

## IST PROJECT 2001-35399



# A Governmental Knowledge-based Platform for Public Sector Online Services

<b>Project Number:</b>	IST-2001-35399
<b>Project Title:</b>	A Governmental Knowledge-based Platform for Public Sector Online Services
<b>Deliverable Type:</b>	Public

<b>Deliverable Number:</b>	D91
<b>Contractual Date of Delivery:</b>	31 January 2004
<b>Actual Date of Delivery:</b>	
<b>Title of Deliverable:</b>	Evaluation of project results
<b>WP contributing to the Deliverable:</b>	WP09
<b>Nature of the Deliverable:</b>	Report
<b>Editor(s):</b>	Nick Adams, Andy McKay-Hubbard
<b>Author(s):</b>	Nick Adams, Andy McKay-Hubbard, Angus Whyte, Ann Macintosh, Elias Spanos, Costas Vassilakis, George Lepouras

<b>Keyword List:</b> SmartGov, e-Forms, public services, knowledge management, evaluation, electronic services,
---

**Project funded by the European Community under the “Information Society Technologies” Programme (1998-2002)**

© Copyright by the SmartGov Consortium.

The SmartGov Consortium consists of:

<b>Partner's Name</b>	<b>Acronym</b>	<b>Role</b>	<b>Country</b>
University of Athens	UoA	Project Coordinator	Greece
T-Systems Nova	TNB	Partner	Germany
Indra Sistemas S.A.	Indra	Partner	Spain
Archetypon S.A.	ARC	Partner	Greece
Napier University	NU	Partner	United Kingdom
General Secretariat for Information Systems	GSIS	Partner	Greece
City of Edinburgh Council	CEC	Partner	United Kingdom



# Table of Contents

<b>EXECUTIVE SUMMARY .....</b>	<b>5</b>
<b>1 INTRODUCTION.....</b>	<b>7</b>
<b>2 EVALUATION METHODS.....</b>	<b>9</b>
2.1 THE PRINCIPLES BEHIND THE EVALUATION.....	9
2.1.1 TWO DISTINCT PILOT SITES.....	10
2.2 ASSESSING THE CONFORMANCE TO THE USER REQUIREMENTS .....	10
2.3 ASSESSING SYSTEM ACCEPTABILITY.....	12
<b>3 CONFORMANCE TO USER REQUIREMENTS.....</b>	<b>14</b>
3.1 ANALYSIS OF THE USER REQUIREMENTS .....	15
3.2 SUMMARY OF USER REQUIREMENTS ANALYSIS.....	46
<b>4 TECHNICAL ACCEPTABILITY.....</b>	<b>48</b>
4.1 RESPONSE .....	48
4.1.1 DEVELOPMENT ENVIRONMENT .....	50
4.1.2 INTEGRATOR.....	51
4.1.3 SERVICE ACCESS BY END-USERS .....	53
4.1.4 COMMUNICATION SERVICES.....	55
4.2 RELIABILITY .....	56
4.2.1 DEVELOPMENT ENVIRONMENT .....	57
4.2.2 COMMUNICATION SERVICES.....	58
4.2.2.1 UPTIME .....	58
4.2.2.2 NUMBER AND TYPE OF MISBEHAVIOURS .....	58
4.2.2.3 NUMBER AND TYPE OF SYSTEM ERRORS.....	58
4.2.3 INTEGRATOR.....	58
4.2.3.1 UPTIME .....	59
4.2.3.2 NUMBER AND TYPE OF MISBEHAVIOURS .....	59
4.2.3.3 NUMBER AND TYPE OF SYSTEM ERRORS.....	59
4.2.4 SERVICE ACCESS BY END-USERS .....	59
4.2.4.1 UPTIME .....	60
4.3 COMPATIBILITY .....	60
4.3.1 DEVELOPMENT ENVIRONMENT .....	60
4.3.1.1 DEVELOPMENT ENVIRONMENT.....	60
4.3.1.2 CLIENT CONFIGURATION .....	61
4.3.2 DEPLOYMENT ENVIRONMENT .....	61
4.3.2.1 DEPLOYMENT ENVIRONMENT.....	61
4.3.2.2 CLIENT CONFIGURATION .....	62
4.3.3 COMMUNICATION SERVICES.....	62
4.3.4 INTEGRATOR.....	63
4.4 EVALUATION OF SMARTGOV INSTALLATION PROCEDURE.....	63
4.4.1 THE INSTALLATION PROCEDURE.....	64
4.4.1.1 FRONT END.....	65
4.4.1.2 INTEGRATOR.....	66
4.4.1.3 INFORMATION INTERCHANGE GATEWAY AND SMARTGOV AGENT.....	67
4.4.1.4 DREAMWEAVER EXPORT TOOL.....	69
4.4.2 EVALUATION .....	70
4.4.3 FUTURE ENHANCEMENTS.....	73
<b>5 USEFULNESS.....</b>	<b>74</b>
5.1 UTILITY AND RELEVANCE.....	76
5.2 USABILITY .....	78
5.2.1 LEARNABILITY .....	78

5.2.2	EFFICIENCY .....	79
5.2.3	MEMORABILITY .....	81
5.2.4	FEW ERRORS .....	82
5.2.5	SATISFACTION .....	83
5.3	ANALYSIS .....	84
5.4	GSIS RUNTIME ENVIRONMENT USAGE .....	85
5.5	CEC RUNTIME ENVIRONMENT USAGE .....	88
<b>6</b>	<b>SOCIAL/ORGANISATIONAL ACCEPTANCE: COST-BENEFIT ANALYSIS .....</b>	<b>92</b>
6.1	SMARTGOV COMPETITIVE ADVANTAGE .....	92
6.2	INDIRECT SMARTGOV BENEFICIARIES – SOCIO/ECONOMIC VALUE.....	93
6.3	SWOT ANALYSIS .....	93
6.3.1	STRENGTHS .....	93
6.3.2	WEAKNESSES .....	94
6.3.3	OPPORTUNITIES .....	94
6.3.4	THREATS.....	94
6.4	PRICING POLICY .....	95
6.5	EVALUATION ANALYSIS .....	97
6.5.1	PROFILE OF INTERVIEWEES .....	97
6.5.2	REDUCTION OF DEVELOPMENT AND DEPLOYMENT TIME FOR E-SERVICES .....	97
6.5.3	LOWERING OF SKILLS OF PERSONNEL FOR CREATING E-SERVICES .....	99
6.5.4	ACCELERATING TESTING PROCESS FOR E-SERVICES .....	100
6.5.5	REDUCTION OF MAINTENANCE COSTS.....	101
6.5.6	REUSABILITY .....	102
6.5.7	IMPROVEMENT OF WORKING CONDITIONS .....	103
6.5.8	IMPROVED SERVICES TO THE CITIZEN .....	104
<b>7</b>	<b>CONCLUSION .....</b>	<b>105</b>
<b>8</b>	<b>REFERENCES.....</b>	<b>108</b>
<b>9</b>	<b>APPENDIX.....</b>	<b>110</b>
9.1	RESULTS FROM CEC USEFULNESS QUESTIONNAIRE .....	110
9.2	RESULTS FROM GSIS USEFULNESS QUESTIONNAIRE .....	116
9.3	OBSERVATIONAL NOTES FROM CEC PILOT .....	122
9.4	COST-BENEFIT ANALYSIS QUESTIONNAIRE.....	134
9.5	CEC RESOURCES FOR SMARTGOV.....	139

## Executive Summary

This deliverable describes the work undertaken in WP09 of the SmartGov Project, and concerns the evaluation of the SmartGov Platform. The overall aim of SmartGov is to specify, develop, deploy and evaluate a knowledge-based platform to assist public sector employees to generate online transaction services by simplifying their development, maintenance and integration with already installed IT systems. The methods used to conduct this evaluation are; detailed questionnaires, field observation, analysis of operational or managerial data and technical-log data. The structure of the evaluation was based upon the framework developed in deliverable D8.1. With respect to this model, the evaluation was categorised into three main components:

**Technical acceptability:** The Platform software was assessed against the user requirements as outlined in deliverable D4.1. Following this we evaluated specifically, the response, reliability, compatibility and installation of the platform. In the conformance to user requirements analysis we find that the overall total of user requirements met either fully or partially is 78%, with more than 80% of the compulsory requirements being met. Evaluation of the installation procedure for the SmartGov software noted that as it stands it works well enough, though for some components specialist knowledge is required that may be beyond that which a public authority, IT engineer might reasonably be expected to have.

**Usefulness:** This was concerned with utility and usability issues relating to how the Platform performed and was perceived by the real users in their native environment. In this we took a holistic view of the Platform in-situ at the two pilot sites detailing the use of the development environment by the Public Authority Staff to create the online pilot services. The results of this evaluation of platform usefulness were generally positive in that the majority of the success criteria were met, taken across the two pilot sites as a whole. However there was a disparity between the two sites, which can be seen in the separate results for each site. This disparity was due to the fact that the service developers in CEC had limited IT experience, whereas in GSIS, the individual service developers had a high level of IT expertise. However, some useful amendments to the platform were drawn from this evaluation.

**Social/organisational acceptability:** This focussed on a cost benefit analysis of the Platform in order to estimate the overall added value of Smartgov. The cost benefit analysis carried out by consortium partner Archetypon, suggests that the platform could provide added value.



## List of Acronyms

<b>Acronym</b>	<b>Explanation</b>
API	Application Programming Interface
BEAN	Java Bean
CEC	City of Edinburgh Council
DSN	Data source name
GSIS	General Secretariat for Information Systems
IIG	Information Interchange Gateway
JDBC	Java Database Connectivity
JDES	Joint Domiciliary Equipment Stores
JSP	Java Server Page
KU	Knowledge unit
LDAP	Lightweight Directory Access Protocol
MVC	Model-View-Controller
RDBMS	Relational Database Management System
RUP	Rational Unified Process
SGA	SmartGov agent
TS	Transaction service
TSE	Transaction service element
TSE Group	Group of transaction service element
OTS	Occupational Therapists
UML	Unified Modelling Language
WAP	Wireless Application Protocol
WML	Wireless Markup Language
XHTML	eXtensible Hypertext Markup Language
XML	Extensible Markup Language
XSLT	Extensible Style sheet Language Template

# 1 Introduction

This deliverable describes the work undertaken in WP09 of the SmartGov Project, and concerns the evaluation of the SmartGov Platform. The overall aim of SmartGov is to specify, develop, deploy and evaluate a knowledge-based platform to assist public sector employees to generate online transaction services by simplifying their development, maintenance and integration with already installed IT systems. In addition to an assessment of the stand-alone software that constitutes the Platform, an evaluation must also consider the trial of the software at the two pilot sites, namely, the General Secretariat for Information Systems (GSIS) in Athens, and the City of Edinburgh Council (CEC).

The structure of the evaluation is based upon the Nielsen Framework (Nielsen 1993) as presented in deliverable D8.1. With respect to this model, the evaluation is categorised into three main components:

- **Technical acceptability:** considers the purely technical aspects. Through it the Platform software is assessed against the user requirements as outlined in deliverable D4.1, and other issues including response, reliability and compatibility are considered. The technical issues relating to the installation of the Platform are also evaluated as part of this procedure.
- **Usefulness:** is concerned with the questions of utility and usability, issues relating to how the Platform performs and is perceived by the real users in their native environment. In this we take a holistic view of the Platform in-situ at the two pilot sites detailing the use of the development environment by the Public Authority staff to create the online pilot services, and the operation of the pilot services through the deployment component of the software.
- **Social/organisational acceptability:** gives a cost benefit analysis of the Platform collating all of the information gathered by the other aspects of the evaluation in order to estimate the overall value of the Platform.

The conformance to user requirements aspect of technical acceptability was undertaken collectively by all of the project partners and is presented in Section 3. The installation procedure was assessed by a single partner not involved in the coding of the software and is presented in Section 4. The usefulness of the



platform with respect to the two pilot sites is described in Section 5, while Section 6 examines the social/organisational acceptability, providing a cost benefit analysis of the platform. Finally this is all brought together and conclusions made in Section 7.

## 2 Evaluation Methods

In this section we outline the methods used to identify, gather and analyse data for this evaluation. The evaluation is presented in two separate sections which require distinct evaluation methods, so this methods section will follow the same structure. We begin with a brief outline of the principles behind the evaluation which provide the structure for the subsequent work.

### ***2.1 The Principles Behind the Evaluation***

As proposed in Deliverable D8.1, this evaluation will adopt a two fold approach: Conformance to user requirements and System Acceptance. The first set of criteria are derived directly from the user requirements of Deliverable 4.1. Thus the evaluation in this sense is a measure of the extent to which each of the requirements has been met. The second approach, system acceptance, is based on Nielsen's model of attributes of acceptance and addresses the success criteria described in deliverable D8.1, developed by the stakeholders for each pilot site. Thus the two approaches of the evaluation are:

1. **Conformance to User Requirements:** based wholly on the user requirements of D 4.1 this is an evaluation of the SmartGov platform and framework in terms of the extent to which those requirements are fulfilled.
2. **System Acceptance:** This is an evaluation of the platform, using the Nielsen model of attributes of acceptance as a structure, in combination with the success criteria developed by the pilot sites.

### **2.1.1 Two distinct pilot sites**

This evaluation is based on pilots being developed at two distinct sites, the City of Edinburgh Council (CEC) and the General Secretariat for Information Systems (GSIS). It is important to appreciate that the users at these sites have completely different roles, working environments and levels of IT experience.

The CEC users are service domain experts, occupational therapists, with very low levels of IT experience and do not currently use IT on a daily basis. They have no prior experience of creating online services, indeed within the whole of CEC there exists no equivalent online service with which to compare the SmartGov services. The development work carried out by this group had to be relatively ad hoc, as and when their existing work constraints permit. This has provided a very good pilot domain to test the SmartGov Platform in relation to non-IT users. However because of this and the complexity of the domain (number of forms, types of users, and extensions needed to be made to the Platform to meet all of the necessary user requirements) a full 4 month 'run-time' of the system has not been possible. Instead positive feedback from the users is provided.

On the other hand the user group at the GSIS pilot site have relatively high levels of IT experience in general, and in particular have prior experience of creating an equivalent online service. The GSIS pilot has produced data from about 100 user survey questionnaires. This data and analysis provided by the University of Athens, is included in this report, and will be further updated on the project website on a continual basis.

These differences between the pilot sites have a significant bearing on the results of this evaluation and so these results are examined in this context.

## ***2.2 Assessing the Conformance to the User Requirements***

The conformance to user requirements evaluates the SmartGov Platform against the user requirements for the software as summarised in D4.1 (Sections 2.3 (User Requirements for the SmartGov Platform) and 3.3 (Electronic Services Requirements Analysis)).

In each case the platform was tested to ascertain the degree to which it could be said to meet the user requirement. This was a matter of confirming that the functionality was present to allow the requirement to be met, and was achieved through detailed examination of the platform with respect to each of the criteria.

All of these were combined in a table and analysed with respect to them being essential/desired (must/should), the environments to which they apply (development or runtime), and the roles of those who should be able to perform the task. Each user requirement is categorised as having been either:

1. Fully met,
2. Partially met,
3. Not met,
4. No longer relevant.

*Fully Met* is defined as the requirement has been implemented in the Platform and is able to be carried out in its entirety by all of the associated user groups to which it pertains. *Partially Met* items either do not fully implement the requirement (but must satisfy at least 50% of the requirement under such circumstances), or require the involvement of user groups other than those to which they directly applied under D41. *Not Met* items are ones where the requirement is not satisfied by the SmartGov Platform.

In addition to the three standard *Fully/Partially/Not met* classifications we introduce a further *No Longer Relevant* category. This is to accommodate user requests which were either unrealistic in that they pertained to something which could not be achieved, were outside the scope of the project, already better implemented by other standard packages (eg. email services are already well met by standard mail servers), or during the process of designing the Platform a strategic decision was made and agreed by the Consortium Partners not to implement the item.

For each user requirement a comment is also given to explain why a certain classification was made.

The classification categories so defined were finally agreed by all of the Consortium Partners, at a meeting during the penultimate month of the project. The results of this analysis are given in Section 3.1.

### 2.3 Assessing System acceptability

As this section is concerned with acceptance it requires the collection of data from a number of stakeholders and various operational sources. The methods used are specific to the criteria being addressed and so they will be described here within the structure used to organise the success criteria.

The section is based on our adjusted Nielsen “attributes of acceptability” model [Nielsen 1993 Page 25]. This is a model for evaluating overall system acceptability through the evaluation of sub-concepts such as practical acceptability, usability etc.

We have adjusted the Nielsen model further, to create a structure for system acceptance in the light of the evaluation work carried out. Figure 2 below shows this structure.

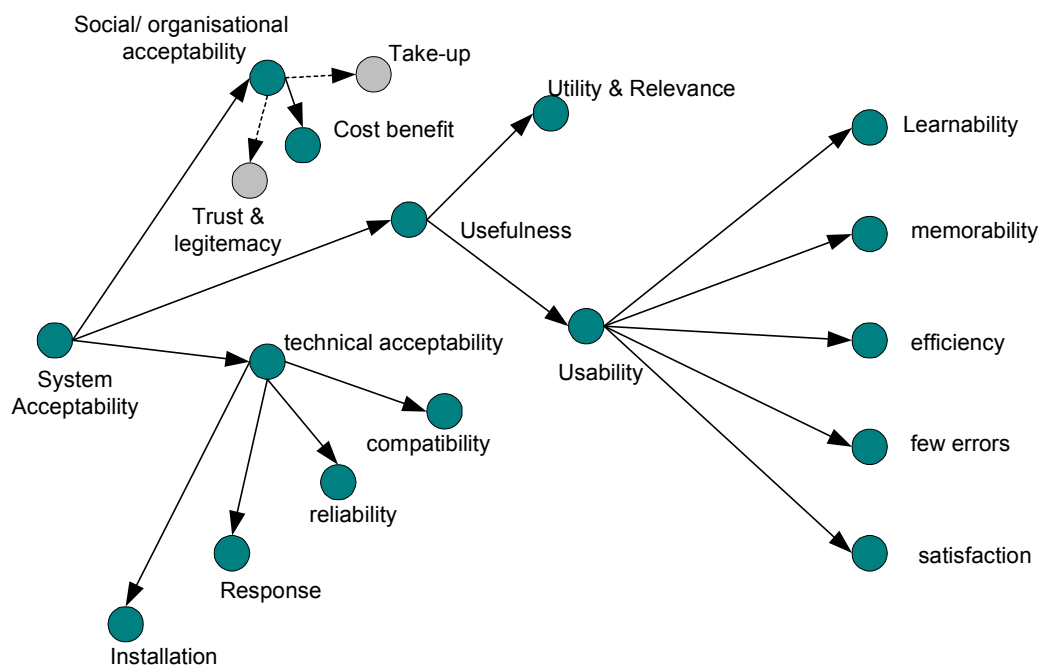


Figure 1 Adjusted Nielsen model

In the evaluation the success criteria are developed and organised under headings derived from each of the nodes in figure 1.

**Technical acceptability:**

For technical acceptance much of this data is concerned with the functioning of the software and is thus in the form of log files and operational data. The technical acceptance evaluation has four sub-sections derived from the model. These are shown below with their respective evaluation method.

**Compatibility**

Data gathered using questionnaires completed by IT staff, installation tests and management stakeholders.

**Reliability**

Operational data from system logs, user questionnaires

**Response**

Operational data from system logs, performance tests, user questionnaires, observations

**Installation**

This process was performed and fully annotated

**Social/Organisational Acceptance and Usefulness:**

These criteria were examined using questionnaires, interviews and observation. Where appropriate some operational data and statistical material has been used. These methods were tailored to address specifically the chosen success criteria, thus providing directly relevant data. Combining questionnaires with observation allows the evaluator to avoid constraining data to preconceived assumptions and enables a broad evaluation of the platform that sits alongside the specific criteria. The individuals interviewed, questioned and observed were the service developers for each pilot site, plus managers and IT staff as appropriate.

Questionnaires were administered at appropriate points in the development cycle so that users of the platform had sufficient experience to be able to provide useful responses. Observation was carried out during sessions of development work performed by these individuals. This was as non-intrusive as possible and followed a method of recording events without attempting to interpret them. Interviews were carried out again at appropriate points in the pilot time line, with relevant individuals.

It is important to note that in both the CEC and GSIS pilots, the individuals concerned had to perform these tasks alongside their everyday work load. The evaluation must be considered in this context.

There follows a list of the generic criteria headings with their evaluation method.

**Table 1 evaluation methods**

<b>Node Heading</b>	<b>Criterion</b>	<b>Instrument</b>	<b>Role</b>
<b>Social/Organisational Acceptance</b>	Cost -benefit	Questionnaires and Interviews	Strategic Management
	Take - up	Not assessed	N/A
	Trust and legitimacy	Not assessed	N/A
<b>Usefulness</b>	Utility and reliability	Questionnaires and interviews	Domain experts
	Usability	Questionnaires and interviews	Domain experts
	Learnability	Questionnaires and interviews	Domain experts
	memorability	Questionnaires and interviews	Domain experts
	efficiency	Questionnaires, interviews and Management data	Domain experts and managers
	few errors	Questionnaires, interviews and observation.	Domain experts
	satisfaction	Questionnaires and interviews	Domain experts

### **3 Conformance to User Requirements**

In this Section the results of the analysis of the SmartGov Platform with respect to the user requirements in D4.1 are presented. Firstly the full analysis of each requirement is given, before the categorisations are summarised and the implications of the analysis are discussed.

### 3.1 Analysis of the User Requirements

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
1.	User-friendly interface	√	√	Must	Fully met	All	Confirmed as part of the user evaluation
2.	Context-sensitive help functionality	√	√	Must	Fully met	All	The development environment offers context-sensitive help, and help relevant to specific contexts can be added to services developed.
3.	Multilingual interface and content	√	√	Should	Fully met	All	Both the development platform and the developed services are fully multilingual.
4.	Multiple access	√	√	Must	Fully met	All	Multiple users may concurrently access the development platform and the deployed services.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
5.	Authentication and access control mechanism	√	√	Must	Fully met	All	Username and passwords are used to authenticate users both in the development environment and the deployed services. The development environment offers role-based access control, whereas for the deployed services access control may limit or allow usage of specific services.
6.	Back up facilities	√	√	Must	Fully met	All within PA	Back-up of data is supported through an inverse document crawler which allows the data housed in the SmartGov repositories to be extracted as XML documents. The document crawler provides a mechanism for reinsertion.
7.	View automated reports	√	√	Must	Fully met	Managers and Service Workers	These can be viewed as tabular output.
8.	Management of statistics	√	√	Must	Partially met	All within the PA	Certain statistics predefined. Other statistics require IT staff involvement to implement.



