ERP Customisation Matrix

Wessel Wessels – iPlan Industrial Engineers

November 2007

A white paper modelling the level of complexities in customising ERP systems.

Contents

Introduction

ERP customisation is an integral part of ERP solutions, and has been a major source of concern for organisations implementing or using ERP systems. For some organisations, customisation proved to be a key enabler to realising ERP benifits. For others though, customisation proved to put the entire project and even the organisation at risk.

ERP customisation decisions need to be conidered carefully as they could be deciding factors to the successful implementation and subsequent operation of the system. There is thus a need for understanding the impact of certain customisation decisions within an organisation. This paper suggests a framework for classifing ERP customisation in terms of the type and method of customisation.

Although customisation has been used to explain many implementation failures, doesn't mean that all types of customisations are undesirable. Case studies show that lack of customisation sometimes causes negative consequences[1].

The implementation and customisation of ERP systems has been an issue since the inception of ERP software. Beyond the actual implementation, customisation affects the organisation in an on-going fashion through system maintenance costs, complexity, and flexibility. For these reasons, many have argued that a "vanilla" implementation, i.e. without customisation, is the "best" way to implement ERP systems. However, when business processes in an organisation cannot be modeled in an ERP system without customisation, the impact of a decision to not customise becomes relevant.

The primary goal of customising an ERP solution is to achieve a fit between the system and the processes it supports. Thus, both the system and the process can be changed or customized to achieve the goal[2].

The best way to identify the gap between the business requirements and the system's architecture or functionality is to conduct a business process blueprint. The blueprint essentially documents how the business should operate. The gap between what the business requires and what the system can do should be identified before a decision is made about customisation.

Once the gap between the business requirement and system functionality has been identified, a decision should be made on who the onus sits with to bridge this gap. Too often, this decision is plauged by complex negotiations on which side should give way, whilst the decision should rather be made from a strategic viewpoint. The degrees of freedom principle could be used to define ERP strategy[3].

This paper is concerned with situations whare the onus of change is on the system, and highlights the fact that all types of customisations and all the customisation methods are not the same.

---

[1] A Davis, *ERP Customisation Impacts on Strategic Alignmentand System Agaility,* Proceedings of the Southern Association of Information Systems Conference, 2005.

[2] Luo W, and Strong DM, *Framework for Evaluating ERP Implementation Choices,* IEEE Transactions in Engineering Management, Vol. 51, No. 3, August 2004.

[3] AJ du Toit, *Managing Discontinuous Change,* iPlan Industrial Engineers, 2007, www.iplan.co.za/downloads/managing_discontinuous_change.pdf

The ERP Customisation Matrix

The ERP customisation matrix is a framework for understanding the implications of certain ERP customisation decisions. The proposed matrix is an overlay of different methods and types of ERP customisation.

Different customisation methods

When implementing an ERP system, there are a number of alternatives methods to change or customise the software solution. This paper regards customisation methods as the "tools" that enable a system to modify, and distinguishes between 4 different types of ERP customisation methods. Furhermore, these ERP customisation methods are likened to the possible modification and alterations available in cars.

1. *Parameters and Switches*:
   The paremeters and switches within an ERP solution allow for "minor" configurations to be made. Using our car analogy, ERP parameters and switches are like the various buttons in a car; you would typically be able to adjust the airconditioner, sound, windows and sometimes even the cruising speed of the vehicle. You would not however be able to adjust the upholstry or the type of engin.

2. *On / Off Selections*:
   Most ERP software packages are developed in a modular approach, where each module contains specific functionality and configurable options. This customisation methed refers to the activation of certain modules within an ERP package. For our car analogy, the On / Off selections relate to the activation of certain functionalities like the a diff lock, or 4x4 drive.

3. *External Applications*:
   External applications are used in addition to an ERP suite, and would typically be purchased from another vendor. External applications could be likened to a trailor or roofrack of a car. These applications should however be compatible with the ERP system.

4. *Coding*:
   An open programming environment is sometimes provided by ERP vendors to modify the system. Coding goes beyond backoffice configurations, it is when changes are made to the actual script of an ERP system. When compared to a car, changing the code of an ERP system would be like changing the suspension or alterations made to the engin or gearbox.

Different types of customisation

In order to understand the impact of a certain customisation decision, it is important to know what should be achieved by the customisation. "What do you want to do, and why do you want to do it?". This paper distinguishes between 4 different types of ERP customisation:

1. *Look & Feel customisation:*
   This type of customisation refers to the user's interpretaion of the system. Look & Feel customisation would typically have no effect on the actual functionality of the ERP solution.

2. *Reporting customisation:*
   Changes made to the way in which the system outputs and measures information is referred to as reporting customisation.

