is assigned responsibility for software process activities. An organization-wide training program is implemented to ensure that the staff and managers have the knowledge and skills required to fulfil their assigned roles.

Level 4 of the model is referred to as “Quantitatively Managed”. The organization sets quantitative quality goals for both software products and processes. Productivity and quality are measured for important software process activities across all projects as part of an organizational measurement program.

An organization-wide software process database is used to collect and analyse the data available from the projects' defined software processes. Projects achieve control over their products and processes by narrowing the variation in their process performance to fall within acceptable quantitative boundaries.

Level 5 of the model is referred to as “Optimizing”. Technical, managerial, and process defects are measured and via causal analysis specific process activities are taken to prevent reoccurrences and improve the overall performance of future projects. Technological improvements are systematically discovered, analyzed and applied and continuous process improvement is formalized.

Documenting the Process

A common mistake made by companies when implementing process improvement is to allow the chosen model to dictate the process design. The CMMI® is a model that needs to be interpreted based upon the business environment and technical needs of the project; it is not a standard that must be implemented exactly as documented. A failure to recognize this might still result in maturity levels being attained; however, the end product is unlikely to be a process suite which complements your operating needs. The correct approach is to balance the recommendations of the model against the business value that the new processes will add. This value added might be recognized by the individual (i.e. improved engineering methods), by the project team (i.e. more accurate estimation), or by the organization (i.e. metrics and project artefacts available for reuse); however, it should not simply be dictated by what the model suggests is good practice or what in principle is a good practice.

Another consideration when documenting your processes is that the majority of process improvement models are designed as auditor’s tools. As such they provide a useful checklist of practices which should be institutionalized within a mature organization but they are not written to easily support the project team in their daily activities. A common trap is for companies to also use the model as a framework for categorizing their process documentation. Separate processes are designed and documented for each process area resulting in a process library that will impress any auditor but which is of limited use for project practitioners whose responsibilities span across process areas. The result is often ‘shelf-ware’ that is seldom referenced and just gathers dust on the bookshelf; for those of you whose process documentation sit on a central server just do a check on the number of hits per day your library attracts.

In contrast we took a role based approach when designing its processes. This entailed identifying the key project personnel that were required to deliver a quality product and then for each designing and sequencing their activities and orchestrating them overall into a coherent workflow. Compliance with CMMI® was a key factor in the process design but not at the cost of operational efficiency. The resultant documentation formed a ‘to do’ list that project personnel could follow on a daily basis. The emphasis was also on readability and usability. Great care was taken to ensure that process definitions were not verbose. Help text, guidelines, and training material were kept in separate files that could be referenced as needed. The global nature of the process deployment across six continents and seventeen time zones also required a process suite that was intuitive to use with corporate-wide visibility. The method chosen was an intranet based GUI solution that provides automated workflows to guide and control project personnel in their activities. The application also acts as a repository for project artefacts with full configuration management, risk management, and process history.

Advertisement – European SEPG 2007 Conference - Click on ad to reach advertiser web site

Process Deployment

The Process Improvement Project

Organizational change forms the basis of any process improvement initiative. Consequently alongside the normal project challenges of designing and deploying a new 'product', in this case new processes, there is the added human dimension of changing existing work practices. The sum of these two factors means that to ensure success the process improvement project must be run using the most rigorous of project management procedures. In CMMI® terms, an organization cannot expect its project teams to function at high levels of maturity if the rest of the organization is still operating at level 1. Management at all levels must practice what they preach; this typically requires a radical cultural change.

To this end a dedicated process management team exists to manage the development, deployment, and ongoing maintenance of its processes. This Technology Process Group (TPG) was staffed by recognized discipline experts and reported into a steering group of senior managers who provided sponsorship, direction, and visible endorsement of the TPG activities. Quarterly progress reviews were held to provide the necessary oversight.

On an annual basis the TPG develops an organizational process improvement plan based upon the business goals of the company. This plan outlines the process improvement objectives for the following year, the approach that will be taken to achieve those objectives and the resources required. Once approved by the steering group this plan is then broken down into manageable tasks and activities to form a Work Breakdown Structure / Schedule that is used to track progress. As with any other project changes to the original requirements do occur and these are controlled by maintaining the developed processes under full configuration management. An independent Standards Compliance function was also established to ensure that the TPG was abiding by its own rules.

Overcoming Resistance to Change

There are a number of factors to consider when implementing a process improvement program not least of which is the resistance to change that you will undoubtedly encounter. Force field analysis is a management technique developed by Kurt Lewin for diagnosing such situations (1946). Lewin assumes that in any situation there are both driving and restraining forces that influence the success of any proposed change. The TPG conducted just such an analysis at the commencement of its process improvement program. (Exhibit 2)

The driving forces for process improvement were provided by the close mapping to business goals, senior management sponsorship, and the consensus by all stakeholders for improved customer satisfaction, productivity, and reduced costs. The restraining forces were less easy to quantify and consisted of a spectrum of people issues from overcoming a natural inertia for change, through issues of self-interest, to a lack of understanding. In a situation such as this just increasing the pressure of the driving forces, for example through management dictates, seldom works. At best a reluctant compliance is achieved and at worst no change occurs as individuals and teams dig their heels in and passively resist. The solution is to understand the reasons for opposition and apply management techniques to address concerns and thus weaken resistance.

At the start of the process improvement initiative, stakeholders were identified and process improvement plans and objectives were negotiated and agreed. Individual performance goals were then established for senior management and these were cascaded down through lower levels of management to the engineering level. Initially simple measures of success were used

such as the percentage of personnel trained on the new processes and the adherence to process compliance. Later more sophisticated measures relating to project performance were introduced.

The next technique was to involve the practitioners, those who would use the processes on a daily basis, in the process design. This participation and involvement not only ensured that the processes deployed would meet operational needs it also provided them with a feeling of process ownership. It therefore precluded the feeling that the new processes were being imposed upon them and avoided a potential source of resistance. This involvement is an ongoing theme of the process improvement initiative. A formal mechanism remains in place for anyone on a project to propose a justification for tailoring and customizing the standard processes.

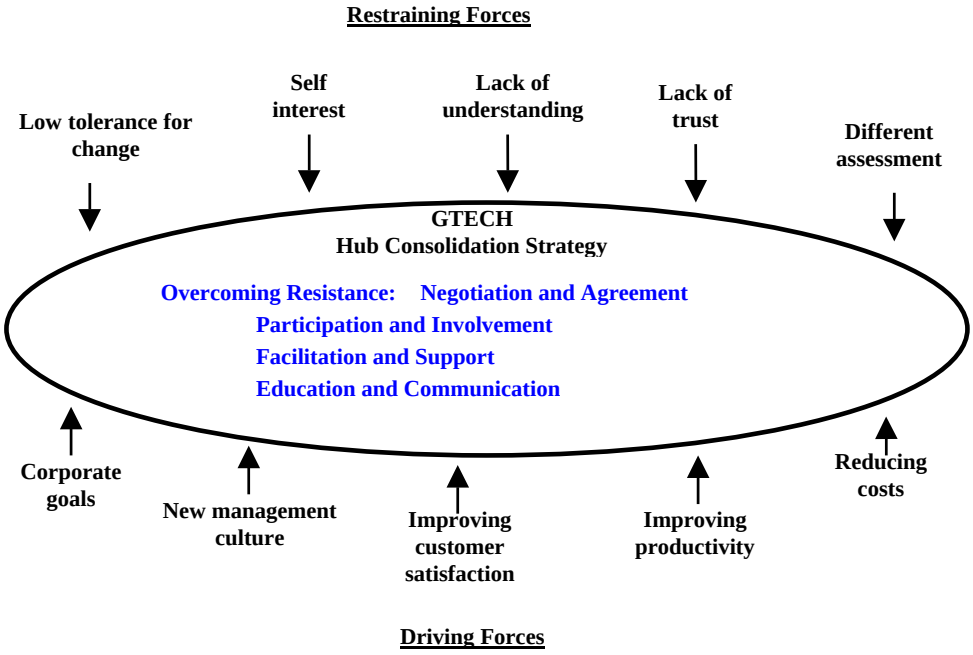


Exhibit 2 – Forces Influencing the Deployment of Process Improvement

To facilitate and support the institutionalisation of the new processes dedicated TPG resources are located in each development centre. These individuals provided mentoring and coaching support to assist the adoption of the new processes into the working environment. Specialist discipline support in areas such as estimation techniques and configuration management are also provided as needed. A standards compliance function was also added to every project to ensure that processes were being followed. To address any potential resistance such as "I've not got a budget for that" all TPG resources were provided at no direct cost to the projects.

A key factor when rolling out process level improvements is to ensure that the people you expect to follow them have the necessary underlying skills. An extensive organizational training program was rolled out concurrently with the process improvement program. This education initiative included training such as; process and CMMI®, project management (i.e. PMBOK), discipline specific, tools usage, and soft skills (i.e. team building). A variety of delivery methods was utilized including; instructor led, web based, and 'Train the Trainer'. The latter of these consisted of empowering the project teams themselves to take on responsibility for their own training.

Vital to the success of any project is the ability to maintain effective two-way communication channels. The TPG utilises periodic newsletters and its own website to distribute general information on both project issues and successes. Affinity groups were established to facilitate discussion on specific process areas. As processes were modified and updated formal release

notes and delta training were utilised to ensure a smooth transition. Meetings with all levels of management were effectively used for the purposes of both progress reporting as well as to address in a timely manner any issues that had arisen.

Finally, despite every effort to facilitate change, some people and/or teams may refuse to conform. In such circumstances, and as a last resort, management intervention might be the only option. The reassignment of staff can be one alternative so as to prevent the process improvement initiative from stalling.

Process Deployment Costs and Benefits

Process Deployment Costs

Process improvement does not come free of charge although the investment made does reap major rewards. Expenditure is firstly required for the recruitment and training of personnel to manage the process improvement program. Processes need to be designed and documentation created. Deployment requires the production of training materials and the allocation of instructors to conduct classes. Support personnel are required to assist project staff climb the learning curve associated with the implementation of new working practices. External consultants are needed to help fill knowledge gaps and conduct assessment services against the model to judge progress. Opportunities might also exist to purchase tools to help with automation. However commercial off the shelf (COTS) products will, almost as a rule, require customisation to your own needs and thereafter be maintained.

A study by the Software Engineering Institute (SEI) *Benefits of CMM-based Software Process Improvements: Initial Results* (1995) suggests that the yearly average cost for companies undertaking software process improvement was US$ 1,375 per software engineer. A more recent survey *CMMI Market Trends* (2005) indicates that companies who invest between 5% and 7% of their development budget have the greatest chance of sustaining a successful process improvement program.

Process Deployment Benefits

The Software Engineering Institute (SEI) *Benefits of CMM-based Software Process Improvements: Initial Results* (1995) also indicated the rewards that can accrue from a process improvement program. (Exhibit 3) It should however be noted that such benefits usually lag investment by months or years. Greater returns are realized as the experience of operating at a particular maturity level increase.

Category	Range	Median
Productivity Gain per Year	9% - 67%	35%
Early Defect Detection Gain per Year	6% - 25%	22%
Yearly Reduction in Time to Market	15% - 23%	19%
Yearly Reduction in Post-Release Defect Reports	10% - 94%	39%
Business Value (savings/cost of SPI)	4.0 - 8.8:1	5.0:1

Exhibit 3 – Benefits of Process Improvement

Less quantifiable, but equally valuable, business advantages are also achieved. Firstly deploying a standard project process is recognition that an organizations development practices are valuable intellectual property that must be defined, documented, and secured. If an individual leaves an organization and takes with them undocumented knowledge disruption and the need for reinvestment can result. Standardizing project processes irrespective of where development work takes place also provides for; more effective resource utilization, faster project start-up and less re-training, improved teamwork and employee morale, and increased customer confidence as project successes become repeatable. The availability of a historic project information repository also provides a knowledge base that facilitates not only the reuse of work products but also enhances project management capabilities by rendering estimations based upon accurately identified historical evidence. Quantitative management techniques can also be utilized to enhance risk management via exception reporting by benchmarking current project performance against established historic norms (i.e. defect detection profiles). Finally, the visibility afforded by an intranet based process improvement implementation facilitates real-time management decision making.