No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
9.	Management of predefined TSEs	√		Must	Fully met	Domain Expert and IT Staff	The development environment facilitates creation, editing and use of predefined TSEs. It is possible to view all of the TSEs, however a sort or search facility would greatly enhance the capability.
10.	Reuse and adaptation of previous work	√		Must	Fully met	Domain Expert	Forms can be re used across services; TSE groups may be reused across forms and services; TSEs may be reused across services, forms, and TSE groups. Knowledge Units and the taxonomy are always available.
11.	Creation of new TSEs	√		Must	Fully met	Domain Expert and IT	TSE creation is allowed by the development environment.
12.	Definition of constraints and validation checks	√		Must	Fully met	Domain Expert and IT Staff	Validation checks may be defined and associated with SmartGov platform elements.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
13.	Modelling of inter-element relations	√		Must	Fully met	Domain Expert and IT Staff	A service is associated with forms, and each form is linked with the TSEs and TSE groups it contains. TSE groups are connected with the TSEs they comprise of. All elements may be related to KUs (links among KUs may be established as well) and all elements may be attached to taxonomies. Finally, links between semantic and visual form elements may be established through an integration procedure.
14.	Attachment of domain knowledge on form elements	√		Must	Fully met	Domain Expert and IT	Form elements may be linked with KUs, which represent domain knowledge.
15.	Composition of manual and instructions for the end users	√		Must	Fully met	Domain Expert and IT Staff	Instructions can be provided to users through associated KUs, from which developers can compose manuals.

No	Functional Specifications	Environment		Must/Should	Met?	User Role	Comments
		Dev.	Run				
16.	Composition of documentation about the implemented service	√		Must	Partially met	Domain Expert and IT Staff	Appropriate KUs may be composed to document the implemented service internally. External documentation of services is outwith the scope of SmartGov.
17.	Connections with third party systems	√	√	Must	Fully met	Domain Expert and IT Staff	The expertise of Domain Experts is needed to provide the provide specifications for data. The IT staff need to write code to link the Platform to the 3 <sup>rd</sup> Party Systems. This is however facilitated through documented source code templates that can be simply tailored to requirements.

No	Functional Specifications	Environment		Must/Should	Met?	User Role	Comments
		Dev.	Run				
18.	Management of user accounts	√	√	Must	Fully met	Service Workers	The SmartGov platform integrates functionality for management of platform user accounts (including both authentication and access control). Management of end-user accounts (for accessing the deployed services) can be done through a GUI user account management tool which has been developed.
19.	Definition of the information to be recorded in the log files	√	√	Must	Fully met	Service workers and IT Staff	Any logging information outside that which is supported by standard web-server installations can to be coded by IT staff. As we cannot foresee all logging information one may want some coding will be necessary.
20.	Definition and editing of service process models	√		Should	No longer relevant	Managers, Domain Expert and Service Workers	A strategic decision was made not to support process models in the Platform. There are a plenty of commercial packages available in the market if this is required.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
21.	Browsing of service process models	√		Should	No longer relevant	Managers, Domain Expert and Service Workers	A strategic decision was made not to support process models in the Platform.
22.	Web-enabled interfaces	√	√	Should	Fully met	Managers, Domain Expert and Service Workers	All operations within the SmartGov environment are performed through web interfaces. Deployed services are accessed through web interfaces also.
23.	Design, store and use standard look and feel for forms	√		Must	Partially met	Domain Expert and Service Workers	Standard look and feel forms can't be stored within platform as all design aspects relating to the look and feel of forms are created with 3 <sup>rd</sup> party applications that are used in conjunction with the SmartGov Platform. However the look and feel of functional components on forms is regulated by the Platform.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
24.	Detect data duplication within a service or across services	√	√	Should	Not met	Domain Expert and Service Workers	No mechanism exists to perform this check. Staff are able to re-use components from other services, however the value of trying to detect data duplication or means of doing it reliably are at best questionable.
25.	Associate forms with time periods and specify related actions	√		Must	Fully met	Domain Expert and Service Workers	Validation rules within the platform enable this.
26.	Specify deadlines for document submission	√		Must	Fully met	Domain Expert and Service Workers	A service has an <i>expiration date</i> which can be used as a hard deadline for the service. Nothing exists at the individual user level, however account restrictions/expiration of service can be set to enforce this.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
27.	Specify trigger conditions when certain dates arrive	√		Must	Partially met	Domain Expert and Service Workers	Not directly supported, as it requires IT staff to write code to implement it. However the development environment supports the insertion of such code. Trigger conditions and associated actions can vary a lot and we cannot foresee them all.
28.	Provide checks for dates that appear on forms (special case of "definition of constraints & validation checks")	√		Must	Fully met	Domain Expert and Service Workers	Date validity is automatically checked. The function library for validation checks includes adequate functionality for coding all date-related checks.
29.	Store PA worker details and associate them with services	√		Must	Fully met	Domain Expert and Service Workers	PA worker details may only be indirectly stored using taxonomies or KUs.
30.	Leverage client awareness for the new services	√	√	Should	No longer relevant	Domain Expert and Service Workers	As a decision was made not to store or manage client details within the SmartGov Platform, no provision can be made for leveraging client awareness.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
31.	View material associated with a form	√	√	Must	Fully met	Domain Expert Service Workers and End Users	The form editing and presentation pages provide links to each individual element associated with the form.
32.	Provide context sensitive guidance on form-filling	√	√	Must	Fully met	Domain Expert and Service Workers and IT Staff	No mechanism exists to support context-sensitive guidance. Attaching of KUs to specific locations on forms allows context sensitive help to be provided.
33.	Associate help and guidance with forms and their uses	√		Must	Fully met	Domain Expert and Service Workers	The development environment provides all facilities for creating help items (KUs) with suitable content and associating them with forms.
34.	Provide guidance on scannability of forms	√		Should	Not met	Domain Expert and Service Workers	No feasible method for assessing form scannability exists.
35.	Automatically check forms for scannability	√		Should	Not met	Domain Expert and Service Workers	No feasible method for assessing form scannability exists.



No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
36.	View process model details		√	Must	No longer relevant	End Users	The SmartGov Platform does not directly support process models.
37.	Access advice and guidance on good form design	√		Should	Fully met	Domain Expert and Service Workers	The SmartGov Platform includes pre-made KUs with appropriate information. Organisations may add their own KUs or modify the existing ones.
38.	Assess actual use of data items	√	√	Should	Not met	Domain Expert and Service Workers	No standard criteria for assessing actual use of data items exist, thus no relevant functionality may be incorporated in the SmartGov Platform.
39.	Identify data on forms that may already be known	√		Must	Not met	Domain Expert	The Platform does not enable this to be supported. Without fully integrated back-office systems and procedures in place for sharing data this requirement is not realisable.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
40.	Design form formats readable by people with sight impairment	√		Must	Partially met	Domain Expert	Provision of guidelines to support this can be achieved through KUs, however these formats cannot be enforced, and there is no explicit support of this by the SmartGov Platform. Form designers should follow the W3C guidelines ( <a href="http://www.w3.org/WAI">www.w3.org/WAI</a> ) when designing the visual part of the form. They can also use an online evaluator for forms (bobby). The platform allows for linking different form visuals (for sight impaired and non-sight impaired) to the same form semantics so any desired style can be accommodated.
41.	Maintain versions of forms	√		Must	Fully met	Domain Expert and Service Workers	Old form versions may be exported (or backed-up) and reloaded when appropriate. They can also be made available on-line through a parallel SmartGov installation.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
42.	Access form-editing facilities	√		Must	Fully met	Domain Expert and Service Workers	Editing of form functionality exists through TSE specification in the development environment. A basic html editor has also been incorporated within the Platform that enables management of form layouts, so this requirement is satisfied fully.
43.	Access simple rules of thumb for identifying complicated text	√		Should	No longer relevant	Domain Expert	"Complicated text" varies depending on the domain context and the target group (e.g. if a service is addressed to accountants, legislative data can be directly used; this is not the case when the service is addressed to citizens) and thus would be impossible build simple rules of thumb for identifying it into the platform in advance.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
44.	Select colours in user interfaces	√	√	Must	Partially met	All Users	IT staff are required to edit Cascading Style Sheets (CSS). Only a single colour/font scheme is supported by default. However as the Platform fully supports cascading style sheets editing of a single file by IT Staff is all that is required to change the look and feel of the whole environment.
45.	Access guidance on readable colour combinations	√		Must	Fully met	Domain Expert	This is catered for through built-in KUs, that provide such advice. Domain Experts are able to edit and complement these according to organisational requirements.
46.	Edit colours in form design	√		Must	Fully met	Domain Expert	A basic html editor has been incorporated within the development environment of the Platform that enables management of form layouts.
47.	Select font sizes in end user interfaces	√	√	Must	Fully met	All Users	A basic html editor has been incorporated within the development environment of the Platform that enables management of form layouts.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
48.	Define font sizes in form design	√		Must	Fully met	Domain Expert	A basic html editor has been incorporated within the development environment of the Platform that enables management of form layouts.
49.	Enter indications clearly show which data are required at each stage	√		Must	Fully met	Domain Expert	Mandatory messages can be generated automatically.
50.	View which data has been entered and which remains to be entered		√	Must	Fully met	End Users	Remembers where you are for an active session.
51.	Tag data input against each stage in a process	√		Must	No longer relevant	Domain Expert	The platform does not directly support process modelling.
52.	Receive email messages from end users		√	Must	No longer relevant	Domain Expert and Service Workers	Since organisations have their own email system for communication and free services exist on the web that allow users to have their own email accounts, there is no advantage to implementing such a facility within SmartGov.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
53.	Include e-mail facilities in services	√		Must	No longer relevant	Domain Expert	Email facilities can be partially included through use of 'mailto' links added to forms. This is done by hand. when designing the form in the integrated editor. Fully-fledged email facilities are not relevant.
54.	Specify notifications for interested parties (service workers and end users) about approaching deadlines	√	√	Must	Partially met	Domain Expert and Service Workers	The SmartGov Platform has no knowledge of the end-users' details, thus no provision can be made for notifying end users of approaching deadlines. For service workers registered in the SmartGov user database, emails can be sent.
55.	Provide means for users to enter dates without typing	√		Must	Fully met	Domain Expert	Functionality has to be designed in TSEs and 3 <sup>rd</sup> party software is required to implement the form layout.. With the completion of an integrated html editor (work in progress) this will be fully met.

No	Functional Specifications	Environment		Must/Should	Met?	User Role	Comments
		Dev.	Run				
56.	Provide details of where to go for extra help if the user gets stuck	√	√	Must	Partially met	Domain Expert and Service Workers	While the functionality exists to make this possible through KUs, such information in the runtime environment is dependant on the transaction service being developed; and cannot be foreseen in advance.. For the development environment the online help provides this.
57.	Provide print facilities so that paper versions of forms look the same as the online version	√	√	Must	No longer relevant	Domain Expert and Service Workers	A strategic decision was made to restrict the platform to standard HTML only for document presentation. As a result this requirement cannot be supported. A method of overcoming this weakness however, could be by the creation of a summary form at the end of a service, which the user can print off.
58.	Define/view how each piece of data will be used	√	√	Must	No longer relevant	Managers and Domain Expert	The platform does not directly constrain the usage of data elements. It is not appropriate to enforce or disallow data usage.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
59.	Include graphics in form design	√		Must	Fully met	Domain Expert and Service Workers	A basic html editor has been incorporated within the development environment of the Platform that enables management of form layouts.
60.	Include offline activities in process models	√		Must	No longer relevant	Domain Expert	A strategic decision was made not to support process models in the Platform.
61.	Save sessions to be resumed later		√	Must	Fully met	End Users	Documents may be saved and retrieved later for completion of editing.
62.	Switch from on-line to off-line mode and vice versa		√	Must	Not met	End Users	No provision for off-line form completion is made. This is compensated for by offering provision to store documents and resume their editing at a later stage.
63.	Define when electronic signatures will be produced, accepted or validated	√		Must	No longer relevant	Domain Expert	Browsers provide this support so it is not necessary to be duplicated within the SmartGov Platform.



No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
64.	Define user classes and authority to access forms	√		Must	Fully met	Managers and Domain Expert	Within the development environment, access control is based on groups. Within the service delivery environment, access by users can only be granted or denied at the service level.
65.	Specify shortcut codes and their meanings	√		Should	Fully met	Domain Expert	Shortcut codes and their meanings may be defined in terms of elements drawing their values from specific lists.
66.	Define system use metrics	√		Must	Partially met	All within the PA	System use metrics may be defined and stored in KUs. System use metrics rated as "important" are always collected.
67.	View system use reports in comprehensible form (graph, tables etc)		√	Should	Partially met	All within the PA	System use reports are presented in tabular format.
68.	Design end-user surveys	√		Should	Fully met	Managers and Domain Expert	End-user surveys may be designed and deployed as transaction services.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
69.	Define efficiency and effectiveness metrics	√		Should	Partially met	Managers and Domain Expert	Efficiency and effectiveness metrics may be defined and stored in KUs. Efficiency and effectiveness metrics rated as “important” are always collected.
70.	View efficiency and effectiveness metrics in comprehensible forms (charts, tables etc)		√	Should	Partially met	Managers and Domain Expert	Efficiency and effectiveness metrics are presented in tabular format.
71.	Schedule notifications and announcements for end-users	√	√	Must	Partially met	Domain Expert and Service Workers	Announcements can only be made to end users through running services.
72.	Schedule notifications and announcements for PA users	√	√	Must	Partially met	Domain Expert and Service Workers	Announcements can only be made to PA users through running services.

No	Functional Specifications	Environment		Must/Should	Met?	User Role	Comments
		Dev.	Run				
73.	Access and use data entered by the end-users		√	Must	Fully met	Service Workers	Services can be configured (through xsl) to generate an html page for each document submitted. Alternatively IT staff can write code to link the Platform to the 3 <sup>rd</sup> Party Systems using documented source code templates that can be simply tailored to requirements.
74.	Get guidance on what to do next		√	Must	Fully met	Service Workers	This is addressed via help items, both in the development environment and in the service delivery environment.
75.	State which documents are required from end users	√		Must	Fully met	Domain Expert	Statements may be incorporated in the visual form design or within KUs.
76.	Access records for submitted forms		√	Must	Fully met	Managers and Service Workers	Services can be configured (through xsl) to generate an html page for each document submitted. Alternatively IT staff can write code to link the Platform to the 3 <sup>rd</sup> Party Systems using documented source code templates that can be simply tailored to requirements.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
77.	Check if required documents have been submitted		√	Must	No longer relevant	Service Workers	This was deemed to be beyond the scope of the project. It is possible for IT staff to code such functionality into deployed services.
78.	Specify which forms should have hard copies retained	√	√	Must	Partially met	Domain Expert and Service Workers	This designation can be made through appropriate KUs. The SmartGov Platform does not print forms, however IT staff may provide code to automatically print specific forms upon submission by end-users.
79.	Specify lists of permissions for data	√		Should	Fully met	Managers and Domain Expert	Access permissions may be set for items within the development environment.
80.	Define fields that draw values from specific lists	√	√	Must	Fully met	Domain Expert IT Staff and End Users	Fields drawing values from specific lists may be defined.
81.	Define sections on forms for grouping of related fields	√		Must	Fully met	Domain Expert IT Staff and End Users	Fields may be grouped using TSE groups.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
82.	Define repeating fields/field groups for implementing tabular forms	√		Must	Fully met	Domain Expert IT Staff and End Users	Repeating groups are fully supported by defining appropriate TSE groups.
83.	Define tables with a fixed number of rows	√		Must	Fully met	Domain Expert IT Staff and End Users	Tables with a fixed number of rows can be created by defining appropriate TSE groups.
84.	Define tables with dynamically varying number of rows	√		Must	Fully met	Domain Expert IT Staff and End Users	Tables dynamically varying number of rows can be created by defining appropriate TSE groups.
85.	Define fields with pre-filled values	√		Must	Fully met	Domain Expert IT Staff and End Users	Default values are supported. More elaborate pre-filling of field values can be implemented using pre-made documents or providing code that is attached to the document retrieval service, but this has to be hand-crafted by IT Staff.
86.	Define display-only fields with constant values	√		Must	Fully met	Domain Expert IT Staff and End Users	Fields may be designated as 'read-only'.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
87.	Define display-only fields that present calculation results	√		Must	Fully met	Domain Expert and IT Staff	Automatic calculation formulas may be entered both as SmartGovLang rules or as native Javascript code.
88.	Retrieve values from registries for pre-filling fields	√		Must	Fully met	Domain Expert and IT Staff	IT staff can write code to retrieve values from registries housed in 3 <sup>rd</sup> Party Systems using documented source code templates that can be simply tailored to requirements.
89.	Designate which pieces of associated information should be available for end-service users	√		Must	Fully met	Domain Expert IT Staff and Service Workers	Only elements associated with the visual form design are available to end-users. All other elements are concealed.
90.	Designate the methods through which end-users will access the extra information (descriptions, examples, etc) in the deployed service	√		Must	Partially met	Domain Expert IT Staff and Service Workers	Extra information is always presented as a pop-up window.
91.	Define allowable value types for fields	√		Must	Fully met	Domain Expert and IT Staff	Achieved through validation rules or fields drawing values from specific lists.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
92.	Define maximum length for text fields	√		Must	Fully met	Domain Expert and IT Staff	An appropriate property exists for fields.
93.	Define minimum and maximum values for arithmetic fields	√		Must	Fully met	Domain Expert and IT Staff	A suitable validation check may be coded. Since this validation is quite common, a 'compact' form has been provided to facilitate the work of developers.
94.	Define set-oriented checks for tabular sections, such as uniqueness, cardinality, etc.	√		Must	Fully met	Domain Expert and IT Staff	Uniqueness and cardinality checks can be entered through appropriate properties of TSE groups. All types of checks may be coded through the function library of the SmartGovLang.
95.	Designate whether a validation check should be performed in the user interface environment (e.g. web browser) for early error detection, or only in the back-end	√		Must	Fully met	Domain Expert and IT Staff	An appropriate designation can be entered for each validation rule.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
96.	Define appropriate error messages whenever validation checks fail	√		Must	Fully met	Domain Expert and IT Staff	Error messages can be defined.
97.	Define warning conditions, i.e. cases that need the user's attention but do not inhibit the continuation of a submission	√		Must	Fully met	Domain Expert and IT Staff	Validation checks may emit warning messages. Informational messages are also supported, but are not generated automatically.
98.	Designate the format of the protocol numbers that will be assigned to statements upon submission	√		Must	Fully met	Domain Expert and IT Staff	A validation check may automatically assign the protocol number in any desired format.
99.	Designate whether submitted documents may be withdrawn (deleted) or edited	√		Must	Fully met	Domain Expert and IT Staff	The appropriate designation can be entered through the transaction service properties.
100	Facilitate generation of customisable login screens	√		Must	Fully met	Domain Expert and IT Staff	A standard login screen is offered. Any customisation has to be made by the organisation's IT staff.



No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
101	Facilitate generation of user registration forms, user authentication procedures and generation of user identification credentials	√		Must	Fully met	Domain Expert and IT Staff	User registration forms can be implemented as SmartGov transaction services, user authentication procedures can be customised by replacing the built-in authentication module with a custom, organisation-specific one, and generation of user identification credentials can be coded to be performed automatically through validation rules.
102	Select the platform that will be used to deploy the service	√		Must	Fully met	Domain Expert and IT Staff	The dissemination server through which services are deployed can be selected.
103	View list of available services and select the desired ones		√	Must	No longer relevant	End Users	There is a list of services in the development environment, but as SmartGov supports multiple deployment servers this is not relevant.
104	View error messages together with the fields involved in the respective validation checks		√	Must	Fully met	End Users	Error messages can be set to appear next to relevant fields, indicating the type of error that occurred.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
105	Produce “exports” of the submitted data		√	Must	Fully met	IT Staff and Service Workers	Services can be configured (through xsl) to generate an html page for each document submitted. Alternatively IT staff can write code to link the Platform to the 3 <sup>rd</sup> Party Systems using documented source code templates that can be simply tailored to requirements.
106	Automatically inform users for “post submission” detection of errors		√	Must	No longer relevant	IT Staff and Service Workers	Detecting late errors is a procedure outside the SmartGov Platform. Moreover, the SmartGov Platform has no knowledge of the details for communicating with the end-users of the services.
107	Access facilities for correcting “lately detected” errors		√	Must	No longer relevant	End Users	Detecting late errors is a procedure outside of the SmartGov Platform.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
108	View complete image of form before submittal		√	Must	Partially met	End Users	No "single page" view is provided, since it is impractical for large services. The user can, however, browse through the distinct forms and inspect the values in the different fields before submittal. It is also possible to add a summary form at the end of a service.
109	Associate form elements with tasks in the process model	√		Must	No longer relevant	Domain Expert and IT Staff	The platform does not support process modelling.
110	Define communication channels with separate IT applications (possibly through data repositories)	√		Must	Fully met	Domain Expert and IT Staff	The expertise of Domain Experts is needed to provide the provide specifications for data. The IT staff need to write code to link the Platform to the 3 <sup>rd</sup> Party Systems. This is however facilitated through documented source code templates that can be simply tailored to requirements.

No	Functional Specifications	Environment		Must/ Should	Met?	User Role	Comments
		Dev.	Run				
111	Associate form elements with decision points in the process model	√		Must	No longer relevant	Domain Expert and IT Staff	The platform does not support process modelling.
112	Define form elements whose contents are conditionally dependent on other data	√		Must	Fully met	Domain Expert and IT Staff	Validation rules are allowed to set the values of form elements, taking into account values of other elements.
113	Associate authorised roles with items in external databases	√		Must	No longer relevant	Domain Expert and IT Staff	This is not directly supported by the SmartGov Platform. It is the responsibility of external databases to set their own access privileges.

