STRATEGY BASED ERP-SYSTEM IMPLEMENTATION APPROACH

© 2017 г.   D.Yu. STEPANOV

Moscow technological university (MIREA)
mail@stepanovd.com

How to make project of ERP-system implementation be under control? First idea coming to mind is predicting each step, knowing ways how to perform next steps and estimating weak and strong points of each way. Ideally manager must just combine methods together covering all activities within the project. Thus, we are talking about assets or accelerates making implementation project into order by using proven techniques. But which implementation areas can be covered? The answer is in this paper.

Objective

The objective of this paper is considering basic strategies to make ERP-system implementation process be trackable, foreseen and simple as much as it's possible.

Implementation project decomposition

To integrate people resources, work activities, deadlines, costs and any other project KPIs, manager should have strong understanding what to do. Otherwise, project activities will be never aligned with schedule and delivered. According to definition, system contains elements cooperating with each other and having final goal. Taking it into account the first step for simplification of implementation project is decomposition. Thus, experts suggested to split project by levels, stages, activities and tasks. [1]. Obviously, each task becomes more trackable, controllable and manageable. Uniting activities into groups will allow applying different methods to make them be delivered. Let's clarify what is strategy. Strategy has description of the way how specific task will be fulfilled. There is also explanation why suggested method is most suitable. However, there is no step-by-step instruction to implement proposed method.

Analysis strategy

Let's start from analysis strategy, defining the way how requirements would be identified and write down. Oral meetings and calls allows online interacting with a customer. Prototyping and workshops are about to demonstrate already existing information system to gather requirements (tab. 1). If it's needed, document flow and supporting documentation can be analyzed. Moreover, to find out current way of working there is possibility to watch how employee is doing his job during the day.

Design strategy

Once requirements identified and prioritized, they must be designed. Business Control Model (BCM) allows to show how company is working in general, ARIS Value Added Chain Diagram (VACD) illustrates a sequence of most important processes, while Integrated Definition 0 (IDEF0) enriches VACD by means of restrictions and resources. They are so-called high level graphical notations to project business processes. If more detail description of processes needed, then low level notations can be used. They are ARIS extended Event Process Chain (eEPC) to simulate processes using initialization events, responsible, documents and applied information system, Data Flow Diagram (DFD) is to show document

lifecycle, Integrated Definition 3 (IDEF3) is to demonstrate processes from time dependence point of view and swim lane based methods like UML Activity Diagram (AD), BPMN Swim Lane Diagram (SLD), Cross Workflow Diagram (WFD) [2]. Using such notations processes can be projected in AS-IS and TO-BE states.

Table 1

Strategy and methods of ERP-system delivery

Strategy	Method	Advantage	Disadvantage
Analysis	**Oral meeting / interview**	Face to face discussion	Time consuming
	Web meeting / call	Accessible, cheap	Misunderstanding
	Workshop*	System interface demonstration	Client cannot try application
	Prototyping*	Customer can test system	Expensive
	Documentation analysis	Overview of all processes	Often not allowed
	Watching operation execution	Real way of working	Time consuming, limited number of processes to check
Design	High level notation BCM	General overview of application	Demonstration purpose only
	High level notation ARIS VACD*	Quick modelling	Only for high level overview
	High level notation IDEF0	More attributes for process description	Time consuming
	Low level notation ARIS eEPC	Event based description	Time and efforts consuming
	Low level notation DFD	Data lifecycle overview	Implicit process description
	Low level IDEF3	Process dependencies	No responsible available
	Low level UML AD*, BPMN SLD, Cross WFD	Modelling processes per responsible	For Cross WFD it's process simplification
	Modelling in AS-IS* & TO-BE	Illustrative	Time consuming
Authorization and roles	**Based on business roles***	Logical	Hard to maintain
	Based on required transactions	Easy to maintain	Hard to track changes
Technical infrastructure	**Multilevel technical landscape***	Safe	Requires more efforts to support
	Single level technical landscape	Cheap	Can ruin production system
Change	Process change	Usage in training instructions	AS-IS, TO-BE modelling required
	Technology change*	Can be utilized in training	AS-IS, TO-BE modelling required
	People change	People resource planning	Estimation can be wrong
	Knowledge change	Training requirements identification	If no up-to-date user instructions, then estimation can't be done
	Structure change	People reallocation	Can be over/under estimated
	Full change scope	All change aspects considered and covered	Time consuming
Development	Customer development	Less expansive, no resource required	Conflicts if bad project climate
	Supplier development*	No conflicts	Require more resources
	External development	No resource required	Misunderstanding can happen
Migration	Single production migration	Less efforts	No approach for test data preparation
	Based on V-model*	Iterative data migration	More efforts and time needed
Training	End user training	Direct knowledge transfer	Time consuming, geographical restrictions
	Key user training*	Further end user training by key user	Some system details can be missed or misunderstood
	External training	No local resource involved	Expensive, no local specific can be trained
	Web training	Accessible, cheap	Misunderstanding
	Self-training	Almost no efforts needed	Low motivation
Testing	Single testing in form of system, integration or end to end test	Less efforts and time needed	Some integration points can be missed
	Based on V-model*	Full scope of system testing	More efforts required
Cutover	**Sequential***	Less efforts for support	Business can stop if system disaster
	Parallel	Safe	More efforts required
	Trial operations	More time for testing and training	Much more efforts for support
Support	**Multilevel support***	Progressive knowledge transfer	Expensive
	Based on support period	Easy to establish	Sharp transfer of responsibility

Authorization, infrastructure and change strategies

Authorization to application can be granted upon business roles, that is when all business needed transactions are assigned to single role and role is assigned to a user. However, it's also possible to assign certain business transactions to user directly. Technical landscape for information system can be organized upon classical three level approach or by means of combination of these levels. The approach means there will be three dedicated systems available, they are: development, quality assurance (QA) and production. Once modification of information system needed, then it should be updated sequentially in development and QA systems before changes will be transported to destination production system. Change strategy usually covers questions related to organizational structure, positions and personal responsibility changes once information system delivered. Process change means business process reengineering, people – increasing or decreasing number of employees, technology – the way how transaction will be executed in applied information system, knowledge – trainings to be arranged to cover technology change, organizational structure – altering corporate structure, enterprise management – using new approaches to manage enterprise.