3. *Workflow customisation:*
   Workflow customisation is changes made to the automation of work among users where the system is intelligent enough to act based on the definition of work types, users and tasks, and the recognition of dynamic processing conditions[4].

4. *Functional customisation:*
   Changes made to the way in which a process or module functions is referred to as functional customisation.

The overlay

The ERP customisation matrix is an overlay of the method and type of customisation, and could be used to classify the different ERP customisation decisions in an organisation. The Table below provides a detailed breakdown of the description, motivation and impact of each of the suggested customisation decision:

[4] Gartner Research, *The Gartner Glossary of Information Acronyms and Terms,* 2004

Types of customisation		Customisation method			
		Parameters & Switches	On / Off Selections	External applications	Coding
Look & Feel	Description	Configure screen layout with logos, colours and fields	Selection of fields, menus and business forms to be used (Dashboard options)	Applications that provide a user friendly front end whilst restricting user access to the ERP back end	Screen painters for designing custom screens
	Motivation	Personal preference	System functionality or process should be restricted to a user or group	Essential look & feel options that will determine ERP success	Unique system dashboard and forms required that are not catered for
	Impact	Look & feel can often be personalised without any business impact	Look & feel selections can determine user access to certain functionality	Could be essential business solution as the application governs user access to the ERP	Same impact as external applications; maintenance issues should be considered
Reporting	Description	Configure existing report templates with filter options and field selections	Selection of standard reports to be used within a solution	Reporting applications that bolt onto an existing data source	Scripts are used to extract custom designed reports from a data warehouse
	Motivation	Report information presented in a useful manner	A report does not apply to the business	Application will add value to the business as existing reports are insufficient	Existing reports does not display essential information which a custom report will
	Impact	Report configuration determines how useful the information displayd is	Not having a certain report will result in users maintaining separate calculations	Applications enables custom reports which can be saved as updatable templates	Requires in depth database knowledge and a universal coding style
Workflow	Description	Approval routings and notifications for a specific process	Whether or not a solution, module or process has workflow functionality	Applications that facilitate workflow and provide a user friendly front-end	Custom designed workflow within a solution
	Motivation	The processing sequence must include certain users	A process must be completed in a specific sequence	The existing workflow capabilities are not sufficient	Custom workflow functionality will provide a competitive advantage
	Impact	Workflow governs the users working on a process	Workflow impacts the productivity of a business	Applications could add value to user friendliness and business productivity but adds complexity to the solution	Custom coded workflow could provide synchronisation across supply chains but requires a great amount of maintenance
Functional	Description	Configuratoin of a process	Selection of modules to be included in a solution	Applications that add some sort of additional functionality	Bespoke software (Custom designed solutions)
	Motivation	Functional configuration will provide a fit with the requirements of the business	The functionality is required by the organisation	The current system does not cater for an essential functionality	Customisation is essential to the organisations and results in a competitive advantage
	Impact	Module configuration determiones how a system works. User acceptance often depends on functional configuration	Functionality selection impacts solution scope, acquisition cost and implementation time	Could add considerable value but might require users to work on multiple fronts which impacts training and focus.	Custom coded functionality could provide a competitive advantage but severely impacts the implementation and upgradability

The scale of customisation

As noted earlier in this paper, conventional wisdom holds that ERP systems should be implemented without customisation, because modification is a risk factor that contributes to project failure. On the other hand, customisation increases the degree of fit between the ERP system and the business requirements. Organisations want to know how much and what kinds of tailoring pose a threat to project success.

Through new releases ERP vendors fix bugs and add new functionality. Consequently, because the customisation done by an organisation may be affected by the new release, it may be very difficult to implement a new release while preserving customisation investments. Research has shown that ERP system customisation is related to upgrade difficulties[5]. And, indeed, some organisations have resolved to eliminate modifications after finding that "upgrading" required a complete re-implementation of ERP software.