**Table 2: Analysis of the Platform against the user requirements.**



### 3.2 Summary of User Requirements Analysis

The above classifications are summarised in the following table.

	Fully Met	Partially Met	Not Met	No longer Relevant	Total
<b>Counts</b>	71	15	6	21	113
<b>Must (%)</b>	68	15	3	15	85
<b>Should (%)</b>	35	18	18	29	15
<b>Total %</b>	63	15	5	17	100

**Table 3: Summary of categorisations of the user requirements.**

It can be seen from these results that over 80% of the compulsory (must) requirements are met either fully or partially, with the majority of the remainder being deemed *No Longer Relevant*, and more than 50% of the desirable requirements (should) have been met at least partially.

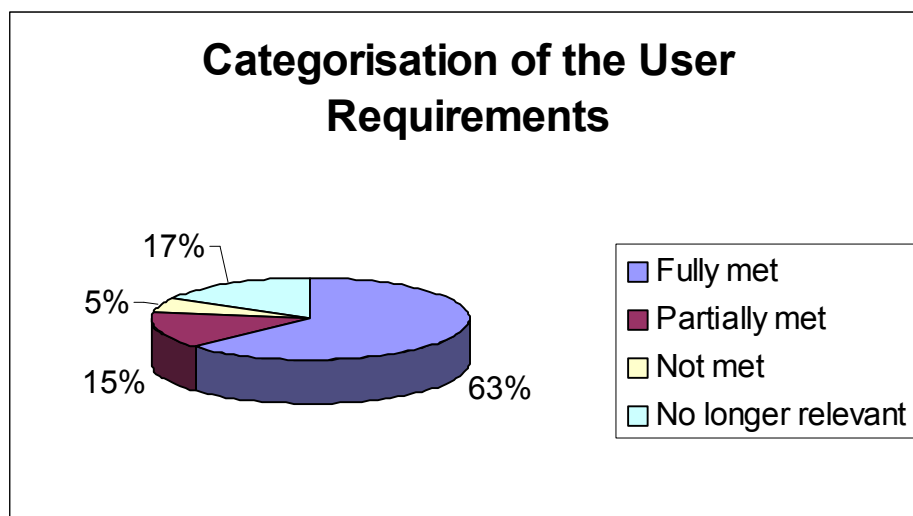
Of the compulsory items classified as *No Longer Relevant* a significant proportion (nearly one third) relate to process modelling. While being a desirable facility to incorporate within the SmartGov Platform, a strategic decision was taken early on in the project by the technical partners not to include it because it was considered that there would be insufficient time available within the project to implement it.

The other compulsory *No Longer Relevant* items relate either to requirements that are better handled by commercially available IT components such as email (item 52) or, with hindsight were unrealistic and should not have been specified as a requirement in D4.1. An example of these is item 106 – “automatically inform users of post-submission detection of errors,” which are likely to occur after the user has logged off and departed, when the back-office staff process the submission.

There are a number of compulsory items that are *Not Met*. These are requirements which were deemed relevant to the scope of the project and important to include, but for various reasons the objectives were not satisfied. The two most significant ones of these were the ability to print the completed form (item 57), and an offline mode so that the user did not have to remain connected while filling in the forms (item 62). The latter requirement while very

desirable indeed especially in areas where internet connections are unreliable or expensive, really only caters for an extreme mode of operation and does not impact at all on the operation of the Platform under normal circumstances. Non-inclusion of a print facility is potentially more serious as it makes it hard for the user to retain a hard-copy of their submission for filing purposes or in the case of a failure occurring with equipment during or after their transaction. While to some extent a record can be made through use of the browser's print menu this is a weakness that needs to be addressed in any future releases of the platform.

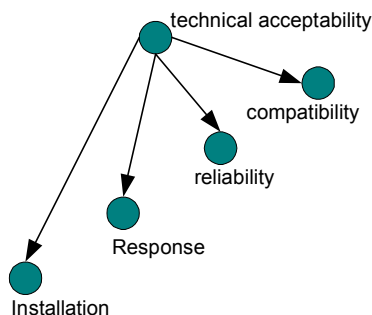
The overall totals for each of the classifications of the user requirements are more clearly understood when viewed as a chart (Figure 2).



**Figure 2: Summary of categorisations of the user requirements.**

From this it can be seen that the overall total of user requirements met either fully or partially is 78%.

## 4 Technical Acceptability



**Figure 3 technical acceptance nodes**

This Section is concerned with the evaluation of the four technical nodes from the adapted Nielsen model Figure below. The nodes; response, reliability and compatibility are examined by the consortium partner The University of Athens. The fourth node, Installation, is examined by the consortium partner Napier University.

### 4.1 Response

This subsection evaluates the performance of the SmartGov platform. Since the SmartGov platform is a complex software system consisting of various components, the performance of each component is reviewed separately. The communication services however, although used as an integral part of the service delivery environment, are examined both in the context of the service delivery environment and separately since (a) they may be used as a communications system for platforms other than the SmartGov service delivery environment and (b) the reviewers have expressed their interest in the performance analysis for these components.

Note that the results of any kind of performance analysis heavily depends on the overall setup of the installation, involving:

1. the number of machines used for the support of the different tasks in the SmartGov architecture (database servers, application servers, communication services, client machines etc)
2. the configuration of each machine (processor speed, amount of RAM, disk speed etc)
3. the speed of the network interconnecting the machines.

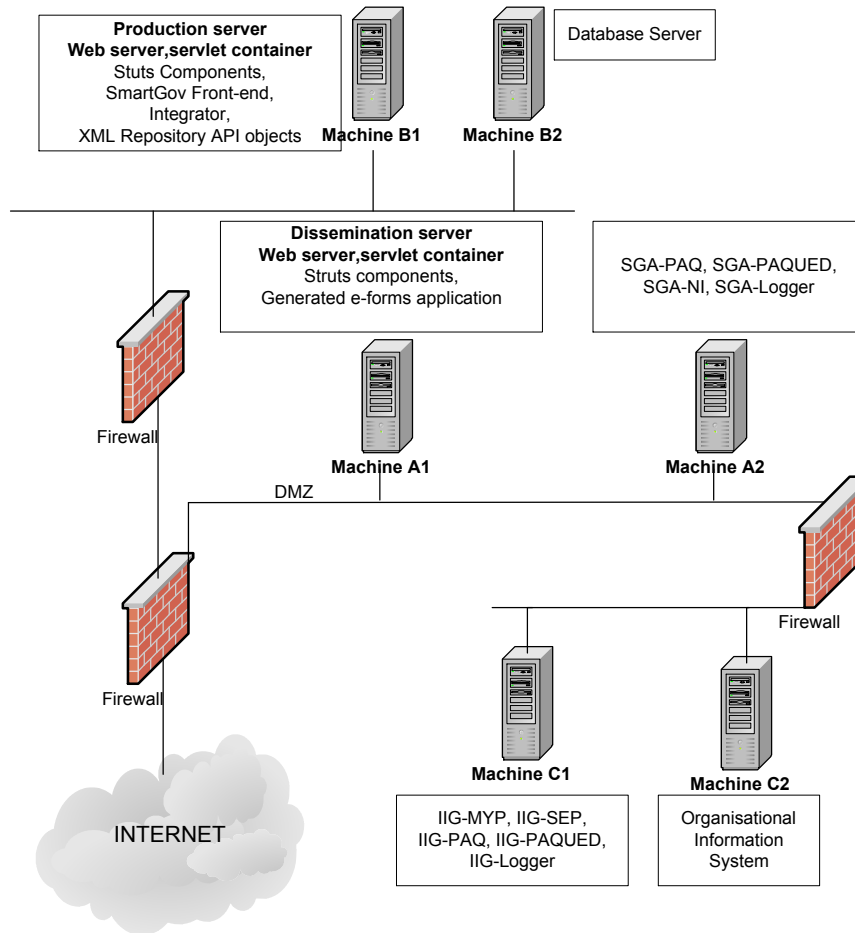


The optimal setup suggested for the SmartGov platform installation has been presented in [D51-61] and is depicted in Figure 4. This setup uses five machines to support various SmartGov tasks; these machines should be interconnected with high-speed networks. However, it is expected that a number of public authorities will opt for sub-optimal setups with fewer machines, due to limited resources. Thus, in our performance evaluation we used such sub-optimal configurations with either one or two machines supporting all the SmartGov services, in order to obtain results which will be valid for small and medium installations. Metrics for larger installations will be given wherever appropriate. Similarly, we have avoided using machines with extreme configurations (very fast processors, vast amounts of memory or extravagant disk subsystems), confining the performance evaluation environment machines to the following characteristics:

1. processor speed up to 2.2 GHz
2. installed memory up to 512 Mbytes
3. ordinary IDE disks with rotational speed equal to 7200 rpm.

Finally, the speed of the interconnecting network was always 100 Mbps.

In the rest of this sub-section, the performance of the SmartGov development environment, the performance of the services generated by the SmartGov platform (and accessed by the end-users), the performance of the Integrator and the performance of the communication services are analysed.



**Figure 4 – Optimal SmartGov topology**

#### 4.1.1 Development environment

The development environment is used by the SmartGov platform stakeholders to create the SmartGov platform objects that make up the transactional services (TSEs, TSE groups, KUs etc). Its usage is highly interactive with platform users issuing requests to the application server when certain actions are performed (effectively when a different page of the interface is requested or objects/data are saved), and working for some time with each page of the interface after it has been fetched and displayed.

Since the development environment is interactive, it is crucial that its performance is acceptable by the users. The acceptability criterion set in D81 states that "75% of actions (are completed in) less than 1 second". The threshold of one second is important in interactive systems, since studies have concluded that users do not observe any difference if actions are completed in any smaller time scale (e.g. it is indifferent for the user of an interactive system if an action is completed in 0.1 seconds or 0.8 seconds). By analysing the application server

logs and observing the user actions and screens, the following performance data have been derived:

1. the application server needs (in all cases) less than 0.02 seconds to formulate the web page that is sent to the user's computer.
2. the user's computer needs 0.187-0.632 seconds to render the page on the screen.

Thus, the overall time needed for a complete cycle "page request-page generation-page rendering" ranges from 0.207 to 0.652 seconds with a single user accessing the platform. For multiple access by 15 users, in the *worst case* that all users will request simultaneously a new page, the time needed to display the page will range from 0.487 seconds to 0.932 seconds<sup>1</sup>.

The only case that the application server needs more time to carry out the request and send the results back to the user is the generation of taxonomy files (effectively a compilation of the taxonomy tree to an easily browsable structure). This operation takes from 15 seconds to 32 seconds to complete, depending on the size of the taxonomy and the machine speed. However, it has to be noted that taxonomy generation is a quite infrequent action, accounting for less than 0.75% of the actions executed in the SmartGov development environment.

Based on the above, 99.25% of actions in the development environment are completed in less than 1 second, thus the development environment performance criterion is met.

#### 4.1.2 Integrator

The Integrator module undertakes the compilation and deployment of services developed using the SmartGov platform to produce executable images that can be run in the context of a web application server (the SmartGov project uses Apache Tomcat for this purpose). If the service is compiled successfully, the Integrator arranges for it to be deployed to the web application server.

When both the semantic and the visual part of a service have been developed, the integration step commences. Typically, a number of integration attempts fail, due to incomplete or erroneous definitions of SmartGov platform elements (e.g. a numeric element may be provided with a non-numeric default value); in these cases, the service designer should use the SmartGov platform to correct the offending definitions and then retry the integration step. Other potential sources

---

<sup>1</sup> Note that only the time required by the server to formulate the web pages is multiplied by the number of users, since each page is rendered on a different computer.

of integration failures are the mismatch between the semantic and the visual part<sup>2</sup>, omission of required tags (e.g. the FORM\_BEGIN tag) and, finally, failure to provide a visual part for some forms. These errors are usually corrected by editing (or providing) the visual part of the forms.

After successful compilation and deployment, the service is used in a test environment to ensure that validation checks have been correctly expressed and ascertain that the overall service image matches the original design. If any amendments are required, the respective semantic or visual elements should be updated, and the compile-deploy-use-modify cycle is performed again.

For this procedure to be efficient, it is important that errors are detected as early as possible. This is particularly true for the first class of errors, i.e. the ones that prevent the successful service compilation, since only one error of these classes can be flagged at each compilation. The second class of errors includes an interactive part (use and assessment of the running service), which is by nature more lengthy, and enables the identification of multiple required adjustments within a single compile-deploy-use-modify cycle; consequently, for this scenario the performance of the Integrator is not as critical as in the previous one. Note that in the latter case service compilation and deployment is successful, thus the time needed is the maximum time required by the Integrator module.

Another important performance aspect of the Integrator pertains to the time needed to replace a running service on the publicly accessible deployment server. This is critical, in order to enable small corrections to be made to a running service, without interrupting normal service delivery and causing inconvenience to service users. For major modifications, the delivery of the service should be stopped before the new service image is deployed (effectively, the new image may be deployed only after the active service user pool has been drained). This duration is a fraction of the full compilation-deployment task duration.

All the aforementioned performance aspects of the Integrator were analysed using three configurations, grossly mapping to setups that may be employed by small, medium and large PAs. The "small" configuration uses a single machine for the XML repository, the Integrator module and the deployment server<sup>3</sup>. The "medium" configuration uses a one machine for the XML repository and the

---

<sup>2</sup> Normally, this is not bound to happen if the visual forms have been tagged through DreamWeaver or the SmartGov simple HTML editor, using an up-to-date tag set. However, the possibility of using outdated tag sets or manually entering tag names still remains, and this is a potential source of errors.

<sup>3</sup> Note that this configuration includes potential security hazards, since the Integrator and the XML repository are installed on a publicly accessible machine; thus this configuration is not recommended.

Integrator and a second machine for the deployment server. Finally, the “large” configuration uses three distinct machines, one for each role. For each setup, services of varying sizes were compiled, starting from simple two-form services with 40 TSEs approximately, to more complex five-form services with more than 100 TSEs. The results of the performance analysis are summarised in Table 4.

<b>Task / Configuration</b>	<b>“Small”</b>	<b>“Medium”</b>	<b>“Large”</b>
Detection of “class 1” errors (preventing successful service compilation)	1 – 5 sec	1 – 5 sec	1 – 3 sec
Successful compilation and deployment (including compilations/deployments for detecting “class 2” errors)	15 – 43 sec	13 – 39 sec	11 – 36 sec
Deployment only	3 – 6 sec	2 – 4 sec	2 – 4 sec

**Table 4 - Performance of the Integrator module**

The figures in the table above indicate that the Integrator module fully satisfies the performance criterion specified in D81, stating that “75% of actions (are executed in) less than 60 seconds”, since 100% of the actions are executed in less than the specified time limit.

Regarding the “hot replacement” of running services, it has been determined that no disruption to service delivery occurs; some users may only experience a small delay (in the range of 3-5 seconds), which is due to process of compiling the service JSPs upon their first use.

#### **4.1.3 Service access by end-users**

The pilot service developed for GSIS has been made available for internal users to evaluate since late October and to all registered TAXISnet users (approximately 180.000 individual tax payers and professional accountants) since early January. For internal users, both the server performance (i.e. the time needed by the server for formulation of the service pages) and the “end-to-end” time (i.e. the time elapsing from the user’s click on the navigation item to the displaying of the page on the browser) were evaluated. For external users, only the server performance data could be directly measured, but the evaluation sheet presented to the users contained a specific question regarding the service performance (“Was the system response satisfactory?” accepting ratings from “Very satisfactory” to “Not satisfactory at all”). In both the internal user evaluation and

the external user evaluation, a sub-optimal hardware configuration was selected, with a single machine hosting all pertinent SmartGov services (deployment server, IIG and XML document repository; the user database resided on the central TAXISnet server, to which the SmartGov platform was linked). Users always accessed the service from dedicated client machine, separate from the server.

Regarding the time needed by the server to formulate the service pages, this has been measured to range between 0.23 msec and 1.04 msec, depending on the number of TSEs on the page and the number of rows in repeating groups<sup>4</sup>. Thus, with an average of 0.635 msec per page, the server can generate 1,575 pages per second. Given that -in the tested service- users spent 53.2 seconds on average in a single page before moving to a next one, the server can efficiently withstand approximately 70,000<sup>5</sup> concurrent service users, without the appearance of any bottlenecks. This is considered a very satisfactory metric, since the maximum number of concurrent users noted for the GSIS services is not expected to exceed 4,000.

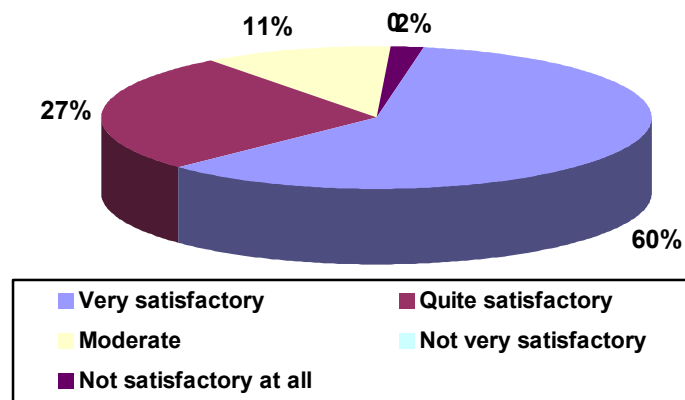
The "end-to-end" time for internal users has been measured to range between 0.114 and 0.378 seconds, depending on the page complexity (number of elements, structure of nested HTML tables etc), and the hardware configuration of the client computer. Since the "end-to-end" time is below the 1-second threshold (which is significant for interactive applications), the service performance is considered to be quite adequate.

Finally, the end-users' responses breakdown to the system performance evaluation question is depicted in Figure 5.

---

<sup>4</sup> In these statistics, the first reference to a page (which requires a compilation of the java server page) has been excluded.

<sup>5</sup> The multiplication  $1575 * 53.2$  gives 83,790 users; however servers approaching their capacity present saturation effects leading to reduced performance, so the limit of 70,000 is adopted as a more realistic capacity threshold.



**Figure 5 - End users' evaluation of system responsiveness**

The breakdown shows that end-users are contented with the service's performance, since an 87% of them rates the performance as "quite satisfactory" or better.

#### 4.1.4 Communication services

The SmartGov communication services (SmartGov Agent-Information Interchange Gateway) are an integral part of SmartGov service delivery since they cater for storage and retrieval of XML documents submitted by users, as well as user authentication. The SmartGov communication services also act as wrappers to procedures of creating pre-populated XML documents through registry querying and populating organisational system registries using the data in the documents submitted by the service end-users.

The performance of the communication services was analysed using three configurations, mapping to "small", "medium" and "large" installations. In the "small" configuration, a single machine was used to host all pertaining SmartGov services (SmartGov Agent, Information Interchange Gateway, Organisational Information System). In the "medium" configuration, one machine hosted the SmartGov Agent, and a second machine hosted the Information Interchange Gateway and the Organisational Information System. Finally, in the "large" configuration, a different machine hosted each SmartGov service. In all cases, a database was used as the organisational information system.

Performance evaluation was conducted for the following services: user authentication, XML document retrieval, XML document storage, creation of pre-populated documents and storage of XML documents into the organisational information system. Note that the creation of pre-populated documents is executed *in place of* XML document retrieval when initial documents are created, whereas storage of XML documents into the organisational information system is

executed *together with* XML document storage (thus in the latter case the overall time is the sum of the two individual times). The results of the performance evaluation are listed in Table 5.

<b>Service / configuration</b>	<b>"Small"</b>	<b>"Medium"</b>	<b>"Large"</b>
User authentication	0.63 sec	0.52 sec	0.35 sec
XML document retrieval	0.66 sec	0.57 sec	0.38 sec
XML document storage	0.82 sec	0.70 sec	0.59 sec
Creation of pre-populated documents	0.72 sec	0.65 sec	0.49 sec
Storage of XML documents into the organisational information system	0.18 sec	0.18 sec	0.11 sec

**Table 5 – Performance evaluation of Communication Services**

The results in the table show that the communication services satisfy the performance success criterion set in D81 "75% of actions (are executed in) less than 5 seconds".

Although the performance of the communication services fulfils the performance success criteria, further optimisations of this software component are still possible; currently, the communication services modules (the SmartGov agent and the Information Interchange Gateway) read and process their configuration files upon the receipt of each individual request. This approach gives greater flexibility to the system administrators, allowing for changes to configuration files to be done at any time and take effect immediately without disrupting the operation of the communication services; on the other hand, reading and processing of configuration files has been measured penalise performance by 0.25-0.42 seconds, depending on the size and complexity of configuration files. Thus for large installations, variants of the communication services may be developed which will read and process configuration files on startup only, storing their contents into efficiently searchable memory structures. This approach will enhance the performance of the communication services, necessitating however the restarting of the relevant servers when changes to the configuration files need to take effect.

## **4.2 Reliability**

In regard to the SmartGov's reliability the following characteristics were identified as measurable aspects of the system's reliability:



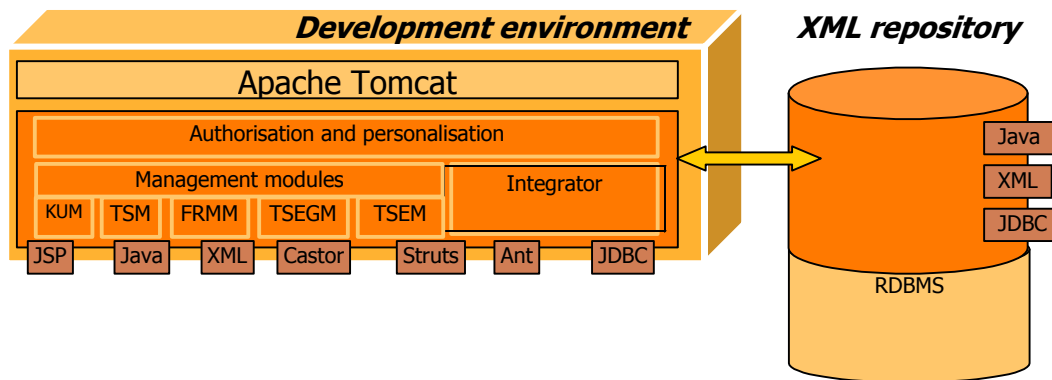
- Uptime,
- number and type of misbehaviours,
- number and type of system errors

Uptime refers to the elapsed time that the system has been running since it was last started and is represented as a percentage of the running time per total time. Misbehaviours refer to incidents where the system behaved differently than it should. Finally, errors refer to incidents where the system did not work at all and can be characterised as critical when the system had to be restarted and non-critical when only the process had to be repeated.