Final Words

Here are some final words of advice for anyone that is considering travelling along the process improvement highway. Whilst they may not guarantee that you reach your desired destination they will prevent you from falling into some of inevitable potholes that you will encounter along the way. Firstly, the time to achieve a maturity level depends entirely upon the extent of senior management sponsorship. The commitment required must be more than just providing the necessary budget. Senior management must be seen to proactively support the process improvement program. When something goes wrong, as inevitably it will, they must support an adherence to the deployed process and not resort to crisis management techniques.

The TPG must be staffed with recognized leaders and discipline experts. Reassigning specialists off projects might in the short term appear to be folly but will over time earn considerable dividends in terms of the quality and uptake of the processes deployed. The process improvement program must be treated as a high priority project with enforced accountability and high-visibility status reporting. Treat the deployment of processes as if it was a product delivery to your most demanding and important customer.

Do not adopt the maturity model as your process. All process improvement models need to be interpreted based upon the specific needs of your organisation. It is your responsibility not that of the model to tell you how your business should be run.

Use the best technology available to develop and deploy your processes. However, do not be seduced by the allure of a 'silver bullet' solution. Just as you cannot loose weight by buying a set of bathroom scales an organization cannot become more efficient by simply purchasing a tool. An individual has to change their lifestyle to see results and an organisation must change its work practices to recognise benefits in terms of improved productivity and quality.

Ensure accountability via standards compliance and periodic assessments (internal and external) to check progress. Then report results to all stakeholders. The use of external consultants helps avoid blind spots however do not abdicate responsibility to them for the process improvement program. You know your business needs better than they do and at the end of the day accountability for success or failure rests with you not them.

References

- Capability Maturity Model® Integration (CMMI^{SM}), Version 1.1 (2002). Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213
- Key Practices of the Capability Maturity Model^{SM}, Version 1.1. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213
- The CHAOS Report (1994). The Standish Group International, Inc.
- Extreme CHAOS (2001). The Standish Group International, Inc.
- Frontiers in Group Dynamics (1946). Kurt Lewin
- Benefits of CMM-Based Software Process Improvement: Executive Summary of Initial Results (1995). Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213
- CMMI^{SM} Market Trends (2005). Andrew Griffiths, Lamri, CMMI^{SM} Made Practical Conference.

Strategic Modeling for Rapid Delivery of Enterprise Architecture

Clive Finkelstein, cfink@ies.aust.com
Information Engineering Services Pty Ltd, http://www.ies.aust.com

Introduction

In this article I will discuss how a Strategic Model is developed for rapid delivery of Enterprise Architecture, based on business planning statements from Col 6 (Why) Row 2 (Owner) of the Zachman Framework for Enterprise Architecture. I will also discuss how to identify from a data model relevant business activities or processes in Col 2 (How) Row 1 (Planner). These will enable activities or processes to be prioritized so they can be used for later rapid delivery of priority systems into production, using workflow models and automatically-generated directly-executable business process management (BPM) languages such as business process execution language (BPEL). I will show how project plans and associated project maps can be derived from a data model to manage the rapid delivery of priority business activities and processes into production as systems. These project plan and project map derivation concepts have not previously been used in data modeling.

The Zachman Framework for Enterprise Architecture

Enterprise Architecture addresses the questions: “What”; “How”; “Where”; “Who”; “When”; and “Why”. These questions should be capable of being answered from the different perspectives of management and staff levels in an enterprise. The details of interest to senior management – as the “Planners” and “Owners” of the enterprise – are likely to be different from the detail needed by middle managers, business experts and IT staff – as the “Designers” and the “Builders” – of the business processes and support systems that are used for financial and other reporting.

These six questions are represented by columns in a matrix, where the different perspectives of “Planner”, “Owner”, “Designer”, “Builder” and “Subcontractor” discussed above are represented as rows. This matrix is provided by the *Zachman Framework for Enterprise Architecture*, as shown in figure 1. The *Zachman Framework for Enterprise Architecture*, developed by John Zachman, is used world-wide for the management of internal controls and alignment of information systems with business and IT resources of Government, Defense and Commercial enterprises. The Zachman Framework is the Enterprise Architecture foundation used by US Department of Defense (as DoDAF – the DoD Architecture Framework) and by US Federal Government Departments (as FEAF – the Federal Enterprise Architecture Framework). The DoDAF and FEAF are based on Enterprise Architecture, as mandated by the Clinger-Cohen Act of 1996. While Enterprise Architecture has previously been considered to be an IT responsibility, when it is also used by senior management it enables precise *Governance Analysis.* It also provides a *Business Transformation Enablement* capability.

The Designers, Builders and Subcontractors (often outsourced) work with business experts who understand the business processes of the enterprise. Based on their business knowledge, IT staff design and build systems and databases that support those processes. They provide the data, information and processing needed for day-to-day operational functioning of the enterprise. They are represented by the bottom three rows of the Zachman Framework in figure 1.

	What	How	Where	Who	When	Why
PLANNER Objectives/Scope						
OWNER Conceptual						
DESIGNER Logical						
BUILDER Physical						
SUBCONTRACTOR Out-of-Context						
FUNCTIONING ENTERPRISE	Data	Function	Network	Organization	Schedule	Strategy

Figure 1: The Zachman Framework for Enterprise Architecture.

What is a Strategic Model?

To illustrate what a strategic model is, I will use business planning statements from the *Project Management Division* of a hypothetical company – XYZ Corporation – as a catalyst to identify the data and information in a strategic model needed to support management in implementing those business plans in the enterprise. I will discuss the conduct of a facilitated strategic modeling session to develop a strategic model based on the plans. I will show you how to identify from the resulting strategic model relevant business activities or processes. These can be prioritized by management so they can be used for later rapid delivery of enterprise architecture. I will show you how to derive project plans and associated project maps from a data model to manage the delivery of these priority business processes into production.

High-level views of an enterprise are shown by horizontal slices at the top of each cell of the Zachman Framework for Enterprise Architecture. This is illustrated in figure 2, now showing representative models that apply to each cell. The numbered columns indicate implementation sequence, keyed to the sequence of the numbered steps following the figure.

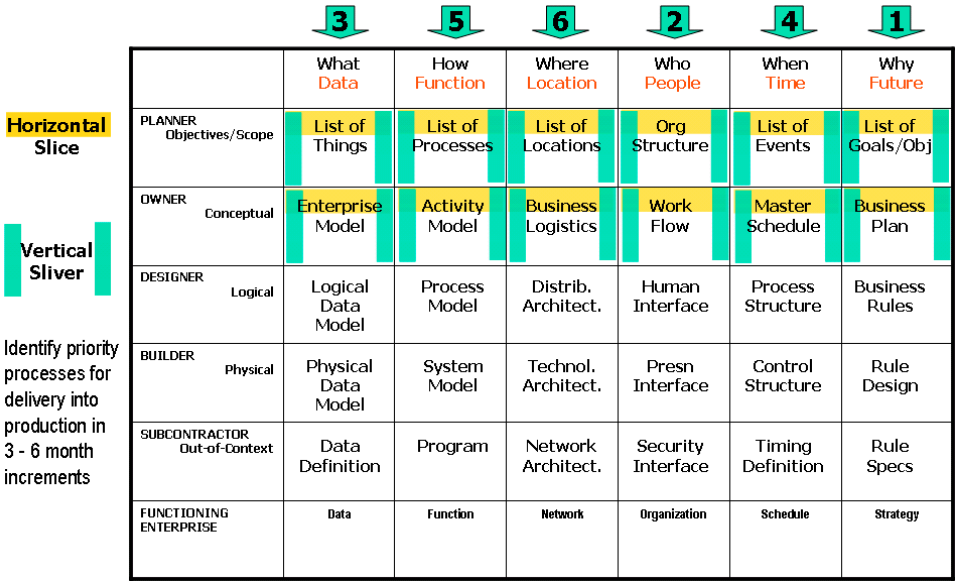


Figure 2: High-level priority areas are identified as “vertical slivers” for rapid delivery

1. Col 6 (Why) focuses on the business plans for the future, to understand the directions for that future.
2. Col 4 (Who) identifies managers in the enterprise responsible for implementing the business plans. It identifies the staff who will help them (as business experts) to transform the enterprise for that future.
3. Col 1 (What) uses the managers and business experts from Col 4 to help identify the data and information needed to support the enterprise based on its transformation plans for the future.
4. Col 5 (When) identifies major business events that initiate business activities. This focus on business events based on the business plans can be used to identify reusable business activities. These can have the greatest potential transformation impact on the enterprise.
5. Col 2 (How) addresses business activities identified from Col 5 and also Col 1. It defines activity models and then uses activity based costing to evaluate costs and other benefits of alternative strategies for business transformation that were defined in Col 6.
6. Col 3 (Where) then addresses the locations of the enterprise to participate in this transformation.

The high-level focus of the horizontal “slice” at the top of each cell also enables priority business activities or processes to be identified as areas for transformation, that need to be implemented first. These are shown as vertical “slivers”.

From the Planners’ and Owners’ perspectives in Rows 1 and 2 we see that vertical slivers in each cell enable greater detail to be defined in priority areas, in the sequence as numbered in figure 2. These areas progress to detailed definition. This is represented by the depth of the vertical sliver in each cell. It leads to rapid implementation using appropriate modeling tools and technologies. Thus priority business activities can be delivered into production early – before other, less important areas that can wait until later.

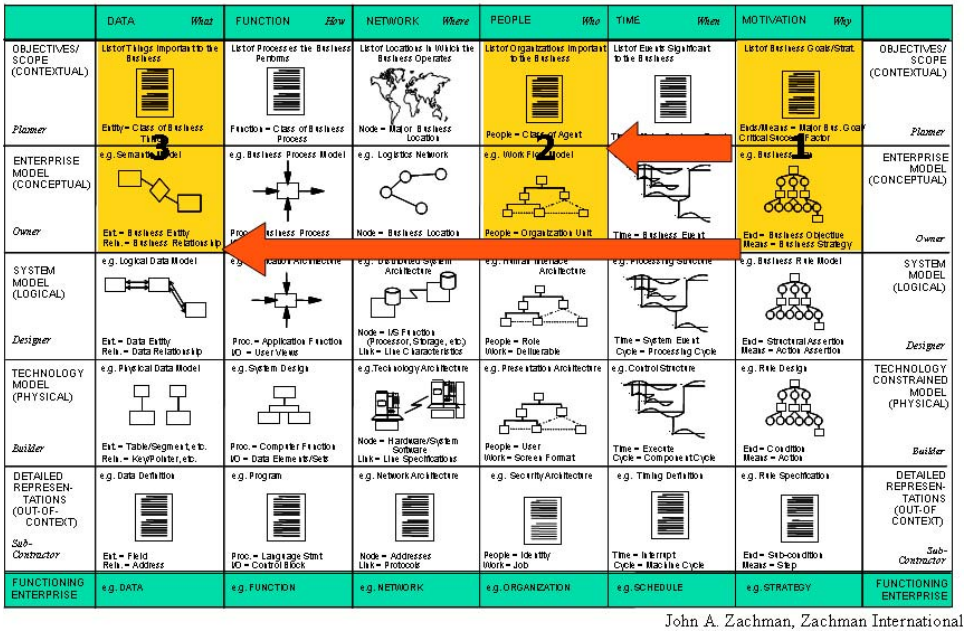


Figure 3: Business Plans in Col 6 (Why) are the catalyst to identify data in Col 1 (What)

In terms of the Zachman Framework for Enterprise Architecture, development of the strategic model shown in figure 3 will take the following numbered steps;

1. Strategic plans in Col 6, Rows 1 – 2 will be used as the input catalyst of strategic needs to transform the enterprise for the future.
2. Managers and business experts are identified from the organization structure in Col 4 (Who) Row 1 (Planner) to participate in the strategic modeling facilitated session.
3. Strategic modeling is used to identify "Things" of interest to the Planner in Col 1 (What), Row 1. It is used to develop a strategic model in Col 1(What), Row 2 for the Owner. This is a high-level data model that addresses the strategic data needs for the future based on the business plans.
    - Strategic business plans provide input to strategic modeling, to develop a strategic model.
    - Analysis of the strategic model produces an enterprise architecture portfolio plan (EAPP) for rapid delivery of priority business activities into production.
    - The strategic model, the EAPP and tactical business plans all provide input to more detailed tactical modeling to develop logical data models as tactical models.
    - The EAPP, tactical models and operational business plans also provide input to still more detailed operational modeling to develop logical data models as operational models.
4. Figure 4 shows that Strategic Business Plans identify *Information Requirements* of management and provide input to the data modeling phase as discussed following the figure.