Development, migration, training and testing strategies

Often implementation of corporate information system requires modifications. Thus, standard functionality of the information system must be enriched by customer developments. Such modifications can be developed by client programmers, supplier developers or any other third party involved in implementation project. There are different implementation projects, for example, implementation from scratch or roll out. Depending on the project type, data migration strategy can also vary. Single migration approach is used, when data should be migrated only once and directly to production environment. V-model based method is also relevant, according to this method data must be migrated sequentially to different systems increasing data volume per iteration, finally 100% of data will be migrated to destination system. To cover knowledge change trainings must be organized and executed. Training strategy can include end user courses, when participants are end users unlike key user trainings, where key business persons are lectured. Courses can be lectured internally or externally, that means by local lecturer or from third party. Users can be trained in classes, via internet or by themselves. Testing strategy describe the ways of information system will be tested. There are unit (exploring certain program modules), system (testing all modules of entire system) and integration tests (examining only integration points within systems). By acceptance test most experts means end to end testing of business processes against unit, system and integration tests. Classical V-model can be interpreted as a set of various tests. Thus, V-model includes unit, system, integration and acceptance tests.

Cutover and support strategies

Cutover strategy can be organized different ways. Sequential cutover implies once go live happened, all historical systems must be switched off or set in read only mode. Parallel cutover envisages handling both new and old information systems until it would be clear newly implemented system is reliable. Once it's clear sequential approach will be taken. Trial operation method denotes combination of parallel and sequential approaches. However, unlike pure parallel approach trial operation method takes place much earlier at a stage of testing but not go live. Once go live happened, implemented information system must be supported. If after going live support is pure responsibility of supplier, then we are talking about support period approach. Nevertheless, if responsibility is shared between customer and supplier experts, then it's multilevel support approach. The approach indicates levels of support, for instance, if there is open question, then it's customer key user responsibility to answer, simple system reconfiguration required – customer support team enabled, system bag – supplier side will be involved in defect solutioning [3].

Conclusions

Having described methods, let's summarize proposed approach. Most preferable methods were marked in bold in tab. 1 The most effective of analysis techniques are oral

meeting as well as prototype demonstrating. First one allows discussing all issues company have, last one – demonstrating system not spending too much time. For designing ERP-systems ARIS VACD notation does not require much efforts and it's justified for high level overview. However, for low level description UML AD is suggested due to necessity of business processes details and responsibility tracking. There will be a lot of users handling ERP-system, in this case authorization method based on business roles can be suggested. Yes, it's hard to maintain business roles, however it guarantees visibility and trackability of granted access. When there is high volume of transactional data in the system, then any modification requires a lot of regression testing to escape system disaster after updates. The only multi-level technical landscape allows such kind of testing to ensure changes do not ruin application. Business change activities covers all changes within enterprise, however the most critical is technology due to nowadays transactional data is stored electronically in information systems, moreover implicitly it takes into consideration also business processes and people changes.

To localize information system modification will be needed. Application changes can be done by supplier while build stage. Of course, it will increase expenses, however if there are lots of system updates, then it's preferable to have local programmer. Moreover, developer can be involved only for a phase of implementation project but not all stages. One of the task during implementing ERP-system is training end users how to deal with system. If there is no available resources or end users are allocated geographically, then key user training is most effective. V-model is used often as a basis for testing and data migration. It allows testing application from different aspects and requires involvement key and end users. Thus, future system users already learn the system before going live. V-model obliges incremental data migration for each testing, thus at least three migration rehearsals take place before production cutover. Due to it most typical and critical issues are caught before final data migration. Sequential cutover to use newly developed information system means blocking all other systems except implemented one. Form business point of view, this approach is very risky, due to system can just not be working, then all business operations will stop. However, this method saves much efforts and make persons take implemented information system more seriously. Moreover, according to implementation methodology each production start must be supported by continuity, contingency and disaster recovery plans. Thus, there is clear understanding how to manage if any issues.

Once system went live, it must be supported by customer. Knowledge should be transferred from project team to client employees. The only way how to do it progressively is organizing multilevel support desk. Starting from integration testing, customer employees will be involved in user support and defect resolution. Thus, test per test they will take enough knowledge to resolve most issues after post go live period. Mentioned above methods form implementation strategy, when each participant knows what and how to do. Strategy should be prepared before implementation project started to estimate efforts more accurate. Typically, each implementation stage has lessons learnt session, where all raised issues are discussed to escape them next phase. Strategy based approach for ERP-system implementation or so-called delivery plan is a basis of project as it provides clear vision which activities and efforts required.

REFERENCES

1. TOGAF Version 9.1. A pocket guide / Andrew Josey etc. – UK.: Van Haren Publishing, 2011. – 160 p.
2. *Stepanov D.Yu.* Analysis, design and development of enterprise resource planning systems: theory and practice // Russian technological journal. – 2015. – vol.8, №3. – p.227-238 (in Russian).
3. *Hamilton S.* Maximizing Your ERP System: A Practical Guide for Managers. – NY.: McGraw-Hill, 2002. – 350 p.