Each of these aspects is used to characterise the reliability of each of the SmartGov's components as well as the developed pilot applications.

#### 4.2.1 Development environment

The development environment is based on Apache Tomcat web server, as it is depicted in Figure 6.



**Figure 6 – Development environment architecture**

The user interface of the development environment consists of HTML pages that are being dynamically created by Java Server Pages running at the web server. Misbehaviours that were identified during the integration phase were subsequently corrected by new versions of the components. During the pilot application development phase, the uptime for the development environment was 100%, meaning that no critical system errors arose. Additionally neither new misbehaviours appeared, no non-critical system errors emerged. Overall the development environment has proven to be very reliable which can be attributed to both the extensive testing during the integration phase and the stability of the underlying third-party software products.

As a final point, an issue was identified in the operation of the SmartGov front-end, which cannot be classified as a subject for the SmartGov front-end itself, but

rather as an overall platform setup matter. The symptom of this issue was the inability of users to login to the SmartGov development environment, if:

1. the user credentials repository was stored in a MySQL database
2. no user login actions to the SmartGov development environment were performed for a period of eight hours.

Initially, the problem was addressed by restarting the Tomcat server hosting the development environment. Finally the root of the problem was associated with the default configuration of the MySQL database, which closes connections after an inactivity period of eight hours. Changing the inactivity period after which connections are closed to 31536000 seconds (one year)<sup>6</sup> has fully resolved the problem.

## **4.2.2 Communication services**

Communication services include the SmartGov Agent (SGA) and the Information Interchange Gateway (IIG). The two software components were extensively tested at GSIS pilot application where communication with back-end, legacy systems was necessary.

### **4.2.2.1 Uptime**

No critical system errors were reported during the testing phase of the pilot applications, resulting in an uptime of 100%.

### **4.2.2.2 Number and type of misbehaviours**

A bug was identified in the storage to third party systems component, which failed when no value at all was provided for specific TSEs. The bug was corrected in a subsequent release of the communication services.

### **4.2.2.3 Number and type of system errors**

No system errors occurred during the testing phase of the pilot applications.

## **4.2.3 Integrator**

The Integrator component is responsible for reading the forms created at the development environment, retrieving the corresponding Transaction Service Elements and Knowledge Units, producing the resulting e-service and uploading it

---

<sup>6</sup> This has been accomplished through adding the lines  
set-variable = wait\_timeout=31536000  
set-variable = interactive\_timeout=31536000  
to the c:\my.cnf file, or changing the values in the corresponding existing lines.

at the deployment server. The Integrator is executed whenever the developer wants to upload a new service at the deployment server.

#### 4.2.3.1 Uptime

The “uptime” criterion is not applicable to the Integrator component, since it is not a system service but rather a program invoked when the relevant task should be performed.

#### 4.2.3.2 Number and type of misbehaviours

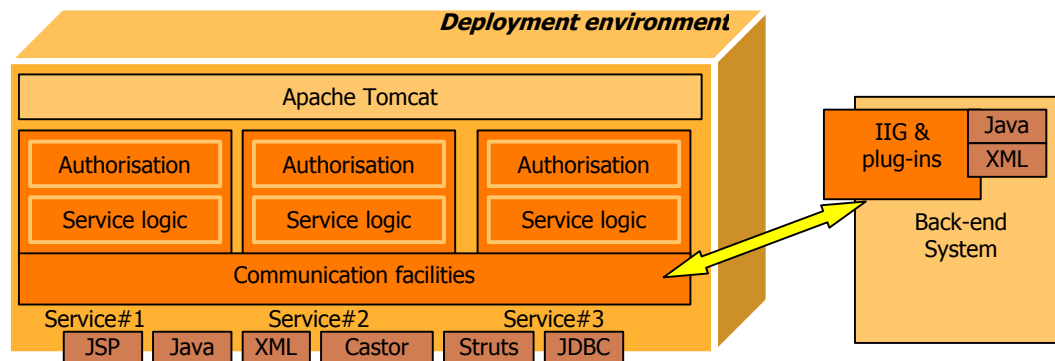
Though the integrator always presented the expected behaviour in terms of the final result type (i.e. compilation/deployment rounds that should succeed indeed succeeded and compilation/deployment rounds that should fail did fail), in the cases of failure the emitted messages were not as informative as expected. For instance, the lack of a required default value was reported as a “null pointer exception”, and the omission of a required tag (e.g. the FORM\_BEGIN tag) from a form was reported as negative string index exception. In all cases however, the object containing the offending definitions was accurately reported, and after some experience was amassed, the staff conducting the integration could pinpoint the exact error fairly efficiently. A checklist to which the staff conducting the integration could refer to in the case of errors has also proven helpful.

#### 4.2.3.3 Number and type of system errors

No system errors occurred during the testing phase of the pilot applications.

#### 4.2.4 Service access by end-users

The deployment environment is based on Apache Tomcat web server, as it is depicted in Figure 7.



**Figure 7 – Deployment environment architecture**

#### **4.2.4.1 Uptime**

This part of the evaluation considers only the GSIS pilot application (e-VAT), since the CEC pilot application has not been running for a sufficiently long time to enable the production of sound statistical data. The GSIS pilot application service has been running for three months, without any problems either in the form of misbehaviour or in the form of system errors appearing. So far, the e-VAT service has proven very reliable and has continued to operate in all cases. It has to be noted that the connection to the TAXISnet back-end for the purposes of performing user authentication and retrieving values for creating pre-populated documents has also operated in a trouble-free manner for a period of 2 months.

### **4.3 Compatibility**

The section “compatibility” discusses the hardware and software platforms on which the SmartGov components have been tested and have been found to operate. In the cases that some SmartGov software component is remotely accessed by a client machine (e.g. the development environment is accessed from the SmartGov stakeholders’ machines), both the server and the client configurations are discussed.

#### **4.3.1 Development environment**

The SmartGov development environment is installed on the SmartGov development server and accessed by the SmartGov platform stakeholders’ computers through a web browser.

The SmartGov development environment is implemented using Java Server Pages (JSPs). The JSPs implementing the front-end environment need to access a relational database for the management of the user credentials and the XML repository for the management of SmartGov objects (TSs, KUs, TSEs etc). From the tests that have been conducted, the following hardware/software compatibility lists have been derived:

##### **4.3.1.1 Development environment**

The development environment has been tested to operate on the following platforms:

1. Hardware: PC; OS: Windows platform (2000 server); JSP container: Tomcat 4.1.x; SmartGov object management: XML repository v.2 on top of MySQL 4.x or SQL Server 2000; User credential management: MySQL 4.x or SQL Server 2000 or Oracle 8.x

2. Hardware: PC; OS: SuSE linux; JSP container: Tomcat 4.1.x; SmartGov object management: XML repository v.2 on top of MySQL 4.x; User credential management: MySQL 4.x
3. Hardware: Sun SPARC; OS: Sun Solaris; JSP container: Tomcat 4.1.x; SmartGov object management: XML repository v.2 on top of MySQL 4.x; User credential management: MySQL 4.x

In the latter two configurations, the MySQL DBMS storing the user account database must be configured to be case insensitive for table and column names.

This compatibility list is a superset of the initially targeted compatibility list, which included only the first item.

#### **4.3.1.2 Client configuration**

The following configurations can be used for accessing the SmartGov development environment:

1. Windows platform (NT, 2000, XP, 95, 98 or ME) with Internet Explorer 5.5 or newer; or
2. Windows platform with Netscape Communicator 6.0 or higher; or
3. Sun SPARC Solaris platform with Netscape Communicator 6.0 or higher; or
4. PC with Linux<sup>7</sup> and Netscape Communicator 6.0 or higher; or
5. Macintosh with MacOS X and Netscape Communicator 6.0 or higher; or
6. Macintosh with MacOS X and Safari browser

This compatibility list is a superset of the initially targeted compatibility list, which included only the first two items.

#### **4.3.2 Deployment environment**

The deployment environment provides a framework for users to access the services created using the SmartGov platform. The services comprise of automatically generated Java Server Pages and pre-made Java libraries and store and retrieve XML documents to an XML repository. The services are accessed through a web browser. From the tests that have been conducted, the following hardware/software compatibility lists have been derived:

##### **4.3.2.1 Deployment environment**

The deployment environment has been tested to operate on the following platforms:

---

<sup>7</sup> SuSE linux was used in the tests

1. Hardware: PC; OS: Windows 2000 server; JSP container: Tomcat 4.1.x or Tomcat 5.x; XML storage and retrieval: XML repository v.2 on top of MySQL 4.x or SQL Server 2000; or
2. Hardware: PC; OS: SuSE linux; JSP container: Tomcat 4.1.x or Tomcat 5.x; XML storage and retrieval: XML repository v.2 on top of MySQL 4.x
3. Hardware: Sun Sparc; OS: Sun Solaris 8; JSP container: Tomcat 4.1.x or Tomcat 5.x; XML storage and retrieval: XML repository v.2 on top of MySQL 4.x

Note that when Tomcat 4.1.x is used as a JSP container, the services must be manually deployed, due to a bug in the undeployment feature of Tomcat 4.1.x.

This compatibility list is a superset of the initially targeted compatibility list, which included only the first item.

#### **4.3.2.2 Client configuration**

The following configurations can be used for accessing the SmartGov development environment:

1. Windows platform (NT, 2000, XP, 95, 98 or ME) with Internet Explorer 5.5 or newer; or
2. Windows platform with Netscape Communicator 6.0 or higher; or
3. Sun SPARC Solaris platform with Netscape Communicator 6.0 or higher; or
4. PC with Linux<sup>8</sup> and Netscape Communicator 6.0 or higher; or
5. Macintosh with MacOS X and Netscape Communicator 6.0 or higher; or
6. Macintosh with MacOS X and Safari browser

This compatibility list is a superset of the initially targeted compatibility list, which included only the first two items.

#### **4.3.3 Communication services**

The communication services provide the linkage between the SmartGov deployment environment and third party information systems. The communication services have run successfully on all the deployment environments listed in 4.3.2.1. Regarding the third party systems, the communication services have been successfully tested to run on:

1. Windows platform (NT, 2000, XP, 95, 98 or ME)
2. Sun SPARC Solaris platform
3. PC with SuSE Linux
4. Macintosh with MacOS X

---

<sup>8</sup> SuSE linux was used in the tests

In particular for database connectivity, the SmartGov back-end communication services have been tested to successfully interoperate with the following DBMSs:

1. Oracle 8.0.x, Oracle 8.1.x, Oracle 9.x
2. SQL Server 7 and SQL Server 2000
3. MySQL 3.x and MySQL 4.x

As a note for database connectivity, non-English characters (i.e. characters that do not fall within the standard ASCII alphabet, having ASCII codes greater than 127; this includes accented characters and characters used in other languages e.g. Greek, Cyrillic, etc) may need special handling in the custom modules that will be developed. In more detail, if the JDBC driver provides automatic conversion to UTF-8 characters (e.g. Oracle 9.x), no special handling is necessary. If however the JDBC driver does not convert automatically non-English characters to the appropriate UTF-8 code page, this must be done programmatically within the custom module.

#### **4.3.4 Integrator**

The Integrator component has been built as a set of Java Server Pages and libraries. The Integrator reads the SmartGov definition from the XML repository, compiles the service and deploys it to the deployment server. The integrator has been tested and found to operate successfully in the following configurations:

1. Hardware: PC
2. OS: Windows 2000 server or workstation
3. XML repository: 2.0, operating on SQL server 2000 or MySQL 4.x
4. Deployment server: Tomcat 5.x

The compatibility list exactly matches the target compatibility list.

### ***4.4 Evaluation of SmartGov Installation Procedure***

The complete SmartGov Platform is packaged as the following set of components

- a. Front End
- b. Integrator
- c. Information Interchange Gateway and SmartGov Agent (IIG/SGA)
- d. DreamWeaver add-in to facilitate creation of forms (DWExport)

The components (a) and (d) relate to the design environment, the Integrator performs the compilation of designed services so that they can operate in a

runtime environment, and the IIG/SmartGov Agent (SGA) support communications with the runtime services.

This evaluation was performed by Napier University, who while being a member of the Consortium, played no part in actually coding the software, but importantly, their skills and experience were comparable to a typical IT support engineer seeing the SmartGov software for the first time.

In this Section we consider the installation procedure for the software as documented in the manual (D8.2). The installation methodology for each of the above components will be considered in turn, their strengths and weaknesses will then be evaluated. Finally future enhancements that could be made to subsequent releases will be discussed.

#### 4.4.1 The Installation Procedure

The installation requires software components to be installed in a specific sequence, with dependencies on both SmartGov and 3<sup>rd</sup> party software components. These requirements are summarised in the following two tables.

Required Component	SmartGov Component			
	Front End	Integrator	IIG/SGA	DWExport
<i>Java SDK 1.4.2 or higher</i>	√	√	√	√
<i>Tomcat 4.1 server</i>	√	√		
<i>Tomcat 5.0 server</i>			√	
<i>SQL dbms</i>	√	√	√	√
<i>Struts</i>	√	√		
<i>Dreamweaver</i>				√

**Table 6: Required 3rd party software.**

Required Component	SmartGov Component			
	Front End	Integrator	IIG/SGA	DWExport
<i>SG Integrator</i>				√
<i>SG Front End</i>				√

**Table 7: Dependencies on SmartGov components.**



DWexport does not strictly depend on either the Integrator or the Front End tool to operate, however without the installation of the Integrator it is unlikely to have access to a valid properties file specifying the database connections it needs to make, and without the Front End tool, services cannot be created and added to the SmartGov repository.

Each component of the software is bundled separately, and is installed in isolation to the others (subject to the pre-requirements described above). Their packaging and method of deployment vary. Their key differences are summarised in the Table 8.

	<b>Front End</b>	<b>Integrator</b>	<b>IIG/SGA</b>	<b>DWExport</b>
<i>Bundled as</i>	6 files	2 files	2 files	4 files
<i>Size</i>	10.0MB	22.0MB	2.18MB	3.38MB
<i>Installer</i>	×	√	√	×
<i>Latest release</i>	17 Oct 2003	31 Oct 2003	31 Oct 2003	30 Oct 2003

**Table 8: Summary of SmartGov Platform components installation.**

An initial observation is that only half of the four components come provided with an installation utility. Considering such items are usually the norm nowadays this might be seen as a disadvantage. However the other components are archived as Zip files, and so can be readily extracted by standard, commonly available tools. We shall reserve comment on the relative merits of the two approaches until the next Section.

The full installation procedure for each component in turn will now be detailed.

#### **4.4.1.1 Front End**

The installation procedure is as follows:

1. Create and initialise structure for front end databases:
  - a. Unzip the archive: sqlScripts.zip
  - b. Execute three sql scripts within chosen SQL dbms to create the three Front End SmartGov databases required.
2. Install and configure the Front End application:
  - a. Unzip the archive SmartGov.zip in the Tomcat webapps directory.
  - b. Set correct properties file for desired SQL dbms by renaming appropriate file smartgov.properties.XXX to smartgov.properties.

- c. Edit file: SmartGov\scripts\integrator.properties file (or copy it from the Integrator's configuration file of the same format)
3. Initialise and populate the SmartGov repository with the knowledge units and taxonomy data.
  - a. Initialise the repository by executing the batch file: createXmlRepository.bat
  - b. Upload pre-defined platform data by executing the batch file: documentCrawler.bat
    - i. Unzip the archive NapierKnowledge.zip
    - ii. Locate the taxonomy definitions in the directory: NapierTaxonomy
    - iii. Locate the knowledge units definitions in the directory: NapierKUs
    - iv. Upload them into the repository
  - c. Login to the Front End as 'user\_expert', enter the taxonomy and choose 'Generate File' option to complete the procedure.

#### 4.4.1.2 Integrator

Installation of the Integrator is achieved through an executable installation utility. To complete the procedure there is a requirement to navigate through 10 screens. These screens are as follows:

1. Select installation set: Usually just choose Normal installation.
2. Select installation directory: This is just to hold uninstall information and any arbitrary choice can be made.
3. Select shortcut folder: This provides all conceivable options as to where to locate the shortcuts. The default of a separate program group is the most useful.
4. Select Tomcat servers:
  - a. Development server specified by its root directory
  - b. Deployment server specified by machine name and port number.
5. Configure connection parameters for XML repository:
  - a. Class name: This an internal SmartGov class and needs to be left as is.
  - b. Database type: Accepts only *Microsoft SQL Server 2000* and *MySQL 4.x*.
  - c. Classpath root folder: This is the path in the directory tree for the root directory of the JDBC driver. Here care must be taken in entering it as in contrary to the Windows/DOS convention of using a backslash as path separator the Unix forward slash separator must be used.

- d. DataSource class: The name and class-hierarchy of the Java class in the JDBC driver used to access data.
- c. Database name and login details.
6. Input and output file storage.
  - a. Location to temporarily store generated services prior to deployment.
  - b. Location of xhtml files: These are the tagged html forms used in the service.
7. Location to store SGA configuration file.
8. Pre-installation summary.
9. Installation screen.
10. Installation completed screen.

#### **4.4.1.3 Information Interchange Gateway and SmartGov Agent**

Preparatory SmartGov configuration is required prior to installation of the IIG/SGA. This consists of creating up to four pending action queue databases, creating and populating a login database to allow users to access services, and a runtime XML repository to hold submitted documents. The installation instructions detail the following procedure for this:

1. Create SGA/IIG pending action queue (PAQ) DBs.
  - a. Log into your chosen DB client and create an empty database by hand.
  - b. Run the appropriate script for the DBMS from those listed in Appendix A of the Installation Guide to set up its structure.
2. Create and populate the service login DBs.
  - a. Log into your chosen DB client and create an empty database by hand.
  - b. Run the appropriate script for the DBMS from those listed in Appendix B of the Installation Guide to set up its structure.
  - c. Add data to two tables by hand:
    - i. SGUserData: Create username, password and full name entries to set up each user.
    - ii. SGUserServices: Specify the services that a user can access by creating userID, serviceName for each service they should be allowed to use.
3. Create and configure the SmartGov runtime repository.
  - a. Log into your chosen DB client and create an empty database by hand.

- b. Execute the proprietary XML Store creation manager provided at the Integrator installation stage of the platform.
  - i. Create 3 indexes each requiring two text values that must be copied from the user manual.
  - ii. Configure a connection to the database server. Requires 8 values, as per step 5 of the Integrator installation.

The installation of the IIG/SGA is then undertaken through an executable installation utility. Navigation through 15 screens is required for the procedure. These screens are as follows:

1. Select installation directory. (as per Integrator installation)
2. Select shortcut folder: (as per Integrator installation)
3. Choose IIG ports:
  - a. IIG port number.
  - b. SSL IIG port number.
  - c. Allowed SGA IP address: This is the allowable IP address that SGA can connect to.
4. EntraPAQ database connectivity configuration.
  - a. Database name and login details
  - b. DataSource class: The name and class-hierarchy of the Java class in the JDBC driver used to access data.
  - c. Connection string: The URI required to make the connection to the specified database through the JDBC driver. It includes server name, port, and certain dbms specific parameters. The syntax required for MySQL and MS SQL Server are fully documented in the installation manual.
5. AdelantePAQ database connectivity configuration (as per step 4 above).
6. IIG XML repository connectivity configuration (as per Integrator installation step 5 for development XML repository).
7. Login database connectivity configuration (as per step 4 above).
8. Configure the log listeners:
  - a. IIG: Supply hostname, and port
  - b. SGA: Supply hostname, and port
9. SGA – EntraPAQ database connectivity configuration (as per step 4 above).
10. SGA – AdelantePAQ database connectivity configuration (as per step 4 above).
11. SGA – Target IIG configuration.

- a. Plain text IIG host.
- b. SSL IIG host.

There is a small amount of confusion here as though the default in both cases is 'localhost' you are instructed in the text not to use this value, and must in fact specify an IP address.

12. SGA – NI port.
13. Pre-installation summary.
14. Installation screen.
15. Installation completed screen.

#### **4.4.1.4 Dreamweaver Export Tool**

The installation procedure for the tool is very simple:

1. Extract the two zip files (DWExport.zip and libs.zip) to a temporary directory.
2. Ensure JAVA\_HOME system variable is set to the root directory of the Java SDK.
3. The GUI version of the tool can then be invoked through executing the supplied batch file: run.bat (alternatively a command line/batch version of the tool can be invoked where the 4 required parameters are supplied as arguments).

Once installed its role is to generate tags to identify the TSEs created in the SmartGov FrontEnd to allow associations to be made with them within Dreamweaver.

The export tool has two modes of invocation. One is as a command line call with arguments, and the second through a GUI. To perform the export, the export tool, irrespective of mode of invocation requires the following four parameters:

1. XML repository properties file (configuration for access to XML repository).  
The integrator.properties file is most suitable as it will contain all required parameters. Care must be taken if the xml repository is being hosted by a different machine to supply the correct machine name or ip address.
2. Dreamweaver Installation path: Configuration directory of Dreamweaver installation
3. New folder name: Always 'SmartGov'.

4. Locale to export: This specifies the language support and for English 'en' is the identifier.

#### **4.4.2 Evaluation**

The SmartGov Platform is packaged in the four components as described above. Two are deployed through tailor-made installation utilities and the others are supplied as archives which need to be extracted and then configured through a combination of tools and manual means. At first glance this may appear long-winded and daunting, however once experienced (having installed once or twice before) it is possible to fully install each of the components within 20-30minutes giving a mean installation time of 2-3 hours for the whole platform.

The FrontEnd tool is the main component that does not use an installation utility. Once its files have been extracted the main configuration activity is creation and population of the SmartGov repositories. It performs all of these actions using a number of scripts and batch files. The scripts are executed within the database environment to create structure and access permissions, then the batch file invokes utilities that can be used to upload xml files to populate them.