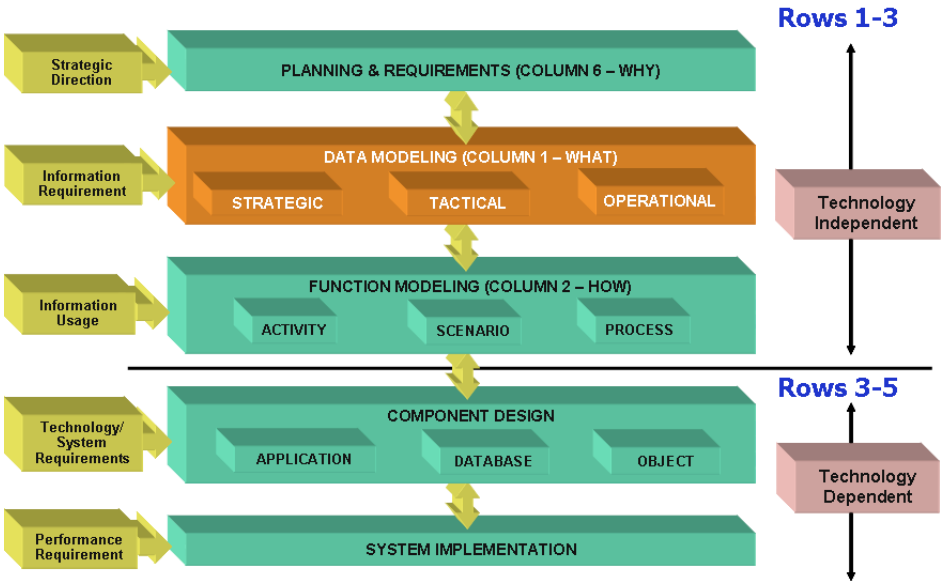


Figure 4: A strategic model is a "picture of the business", defined using data modeling

The strategic priorities that provide input to strategic modeling are identified as follows:

- The strategic plans in Col 6 (Why) are related to parts of the enterprise by using a matrix of planning statements related to the organization structure (Statement – Organization Matrix).
- The Statement – Organization Matrix uses the organization structure in Col 4 (Who), Row 1 (Planner) to identify key managers who are responsible for various statements in the strategic plan in Col 6 (Why).
- These key managers and their business experts participate in the facilitated modelling session (covered next) to develop the strategic model in Col 1 (What), Rows 1 – 2.
- Data modeling is used to develop a strategic model from strategic plans for the rapid development of high-level business data models. These data models will be used to derive project plans for high-priority business activities and processes, to deliver high-Return-on-Investment systems early.

Data modelling helps identify alternatives, leading to business benefits. This provides business justification for technology alternatives and funding approval for the technology and resources for implementation.

The enterprise architecture portfolio plan (EAPP) is a deliverable from strategic modeling and is based on entity dependency analysis, which I will discuss shortly. This establishes clear project plans for priority projects. It leads to detailed development of approved projects. Tactical and operational data models define data bases for rapid implementation, and define metadata for XML repositories, web services and Service-Oriented Architecture (SOA) as I will cover in later articles.

I will now discuss the business planning statements that I will use as catalysts for an example strategic modeling facilitated session. The following statements are for XYZ Corporation, whose corporate mission is as follows:

> "***Mission of XYZ Corporation:*** *We supply products and services to address the needs of our customers, wherever they are located. On their behalf we will research and source the most appropriate products from the world's leading suppliers. We are skilled and*

> *dedicated people working with our customers to satisfy their needs and expectations for our long term mutual benefit. We will provide exceptional service and value so that we will always be their first choice. We will increase the value of our Company and also improve the economic well-being and quality of life of our stakeholders: customers, suppliers and staff."*

Effectively, XYZ acts as a broker for its customers, locating suppliers that provide the products and services that they need. I will use planning statements from the *Project Management Division* of XYZ Corporation. As an experienced project manager, you can therefore take the role of a business expert in the following discussion of an example facilitated strategic modeling session. These statements define the initiation and management of projects based on the following business unit mission:

> *"**Project Management Mission:** We will establish a project to manage research and sourcing of each customer's product requirements, based on their identified needs and location. For each customer project we will assess the customer based on profitability, product pricing, governance and security criteria. Based on guidelines defined with the criteria, we will research and source the most appropriate products from the world's leading suppliers."*

The following statements are not intended to cover all aspects of project management. They are defined only to illustrate the application of strategic modeling principles. There are nine major policy statements in the *Project Management Division* business plan. To illustrate a strategic modeling facilitated session, I will use the first three policy statements as catalysts, defined by XYZ and prioritized as follows:

1. Project Ownership Policy	Priority
Each Project must have a Project Owner, responsible for allocating and managing the project budget	1
2. Project Management Policy	
Each Project must have a Project Manager, responsible for completing the project by the scheduled date, within budget	1
3. Project Authorization Policy	
Projects are only authorized that can achieve Project Objectives by the scheduled completion date, within budget.	2

Conducting a Strategic Modeling Facilitated Session

A strategic modeling facilitated session is attended over 2 days by senior managers and business experts responsible for the XYZ *Project Management Division*, identified from the organization structure in Col 4 (Who) Row 1 (Planner) of figure 2.

If the enterprise architecture focus is enterprise-wide, this will of course include the senior managers of the organization. If the focus is on a particular enterprise area only – such as the *Project Management Division* of XYZ in this example – it will draw on senior managers and business experts of that business unit and other units that interface with it. In this discussion, you can therefore take the role of a project management business expert.

Summarizing the strategic priorities that were identified earlier in relation to figure 3:

- The strategic plans in Col 6 (Why) are related to parts of the enterprise by using a matrix of planning statements related to the organization structure (called a "Statement – Organization Matrix").
- The Statement – Organization Matrix uses the organization structure in Col 4 (Who), Row 1 (Planer) to identify key managers who are responsible for various statements in the strategic plan.
- These key managers and their business experts are used during the facilitated modelling session that follows, to develop the strategic model.

The following paragraphs describe the conduct and results of an example facilitated modeling session with these business managers and their business experts.

A facilitated strategic modeling session draws on expert business knowledge. Two days is a large amount of time for managers and business experts to allocate. It is important that all participants see the modeling session as an opportunity to identify their business needs and relevant business rules. They should *not* see it as a technical data modeling exercise. The session should focus on developing a *"picture of the business"* based on strategic plans and expert business knowledge.

The picture that is built up on a white board does not assume the business participants know anything about data modeling. Instead, any data mapping principles are presented from a business perspective only, to help in documenting the *"business picture"*.

Unless the managers quickly see that they can directly contribute their business expertise to the evolving picture, they will consider that the session is about computers: that it is for IT staff; and not for them. At the first rest or coffee break they will disappear back to their offices with urgent duties they must attend to. If this happens, you will not get them back. You will have lost an opportunity to get them actively involved. This opportunity may not easily present itself again. The result will be that your enterprise architecture initiative will become a typical IT project. It will be limited – as for most IT projects – by the difficulty in determining the business requirements.

As we develop the business picture in the following pages you will see that the relevant data modeling and data mapping principles are introduced in a *non-technical* way to document the business rules that are being represented in the picture. But as it evolves as a strategic model, you will also see that it is a high-level representation of an enterprise model.

1. Project Ownership Policy

I will start with the first planning statement from the project management division of the XYZ Corporation. This is called the *Project Ownership Policy*:

- *Each Project must have a Project Owner, responsible for allocating and managing the project budget.*

The nouns in the above statement are first listed by the facilitator on the white board as shown below. You will recognize these as a "list of things" for Col 1 (What) Row 1 (Planner) as follows:

PROJECT

PROJECT OWNER

BUDGET

PROJECT BUDGET

The first two listed – PROJECT and PROJECT OWNER – are drawn as boxes on the white board. But they are NOT given the technical term “entity”. Instead the facilitator explains that:

> *“The PROJECT box is used to represent all of the details that need to be stored about a Project. It is expressed in the singular for a single project. Many projects can exist; each can be represented in the third dimension coming out from the white board. The single PROJECT box that is drawn on the white board will therefore be used to represent all Projects. Similarly the single PROJECT OWNER box that is drawn on the white board will be used to represent all Project Owners.”*

This is shown in figure 5.

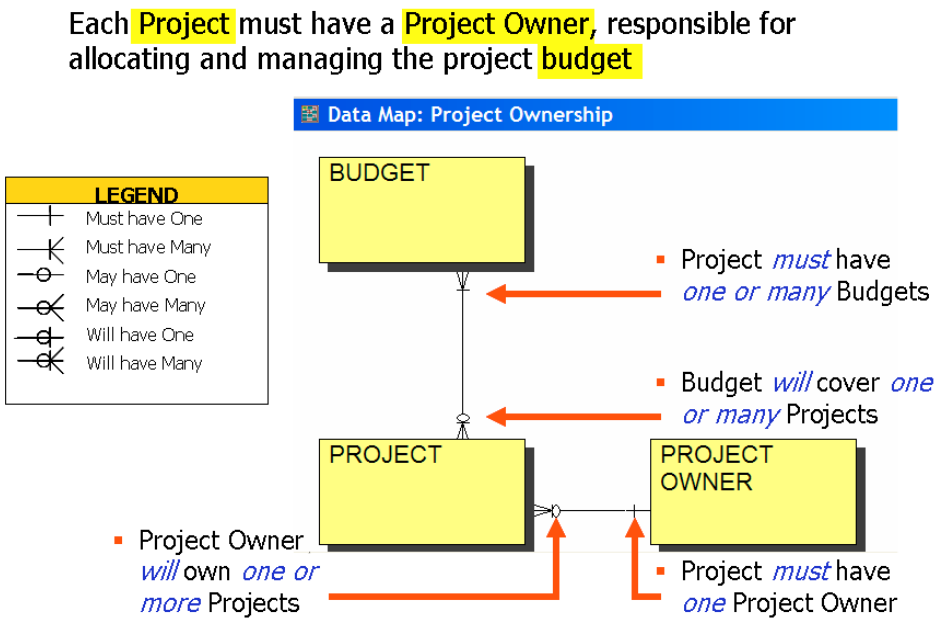


Figure 5: Strategic model fragment from the first planning statement

The facilitator asks if the PROJECT and PROJECT OWNER boxes in figure 5 are related to each other in any way. This question will be answered positively by the business audience. By drawing a line joining them, the facilitator explains that:

> *“This line will be used to show the business rules that are used to manage Projects and Project Owners. Lines that join boxes can also represent management controls, audit controls, security controls, governance controls, communication paths or reporting paths.”*

In this statement the facilitator has explained the concept of association lines in business terms, not technical terms. Next the association degree (cardinality) and the association nature at each end of the line are also defined in business terms, as business rules.

