Formalizing and Implementing e-Government Strategies

Olivier Glassey

There are many frameworks, methods, and capability maturity models available in the domain of information system governance, such as COBIT for IT governance, TOGAF for enterprise architecture, OpenUP or RUP for the software lifecycle, Hermes for project management, and ITIL for service management. This paper looks at these good practice methods, discusses how such could contribute to e-government strategies and describes a practical solution for formalizing and publishing such strategies using Eclipse Process Framework, an open-source tool that provides an extensible and adaptable framework for defining methods and documenting processes.

Prof. Dr. Olivier Glassey
Swiss Graduate School of Public Administration; Swiss Public Administration Network
olivier.glassey@idheap.unil.ch

Introduction

Most states (including Switzerland and its cantons) have written and published documents defining their e-government strategies. There is no question that most of these strategies are well thought out and establish a workable framework for e-government projects. However, I believe that what is lacking in many cases is a formal arrangement of such strategies that would facilitate greater practical implementation of e-government processes and systems and support the development of indicators to measure the strategic alignment of such processes. There are currently many e-government rankings available (such as "The user challenge – Benchmarking the supply of online public services – 7th measurement"[1] of the European Commission). These comparative approaches are very interesting politically as they show where a state stands in terms of e-government and provide incentives for developing better services. Nonetheless, in my opinion neither these rankings nor the classical approaches based on 4 or 5 stages of online sophistication help when it comes to implementing strategies into projects. In this paper I therefore propose a solution based on good practice frameworks or methods taken from the field of information system governance, and founded in particular on the concept of capability maturity models.

Lessons to be learned from IT governance

This paper is too short to permit me to present even a brief overview of existing IT governance methods and frameworks, however, here is a selection that in my opinion provide very interesting sources of ideas and good practices for e-government.

- COBIT[2] (Control Objectives for Information and related Technology) is a set of best practices in the IT governance and control domain centered on IT processes, measures and indicators whose main objective is maximizing the benefits of IT use.
- TOGAF[3] (The Open Group Architecture Framework) is based on four architecture domains (Business, Data, Application, Technology) and provides a method for the development cycle. Its Enterprise Continuum furthermore relies on requirements and on generic architectures to enable future use of solutions.
- RUP[4] is a commercial software development process, whereas OpenUP[5] is the open-source lightweight version of the same tool that was donated to the Eclipse Foundation. They both include iterative development processes based on use cases and scenarios; both center on architecture approaches.

There are many other methods and frameworks that could provide best practices for the implementation of e-government strategies. These include Hermes[6],

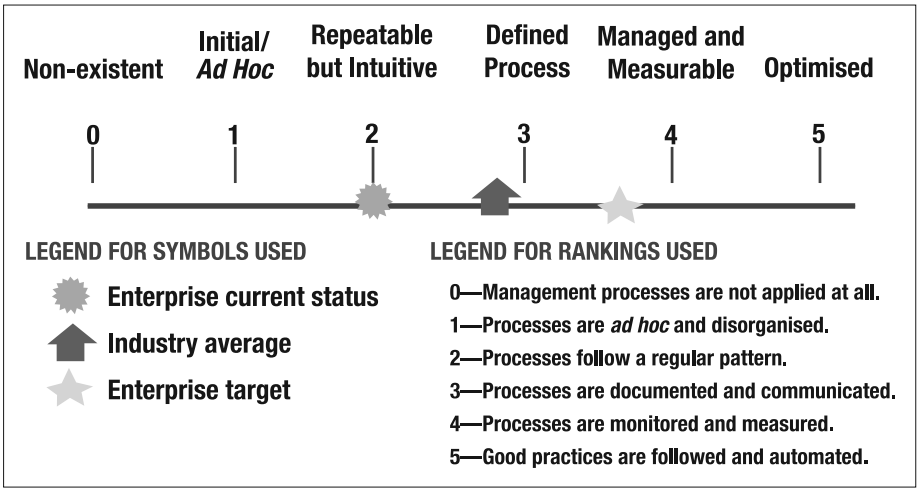

Figure 1: COBIT CMM

an open method for IT project management developed by the Swiss Federal Strategy Unit, ITIL[7] for IT service management and ABRD[8] for Agile Business Rule Development, to name but a few. Although the numerous advantages brought by these best practices could be listed, let me propose two main concepts that should be included when implementing e-government strategies:

– Formalization of requirements and processes whether for project management, software quality and lifecycle, or service management; formalized processes support the development of measurement tools and indicators.
– Use of maturity models: isolated indicators, even when very relevant and applicable, are not sufficient per se, and should be integrated in a capability maturity model. This is discussed in the next section.