This could be simplified greatly, firstly by concatenation of the scripts into a single entity. Secondly, within mySQL, it is possible to populate tables using scripts and as the initialisation data supplied is static in content this could be incorporated into the single script also. The tools then would remain useful for adding any dynamic content created by platform users for example.

In the first release the IIG/SGA configuration was combined with the Integrator installer. While it is necessary to install all components for full functionality this distinction is nice as usually the Integrator and IIG/SGA are deployed on different machines so the separation is convenient, and it also allows the installation to progress in stages and means that IT staff are not faced with a large number of parameters in one go, many of which they may not initially understand the role of and so find it hard to supply sensible values. Of these values probably those relating to the JDBC driver (required by SmartGov to facilitate access to its own and external databases) are probably the most complicated. During installation a knowledge of the following driver settings is required:

1. JDBC driver name
2. JDBC driver file classpath
3. JDBC driver package class name

This is probably beyond the capabilities of the average IT support staff (though many may well be motivated enough to have learned such things or find them out). It would be far better to supply either an intelligent search facility that would allow the user to navigate the directory structure and find the right driver and configure it automatically, or for a small overhead in size, to bundle the drivers of the two databases supported pre-configured and simply allow selection of the desired one through a drop down box. The latter would provide minimal programming effort and greatly simplify the installation process.

Drop-down boxes could also be useful in a hybrid approach. The three parameters listed above are standardised for a particular driver and do not change between installations, so perhaps a drop-down box where the user chose the database type to get the above settings, then a browse facility to actually locate the driver files on the hard disk is the ideal. This then has the advantage of isolating the IT support staff from low level parameters they may not know or have available to hand, while giving them the freedom to choose the most up to date driver for their platform.

The current installation documentation for the Integrator appears confusing. It was originally intended that the Integrator be installed on the development Platform alongside the Front End on a Tomcat 4.1 server, and deploy to a Tomcat 4.1. However, if a Tomcat 4.1 server is used on the deployment Platform problems may arise during deployment, and the manual does not make this clear. The set-up as defined currently requires the development server to be specified by its root directory, and the deployment server by machine name and port number. The latter settings are fine, but if the integrator is to be deployed on the deployment server – which may be desirable under certain circumstances - then this path must be specified in the first item. Furthermore were the deployment server to reside on a different machine to the development one certain parameters later on in the install have to be chosen very carefully.

Next in the Integrator installation comes the JDBC driver configuration. For the first entry in this step the manual advises to leave the value unchanged, as it

relates to some internal configuration parameter for the SmartGov Platform software. Thus the inclusion of this in the installer is somewhat superfluous. It seems to serve only to add to the perceived complexity of the installation process and would be better removed.

There is also an inconsistency in the syntax required for specifying directories between different steps in the installers. In some the instructions specify that '\\' be used as the separator in the file path, whereas for others the '/' notation is required.

A good aspect to the Integrator is that all of its configuration is defined in a single text file. This means that backing up the settings is very straightforward and tailoring the installation (for example if the databases used change) can also be achieved very simply without need to recourse to the full installation utility. Furthermore once a working installation of the Integrator and IIG/SGA have been achieved it is possible to save their configuration and then when reinstalling them all of the screens in the installer can be ignored. However, unfortunately the user still has to navigate through them all as no 'reinstall' option is provided.

In the IIG/SGA installer step 3 mixes up the IIG and SGA parameters together on the same screen and makes distinguishing between IIG and SGA configuration parameters a little confusing. There are also the same difficulties with configuration of JDBC drivers as was seen in the Integrator installer, and this is compounded by the fact that these same parameters need to be entered on at least three occasions. Thirdly steps 9-12 of the process can usually be skipped over in a standard installation, as they are only required if the test SGA client supplied with the software is to be used. However, it should be noted that many parameters, including all of the port numbers, have sensible default values that can usually be accepted without changes.

A minor complication with the split installation of Integrator and IIG/SGA is that there is an overlap with some of their configuration parameters (primarily that the integrator needs to pass some SGA/IIG details onto the services it compiles). To this end the Integrator comes with a set of default SGA configuration files in a zip archive, for use by these compiled (runtime) the services, however if settings other than the defaults are used these need to be modified by hand.



To manage user accounts the FrontEnd tool has a nice account manager. In a beta iteration of the Platform user access to runtime services was granted through amending values to the login database by hand. While this approach to user account creation might be acceptable for a relatively small pilot application, this was deemed extremely impractical on a full-scale service with 1000s of registered users. As a result a GUI-based account manager was developed. within the scope of the project and has now been made available.

A number of minor points and niceties pertaining to the SmartGov installation procedure have been expressed above. Overall, on first encountering the process it appears to be extremely daunting and certainly not for the faint-hearted. However once familiarity has been achieved it is possible to install the platform quickly (in under 2-3 hours). Apart from minor inconsistencies with the file separators throughout the procedure the only real difficulty is in specifying the Java class names when configuring the JDBC drivers.

#### **4.4.3 Future Enhancements**

The installation procedure for the SmartGov software as it stands works well enough, though for some components (specifically the Integrator and IIG/SGA) specialist knowledge is required that may be beyond that which typical IT support staff might reasonably be expected to know. The bundling of the software as a series of discrete components with varying set-up mechanisms (zip or installer) also adds slightly to the installation effort required.

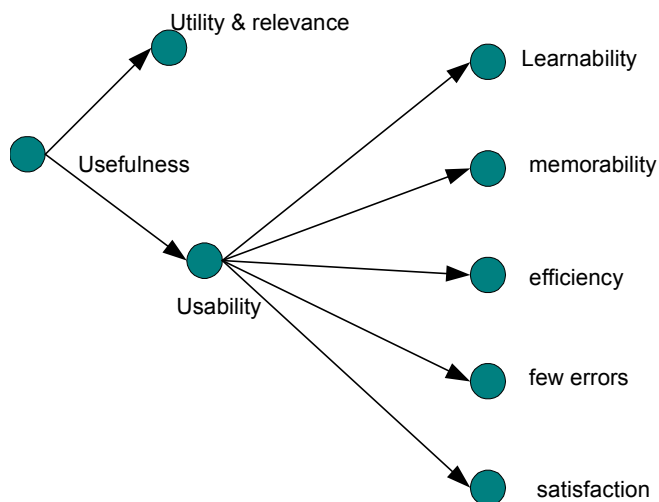
Adopting some of the following simple refinements in future releases of the Platform could enhance the installation process.

1. Amalgamate all of the database creation and population activities into a single script. This would enable a simple installation option for less experienced users. For more advanced configuration the document crawler tool should remain available.
2. Having a drop down list of supported drivers for selection in sections requiring JDBC configuration, that (silently) supplied the appropriate JDBC connectivity parameters. This might be done in conjunction with bundling the supported drivers with the platform or providing a browse facility to locate the installed driver files on the server.

3. Given that all of the above components are necessary, the ideal scenario would be to bundle them as a single unit, having a menu that automatically starts up, and lists the components available to be installed.
4. As a final nicety the option of standard/advanced install options and the ability to do a reinstall that does not overwrite existing configurations, would greatly enhance maintainability of the installed platform.

The first recommendation would reduce the effort involved in installing the Front End Tool, and the second would remove the need to have knowledge of low-level JDBC parameters that could cause difficulty or be easily got wrong. The final two recommendations while being less important would increase the general 'user-friendliness' of the installation process, and give it coherence.

## 5 Usefulness



**Figure 8 Usefulness nodes**

This section is concerned with those criteria that reside under the heading "usefulness" as shown below. The section begins with an outline of training provided to users prior to undertaking the evaluation. This was done to provide a basic level of competence across users so as to facilitate a more meaningful evaluation. This is followed by an examination of each success criterion directly, using the methods described earlier.

A training strategy and session was developed and used to provide basic platform training prior to evaluation of the usefulness of the platform. The session lasted about one working day. The user guide D8.2 was used as a reference manual throughout the training and development period. The document "Smartgov Illustrated" (available on the Smartgov project web site [www.smartgov-project.org](http://www.smartgov-project.org)) was provided as a training manual.

The training session followed the structure and content of the Smartgov Illustrated manual, with the reference to the user guide, D8.2, where appropriate. Users used a live Smartgov system during the training session and were required to perform tasks as the session progressed. A presentation was included with visual material from the user guide to assist trainees. The user guide was limited to those sections of direct relevance to the work of the domain experts (outlined below).

In this session the users were invited to create elements without direct supervision from the trainer, using only the User Guide D8.2 as a reference material. Users were invited to create some of the following elements; KU, TSE, TSE Group, as a test of their system understanding and the user guide. There were no set limits or targets in terms of quantity of elements created.

For the City of Edinburgh Council pilot, there were five persons present as trainees. These were the Occupational Therapists normally responsible for the management of the existing service. As such they satisfy the domain expert role.

Each of the tables below describe the following:

- Criterion: Given as the sub-section heading and listed in the order given stated in D8.1 (section 5.2.2), and numbered accordingly.
- Role: These follow those used in defining the User Requirements: -
  - Managers;
  - Domain Experts;
  - Service Workers;
  - IT Staff;
  - End Users- Internal
  - End users- External
- Metric: As stated in D8.1 section 5.4, with minor updates.
- Data source: question type; yes/no rating, comment, or interview topic. Rating scales were generally of the "How much do you agree with the following statement" format. In some cases 'operational data' (op's data)

is stated instead of a question type, referring to targets based on baseline data gathered by the service concerned.

- The final two columns give results for each site in a format matching that of the criterion.

The questions were compiled into questionnaires and checklists for each site, and for particular roles. Others were used in interview schedules. A minority was based on observational data, gathered from field study.

Note that since 'roles' will vary and in some cases individuals may have overlapping roles, whether formally defined by their job title or informally. We asked questions according to the roles that people play in the pilots, which should normally be the roles that are typically expected of someone in their position in the organisation. For example, if 'domain experts' are particularly knowledgeable about IT they were not classed as 'IT Staff' unless they were expected to perform that role normally.

With each table there is an analysis of the result combined with a description of the data sources & questions, which were adapted to fit each of the above metrics, and to be more specific to the role and the pilot site.

## 5.1 Utility and Relevance

### Reusability of elements (TSEs, KUs etc)

Success Criterion	User role	Data source	CEC result	GSIS result
<i>Stakeholders should rate the reusability "satisfactory" or better in 70% of test cases.</i>	Domain-experts CEC/GS IS	Rating scales	95%	100%

After completing the test service, domain experts are asked to assess the applicability and ease of re-using (a) TSE's and (b) KU's. 'Test case' refers to a domain expert's assessment of one TSE or KU.

In the CEC pilot context the combined results over the four metrics is that respondents rated reusability satisfactory or better in 95% of cases. Thus for CEC this success criterion has been met.

In the GSIS pilot context the combined results over the four metrics is that respondents rated reusability satisfactory or better in 100% of cases. Thus for GSIS this success criterion has been met.

### Relevance of elements (TSEs, KUs etc)

Success Criterion	User role	Data source	CEC result	GSIS result
<i>The relevance of TSEs and KUs should be rated as satisfactory by 70% of users.</i>	Domain-experts	Rating scales	90%	100%

What is being assessed here is whether a domain expert, who thinks there may be a relevant TSE or KU in the system that may be useful for a new service, can retrieve relevant TSE's/ KU's from all those that exist. This can be considered a test of the effectiveness of the taxonomy.

In the CEC pilot, from the responses given in the development environment context, we can state that relevance was rated as satisfactory or better in 90% of cases, thus satisfying the success criteria.

In the GSIS pilot, from the responses given in the development environment context, we can state that relevance was rated as satisfactory or better in 100% of cases, thus satisfying the success criteria.

### Minimal IT input needed for service design and development

Success Criterion	User role	Data source	CEC result	GSIS result
<i>time involved: Less than 10% of design time requires input from IT professionals</i>	Domain experts; IT Staff; Managers	Op's data	11.67%	N/A

The 'design time' includes all time spent on service development by the people involved, as recorded in time sheets. Time sheets were therefore collected and the person-hours counted for IT Staff (those employed in support or internal development roles) and non-IT Staff. Where SmartGov research partners, other

than the pilot sites, contributed to service development tasks this was also identified.

In the CEC pilot context the contribution of IT staff time to the development process was 11.67%. This is over the 10% threshold of the success criteria, however the skills of the domain experts and service workers, in this pilot situation do not extend to simple form design. Thus some tasks in the service design process which may normally be performed by service workers or domain experts, have been performed by IT staff in this pilot site. It should also be noted that some of the IT time does include system set-up and installation, in association with Napier University. As IT support is outsourced by CEC the access to exact timings for individual tasks is limited because of its commercial nature. Within this context then is difficult to argue that the results, though over the 10% criterion, constitute a failure to meet that success criterion.

In the GSIS pilot site the domain expert and IT staff user roles were held by the same individuals, thus in this instance the metric was not appropriate.

## 5.2 Usability

Below are the criteria which conform to the subheadings under Usability.

### 5.2.1 Learnability

Success Criterion	User role	Data source	CEC result	GSIS result
<i>Time to learn how to create and modify a service: 75% of users should be proficient in creating and modifying a specified test service after one days training or self-practice.</i>	Domain experts CEC/GS IS	Rating scale	40%	0%

In the case of CEC, only 40% of users claimed to be confident in using the system after one day's training. While in the GSIS pilot site none of the users claimed to be confident in using the system after only one days training. In this case the success criterion has not been met.

Success Criterion	User role	Data source	CEC result	GSIS result
<i>Friendliness/intuitiveness of development environment: Over 70% of users who have received training should rate the service development environment as "<b>easier to learn</b> than current methods for developing online transaction services".</i>	Domain experts	Rating scale	N/A	100 %

In the context of the CEC pilot, 80% of respondents felt that "SmartGov allowed me to learn technical tasks that I would not otherwise have the technical skills to do confidently". However only 20% of them agreed that "Smartgov was easier to learn than other software I have used to develop transaction services".

This metric is rendered invalid by the fact that 100% of respondents claimed no prior experience of online service development. In this case we can draw no meaningful conclusion in terms of this criterion for the CEC pilot.

Within the context of the GSIS pilot, 100% of respondents felt that "SmartGov allowed me to learn technical tasks that I would not otherwise have the technical skills to do confidently" and 100% agreed that "Smartgov was easier to learn than other software I have used to develop transaction services". All respondents claimed to have over 1 years prior experience of online service development. Thus in this pilot site the criterion has been met.

### 5.2.2 Efficiency

*Note: since the metrics relating to 'creating' and 'modifying' a service are described in very similar terms in D8.1 they have been combined in this table.*

Success Criteria	User role	Data source	CEC result	GSIS result
<i>Time to create &amp; modify a service- Design and deployment time: Service designed and deployed within 2 months</i>	Domain experts; IT Staff; Managers	Ops data	N/A	Less than 2 months

As stated previously, CEC have no comparable online service. Therefore this metric is not appropriate for the site. The measurement is straight forward for the GSIS site and here the success criterion is met.

Success Criterion	User role	Data source	CEC result	GSIS result
<p><i>Effort to create &amp; modify a service</i></p> <p>a) <i>HTML forms.</i></p> <p><i>HTML form development &amp; modification requires no additional effort; simple forms should be capable of development in less than one working day; most complex forms should be capable of development in six working days.</i></p>	<p>Domain experts;</p> <p>IT Staff;</p>	<p>Rating</p>	<p>disagree</p>	<p>agree</p>

For the CEC pilot, the responses show that users did not find the development of simple forms to be possible in just one day. However, there was agreement with the development time for complex forms. The CEC user did not agree that “there is no significant extra effort required when using the smartgov platform, compared with other methods for creating online forms”. So we would argue that this criterion has not been met at the CEC pilot site.

Within the GSIS context users agree with all the questions relating to effort required to produce forms and so the criterion is met.

Success Criterion	User role	Data source	CEC result	GSIS result
<p><i>Effort to create and modify a service:</i></p> <p><i>(b) Service knowledge aspects (TSEs, KUs, validation checks).</i></p>				
<p><i>Effort needed to consider and act on the relevant aspects is estimated to be less than with current working practices for translating business logic into a working service;</i></p>	<p>Domain experts;</p> <p>IT Staff;</p>	<p>Questionnaire with ratings</p>	<p>agree</p>	<p>agree</p>



Success Criterion	User role	Data source	CEC result	GSIS result
<i>Effort needed to make changes to the business rules/procedures is less than with current working practices</i>	Domain experts; IT Staff;	Questionnaire with ratings	agree	agree

From the data in the questionnaire we can state that; when measured at the CEC site, effort is identical in two cases and less in the remaining four. Thus this criterion is met.

From the data in the questionnaire we can state that; when measured at the GSIS site, effort is identical in one case and less in the remaining five. Thus this criterion is met.

### 5.2.3 Memorability

Success Criteria	User role	Data source	CEC result	GSIS result
<i>Users should be able to recall how to perform 80% of specific tasks, following one week of non-use.</i>	Domain experts	Question rating scale	44%	100%

After one week (min.) to 10 days (max.) of non-use, all users should give satisfactory responses for at least 4 of the tasks below ...

- Create a TSE
- Create a TSE Group
- Create a KU
- Create a Form Element
- Add something to the taxonomy

As there are 5 tasks the value for CEC does not make statistical sense, however the result does lie between 40% and 60% and so this criterion has not been met.

In the GSIS pilot context, all users reported that they could recall how to perform 100% of the tasks after 1 week of non-use. The criterion is thus met.

### 5.2.4 Few errors

Success Criteria	User role	Data source	CEC result	GSIS result
<ul style="list-style-type: none"> <li><i>Fewer than 20% of trained users should experience serious problems when creating and modifying specific test services</i></li> </ul>	Domain experts	Rating/ observation at CEC	Initially 40%	0%

In addition at CEC, observational data was also gathered and this helped to understand the CEC result.

*Observation in CEC only:* 5 development platform users, were observed performing the tasks supported (i.e. corresponding to their user role), for 1 hr (possibly spread over several sessions). They were asked to seek help when they felt they needed it. The nature of any problems were noted, and their seriousness categorised as follows:-

- *Serious:* user performed the task wrongly or could not continue without help from IT support staff
- *Moderate:* user had difficulties, but was able to continue or successfully try alternatives
- *Minor:* user was able to continue after encountering the problem – i.e. a minor distraction.

The observations carried out at the CEC pilot site we can show the following:

One week after training:

- Serious error: All users lose information as there is no prompt to save before exiting page.
- Serious error: All users lose information if they press the "logout" button when they intend to press "save"
- Moderate error: Platform appeared to generate spurious windows.
- Minor error: All users press return key after entering login details instead of the "submit" button on the page. This does not work and they have to remember to use the "submit " button.
- Moderate error: All users regularly fail to confirm data in editable fields using the yellow "tick" button.
- Minor error: Two of five users misinterpret the use of the value list function.

Approximately two weeks after training:

- Serious error: less than half of users lose data when they press “logout” button when they intend to press “save”
- Moderate error: All users regularly fail to confirm data in editable fields using the yellow “tick” button.
- Serious error: Two of five users fail to adhere to naming convention.

Approximately one month after training:

- Serious error: one user loses data when they press “logout” button when they intend to press “save”.
- Minor error: All users report inconsistency between naming of certain fields in the platform
- Moderate error: All users regularly fail to confirm data in editable fields using the yellow “tick” button.
- Moderate error: The ‘section visibility’ functionality is not understood – may be a more fundamental design issue? (all users)
- Moderate error: The prev link is not highly visible, nor is its function obvious since it is nowhere near the ‘next link. (all users)

From observation we are able to show a decline in the incidence of serious errors, to the point where, in the final session only one of the five users experienced such an error. Thus, one could argue that less than 20% of trained users experience serious errors when using the platform.

The questionnaires show that in the GSIS pilot none of the users experience serious problems with the platform, and none had any issues with the language used. However, in the CEC context we see that some users did have problems with using the platform, certainly in the initial sessions. All users stated that they had difficulties with the language used within the platform. This later point reflects the difference in the nature of the users at the two sites as discussed previously.

### 5.2.5 Satisfaction

Success Criteria	User role	Question type	CEC result	GSIS result
<i>70% or more of the development environment users should rate it as “preferable to the current methods”.</i>	Domain experts; IT Staff	Rating	N/A	100%

### **5.3 Analysis**

As can be seen from the results for each of the criteria above, there is significant disparity between the results from each of the pilot sites. This disparity is largely due to the fact that the service developers in CEC have limited IT knowledge and experience, whereas in the GSIS site the service developers constitute both IT experts and service domain experts. This presents a useful comparison, as the smartgov platform could be deployed in a wide variety of public authority contexts with a spectrum of user capabilities and experience.

Both pilot sites returned a similarly high evaluation of the utility and relevance of the platform, however in CEC the proportion of IT staff input into the service development exceeded the 10% criteria. As discussed, there are a number of mitigating factors here, so one cannot conclude that this constitutes a failure to meet that criterion. Overall the platform can be said to provide a high level of utility and relevance for the two user groups.

Usability of the platform is assessed through criteria under the five sub-headings;

*learnability, efficiency, memorability, few errors and satisfaction.*

While in both sites users did not feel confident with the platform after only one day's training, thus failing to meet that criterion, there was a marked difference between the sites, when comparing the platform with other methods. This is most probably due to the disparity of levels of IT experience between the two user groups. Overall the evaluation of the platforms learnability was positive within the GSIS context, but negative in the context of the CEC pilot.

The evaluation of the platform's efficiency shows that both sites in general fulfil the success criteria. Again there is some difference between the sites with GSIS giving a more positive assessment than CEC. This difference can, once more, be attributed to the nature of the user groups concerned and the context in which the development work took place.

In terms of memorability, the two sites show vastly differing results. In the CEC site, we find that users were unable to remember how to perform 80% of tasks, while at the GSIS site users were able to remember how to perform 100% of tasks. This may be a function of both relative IT experience and the conditions under which the development work was undertaken.

The criteria regarding the frequency of errors, was wholly met in the GSIS evaluation, with no negative responses. In the CEC pilot, observation was used alongside the questionnaire. The observation shows that though there were serious errors reported initially, these diminished rapidly over the duration of the development period. This would indicate that the platform is not intrinsically prone to user error, but that for the users concerned there is a considerable learning period. From the questionnaires we see that more than 20% of users had to resort to asking for help when using the platform and that only 40% found that the language was clear and easily understood. This may suggest that some work is required on the clarity of language used in the platform.