The facilitator asks if a Project can have only one Project Owner. If the response is positive, a short bar is drawn across the line close to the PROJECT OWNER box. The facilitator explains this by saying that:

> *"We will use this bar to represent the business rule you have just confirmed – that a Project must have only one Project Owner."*

If the reply is negative and there are many apparent Project Owners, the facilitator asks if one is nominated as the *"responsible owner"*, or if a Steering Committee acts as the responsible Project Owner. Next the question is asked for the other end of the line:

> *"Is a Project Owner responsible for only one Project, or for one or many Projects?"*

If the audience response is *"many projects"*, by holding up three fingers the facilitator introduces the concept of the crow's foot symbol to represent *one or many*, with a single finger representing *one*. The crow's foot symbol is added to the end of the line touching the PROJECT box, pointing out that:

> *"The one symbol (the absence of a crow's foot) already touches the PROJECT OWNER box based on the earlier business rule that you defined."*

The next question determines the association nature at the PROJECT end of the line, by asking:

> *"May a Project Owner have no projects at any time?"*

For a positive reply, the facilitator adds a zero to the *many* symbol saying:

> *"This represents the business rule that a Project Owner **may** own zero, one or many Projects at any time."*

On a positive response to: *"must a Project Owner have at least one project at any time?"* a short bar is instead added to the *many* symbol saying:

> *"This represents the business rule that a Project Owner **must** own at least one or many Projects at any time."*

Finally, on a positive response to: *"will a Project Owner eventually have at least one or many projects?"* a short bar and zero are both added to the *many* symbol saying:

> *"This shows the business rule that a Project Owner **will** eventually own one or many Projects"* (see Legend in figure 5.)

Figure 5 shows the result so far, with PROJECT and PROJECT OWNER documented on the white board and the business rules notation discussed above shown in the legend. The business rules that have been agreed by the audience have been written for each end of the line. This is an initial fragment of the strategic model that will eventually be developed for Col 1 (What) Row 2 (Owner).

We will discuss the addition of BUDGET and its relevant notation in figure 6.

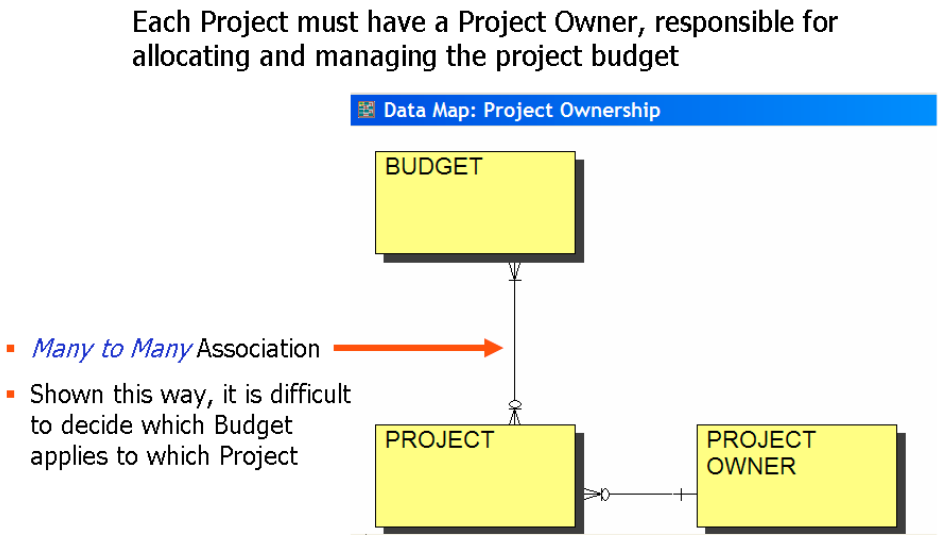


Figure 6 Strategic model fragment with BUDGET added

Referring now back to the list of nouns, BUDGET is added to the white board. On confirmation (from the facilitated modeling session audience) of the business rule that *"Projects have Budgets"*, a line is used to join these two boxes. The relevant business rule questions above are now asked at each end of the line between BUDGET and PROJECT, with the results added to the white board to show that *"a Project must have one or many Budgets"* and *"a Budget will cover one or many Projects"* (see figure 5). It is useful also to have a legend of the association degree and nature symbol meanings prominently displayed in the facilitated session, as in figure 5.

Referring now to figure 6, the facilitator explains in business terms that:

> *"It is hard to see from this business picture which Budgets relate to which Projects. We need to make this clearer by adding another box between these two boxes. This new box is typically named by combining both words – as PROJECT BUDGET. In fact the policy statement that we started with did refer to Project Budgets, although this does not always happen in such a convenient way."*

The result of this decomposition is shown in figure 7.

The facilitator confirms again the business rules for each crow's foot *(many)* that touches PROJECT BUDGET, as well as the *mandatory one* rule (**must**) touching each of PROJECT and BUDGET. These are verbally expressed using business terminology as business rules; technical data mapping terminology should not be used.

Notice that PROJECT BUDGET has been arrowed in figure 7 to show it represents the *Project Budget Management Activity*. The facilitator explains this principle in business terms by saying, that:

> "*Whenever we decompose a many to many association in this way, we are also focusing on the underlying business activity or business processes that are represented by the intermediate box."*

This is a principle that is not generally understood by experienced data modelers, who typically leave *many-to-many* associations in an enterprise model. This squanders opportunities to discuss the business meaning of the business activities or processes uncovered by decomposing *many-to-many* associations as illustrated in figure 7.

In business terms, the facilitator has now also introduced the main concepts of business-driven data mapping. By using non-technical terms, the managers and business experts in the audience now understand that:

- Boxes are *data entities* that are stored during implementation for later reference as data base tables.
- Lines joining related boxes are used to represent management controls, audit controls, governance controls, security controls, communication paths and reporting paths. These lines are called *associations.*
- Symbols on a line touching a box represent business rules that apply to the two boxes joined by the line. They are called association *degree* and *nature.*
- Association *degree* is represented by a crow's foot symbol for *one or many;* with no crow's foot symbol for *one.*
- Association *nature* is represented by a zero across the line for *optional* or **may**; a short bar across the line for *mandatory* or **must**; or a zero and a bar across the line for *optional becoming mandatory* or **will.**
- The result of decomposing a *many to many* line is the addition of an intermediate box to the picture, named from the two boxes it joins. This is called an Intersecting box.
- This intersecting box represents relevant business activities or processes, named initially by adding the suffix "Management Activity" to the intersecting box name.

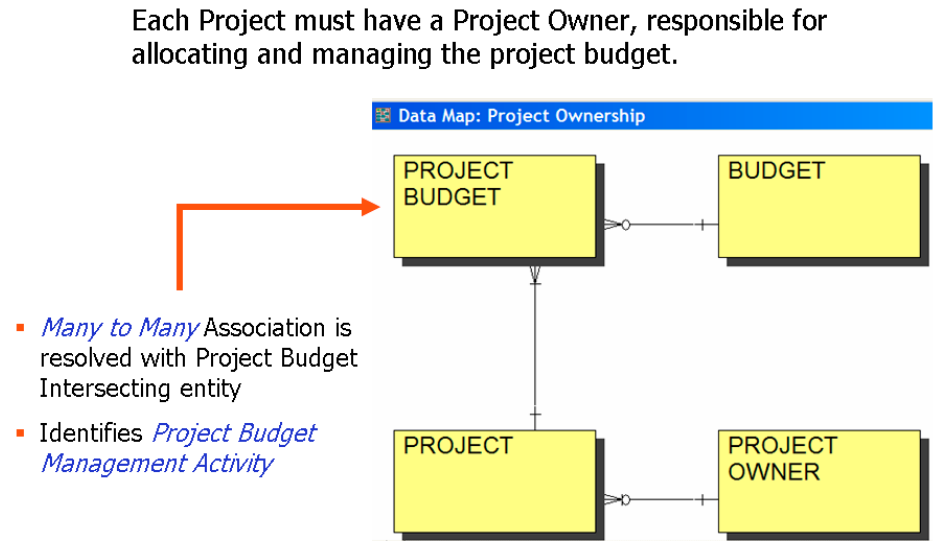


Figure 7: Strategic model fragment after adding the second planning statement

2. Project Management Policy

The facilitator now moves on discuss to the second policy from the planning statements of the Project Management Division. This is a statement of the *Project Management Policy*.

- Each Project must have a Project Manager, responsible for completing the project by the scheduled date, within budget.

Any new nouns in the above statement are listed by the facilitator on the white board. These are potential data entities for Col 1 (What) Row 1 (Planner) as follows:

PROJECT MANAGER

The facilitator has listed PROJECT MANAGER, but not *scheduled date* – as this is descriptive of a PROJECT. It will likely later be identified as an attribute of PROJECT. Attributes are not shown in a strategic data map in Col 1 (What) Row 2 (Owner). They are later defined using business normalization, during more detailed logical data modeling for Col 1 (What) Row 3 (Designer).

Figure 8 now shows the addition of PROJECT MANAGER (from the second policy statement) to the evolving Strategic Model. It shows the business rules that:

> *"A Project Manager may be responsible for one or many Projects";* and *"a Project must have a Project Manager"*.

In technical data mapping terminology this documents the *mandatory one* to *optional many* association between these two entities.

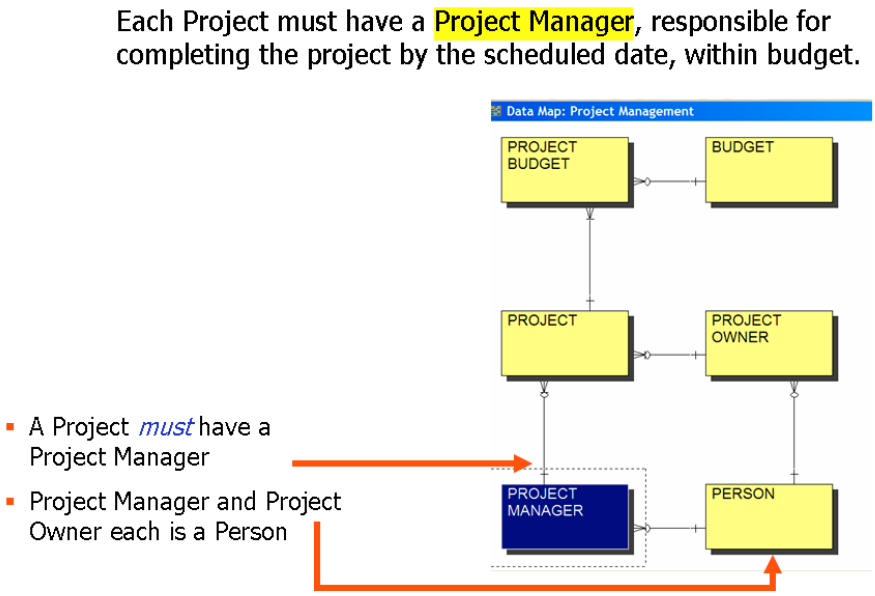


Figure 8: Strategic model fragment with the second planning statement added

Notice that PERSON has also been added in figure 8 for the two implicit business rules: *"a Project Manager is a Person"*; and also *"a Project Owner is a Person"*. Similarly: *"a Person may be one or many Project Managers"* – most likely over time; and also *"a Person may be one or many Project Owners"* – again over time.

3. Project Authorization Policy

The third Project Management Division planning statement is the *Project Authorization Policy*.

> Projects are only authorized that can achieve Project Objectives by the scheduled completion date, within budget.