Capability Maturity Models

The Capability Maturity Model (CMM[9]) was originally developed to assess the ability of government contractors to deliver software systems. The concept of CMM and most notably the idea of describing the maturity of an organization's business processes then burgeoned into other disciplines, such as project management, IT governance, risk management, and human resources. The CMM identifies five levels of process maturity:

1. Initial or ad hoc: processes are undocumented and might change at any time.
2. Repeatable: some processes are repeatable and results may be consistent.
3. Defined: there are sets of defined and documented standard processes.
4. Managed: all processes are defined as are the metrics supporting control and adjustment.
5. Optimized: processes continuously support improving performance.

Fig. 1 shows the COBIT capability maturity model. Although the wording might differ from the above list, the five conceptual levels remain the same in most governance models or methods.

Remaining with the COBIT example I illustrate how a strategy can be formalized, implemented and managed around CMM levels of concept. COBIT 4.1 provides 34 high-level processes covering four domains of IT governance (Planning and Organization, Acquisition and Implementation, Delivery and Support, Monitoring and Evaluation), and four dimensions (Applications, Information, Infrastructure, People). Fig. 2 shows (schematically) strategic objectives (business and IT) and processes of an organization. The maturity level of each of these 34 processes is then measured on the basis of 210 control objectives.

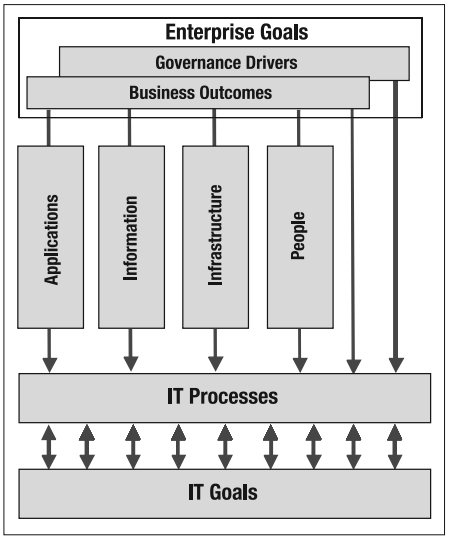

Figure 2: COBIT Goals and Processes

Proposal for a practical solution

Up to this point I have argued that it is necessary to formalize e-government strategies and have listed a few methods and frameworks that might be useful in this context. All well and good, but how should we proceed so as to formalize, publish, implement, and measure such e-government strategies?

This is where the Eclipse Process Framework (EPF[10]) comes into play. EPF is an open source project that aims at providing an extensible and configurable software development and management framework for authoring and publishing. Many libraries and plug-ins are available and implement OpenUP, TOGAF, ABRD, and other frameworks. The Swiss Federal Strategy Unit has developed its own tool on the basis of EPF called PowerUser, which supports the execution of Hermes projects.

EPF is based on libraries, i.e. containers for method content, such as tasks, roles, and work products. These libraries provide step-by-step explanations and describe how specific development goals can be achieved. Processes then use these method elements and relate them into semi-ordered sequences customized for specific types of project. A method configuration is a selection of packages from a method library used for publishing. Once processes have been authored in EPF, they can be published as a website for simple browsing of work breakdown structures, checklists, concepts, and guidelines. The use of configurations allows one sole strategy to be published as several websites each specifically adapted for IT managers, business analysts, project managers, developers, and so on.

Conclusion

To conclude, I would like to suggest a practical approach to formalizing and implementing a text-based e-government strategy, such as those mentioned in the introduction.

– A basic knowledge of the main concepts of COBIT is required to define the business objectives and IT goals; TOGAF can be used to build a suitable business architecture, and OpenUP for developing corresponding business use cases and scenarios.
– Using a selection of concepts that fulfil the requirements, business architects will translate the e-government strategy into a conceptual model and formalize matching processes in EPF.
– A capability maturity model should be defined to measure the success of the strategy and relevant indicators should be used for each process (COBIT is a good source of inspiration when defining such indicators).

Even though this approach is summarized here under just three bulleted points, this is a lot of work and can become quite complex. The entire task should be based on an iterative, incremental approach, for example starting with one strategic objective and a few processes. Thus, business architects will be able to validate the selected concepts, refine the chosen capability maturity model, and master the EPF tool.

1 http://ec.europa.eu/information_society/eeurope/i2010/docs/benchmarking/egov_benchmark_2007.pdf
2 http://www.isaca.org
3 http://www.opengroup.org/togaf
4 http://www.ibm.com/software/awdtools/rup
5 http://epf.eclipse.org/wikis/openup
6 http://www.hermes.admin.ch
7 http://www.itil.org
8 http://www.eclipse.org/epf/openup_component/openup_abrd.php
9 http://www.sei.cmu.edu/cmm
10 http://www.eclipse.org/epf

Olivier Glassey is an assistant professor who works within the Swiss Public Administration Network and leads the digital governance research unit at the Swiss Graduate School of Public Administration.