In the evaluation of satisfaction with the platform, we found that at GSIS users strongly agreed that smartgov was better than current methods. Thus we can conclude that, for this site the criterion was met. However, at the CEC pilot site, all users stated that they did not know whether the platform was better than their current method. This was explicitly a comparison of the smartgov platform with their current, paper-based method. This result can neither be interpreted as positive or negative and so we cannot say whether this criterion is met for the CEC site or not.

Overall, the evaluation of the usefulness of the platform though differing significantly for each pilot site, can be said to be generally positive.

## ***5.4 GSIS Runtime Environment Usage***

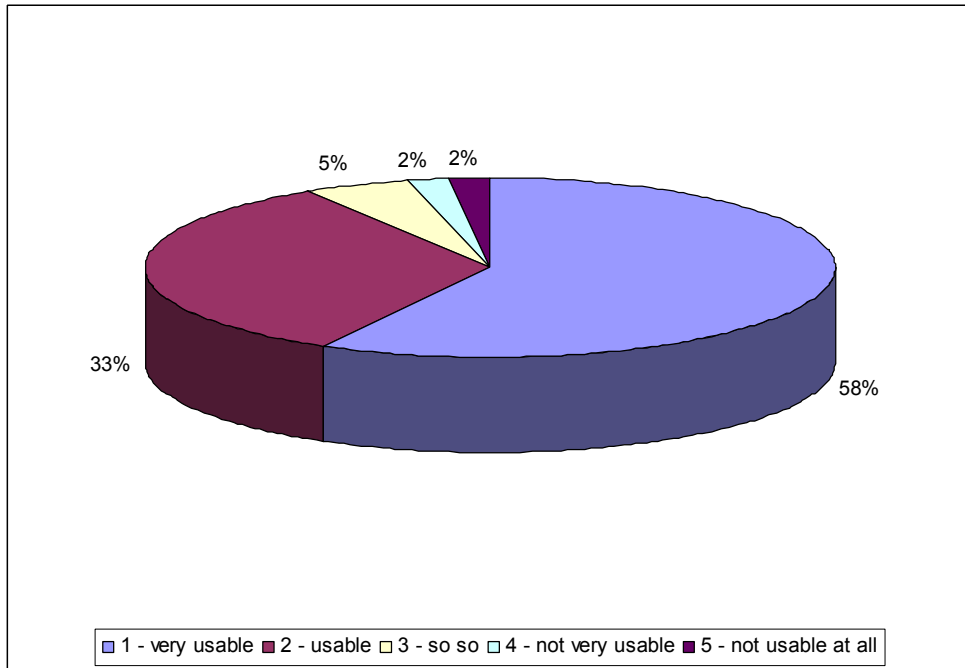
The GSIS pilot application service, end-user evaluation was carried out electronically during December 2003 and January 2004. The evaluation comprised of a small questionnaire that appeared as a final form in the series of forms of the pilot electronic service (e-VAT) (next Figure).

**Figure 9. e-VAT Questionnaire**

The questionnaire comprised of four questions, three of which were closed (with pre-defined answers) and one of them was open. The first question was assessing the perceived usability of the e-service (with possible answers ranging from very usable to not usable at all), the second the system's response (with possible answers ranging from very responsive to not responsive at all) and the third one was asking the users to rate the new service in relation to existing services (with possible answers ranging from much better to much worse). The fourth question was an open one, asking users to give any comments regarding the service they felt necessary. It has to be noted that the questionnaire was itself built with the SmartGov platform and delivered as the service's final page.

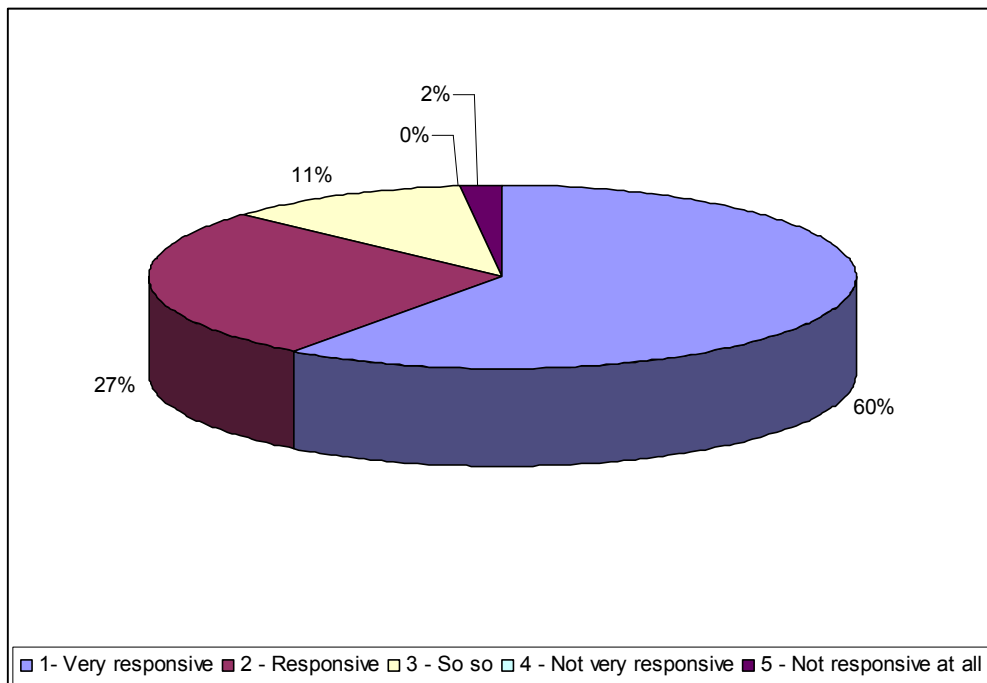
The pilot application had been advertised to selected registered users of the electronic taxation services of the Hellenic Ministry of Finance and in particular to accountants. To this end, the end-user group was already experienced in using electronic services and had used other electronic services of the Ministry of Finance before.

Overall, till the end of January 2004 more two hundred had responded and filled-in the online questionnaire. The reaction of users was more than positive. As depicted in the next diagram, 58% of the users found the service very usable and 33% usable, giving a total of 91%.



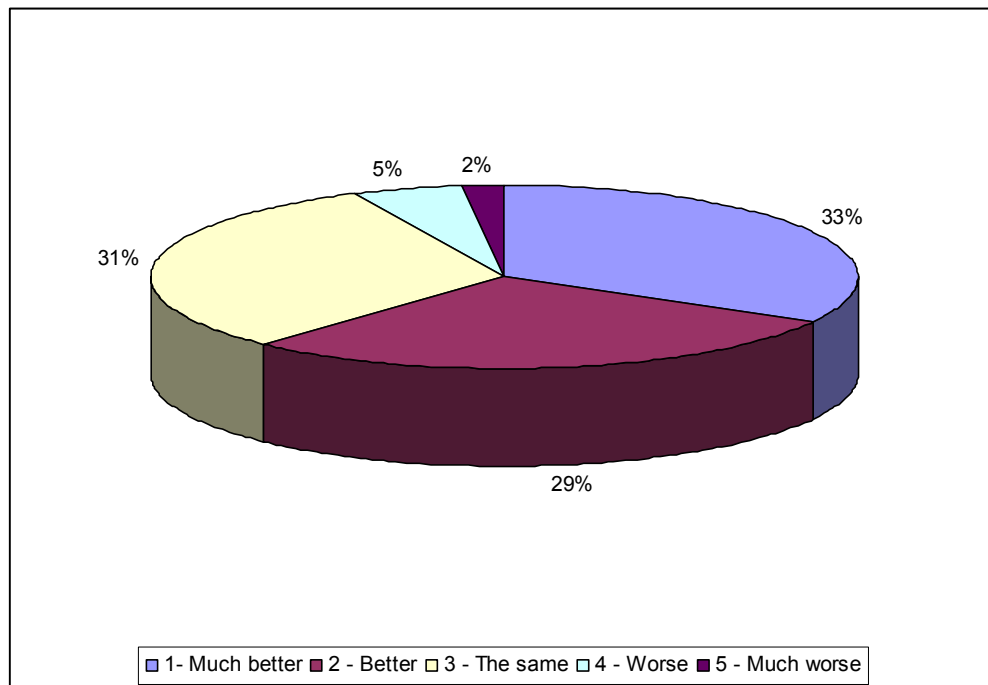
**Figure 10. Perceived Usability**

In regard to the system’s responsiveness, 60% found the system very responsive and 27% responsive, giving a total of 87%, as illustrated in the next diagram.



**Figure 11. System Responsiveness**

However, the most promising reaction was the user's comparison of the new service in contrast to other services they have used before. From the users who responded to the questionnaire, 33% found the service much better than existing services and 29% better. Only 7% considered the new service to be worse than existing services. These percentages are very significant since this was a user group of experienced users that habitually does not embrace new services.



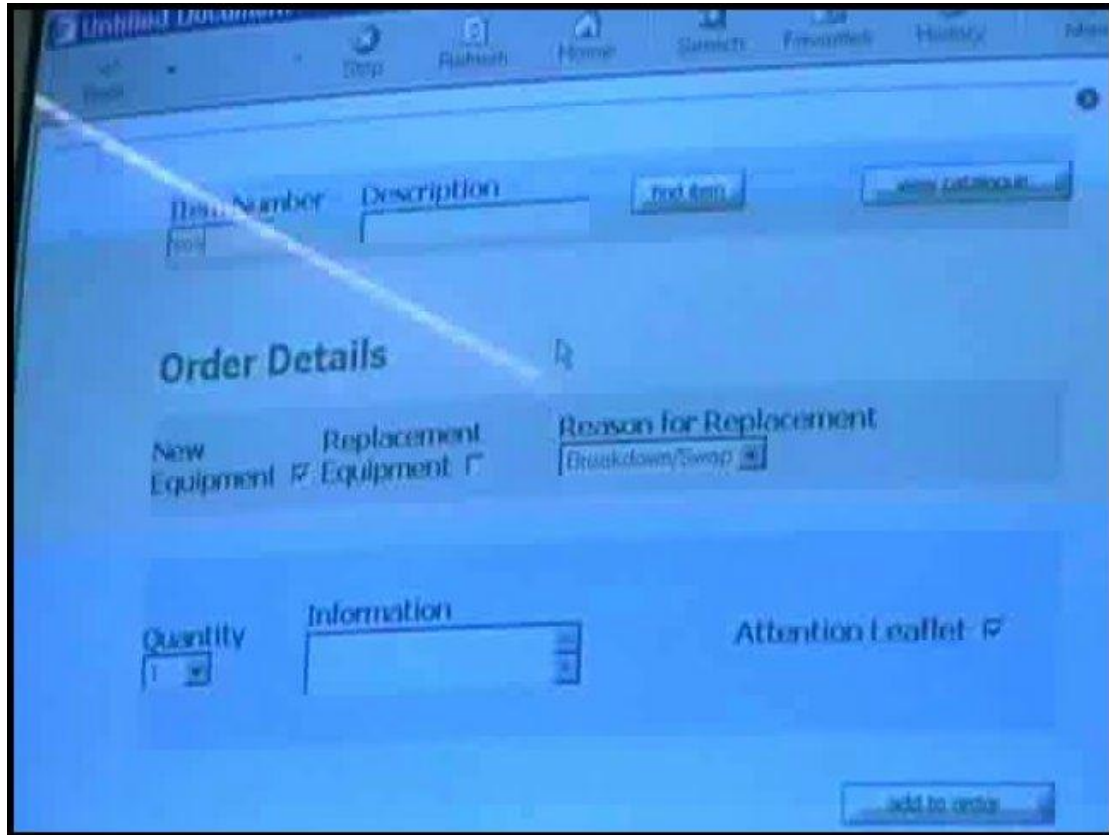
**Figure 12. Comparison of the New Service vs. Existing Services**

Finally, in regard to users' comment, probably the only negative comment was that the service make use of the period (.) as a decimal separator, instead of the comma (,) which is commonly used in Greece.

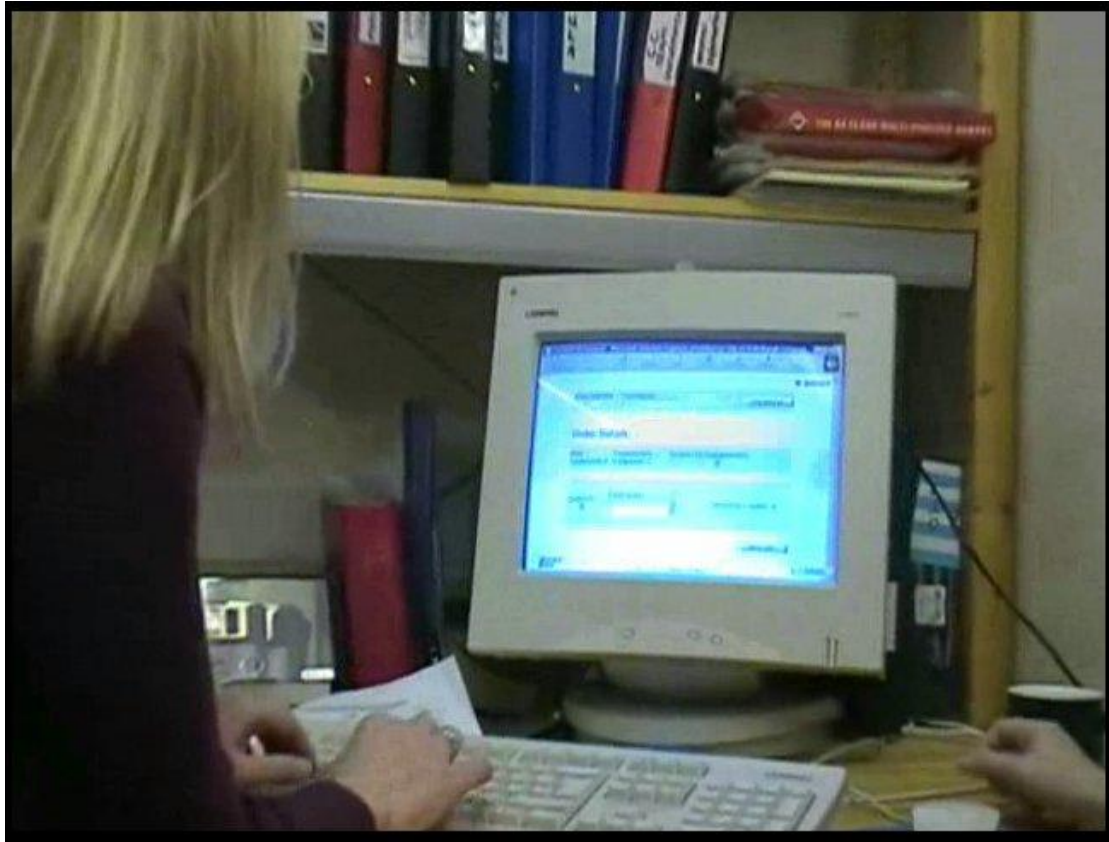
## ***5.5 CEC Runtime Environment Usage***

This section briefly highlights the use of the runtime environment within CEC. It includes pictures of the service in use and quotes from users.





This image shows the Order Details page of the CEC pilot service, where details about the item to be ordered are entered.



This image shows a user completing the Order Details page for the first item being ordered.

The screenshot shows a web browser window with the title 'Microsoft Internet Explorer provided by City of Edinburgh Council'. The page content is titled 'Client details' and features a 'smart' logo in the top right corner. There are three identical form sections stacked vertically. Each section contains the following fields: 'Title' (with a dropdown menu), 'Firstname', 'Surname', 'Address 1', 'Address 2', 'Postcode', and 'City/Town'. To the right of each form section is a link that says 'copy client details'. A red laser pointer is visible on the screen, pointing towards the 'Firstname' field of the top form section.

The above image shows the Client Details page of the CEC pilot service.

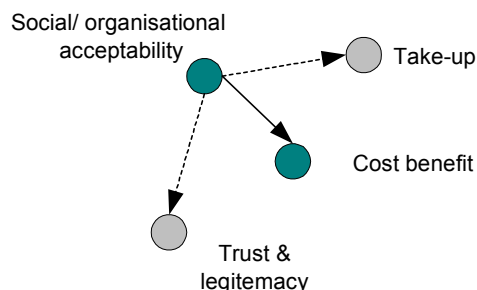
The CEC pilot users were asked to comment on the runtime environment of the CEC pilot service. Below are some of their responses.

“Once you were familiar with what you need to do it was reasonably straight forward.”

“[the produced service] has got rid of extraneous information...But having sufficient knowledge behind...That’s the benefit of the knowledge units.”

“If we were going to create other forms...we would be able to then use those existing knowledge units...The creating of the knowledge units and transaction service elements would happen much more quickly.”

## 6 Social/organisational acceptance: cost-benefit analysis



**Figure 13 social/organisational nodes**

This section presents the cost benefit analysis undertaken by the consortium partner Archetypon. Issues regarding take-up, trust, and legitimacy would be a necessary part of the roll-out of the SmartGov Platform.

In the following sections the cost benefit analysis of the SmartGov platform is performed in terms of:

- Direct/indirect added value,
- Estimation of platform costs,
- Evaluation of the platform by users

### 6.1 SmartGov Competitive Advantage

Despite the huge range of market competitors, SmartGov enjoys one vital competitive advantage. While competitors provide solutions acting in a consultancy role, the SmartGov set of applications will be wholly owned and managed by the local authority staff that uses it. Council staff with reasonable IT skills will be able to publish new e-forms and alter existing ones without having to rely on external consultants. SmartGov will provide local authorities with a specialist and value for money solution to creating and maintaining e-forms in-house.

Furthermore, the unique combination of e-Forms and Knowledge Management in one product at a low price, position the SmartGov solution as a strong new-comer in the existing competitive market.

## **6.2 Indirect SmartGov Beneficiaries – Socio/Economic Value**

The suggested SmartGov platform provides a solution to the public sector for better organisation and improved communication with citizens and businesses. But by implementing the SmartGov solution, one can see potential socioeconomic value that derives to indirect beneficiaries such as the European Union and the European citizens. This added value of the suggested solution can be summarised in the following points:

- SmartGov reduces government expenditures by decreasing bureaucracy
- SmartGov brings government services faster, easier and closer to the citizen
- SmartGov enables Knowledge Sharing
- SmartGov creates a better Europe-wide market information
- SmartGov supports EU's language diversity
- SmartGov improves the working environment of public authorities and businesses that interact with them
- SmartGov contributes to the development of individual skills

## **6.3 SWOT Analysis**

### **6.3.1 Strengths**

- Solution fully owned and managed by customer
- User friendly environment and ease of use
- Holistic approach to administrative processes
- Combination of needs and requirement of all public sector stakeholders
- Reduced price and cost of ownership in comparison with the competitive solutions
- Expertise of the SmartGov consortium to eGovernment solutions
- Specialisation only the public sector
- Knowledge management and e-Form tools are offered in the same application
- Increased service provision to citizens and businesses by the public sector.
- 24x7 service provision
- Increased visibility and transparency of public administration processes

- Improved access to information
- Perception for the citizens and businesses of a tailored service provision
- Reduction of the cost of servicing citizens and businesses
- Reduction of bureaucracy
- SmartGov can be implemented also by very small public authorities

### **6.3.2 Weaknesses**

- Public employees are reticent to new technologies and resistance to change may occur.
- Limited use of electronic signatures from European citizens
- Many public authorities have already implemented e-Form technologies
- Late start in comparison to competition
- Spread of potential customers across Europe makes it difficult to approach them from a central location.
- Internet penetration still low in some member states
- Lack of e-policies and initiatives in the new member states.
- Some public authorities may lack necessary expertise to implement the suggested solution and fully exploit SmartGov advantages.
- Knowledge Management may be considered as an unnecessary expenditure by some public authorities
- The SmartGov Knowledge Management Repository will require an additional time period after the end of the project to be transformed to marketable product.
- Security risks

### **6.3.3 Opportunities**

- Enlargement of the European Union
- Lack of an established standard for e-Forms
- Implementation in the international market of public authorities
- Implementation in the private sector, and specifically intranets.

### **6.3.4 Threats**

- The use of an ERP, CRM or KM system by a local administration may overlap with the scope of the SmartGov development environment.
- The SmartGov framework will be easy for others (vendors and consultants) to copy and use in conjunction with their own services and tools

- The SmartGov framework may not appear to provide the quick-fix recipe books that public authorities seek as they rush to meet tight targets for delivering services on-line.

## **6.4 Pricing Policy**

Two basic functionalities of the SmartGov platform are the e-Forms and the Knowledge Management applications. It could be possible to market the two applications in different packages, namely:

- Basic package: Only the e-Services applications
- Advances package: e-Services and KM applications combined

However, this strategy will not be implemented, as the SmartGov product has stiff competition to face. It is therefore important to differentiate from the competition by offering a strong package, at a low price, which is not available in such a combination as yet. It is possible that some public authority bodies may not use the KM applications at all; on the other hand, some may try it even if it was not in their original requirements. This will increase training and support turnover.

The marketable package will include a free 12-month support, with telephone and e-mail support and free upgrades. Installation of the package will be made by at IT user of the public authority, as it will be designed to be user friendly. However, some customers will require some degree of customisation, and will be charged for that. It is estimated that different sizes of customers will require dissimilar effort for paid support. In order to estimate the support requirements, an assumption has been made based on sales:

- Small local authorities will spend 10% of the price they paid for the package for support purposes (apart from the 12 month free support bundle).
- Medium local authorities will spend 15% of the price they paid for support.
- Large local authorities will spend 20% of the price they paid for support.
- Ministries will spend 45% of the price they paid for support, as they will have greater customisation needs.

The final product will include a comprehensive user and developer manual and further support material will be available at the SmartGov website. However, it is estimated that customers will also require some personal training by SmartGov personnel. These requirements have again been estimated based on sales:

- Small local authorities will spend 25% of the price they paid for the package for training purposes (they will have less users).

- Medium local authorities will spend 45% of the price they paid for training.
- Large local authorities will spend 110% of the price they paid for training.
- Ministries will spend 230% of the price they paid for training, as they will have many users to train and they will require special training courses.