New nouns in the above statement are now listed by the facilitator on the white board. These are potential data entities for Col 1 (What) Row 1 (Planner) as follows:

OBJECTIVE

PROJECT OBJECTIVE

The only new nouns are Objective and Project Objective. These two boxes are added to the business picture on the white board in figure 9. The *many to many* business rule between Objective and Project has already been decomposed in figure 9. The intersecting box *Project Objective* is explained by the facilitator as *"representing the Project Objective Management Activity."*

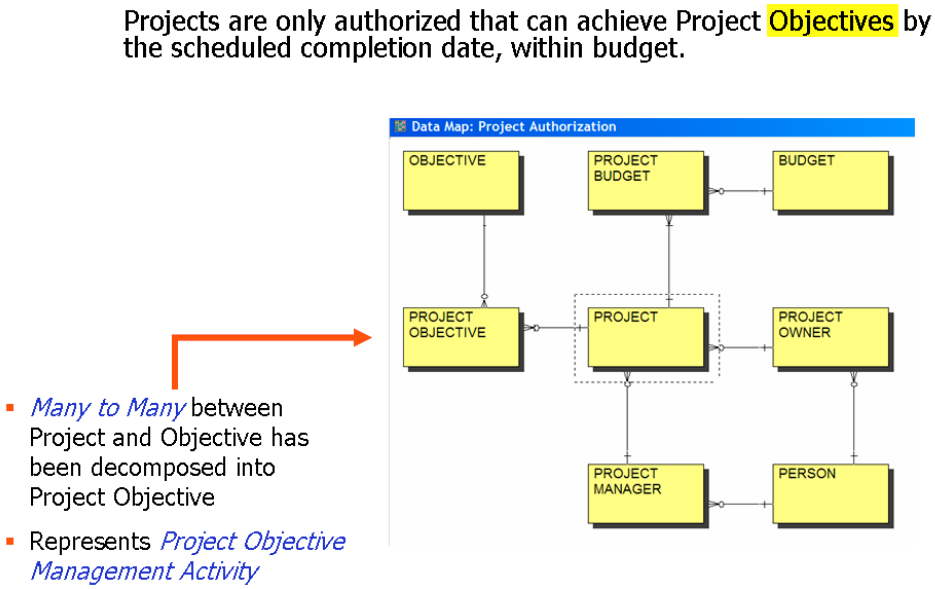


Figure 9: Strategic model fragment with the third planning statement added

Now that data mapping concepts have been introduced using business terminology and business rules, the business audience can also understand the technical data mapping terminology. Where appropriate the facilitator can now use relevant technical terms, relating the technical terminology back to the equivalent business terms. This way the business experts will be able to work actively later with data administration staff that may lapse into technical data mapping terminology from time to time.

In practice, in a facilitated modeling session the facilitator drives the session by using the relevant business statements as catalysts for discussion. *Do not answer these questions yourself as facilitator based on your knowledge of data modeling.* If you do, the strategic model will be viewed by the business audience as *your* solution; not theirs. Under no circumstances should you ever let this happen. You should only be a "mirror": acting to translate their words into a business picture. If you add your ideas to the model on the white board, they will lose interest and they will likely disappear in the next rest break.

During a live facilitated session, you will find that discussions may arise about the correct names to be used in different entity boxes. This is typically because of many synonyms that exist in most organizations. To resolve this terminology dilemma, each suggested term should be defined precisely by the audience. These definitions are unobtrusively captured by a scribe at the back of the room, using Microsoft Word (say) with a laptop computer for reference later to these definitions. Synonyms – *different words that all refer to the same thing* – can be discussed; a common word can then be agreed by the business audience. You will also uncover Homonyms – where *the same word is used to refer to different things*. Words that are unambiguous can then be agreed. These definitions are important to document, as they constitute the metadata that will later be used for implementation utilizing rapid delivery technologies based on XML, web services and Service Oriented Architecture (SOA) using BPM languages.

Deriving Project Plans using Entity Dependency Analysis

We will now discuss the steps involved in deriving project plans from the data map representing the strategic model that we have developed so far. These project plans represent the enterprise architecture portfolio to manage the rapid delivery of priority business activities or processes into production, typically in 3-month increments: derived based on the principles of entity dependency analysis.

It is important to re-emphasize the potential existence of business activities and processes that can be identified from a data map. These are based on the intersecting entities formed by decomposing *many to many* associations, as discussed earlier in relation to figure 7. I said then that many data modelers leave these *many-to-many* associations in an enterprise model. In fact, the decomposition of *many-to-many* associations is what constitutes the difference between an enterprise model and a strategic model. This enables the business importance of these business activities or processes to be considered by business experts and managers.

These intersecting entities are used to identify relevant activity, process or system names for Col 2 (How) Row 1 (Planner). By adding the suffix "Management", "Activity", "Process" or "System" to an intersecting entity name, the relevant Col 2 name can suggest itself, as shown in figure 10.

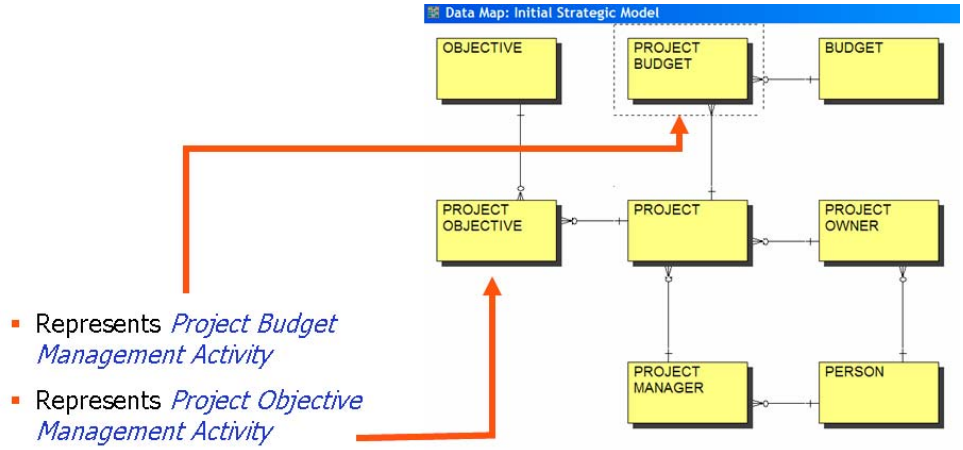


Figure 10: Business activities are identified from intersecting entities that are formed by decomposing many-to-many associations in a data map

This identification of activities, processes or systems defines the "List of Processes" for Col 2 (How) Row 1 (Planner) for prioritization by management. Priority activities or processes can then be used as vertical slivers that are sub-projects for rapid delivery of enterprise architecture, based on entity dependency analysis as discussed shortly.

The strategic model we have developed so far in figure 10 shows the existence of two activities:

- Project Budget Management Activity
- Project Objective Management Activity

Figure 11 now shows a fragment of the strategic data map for the *Project Budget Management* activity. Notice that a number is written close to each entity box. This indicates the sub-project phase number for the Project Budget Management activity, when that entity will be defined in detail to identify data attributes in logical data modeling for Col 1 (What) Row 3 (Designer).

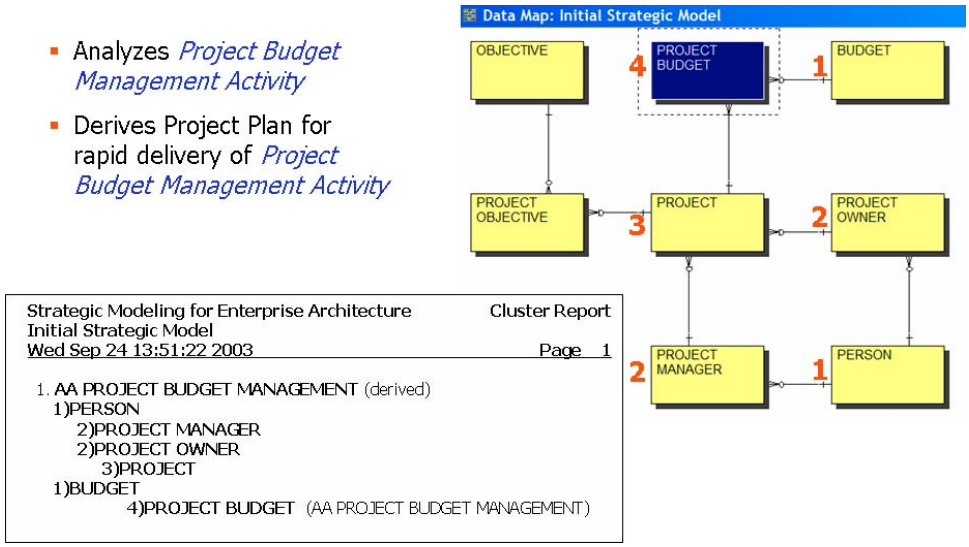


Figure 11: Deriving project plans from a data map

Figure 11 shows a project plan for that activity, derived using entity dependency analysis. This project plan represents a derived sub-project for implementation, called a "cluster". It indicates a vertical sliver for potential rapid delivery, if it is a priority. A phase number precedes each entity – separated from the entity name by a right bracket. Each higher phase number is indented one position further to the right in Outline format, so it can be read as a conceptual Gantt Chart.

The derivation of the project plan shown in Outline Format in figure 11 shows a fragment of the strategic data map for the *Project Budget Management* activity. Notice that a number is written close to each entity box. This indicates the sub-project phase number for the Project Budget Management activity, when that entity will be defined further, to identify detailed data attributes in logical data modeling for Col 1 (What) Row 3 (Designer). This derived project plan is based on entity dependency analysis. Although the project plan in figure 11 has been automatically derived by a modeling tool, the rules and the associated entity dependency analysis steps can be also applied manually. In figure 12 each entity required by the Project Budget Management activity is now highlighted. This diagram schematically shows the required entities that are needed by the vertical sliver that represents the activity. Notice also that each of the entity names is **bold**. The figure 12 comment states that *Project Budget Management is a common, independent activity that is reusable.* We will see this reuse shortly.

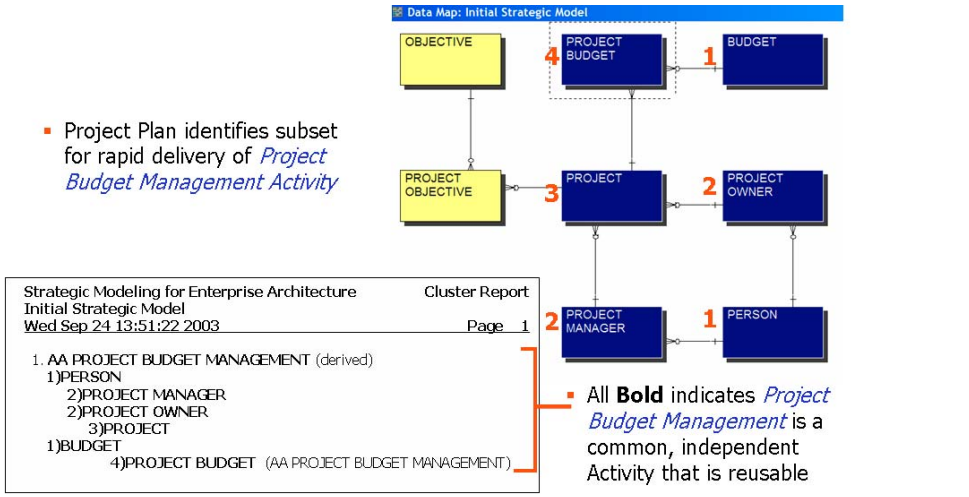


Figure 12: The derived project plan identifies the data needed by the relevant activity

Figure 12 also shows the derived project plan for the highlighted entities in that activity. This project plan represents an implementation sub-project for early delivery if required. A phase number precedes each entity – separated from the entity name by a right bracket. Each higher phase number is indented one position further to the right in Outline format. It can also be read as a conceptual Gantt Chart.

An important point to make about the derivation of project plans from data models is that these derived project plan clusters precisely define "how to build" the relevant databases and processes that are to be implemented rapidly as production systems. The defined phases are used to schedule logical data modeling sessions with relevant business experts who have detailed business knowledge to help identify the required attribute detail in specific entities.

Derivation of a Project Map for Rapid Delivery

We continue our discussion of the steps involved in deriving project plans from the strategic model that has been developed so far. We will see how a project map is derived to manage the rapid delivery of priority sub-projects into production, typically in 3-month increments.

I will now analyze the data map further to identify the data entities that are required by the *Project Objective Management Activity*. Notice in figure 13 that the intersecting entity representing this activity has been highlighted. The project plan for this sub-project cluster is also included in figure 13.

Notice that all of the entities required by the *Project Objective Management Activity* sub-project cluster are highlighted in **bold**, as we also saw earlier. But included in the project plan we also see that BUDGET and PROJECT BUDGET (for the *Project Budget Management Activity*) are also included – but are NOT BOLD. The comment in figure 13 indicates that this shows that "*Project Budget Management is a prerequisite reusable activity that is shared*". This prerequisite sub-project dependency is more apparent in figure 14.

