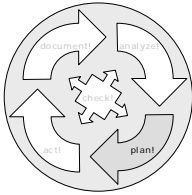


6

Planning: Creating Enterprise Architecture

Architecture takes place between two ears.



Most organizations have a map of their applications environment and infrastructure systems. How up to date and accurate is your map? Is it also used for purposes of planning, i.e. does it include representations of multiple past and future states? Are you able to run gap Analyses of current and target states? Do you have a practice of aligning your project portfolio with the measures derived from the planning of the applications environment and the gap analysis? Does this or some other procedure allow you to ensure that the various measures of technical renovation (e.g. enterprise application integration) harmonize with the IT investment strategies of the departments in your organization and that are outlined in the project portfolio?

6.1

IT Development Planning

Development planning may have a technical and/or business orientation, i.e. it may concern the infrastructure landscape and/or the applications environment. In either case, development planning represents a necessary extension of traditional portfolio management whenever its goal is not only to optimally satisfy the requirements of your clients, but also to plow ones own field, i.e. to establish the future security and stability of the current applications portfolio.

Development planning

Development planning integrates the planning of new projects with the optimization of existing systems, the securing of stability and integration, the closing of gaps, the elimination of redundancies and breaches. The outstanding significance of development planning arises from the elimination of unnecessary heterogeneity and the standardization of infrastructure and application systems. The fact that heterogeneous infrastructure landscapes possess a high degree of complexity and thereby generate high costs has long since been recognized. The heterogeneity of applications environments always comes up for consideration when it comes to eliminating clearly redundant applications in the context of mergers and acquisitions. But have you ever taken

a look at your applications environment independently of such external factors and asked yourself whether it really has to be so complicated?

Example

Let's consider an example from the insurance industry: The classic back office systems for inventory management are being developed on a mainframe in COBOL, the sales systems for mobile sales in C++, and the systems for stationary sales via sales partners in Smalltalk. What is more, the premium calculators on the Internet have an HTML format, while the C++ computing kernels are linked via Java Servlets. The new website has been realized by an external supplier in .NET technology and responsibility for its maintenance is currently being transferred to internal employees. The new generation of inventory management is to be developed in J2EE technology, which is currently being implemented in a pilot project known as *PartnerSystem*.

Our scenario involves no less than six different technologies that are not a matter of programming languages, but of complete development lines, each with its own programming conventions, tools for version maintenance, testing, debugging, and perhaps even for applications analysis. Each line requires support, each requires specialists and each a human resources development program. And all of the lines will have to be technically integrated with one another.

Does this come across to you as an exaggeration or a description of other people's problems? If yes, then skip ahead to the discussion of the significance of development planning in the context of restructuring projects and standard software development. Those of you, however, who can relate to my sketch of an imaginary, but nonetheless realistic applications environment in the insurance sector may wish to stay tuned.

The issue is planned development.

The point here is not to condemn or try to prevent the introduction of new technologies, but to highlight the importance of planned development! Which development lines are necessary? Which have arisen in connection with a strategic plan? Which qualify as technological mistakes or excess baggage and should be jettisoned? Which have arisen because projects were outsourced without explicit architectural specifications on account of staffing shortages? For which of these development lines would it also make sense to outsource the maintenance and continued development (perhaps temporarily)? And which have