The price at which the SmartGov product will be marketed will be based on user licenses. On very rare occasions (such as a very small local public authority) a public servant will also be the IT specialist. Usually the users involved in the operation of the SmartGov platform will include public sector domain experts, IT staff and managers. Therefore, the public authority will require a number of licenses, which will be priced in bundles, as following:

- 1-5 licenses: € 3.000 per user
- 6-12 licenses: € 2.400 per user
- 13-20 licenses: € 1.800 per user
- 21-50 licenses: € 1.500 per user

At this point, it should be mentioned that the SmartGov customers will not carry any costs for additional software, such as RDBMS (unless explicitly requested by the customer) and application server licenses. The cost of the operating system (MS Windows) will be the only additional software cost. Hardware costs will be minimal as well, depending on the scalability and performance of the target system.

Regarding the support packages, after the first 12-month free support offer, an annual support contract will be signed with interested parties, under the following suggested price bundles:

- 1-5 licenses: € 1.000 per annum for all users
- 6-12 licenses: € 2.000 per annum for all users
- 13-20 licenses: € 2.500 per annum for all users
- 21-50 licenses: € 4.000 per annum for all users

For extra support, such as customisation, or for any support outside the contract, the price given will be estimated with an hourly rate of approximately € 100.

The training prices will again follow the same approximate rate when performed in the customer's location. If the training lessons are made in SmartGov's offices, then a discount of approximately 45% will be offered, as customers from different public authorities could be combined in the same lessons.



## **6.5 Evaluation Analysis**

SmartGov added value was evaluated based on a set of pre-defined criteria:

Reduction of development and deployment time for e-services

- Lowering of skills of personnel for creating e-services
- Accelerating testing process for e-services
- Reduction of maintenance costs
- Reusability
- Improvement of working conditions
- Improved services to the citizen

In order to evaluate the criteria, the questions that are listed in the questionnaire of Appendix "Cost-Benefit analysis questionnaire" were asked to the interviewees.

The target group for filling-in the questionnaires were power-users that influence decisions in their organizations but also have expertise in creating –or participating in the creation process of- online transaction services.

In total, 5 persons have been interviewed from the pilot sites GSIS and CEC.

The results of their evaluation are analysed in the following sections.

### **6.5.1 Profile of interviewees**

Interviewees are mainly senior IT employees with long experience in using computers, involved in the electronic service creation/maintenance process of their organization and sufficiently trained to use the SmartGov platform.

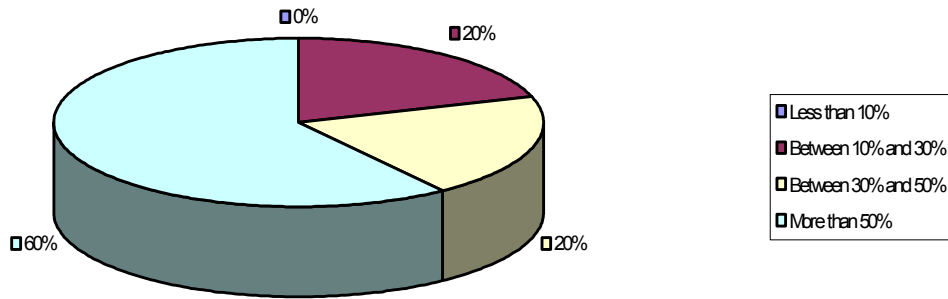
In their respective organizations they are aware of electronic services production and maintenance issues and through their positions they influence decisions towards implementing electronic services.

### **6.5.2 Reduction of development and deployment time for e-services**

The evaluation of this criterion was performed through measuring the improvement in the development process of e-services thus computing the reduction of costs for developing e-services.

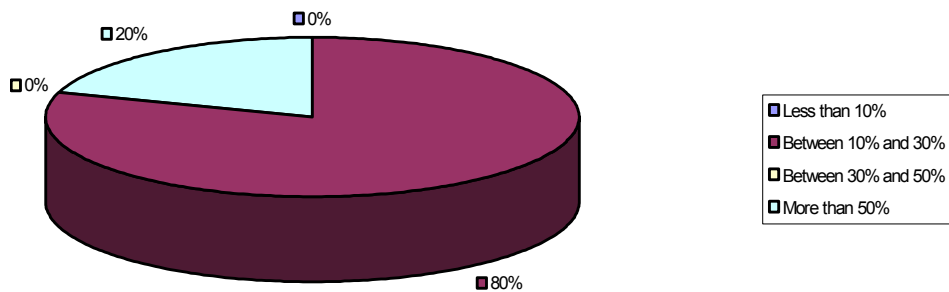
Provided that there exists a variety of services in terms of development complexity, they were separated in simple services (one-screen) and composite (many screens).

For simple services there was major improvement: most users responded to a "more than 50%" reduction of development time with an average of answers "between 30% and 50%" as indicated in the following Figure 14:



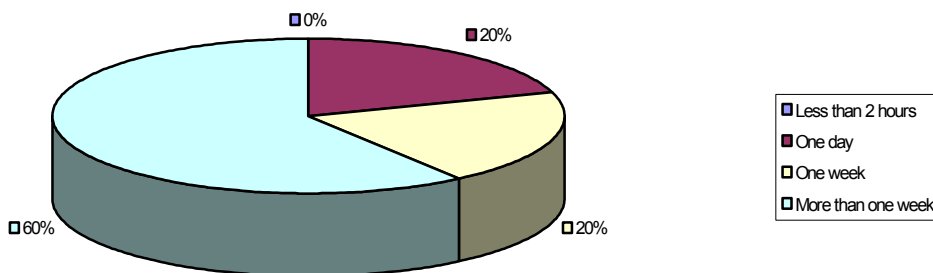
**Figure 14. Reduction of development/deployment time for simple e-services**

For composite services there was also an improvement of “between 10% and 30%” in average as indicated in Figure 15:



**Figure 15. Reduction of development/deployment time for composite e-services**

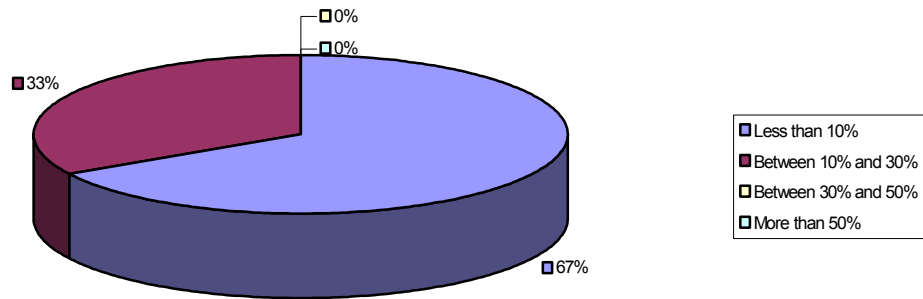
This reduction in developing/deploying e-services results to a major improvement for the Public Authority when combined with the long time-effort required for developing and deploying composite services. This effort is in average more than one person-week as calculated from the following Figure 16:



**Figure 16. Average development costs in time-effort for e-services**

Finally, improvement of connectivity to legacy systems was measured, although there was no sufficient usage of that characteristic of the SmartGov platform by

interviewees, therefore only 60% answered to the relevant question responding mainly with "Less than 10%" improvement:



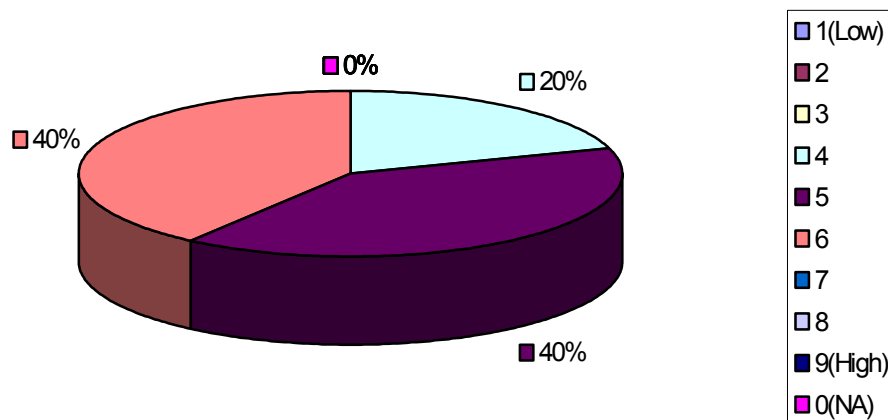
**Figure 17. Improvement in connectivity to legacy systems**

Overall evaluation of the current criterion resulted to the conclusion that SmartGov can improve a lot the development/deployment time for e-services by reducing the effort required by 30-50% or several weeks in terms of person-effort.

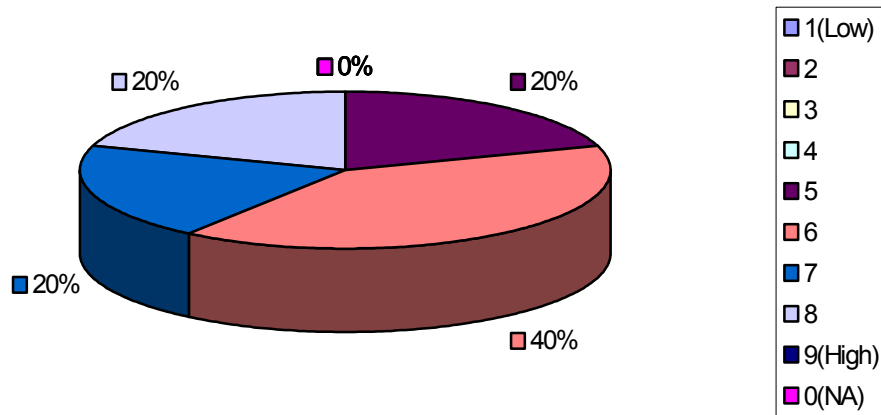
### 6.5.3 Lowering of skills of personnel for creating e-services

In the current criterion there was evaluated the need to lower the skills of personnel that are involved in service creation, thus resulting to lower costs related to the service production process.

Although traditional development of e-services requires advanced skills (programming skills) from the evaluation there was the conclusion that even SmartGov requires average-skilled staff to use efficiently the SmartGov platform in most cases. However there are a lot of cases that low-skilled staff can perform. These results are depicted in the following Figure 18 and Figure 19.

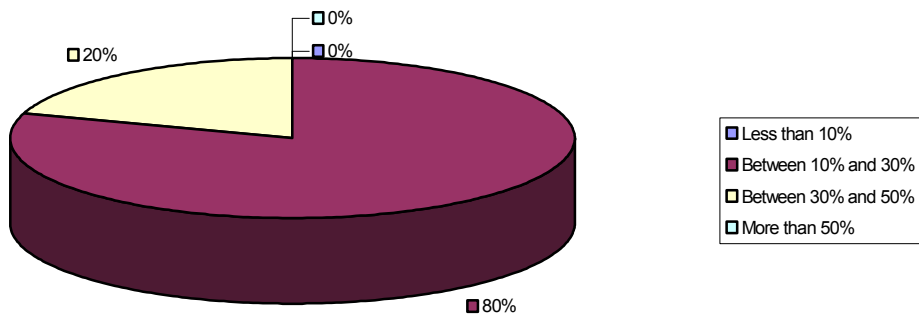


**Figure 18. Average skills for using the SmartGov platform**



**Figure 19. Volume of actions performed by low-level IT staff with SmartGov**

The weight of the current criterion is determined by the reduction of costs for the Public Authority in case of using low-level staff instead of the regular (average level) staff. In that case, there will be an improvement of 10-30% in most cases as indicated in the following Figure 20.

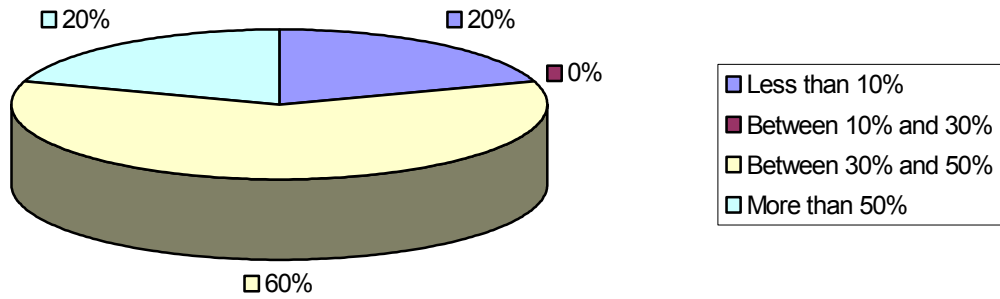


**Figure 20. Reduction of costs when utilizing low-level staff**

### 6.5.4 Accelerating testing process for e-services

The testing phase is often a time-consuming task during the lifecycle of e-service development and deployment. In the current criterion, the added value of the SmartGov platform in respect to reducing testing effort was evaluated.

The evaluation resulted to a "Between 30% and 50%" reduction of testing effort as illustrated in Figure 21 thus improving a lot costs related to that activity.

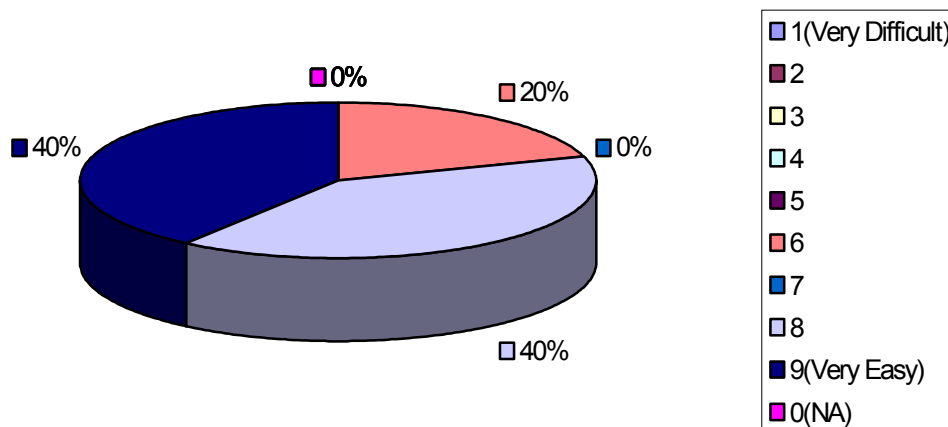


**Figure 21. Reduction of testing effort**

### 6.5.5 Reduction of maintenance costs

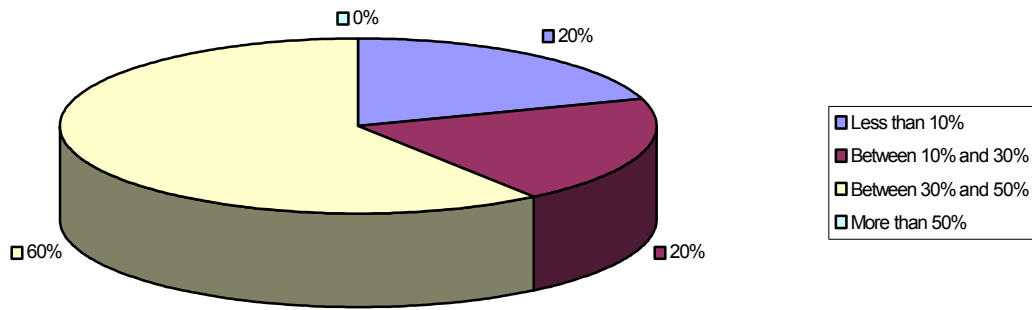
It is often the case that maintenance costs of online services are extremely high compared to the respective development costs. Therefore there was evaluated the value of SmartGov in that area.

Easiness of maintenance through using SmartGov is unquestionable according to the responses of interviewees:

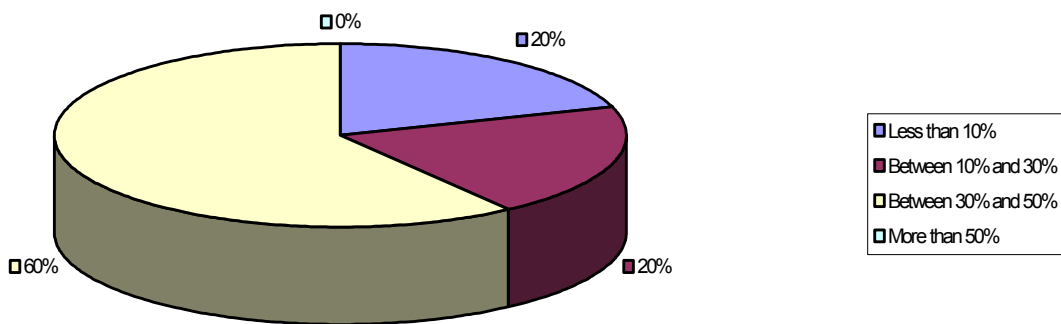


**Figure 22. Easiness of service maintenance with SmartGov**

In terms of effort, maintenance time was reduced in average 30% (see Figure 23) and when combined with the average maintenance effort for e-services -which is more than 30% as from Figure 11- it results to very positive evaluation of the SmartGov added value for the specific criterion.



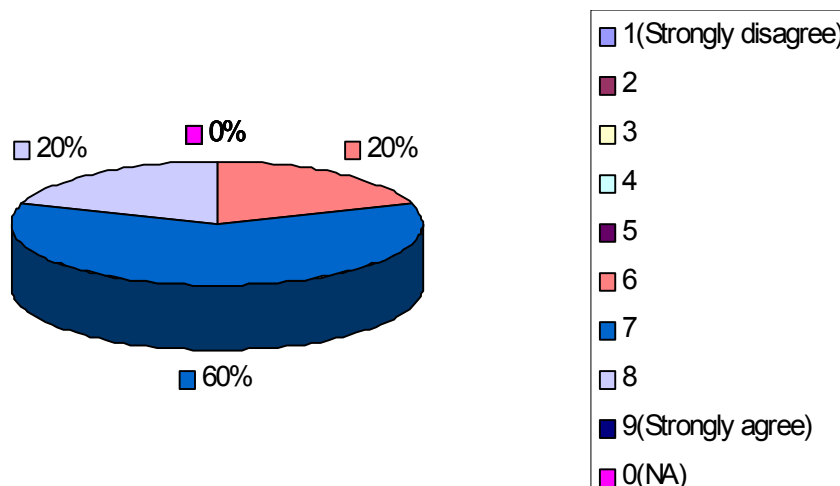
**Figure 23. Reduction of maintenance effort with SmartGov**



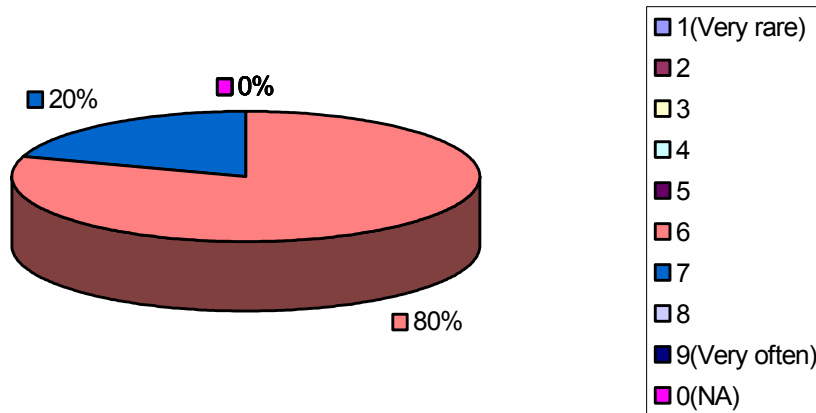
**Figure 24. Maintenance costs compared to development costs for e-services**

### 6.5.6 Reusability

Reusing knowledge and components stored in the SmartGov platform is very useful towards facilitating the service production process when using SmartGov. Based on the evaluation performed, reusability is addressed rather efficiently (more than 60%) through SmartGov.



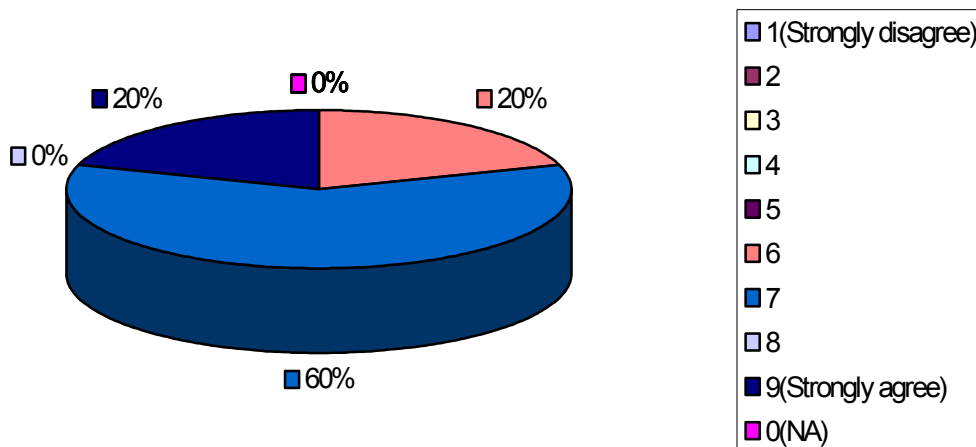
**Figure 25. Easy to re-use knowledge and components**



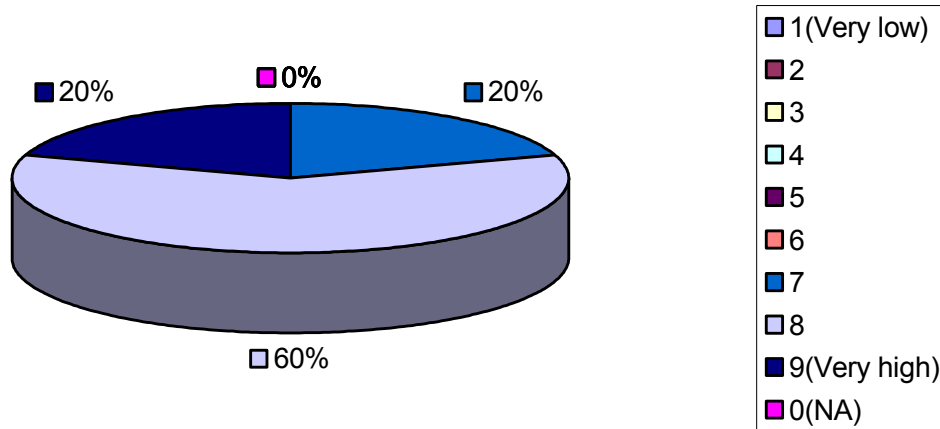
**Figure 26. Reusability frequency**

**6.5.7 Improvement of working conditions**

The value of the SmartGov platform towards improving the working conditions for IT personnel of Public Authorities was definitely accepted by interviewees that responded that they were facilitated in service production -70% satisfaction in average- moreover organized the e-service lifecycle in a better way -80% satisfaction in average- through SmartGov.