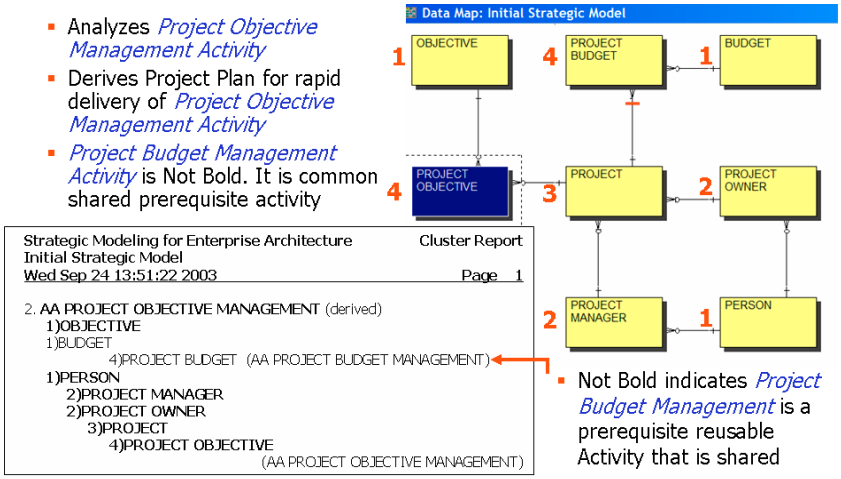


Figure 13: Derivation of project plan for project objective management

Figure 14 shows both the *Project Budget Management* and *Project Objective Management* sub-project clusters together. We can see now that the *Project Budget Management* activity has been reused in *Project Objective Management*. This reflects the mandatory business rule that a *Project* ***must*** *have at least one or many Project Budgets*. The analysis has applied this business rule to mean that *Project Objectives* cannot be managed effectively without also knowing the relevant *Project Budget*. This rule certainly is true for most enterprises.

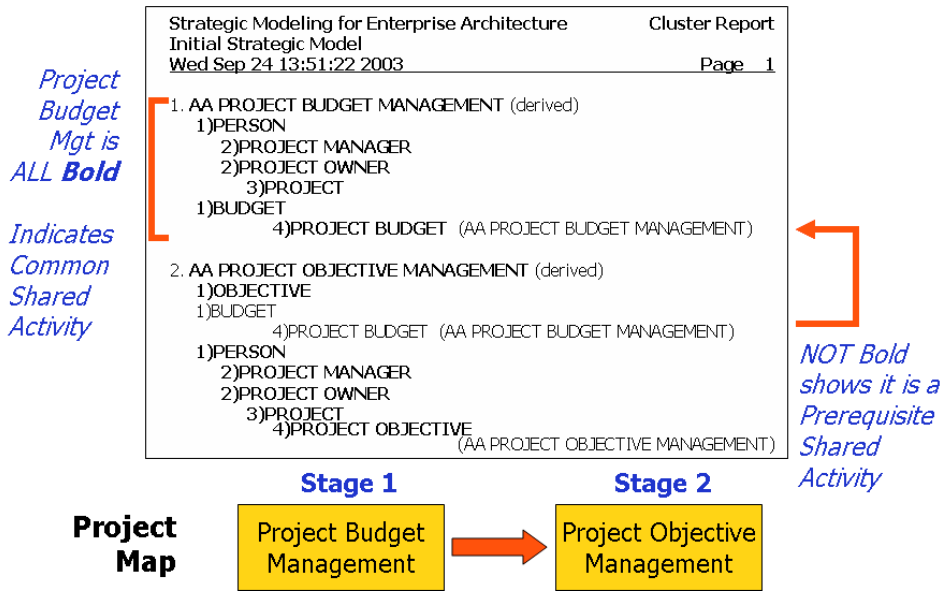


Figure 14: Derivation of project map for rapid delivery of sub-projects

Notice the Project Map in the bottom section of figure 14. Each sub-project cluster is shown as a highlighted (shaded) box. Each box represents all entities and their phases that are contained in the relevant cluster in the top part of the figure.

The Project Map displays the *Stage* in a larger project when the relevant sub-project will be implemented. We can see from above that *Project Budget Management* is a *Stage 1* sub-project; while we see that *Project Objective Management* is a *Stage 2* sub-project. We now know the order in which each sub-project should be implemented. Project plans derived in this way are very powerful. Consider the following bulleted comments:

- The Swedish furniture manufacturer IKEA has furniture showrooms in cities throughout the world. Its furniture is sold in kit form – for "Do-It-Yourself" (DIY) construction. With the enclosed diagram and instructions, the supplied components can be assembled into the finished furniture item.
- Previously in systems development we have built systems like putting together pieces of a jigsaw puzzle – but without a jigsaw picture. We now have that jigsaw picture. This is the strategic data map; it shows us how the data entities "fit together". We also have many sub-project clusters derived from a data map, which show us the phase sequence of entities to build each sub-project.
- And we now have project maps that show us how the sub-projects can be progressively built in Stages. We now see that project maps and project plans in clusters provide the unique *DIY Construction Kit* that has previously been missing. These can provide a *DIY Construction Kit* for systems development of your enterprise.

We have now covered the main principles of strategic modeling and the derivation of project plans and project maps from a data model. These principles can be used to derive and document the Enterprise Architecture Portfolio Plan (EAPP) for any organization … in only 20 days.

This article is based on the book "Enterprise Architecture for Integration: Rapid Delivery Methods and Technologies" (Artech House Mobile Communications Library, ISBN: 1580537138)

Fear of Intervention - How Subordinates Grow to be Entrepreneurs.

Willem van den Ende, willem@willemvandenende.com
Living Software B.V, http://www.willemvandenende.com/

Introduction

Increasing the effectiveness of workers in a software development organization can often be done by introducing some surprisingly simple practices. However, manager-subordinate relations and a history of less than pleasant management interventions can make workers to afraid of interventions to improve their effectiveness.

Using an allegorical story based on various experiences and Diagrams of Effects I will investigate actions that can be undertaken to empower someone in a subordinate position to improve his or her effectiveness.

As a reviewer pointed out, there are also actions that can be undertaken to make the manager more effective. Reading the story again some years after I first wrote it, I believe it contains pointers for both. Tom, the manager also changes his behaviour. And Rick, Toms' subordinate, learns to understand Toms behaviour better as the story unfolds.

I believe that if you want someone else to grow, it helps tremendously if you grow in parallel. Make an effort to change your own behaviour, show that you are making the effort, and be open in your successes and failures. Thus, people who grow together can get into a positive spiral, where successes of one person motivate the other(s) to try more. If a manager changes his or her behaviour, the effect is amplified 'downwards'. Watch how Tom changes the way he spends his time with Rick halfway through the story. Show, don't tell :).

Tom and Rick's history

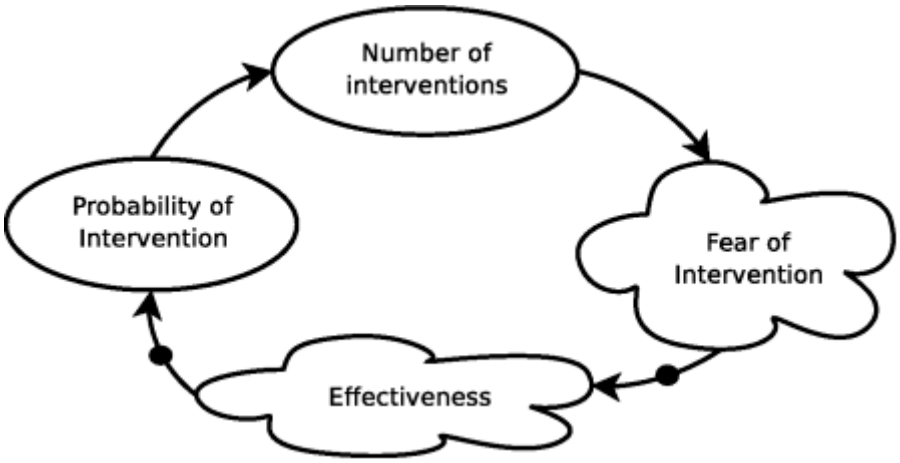
Tom is a manager who used to apply an authoritarian style of management. He is a senior manager with much more experience than his relatively young employees, such as Rick, who tries to grow from programmer to project leader. Using his experience this way has enabled Tom to make swift decisions so his teams are agile and make rapid progress. The number of Tom's subordinates has been growing rapidly, so he does not have enough time to be involved in all the details any more. Therefore, Tom wants Rick and his colleagues to behave in a more entrepreneurial fashion; they have to take risk, invent new business ideas, experiment with these new ideas and generally make more decisions on their own.

In the past Tom has experimented with giving subordinates more latitude. Sometimes the subordinates failed, so Tom had to intervene. For instance, Rick had hired some programmers who quarrelled with Tom a lot. Tom got irritated, so he fired them swiftly without consulting Rick first. Rick was not pleased when he heard most of his development team got the sack. Tom wasn't happy with this intervention either, he just felt he was driven into a corner and had no choice.

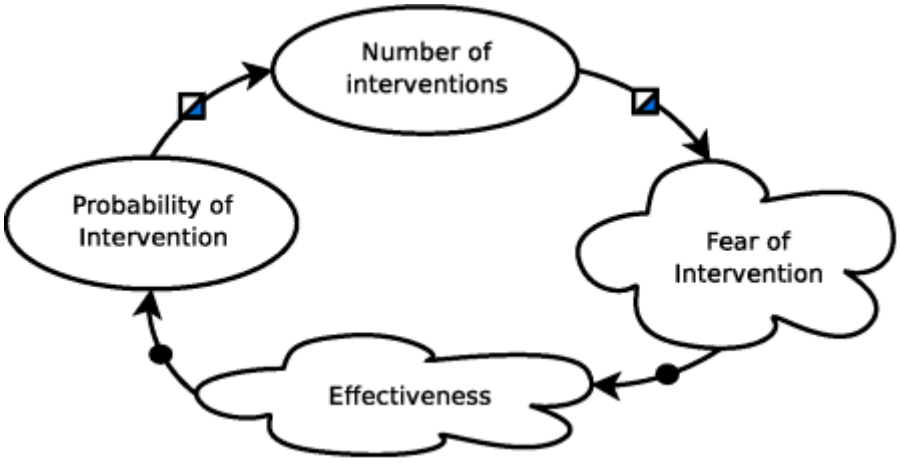
With a new process, plus training and coaching in several key skills, and more experience the subordinates could become more entrepreneurial, so the manager wishes to give it another try.

Now history becomes a problem. Rick is willing to take on more responsibility, but is afraid Tom might do more unpleasant interventions like before. This fear lowers Rick's effectiveness when carrying out his tasks independently: he procrastinates, speculates about possible

interventions, tries to second-guess Toms wishes. When Rick is ineffective, Tom is more likely to intervene, and with each intervention the fear of interventions with Rick and his colleagues rises. This gives us the vicious circle in the following diagram (The symbol ● denotes a negative influence, the absence of this symbol denotes a positive influence):



So, how can Tom and Rick escape from this spiral? Let us focus on two intervention points within the circle. Subordinates can accept an intervention as a signal that they still have something to learn, without letting an intervention result in growing fear. The manager can try to reduce the number of interventions. This results in the following diagram. We use the *management action with open choice effect* symbol ◪ to indicate that someone can choose to let the arrow have a positive influence, a negative influence or no influence at all:



So, both the manager and the subordinates have the choice to not let a history of interventions influence them. Growing this kind of trust takes time and effort from both parties.

Maybe there also is a way to increase the effectiveness of the subordinates, even if we disregard the fear for a moment. The introduction also mentioned training, coaching and experience. Do they have an effect? They probably have, but they do not influence the way people work immediately. Some training is easier applied than others, and some experience only works if one makes the same mistake twice (or even more). Reflecting on experience and learning new behaviour takes time as well.

Before we go onto the next diagram, let's elaborate a bit on possible training, coaching and using the growing experience of Rick and his colleagues.