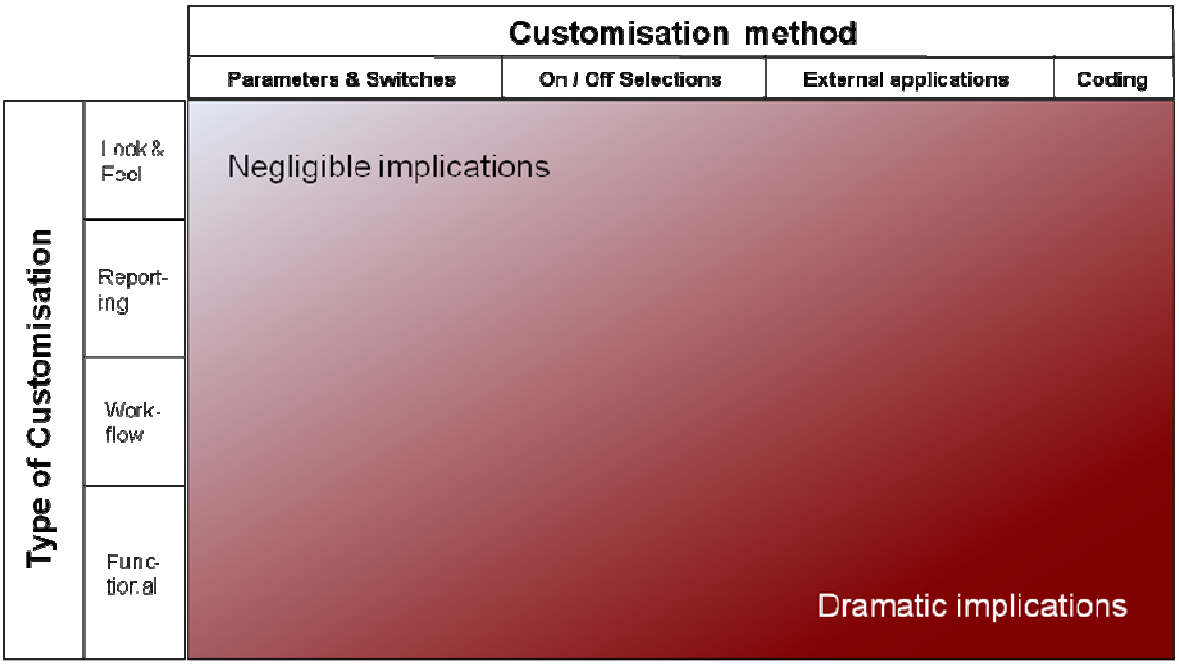
Brehm et al[6] note that the customisation types are likely to affect upgrading in different ways. For example, parameters and switches set during configuration should be unaffected by a new release. This is a major task of an ERP vendor and one of the benefits organisations expect from their solution. The other customisation types require greater effort since more elements of the system is involved. Coding will have to be thoroughly tested and may have to be reprogrammed every time a data field, software function, or variable is changed in a new release.

Evaluation of the ERP customisation matrix reveals a clear ranking in both the methods and types of customisation. The business implication of customisation options are thus ranked from left to right and top to bottom in the matrix.

The figure below illustrates the scale of ERP customisation implications. When thinking about these implications, organisations should realise that the decision making responsibility as well as the customisation skill required should be in relation to the impact it has on business.

[5] ML Markus, and C Tanis, *The Enterprise Systems Experience - From Adoption to Success*, in R.W. Zmud, ed., *Framing the Domains of IT Research: Glimpsing the Future Through the Past*, Cincinnati, OH: Pinnaflex Educational Resources, Inc., 2000, pp. 173-207.

[6] L Brehm, A Heinzl, and ML Markus, *Tailoring ERP Systems: A Spectrum of Choices and their Implications,* Proceeding of the 24th Hawaii International Conference on System Sciences, 2001.



Conclusion

Conventional wisdom holds that "vanilla" implementations" of ERP packages are much more likely to be successful than implementations that require customisation. But many companies have had to modify ERP software to meet essential business needs.

Because customisation is posited as a factor in ERP project success it is important to develop a framework for understanding the options impacts ERP customisation - this paper suggests the ERP customisation matrix.

The proposed customisation matrix is is an overlay of different methods and types of ERP customisation and holds that the business impact of these options could be ranked according to its position on the matrix.

Biblioagraphy

1. A Davis, *ERP Customisation Impacts on Strategic Alignmentand System Agaility,* Proceedings of the Southern Association of Information Systems Conference, 2005.

2. AJ du Toit, *Managing Discontinuous Change,* iPlan Industrial Engineers, 2007, www.iplan.co.za/downloads/managing_discontinuous_change.pdf

3. Gartner Research, *The Gartner Glossary of Information Acronyms and Terms,* 2004

4. L Brehm, A Heinzl, and ML Markus, *Tailoring ERP Systems: A Spectrum of Choices and their Implications,* Proceeding of the 24th Hawaii International Conference on System Sciences, 2001.

5. Luo W, and Strong DM, *Framework for Evaluating ERP Implementation Choices,* IEEE Transactions in Engineering Management, Vol. 51, No. 3, August 2004.

6. ML Markus, and C Tanis, *The Enterprise Systems Experience - From Adoption to Success*, in R.W. Zmud, ed., Framing the Domains of IT Research: Glimpsing the Future Through the Past, Cincinnati, OH: Pinnaflex Educational Resources, Inc., 2000, pp. 173-207.