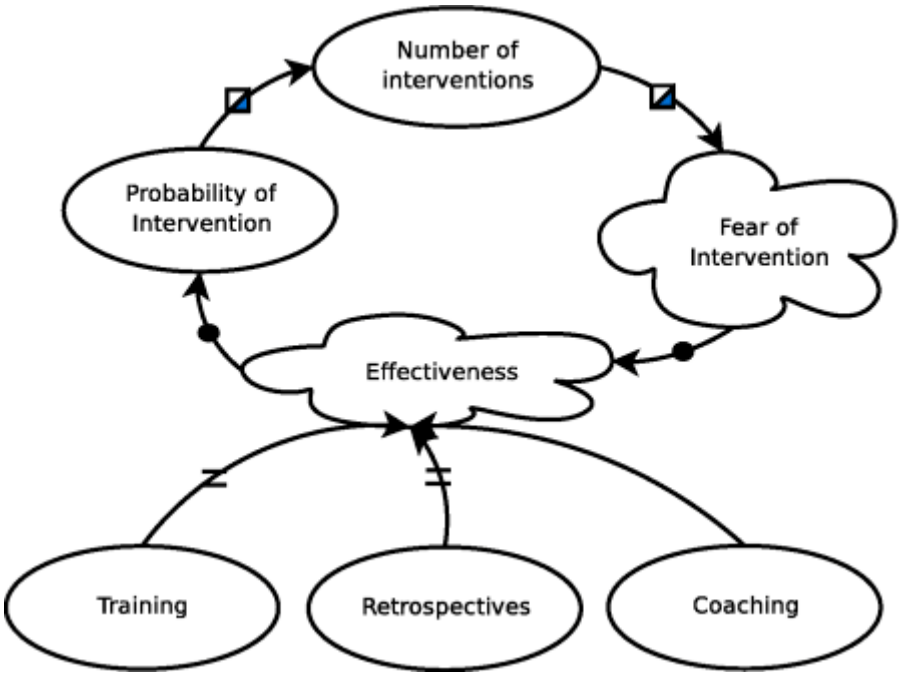
Possible training includes training on communicative skills, sales, and planning. This usually requires Rick to spend some days outside the company in training. After that, Rick needs some time to integrate what he learned into his daily practice. Another option is training on the job by an external coach or by experienced people inside the company. This way, Rick can learn while providing value to the company at the same time.

Tom offers Rick the benefits of his expertise. Tom has many years of experience, which he uses to make decisions. Tom can teach some of his experience Rick and others, so they can make better decisions on their own and Tom can delegate responsibility easier. Rick also decides to take some courses on sales, and Tom and Rick decide together to hire Steve, an external coach to help them introduce a more agile software development process in Rick's projects, so Rick and his developers can improve their effectiveness faster.

Working together on concrete projects is a great way to learn, using the experience gained in each project to improve the next project. With their new process, Rick and Tom can run projects of a few months maximum. With Steve's help they run a retrospective after each project, so they and other participants such as customers and programmers can all learn from their experience and increase their effectiveness. Retrospectives gradually increase effectiveness, it is not something that happens overnight.

After a few one-month projects Tom and Rick get a better grip on planning inside projects and program management around projects. Projects start to run smoother, so effectiveness increases. Rick's fear is only slowly decreasing, since the memories of some interventions are still quite painful.

So training, on-the job coaching and regular project evaluations have a positive effect on effectiveness and rising effectiveness will reduce Rick's fear of intervention, which in turn will make him more effective - a virtuous cycle. But, as indicated before, training and evaluation have influence after some time, which we indicate with the delay symbol = .



Training, evaluations and coaching only work, if they combined are stronger than Rick's fear of intervention. For instance, if Steve were to leave now, it is questionable if Rick's effectiveness would improve. What can be done to reduce Rick's fear more directly?

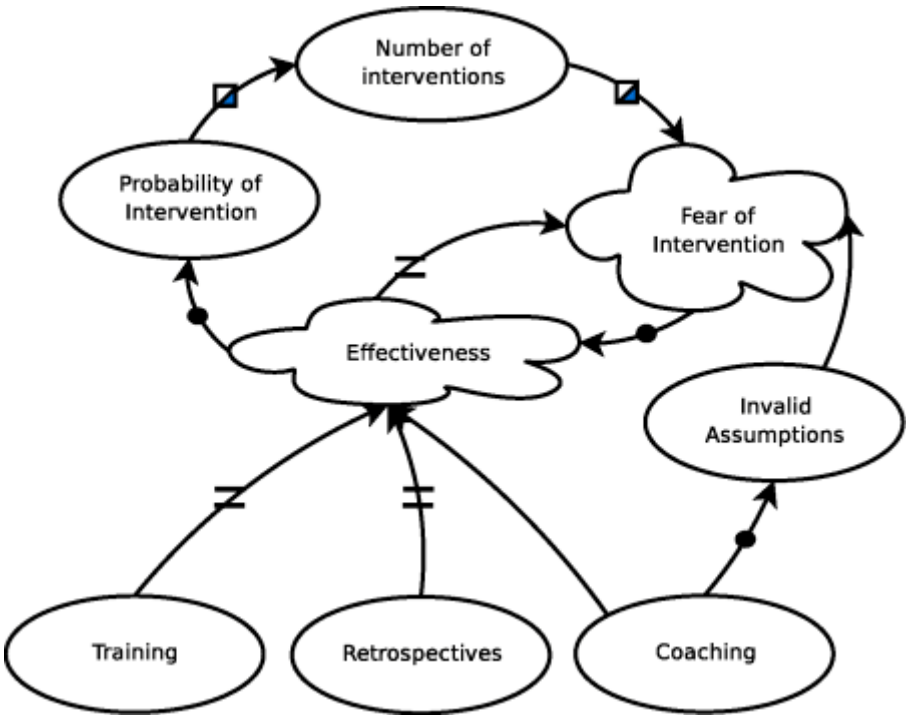
Tom is trying to make the switch from a 'push' model, where he gives orders to people, to a 'pull' model, where people can ask him for advice when they need it. For Rick and Tom this means they have to keep a delicate balance. Rick can ask questions but has to make his own decisions. Tom has to make a big effort to prevent his suggestions from being perceived as orders. For instance, sometimes Tom will keep his opinion to himself, in order to give Rick space to do his own thing, Rick might erroneously perceive this as disinterest by Tom... One way out of this dilemma is for Tom and Rick to make their assumptions explicit, share their fears, hopes and expectations.

One time, Rick and Tom have a meeting where Steve, the coach, is also present. Steve notices Rick is getting angry at Tom. Tom tries to coach Rick and get him to explore several options for the process to use in their next project. Rick assumes Tom is trying to tell him what to do, and Rick starts to resist Tom, because he wants to do the project differently. Steve decides not to say much, but just ask Rick a few questions to make Rick aware of his assumptions. Even though Tom had a strong opinion on the issues, the final decisions would be up to Rick. After a while Rick's' anger dissipates, and he and Tom have a constructive discussion.

After the meeting Rick asks Steve why he didn't say very much and use his weight as an 'process expert' to influence the meeting. Steve says, that like Tom he has a strong opinion on how he likes to run a project, and that he thinks differently about it than both Tom and Rick, but that it is up to Rick to run the project and find out, as much as possible through his own experience, how he likes to run a project. Steve says he is glad to give his opinion and advice, but only if Rick asks for his help. From his experience Steve knows, that if he gives advice to someone who did not ask for help, there is a big chance his advice will be ignored, or even

worse, that the person he is trying to help feels insulted or threatened, and does exactly the opposite of what Steve suggests.

Steve explains to Rick that Tom was trying to do the same in the conversation they just had, give advice only if someone asks for help, and not use his weight as an expert and boss to give orders. He advises Rick, to be more aware of his assumptions, and validate them more often, so Rick does not become angry or afraid because he thinks Tom is intervening unpleasantly again, when Tom is only trying to help Rick.



The diagrams tell us Tom can delegate more responsibility to Rick and his colleagues. Rick can grow to assume more responsibility and behave more entrepreneurial. To get there, there are some things that need to be done: Tom reduces the amount of interventions, and invests time in coaching Rick and his colleagues. Tom and Rick reserve time and money for training, evaluation and coaching. Rick and his colleagues realize that when they are becoming more effective, there will be less interventions, and welcome an intervention, if any, as an opportunity to learn. Rick learns to validate his assumptions before he grows afraid. Rick also takes time to follow training and coaching, and he makes sure everyone in his projects participates in evaluations. But wait a minute, now Rick is reading this, he realizes something....

Rick and Tom are not that different

Rick used to be a programmer, now he is leading projects. Rick is an expert programmer, so his favorite way of leading a project is looking at the code and seeing what happens there. If he doesn't like some of the code, he makes a suggestion to the programmers to do it differently. This works well, he thinks, because with his experience the team can make quick decisions, make rapid progress and be agile.

Now Rick has a problem: the process improvement he is doing together with Tom, all the training courses, and organizing evaluations take up a lot of his time. Now he wishes to do the same thing Tom does: delegate some of his responsibility to his programmers, let them make more decisions on their own. But every time he tries to let programmers do their own design, either nothing happens, or they are constantly at his desk asking questions. What can he do?

After some time Rick realizes he has the same problem Tom has. His 'design suggestions' are taken as orders by the development team. Rick rarely explains the rationale behind his decisions, because his experience enables him to use the patterns he knows intuitively. The programmers think Rick is really clever, so they could never design like Rick.

So, what can Rick do? Apply the diagram to his teams and invest time and money in training, coaching and retrospectives, and let the programmers gain confidence by letting them be more effective.

Now Rick comes full circle. Rick has to treat his programmers in the same manner Tom and Steve are treating to him: give advice only to programmers that ask for his help, trust them and give them as much space as possible, allow them to make mistakes and learn from those. If a programmer gets angry or frustrated with Rick, Rick tries to coach the programmer, to find out which assumptions are in the way.

Luckily, Rick and Tom have spent time learning together. Seeing Tom go experience the transition to a facilitating, 'pull' management style might make it easier for him to change his behaviour too...

Conclusion

Using the diagram of effects for some of the observations in Tom and Rick's story enabled us to find intervention points to break the vicious circle Tom and Rick found themselves in. Training, regular retrospectives and coaching are actions that make Rick more effective and less afraid of interventions.

Hopefully these diagrams inspire you to find your own. Maybe you recognize yourself too in Tom and/or Rick.

I used this diagram to understand the fear of intervention, when I encountered it again at a new client. I had been "there" before, one has to make the same mistake twice (or more) to understand it...

I thank Pascal van Cauwenberghe, Marc Evers, Erik Groeneveld, Nynke Fokma and Peter Schrier for their comments on the first release.

Further Reading

Quality Software Management - I - Systems Thinking by Gerald Weinberg

The Fifth Discipline by Peter Senge

A Methodology to Support Software Release Decisions

Hans Sassenburg
SE-CURE AG, www.se-cure.ch

Software is everywhere and has become a major worldwide industry. We find software embedded, for example, in watches, coffee makers, cars, televisions, airplanes, telephones, reservation systems, and medical equipment. Software not only pervades a multitude of products, but also is an important corporate asset, and demand is increasing. Yet software projects are characterized by schedule and budget overruns and the delivery of unreliable and difficult to maintain software products.

Despite an exponential increase in the demand for and dependence on software, many software manufacturers exhibit unpredictable behavior. It is sometimes difficult to determine, for example, a software product's release date, its features, the associated development costs, or the resulting product quality.

Without knowing the release date, a software manufacturer experiences difficulty planning product promotions, customer training, and maintenance support. Resource utilization across projects may become inefficient and difficult to manage when projects fail to meet schedules. Customers have difficulties planning for the introduction of new software into their organizations when a scheduled release date is missed.

The exponential growth of software suggests that end-users will be exposed to more defects if the software industry is not able to exponentially improve its defect potentials and removal efficiencies. Combined with increased competition and smaller market windows, software manufacturers likely will be exposed to higher levels of uncertainties when releasing software. The decision to release a software product will become an even more complex and important decision.

A Software Release Methodology Supporting Strategic Value

Existing methodologies, models, and standards reveal limited guidance for structuring software release decisions in a methodological way to support strategic value. A decision has strategic value when it has the potential for large prospective financial losses to a software manufacturer or its customers or end-users. Software release decisions often have strategic value due to high costs for reversing the decision. Prospective loss outcomes also may arise long after the decision has been made, for example, in cases where liability leads to lawsuits.

A software release decision can be seen from different perspectives:

- Maximizing behavior. A software release decision is a trade-off between early release to capture the benefits of an earlier market introduction and the deferral of product release to enhance functionality or improve quality. If a software product is released too early, the software manufacturer incurs post-release costs of later fixing failures. If a software product is released too late, the additional development cost and the opportunity cost of missing a market window could be substantial. These two alternatives need to be compared to determine which alternative maximizes economic value.
- Optimizing behavior. A release decision deals with the difficulty of verifying the correct implementation of functional and non-functional requirements. How much testing is needed? Software manufacturers must find the optimal level of information because information has its price in cost and time. In practice, cost and time constraints will normally

be present in retrieving complete and reliable information, so this search for information should be taken into account as an economic activity. This leaves the software manufacturer with the problem of finding the optimal level of information where marginal value equals marginal costs and marginal yield is zero. This optimal level is difficult, if not impossible, to find.

- Satisfying behavior. Decision-making in the real world is often unstructured and normally involves various stakeholders who may, for example, have reasons to release a system or software product due to political or business pressures even though they know it still contains defects. A study of spacecraft accidents, for example, reveals that, although inadequate system and software engineering occurred, management and organizational factors played a significant role, including the diffusion of responsibility and authority, limited communication channels, and poor information flows [Leveson, 04].
- Decision Implementation. A decision is only considered successful if there is congruence between the expected outcome and actual outcome, which sets requirements for decision implementation. In practice, there are many obstacles to the successful implementation of almost any decision, including:
  - the reduced importance of a decision once it is made and implemented
  - the control of the outcome of a decision by stakeholders not involved in its making
  - the development of new situations and problems to command the attention of the decision-makers once the choice has been implemented

Research Findings

This research reviewed these different perspectives from both a theoretical and an empirical point of view by studying practical examples. The results helped to frame a proposed methodology, called release decision methodology, to address a software release decision from different perspectives. (See Appendix [Sidebar].) The methodology, consisting of a coherent set of practices, combines insights from economics, software management, and social psychology disciplines.

Studies in three different organizations validated the methodology. One participating organization is a leading global financial services company that provides financial services and products to retail and business markets. Services include insurance, pensions, occupational health and safety, asset management, investments, leasing, real estate, venture capital, and mortgage finance.

The research examined a project in this organization's IT department, which develops custom systems for internal and external use. The initial estimate for the schedule was 10 months and for the pre-release cash outflows, Euro 15M. Budgets were reserved for the technical infrastructure and post-release cash outflows for maintenance and exploitation. During the first months, the project encountered several setbacks: technical problems surfaced and the development budget turned out to be too optimistic. Progress control was lacking, mainly through the absence of clearly-defined milestones or quality gates. These problems increased, and in November 2001, the project was re-defined. Both Senior Management and Marketing exerted pressure on the Product Development Team to release the product as soon as possible. However, the team was faced with an unstable product under test and had to use a veto several times to postpone a scheduled release date. When the product was released, uncertainty was high because many known problems were not resolved (although not considered critical), and the organization judged that continued testing would reveal more defects, including potentially critical ones, that could severely hamper the correct functioning and stability of the product.

After the product was released, a special task force assumed responsibility for temporarily performing corrective maintenance activities. This team needed more than a year to resolve the known and newly-detected defects. Despite the original requirement to develop a maintainable product, the organization decided in 2004 to start a pre-study toward a totally new product to replace this product because corrective maintenance and functional enhancements proved difficult and costly. In other words, the early release of the product saved the organization additional testing cost, but the post-release maintenance cost turned out to be significantly higher than expected. A retrospective review of this project using the software release methodology enabled this organization to assess the project from a release decision point of view.

Figure 1 illustrates how the organization scored on the identified practices in the methodology. Lack of a product development strategy (release definition) and lack of information as input to the decision-making process (release information) led to a poorly structured release decision-process without consensus among the stakeholders involved (release decision). Sufficient financial resources "saved" the organization on the short-term by patching the released software.

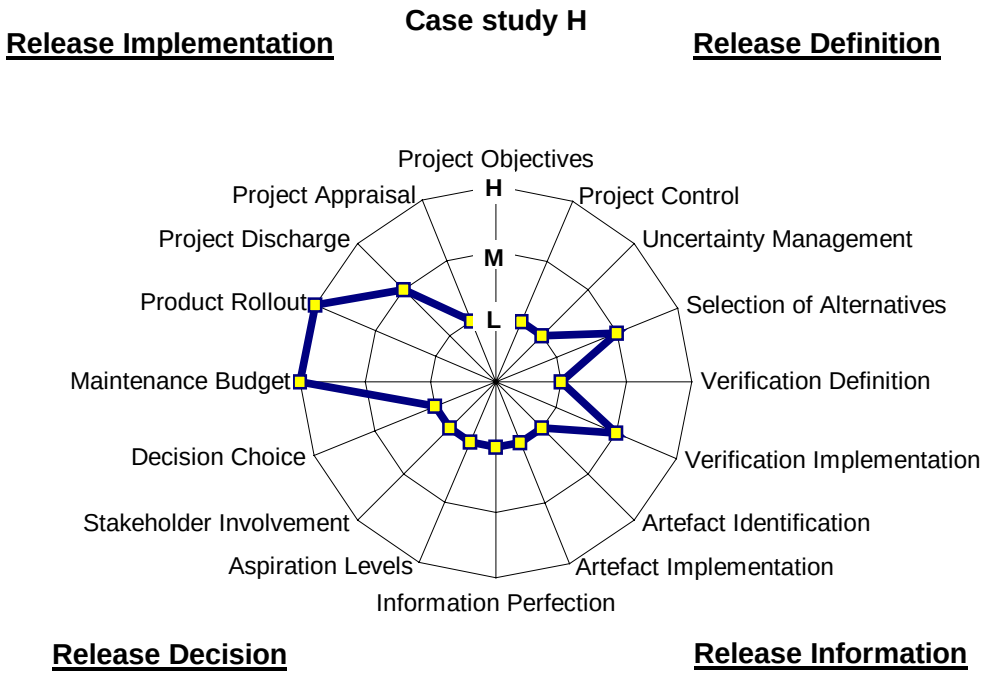


Figure 1: Radar Presentation of Case Study Results.

Based on these validation results in a practical setting, the software release methodology displayed a descriptive and a judgmental character, and it can therefore support understanding, analyzing, assessing, and improving the capability of software manufacturers in this problematic area, at least in the studied environments. Currently, ongoing research in software manufacturer environments is under way to study the effects of applying the methodology at the start of projects to proactively aim for release decision success.

Reference:

Leveson, N.G. "The Role of Software in Spacecraft Accidents." *AIAA Journal of Spacecraft and Rockets, 41, 4,* 2004.

For further information, see the full study at
http://dissertations.ub.rug.nl/faculties/eco/2006/j.a.sassenburg/

Appendix: Release Decision Methodology

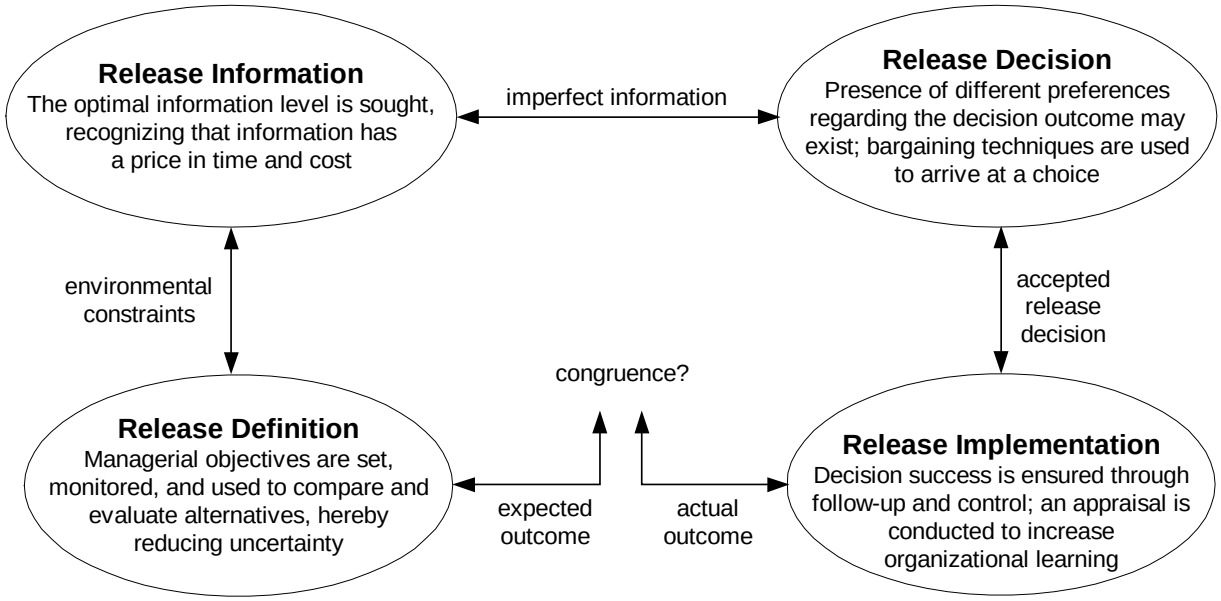


Figure 2: Release Decision Methodology.

In the release decision methodology framework, four areas in the software release decision-making process are distinguished, each addressing the process from different perspectives. A process area is defined as a cluster of related practices that, when performed collectively, achieve a set of goals considered important for establishing process capability in that area. Each process area consists of four relevant practices, describing "what" is to be accomplished but not "how." Through this approach, the descriptions of practices still offer the possibility for interpretation and customization to the external market environment and to internal strategic and functional characteristics of a software manufacturer organization.

Identified process areas are (see Figure 2):

1. *Release Definition*. Decision-making is mainly viewed from a quantitative perspective, assuming that information is near perfect: complete and reliable. It emphasizes the *maximizing behavior* approach with emphasis on mathematics, economics, and statistics. In software release decisions, decision-making from a quantitative perspective is concerned with the definition and control of a product development strategy setting the managerial objectives with their priorities and ensuring they are attainable. The availability of a product development strategy enables the comparison and evaluation of different release alternatives, answering the question: which alternative maximizes economic value?
2. *Release Information*. This process area is concerned with the search for alternatives during product development, for example, the identification and collection of information that is needed to compare and evaluate different release alternatives. This search is derived from the formulated product development strategy. Decision-making is also viewed from a quantitative perspective, but with the recognition that information is imperfect in the sense that not everything can be expressed in numbers and that information has its price in time and money. For this process mathematics, economics, and statistics still play an important role, but the maximizing behavior approach is extended with an *optimizing behavior* approach: what is the optimal volume of information? Insufficient information increases uncertainty and hampers the decision-making process, whereas too much information is a waste of scarce resources. There is an optimum above which the cost for searching for more information exceeds the benefits.

3. *Release Decision*. Decision-making is viewed from a psychological, sociological and socio-psychological perspective, addressing factors that influence individual and group behavior. It recognizes the imperfections of information, and stakeholders involved in the choice will possibly have different preferences with respect to the decision outcome; an open decision-making process. The challenge is to use a judgmental strategy to reach a decision outcome that meets the formulated objectives and is agreeable to all stakeholders involved. The concept of optimizing behavior is extended with a *satisfying behavior* approach: which outcome satisfies the needs of all stakeholders involved?
4. *Release Implementation*. Decision-making is viewed from an implementation perspective once a decision has been made and is implemented, assuming a successful decision requires follow-up and control. For software release decisions, it is necessary to identify the factors that ensure congruence between the expected and the actual outcome. To increase organizational learning, the decision-making process and its outcome should be evaluated.

METHODS & TOOLS is published by **Martinig & Associates,** Rue des Marronniers 25,
CH-1800 Vevey, Switzerland Tel. +41 21 922 13 00 Fax +41 21 921 23 53 www.martinig.ch
Editor: Franco Martinig ISSN 1661-402X
Free subscription on : http://www.methodsandtools.com/forms/submt.php