**Figure 27. Facilitation in service development**



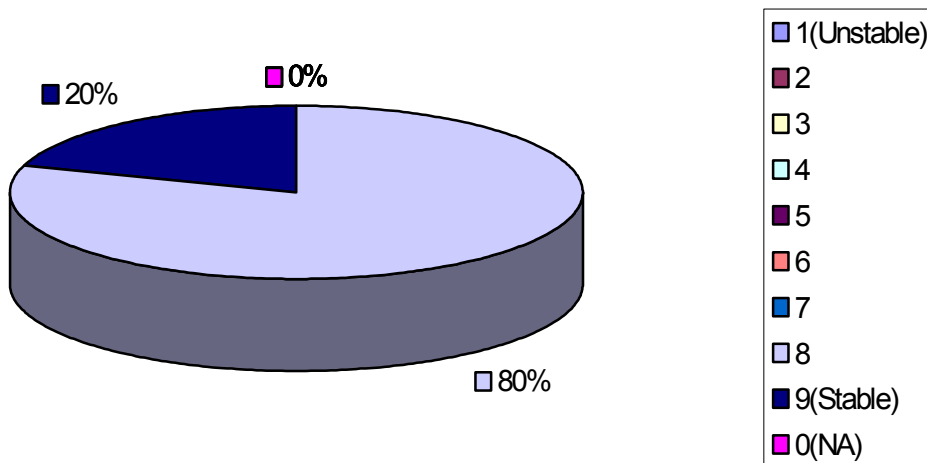
**Figure 28. Better organization of e-service life-cycle**

### 6.5.8 Improved services to the citizen

The provision of services to the citizen through the SmartGov platform was evaluated by means of:

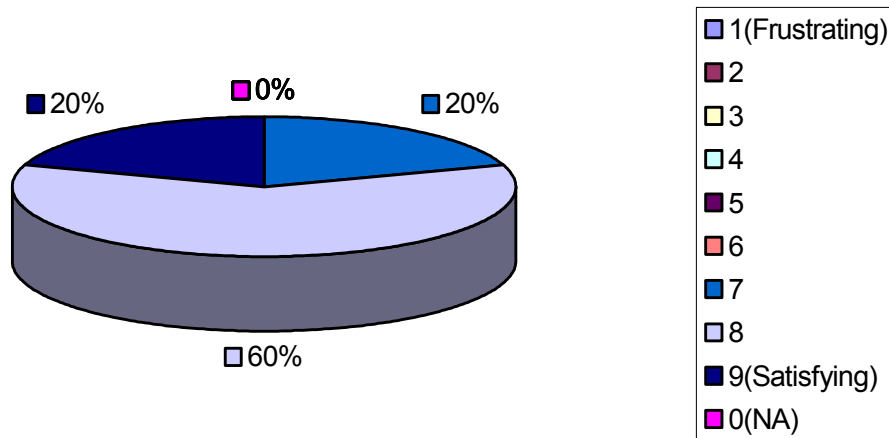
- Stability of produced online services
- Overall impression of citizens from using the produced electronic services

In both criteria, the evaluation resulted to a more than 80% satisfaction, which is highly appreciated basically with regards to stability and less for the overall impression to the citizen –as this mainly depends on the creativity of the person who used the SmartGov platform to produce the e-service.



**Figure 29. Stability of produced online services**





**Figure 30. Overall impression of citizens**

## 7 Conclusion

This work has been concerned with the evaluation of the SmartGov Platform including the two trial sites where pilots were deployed. The structure of the evaluation has been based upon the Nielsen Framework as presented in deliverable D8.1, and considered; technical acceptability which looked at the Platform’s conformance to the User Requirements as stated in D4.1; usefulness which evaluated the Platform in-situ at the two Public Authority sites where it was deployed; and social/organisational acceptability that collated all of the information gathered by the other aspects of the evaluation in order to estimate the overall value of the Platform.

In the conformance to user requirements analysis it was found that the overall total of user requirements met either fully or partially was 78%. More than 80% of the compulsory requirements were met, and of those that were not, the only requirement to have significant impact on the functionality of the Platform was the non-inclusion of a form print facility thus making it hard for the user to retain a hard-copy of their submission for filing purposes, or in the case of a failure occurring with equipment during or after their transaction.

Evaluation of the installation procedure for the SmartGov software noted that as it stands it works well enough, though for some components (specifically the

Integrator and IIG/SGA) specialist knowledge is required that may be beyond that which IT support staff might reasonably be expected to know. The bundling of the software as a series of discrete components with varying set-up mechanisms (zip or installer) also adds slightly to the installation effort required. As a result of the analysis a number of suggestions for enhancements in future releases were recommended.

The results of the evaluation of platform usefulness are generally positive in that a majority of the criteria have been met as taken across the two pilot sites as a whole. However there is a disparity between the two sites, which can be seen in the separate results for each pilot site. This disparity is, as mentioned previously, due to the fact that the service developers in CEC have limited IT knowledge or experience, whereas in the GSIS site, the individual service developers have a high level of IT expertise.

These results are summarised as follows. Both pilot sites returned a similarly high evaluation of the utility and relevance of the platform, however in CEC the proportion of IT staff input into the service development exceeded the 10% of time, criteria. Overall the platform can be considered useful and relevant.

With regard to usability, while neither sets of users felt confident with the platform after one day's training, there was a marked difference between the them, when comparing the platform with other methods. This is due to the disparity of levels of IT experience between the two user groups. Overall the evaluation of the platforms learnability was positive within the GSIS context, but negative in the context of the CEC pilot. The evaluation of platform efficiency shows less disparity, with both sites fulfilling the success criteria overall. In terms of memorability, the two sites show vastly differing results. In the CEC site, we find that users were unable to remember how to perform 80% of tasks, while at the GSIS site users were able to perform 100% of tasks. This may be a function of both relative IT experience and the conditions under which the development work was undertaken. The evaluation of frequency of errors was wholly positive in the GSIS evaluation, with no negative responses. In the CEC pilot however, results from observation and questionnaire indicate that the platform is not intrinsically prone to user error, but that for the users concerned there is a considerable learning period. In the evaluation of satisfaction with the platform, we found that at GSIS users strongly agreed that smartgov was better than

current methods. However, at the CEC pilot site, all users stated that they did not know whether the platform was better than their current method.

Overall, the evaluation of the usefulness of the platform though differing significantly for each pilot site, can be said to be generally positive. However, some recommendations can be drawn from this evaluation. The experience of the users from the CEC pilot site would suggest that some further work could be done to improve the memorability of platform functionality. These users also stated that they would have liked some overview of the service in development and that the language used could have been clearer. Further work would also be recommended to address these issues.

The relevance and utility of SmartGov (runtime) services could only be assessed in the pilots in terms of anticipated improvements rather than through experience. The involvement of service workers in application development means, at least in principle, that the Knowledge Units written by them reflect their knowledge and experience of what their colleagues need to know, and when. Further research is needed to assess whether, in practice, the Knowledge Units developed are relevant and useful as and when service workers use the forms the KUs are associated with. Their relevance and utility should not be considered a foregone conclusion since, as many workplace studies of systems have shown (Crabtree, 2003), 'keeping the work flowing' commonly requires tacit knowledge of current practices, which goes unacknowledged when workers describe their everyday practice.

The cost benefit analysis carried out by consortium partner Archetypon, suggests that the platform can provide added value to the deployment public authority, beyond its mere functionality. They summarise the added value in the following points:

- SmartGov reduces government expenditures by decreasing bureaucracy
- SmartGov brings government services faster, easier and closer to the citizen
- SmartGov enables Knowledge Sharing
- SmartGov creates a better Europe-wide market information
- SmartGov supports EU's language diversity
- SmartGov improves the working environment of public authorities and businesses that interact with them

- SmartGov contributes to the development of individual skills

It is also felt that, despite the huge range of market competitors, SmartGov enjoys one vital competitive advantage. While competitors provide solutions acting in a consultancy role, the SmartGov set of applications will be wholly owned and managed by the local authority staff that use it. Council staff with reasonable IT skills will be able to publish new e-forms and alter existing ones without having to rely on external consultants.

Furthermore, the unique combination of e-Forms and Knowledge Management in one product could position the SmartGov solution as a strong new-comer in the existing competitive market.

The SmartGov platform is a knowledge-based platform to assist public sector employees in the generation of online transaction services. Development, maintenance and integration with IT systems are simplified through the use of the platform. However, the platform would benefit from further development in the recommended areas.

## 8 References

- [Crabtree 2003] Crabtree, A. (2003) Designing Collaborative Systems: A Practical Guide to Ethnography, London: Springer-Verlag
- [D41] Katifori, A., Charissi, A., Lepouras, G., Rouvas, S., Vassilakis, C., Adams, N.J., Fraser, J., Heredia Larios, S., Boukis, G., Gorilas, S., Cañadas, R., and Laskaridis, G. (July 2002) User Requirements, Services and Platform Specifications, SmartGov Project Deliverable 4.1, available at <http://www.smartgov-project.org/index.php?category=results&langid=eng>
- [D51-61] Gorilas, S., Fernadez, P., Pariente, T., Vassilakis, C., Katifori, A., Charissi, A., Lepouras, G., Rouvas, S., Adams, N.J., Fraser, J., and Macintosh, A. (February 2003) Low-level Specifications of SmartGov Services and Applications and the Knowledge-Based Core Platform, SmartGov Project Deliverable 5.1-6.1, available at <http://www.smartgov-project.org/index.php?category=results&langid=eng>
- [D62] Gorilas, S., Fernadez, P., Pariente, T., Vassilakis, C., Katifori, A., Charissi, A., Lepouras, G., Adams, N.J., Fraser, J., Macintosh, A., and Stoumpos, V. (July 2003), Implementation of SmartGov Services and Applications, SmartGov Project Deliverable 6.2, , available at <http://www.smartgov-project.org/index.php?category=results&langid=eng>
- [D81] Cañadas, R., Fernández, P., Heredia, S., Gerogianakis, S., Fatourou, E., Kattoulas, P., Kavadias, G., Macintosh, A., Whyte, A., McKay-Hubbard, A., Adams, N.J, Vassilakis, C.,

Lepouras, G., Katifori, A., Stoumpos, V., Pipis, A., Bogiatzis, L., and Haston, S. (October 2003) Trials Configuration and Evaluation Process, SmartGov Project Deliverable 8.1, , available at <http://www.smartgov-project.org/index.php?category=results&langid=eng>

[Nielsen 1993]

Nielsen (1993) Usability Engineering, 1993 ISBN 0-12-518406-9 Academic Press Page 25

## 9 Appendix

### ***9.1 Results from CEC usefulness questionnaire***

Below is an example of the questionnaire used to gather data on the usability of the development environment. Five domain experts, who were responsible for the development of the CEC pilot service, completed the questionnaire. The responses, marked in red text, have been indicated using numbers corresponding to quantity per tick box. Text remarks, where appropriate are included verbatim.

Note: The CEC version was administered after observing the domain experts (Occupational Therapists) working on development tasks, as the development neared completion.

#### **Service Developers : CEC Pilot**

December 2003

Your Name

---

Please note we will be taking notes of any problems you experience with the software, and any comments you may want to make. Your name will only be used to match our notes with your responses to this questionnaire. We will provide a copy of both to you, when we have written up our notes. However your name will **not** be used in any research reports.

Most questions have the format shown below. Please read the statement, decide which option is closest to your own views, and tick the appropriate box.

We would also appreciate any comments you may have, to add to your response. We will note what you say and ask you to check our notes for accuracy later.

#### **Example**

Q. A statement about the SmartGov service development...

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

### Remembering what to do

Q1. I was able to **create a TSE** without needing to ask for help or refer to the manual/training materials.

1 Strongly agree  3 Agree  Neutral  1 Disagree  Strongly disagree  Don't know

Q2. I was able to **create a TSE Group** without needing to ask for help or refer to the manual/training materials.

Strongly agree  3 Agree  1 Neutral  1 Disagree  Strongly disagree  Don't know

Q3. I was able to create a **KU** without needing to ask for help or refer to the manual/training materials.

Strongly agree  1 Agree  1 Neutral  2 Disagree  1 Strongly disagree  Don't know

Q4. I was able to create a **Form Element** without needing to ask for help or refer to the manual/training materials.

Strongly agree  1 Agree  Neutral  3 Disagree  Strongly disagree  1 Don't know

Q5. I was able to add something to the taxonomy without needing to ask for help or refer to the manual/training materials.

Strongly agree  2 Agree  1 Neutral  1 Disagree  Strongly disagree  1 Don't know

**Re-using Transaction Service Elements and Knowledge Units**

Q6. Elements can be easily applied to different services if the elements share the same name

Strongly agree  4 Agree  1 Neutral  Disagree  Strongly disagree  Don't know

Q7. It was easy to use the software to re-use elements.

Strongly agree  2 Agree  1 Neutral  1 Disagree  Strongly disagree  1 Don't know

Q8. The KUs were still relevant after re-using them in another service, with only minor changes.

1 Strongly agree  2 Agree  1 Neutral  Disagree  Strongly disagree  1 Don't know

Q9. It was easy to use the software to re-use the KU's

1 Strongly agree  2 Agree  2 Neutral  Disagree  Strongly disagree  Don't know

**Finding relevant Transaction Service Elements and Knowledge Units**

Q10. When I looked for a relevant TSE I could easily find it

2 Strongly agree  2 Agree  1 Neutral  Disagree  Strongly disagree  Don't know

Q11. When I looked for a relevant KU I could easily find it

1 Strongly agree  2 Agree  1 Neutral  1 Disagree  Strongly disagree  Don't know



**Learning to use SmartGov**

Q12. I felt confident in creating and modifying a SmartGov service after:-

- Less than one working day (7 hrs) of training and practice
- 2 About one working day
- More than one working day
- 3 I still do not feel confident after \_?,2,3\_ days.

Q13. Using SmartGov has allowed me to learn technical tasks that I would not otherwise have the technical skills to do confidently.

- 2 Strongly agree
- 1 Agree
- 1 Neutral
- 1 Disagree
- Strongly disagree
- Don't know

Q14. Smartgov was easy to learn

- Strongly agree
- 1 Agree
- Neutral
- 2 Disagree
- 2 Strongly disagree
- Don't know

Q15. How much previous experience have you had in developing online transaction services, or web database applications?

- 5 None
- 1-3 months
- 4-6 months
- 6months- 1year
- 1 year +

**Using SmartGov regularly**

Q16. I could work out what I should do next without help from (SmartGov) IT Support Staff

- Strongly agree
- 2 Agree
- 1 Neutral
- 1 Disagree
- 1 Strongly disagree
- Don't know

Q17. The pages appeared on the screen quickly enough

Strongly agree  4 Agree  Neutral  1 Disagree  Strongly disagree  Don't know

Q18. The language was clear and easy to understand

Strongly agree  2 Agree  Neutral  1 Disagree  2 Strongly disagree  Don't know

### Your overall views about SmartGov

Q19. I am confident that data that gets entered on SmartGov forms will be stored safely

Strongly agree  Agree  2 Neutral  2 Disagree  Strongly disagree  1 Don't know

Q20. I am confident that data will not be put to any improper use or disclosure

Strongly agree  1 Agree  1 Neutral  Disagree  Strongly disagree  3 Don't know

Q21. Overall, I prefer using SmartGov rather than maintaining a system based on paper forms.

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  5 Don't know

Q22. What are the most likely benefits of SmartGov for your own work ?

1 "online ordering: speed of transactions , contact with resource"

2 "Less paper, speedier response-avoiding current paper based system (see monitoring form). As a manager, benefit would ensure that workers will have some information to assist in completing process (KUs)"

3 "online ordering-ability to browse, less time spent completing paper version, more efficient quicker response from JDES/ posting and faxing"

4 "it will help to inform and store business information that may be applicable to future changes or developments"

5 "speed of communication -> service delivery and tracking service delivery. Guides people completing form to reduce errors and omissions."

Any disadvantages?

- 1 "will depend on clarity of screens and speed of transaction"
- 2 "some staff may not find computerised system useful."
- 3 "may be difficult for staff not confident with PC, may be problem if system fails, access to PCs for staff not a problem with paper system."
- 4 "appears to be time consuming at present."
- 5 "Need the computers to be available for staff to use. Not readily available at present."

Q23. What are the most likely benefits of SmartGov for service users ?

- 1 "can be used in the field, catalogue on screen linked to ordering online. Less likelihood of failure/ mistakes in completing order. Instant notification on lack/problems of/with resource. Booking deliveries and confirmation of this."
- 2 "hopefully speeding up process of ordering, progress chasing. Should provide quality information for user use (KUs)"
- 3 "hopefully faster access to provision, response to enquiries. Could be accessed from mobile office/laptop. Ability to browse catalogue- few paper versions around."
- 4 "Hopefully decreased duplication of forms and reduce unnecessary extra service delivery."
- 5 "faster response times for delivery and uplift. Reduce delays relating to incomplete information."

Any disadvantages?

- 1 "Limited access to PCs in work place may delay activity."
- 2 "Computer literacy of staff is variable as we have not yet completed the forms -difficult to know though...may be quite simple."
- 3 "initially, staff may lack confidence. Initially staff may require time to find way around system."

- 4 "no hardware or hardware problems- ie system crash"  
5 "No"

*Thank you for your help.*

## **9.2 Results from GSIS usefulness questionnaire**

### **Service Developers : GSIS Pilot**

December 2003

Your Name \_\_\_\_\_ \*\*\*\*\*

Please note we will be taking notes of any problems you experience with the software, and any comments you may want to make. Your name will only be used to match our notes with your responses to this questionnaire. We will provide a copy of both to you, when we have written up our notes. However your name will **not** be used in any research reports.

Most questions have the format shown below. Please read the statement, decide which option is closest to your own views, and tick the appropriate box.

We would also appreciate any comments you may have, to add to your response. We will note what you say and ask you to check our notes for accuracy later.

#### **Example**

Q. A statement about the SmartGov service development...

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

### Remembering what to do

Q1. I was able to **create a TSE** without needing to ask for help or refer to the manual/training materials.

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q2. I was able to **create a TSE Group** without needing to ask for help or refer to the manual/training materials.

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q3. I was able to create a **KU** without needing to ask for help or refer to the manual/training materials.

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q4. I was able to create a **Form Element** without needing to ask for help or refer to the manual/training materials.

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q5. I was able to add something to the taxonomy without needing to ask for help or refer to the manual/training materials.

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

### Setting up Forms

Q6. I am confident that, we can produce simple forms with SmartGov & Dreamweaver in no more than one working day.

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q7. I am confident that we can produce complex forms with SmartGov & Dreamweaver in no more than six working days.

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q8. There is no significant extra effort needed to design or modify a form when using SmartGov, compared with other online forms I have worked on;

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

### **Re-using Transaction Service Elements and Knowledge Units**

Q9. Elements can be easily applied to different services if the elements share the same name

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q10. Re-using elements was easier to do with SmartGov than the current methods

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q11. The KUs were still relevant after re-using them in another service, with only minor changes.

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q12. Re-using documents by linking them to forms was easier with SmartGov KUs than with the current methods.

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

### **Finding relevant Transaction Service Elements and Knowledge Units**

Q13. When I looked for a relevant TSE I could easily find it

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q14. When I looked for a relevant KU I could easily find it

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

### **Learning to use SmartGov**

Q15. I felt confident in creating and modifying a SmartGov service after:-

Less than one working day (7 hrs) of training and practice

About one working day

More than one working day

I still do not feel confident after \_\_\_ days.

Q16. Using SmartGov has allowed me to learn technical tasks that I would not otherwise have the technical skills to do confidently.

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q17. Smartgov was easier to learn than software I currently use for developing online transaction services.

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q18. How much previous experience have you had in developing online transaction services, or web database applications?

None  1-3 months  4-6 months  6months- 1year  1 year +

### Using SmartGov regularly

Q19. I could work out what I should do next without help from (SmartGov) IT Support Staff

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q20. The pages appeared on the screen quickly enough

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q21. The language was clear and easy to understand

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

### Your overall views about SmartGov

Q22. I am confident that data that gets entered on SmartGov forms will be stored safely

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q23. I am confident that data will not be put to any improper use or disclosure

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know



Q24. Overall, I prefer using SmartGov rather than the current methods for developing online transaction services.

Strongly agree  Agree  Neutral  Disagree  Strongly disagree  Don't know

Q25. What are the most likely benefits of SmartGov for your own work ?

Well constructed and easy development

Any disadvantages?

Possible limitations on forms flow

Q26. What are the most likely benefits of SmartGov for service users ?

If many services are developed, the similarity of the user interface.

Any disadvantages?

N/A

*Thank you for your help*

## **9.3 Observational notes from CEC pilot**

### SMARTGOV FIELD NOTES

Andy

#### **Description of situation**

These are sessions in which the Occupational therapists, who constitute domain experts in the CEC pilot situation, use the smartgov platform to develop their online service. They have developed the logic in a paper based format previously and use this as a basis for the creating of various transactional service elements (TSE), groups of these(TSE Group), form elements (Form) and Knowledge Units (KU). They have previously received basic training in the use of the platform.

#### **Location**

The work is carried out in a computer training room provided by CEC. The room contains several networked PCs which are able to connect to the smartgov server. All have Internet Explorer 5 as their browser.

#### **Activity**

The creation of various elements to develop an electronic transaction service.

#### **People**

Five OT staff members (Domain experts)

#### **Time frame**

These sessions took place on three separate occasions: 14/11/03, 26/11/03 and 05/12/03. Each session lasted for a minimum of three hours.

#### **Observation**

There follows three sets of observational notes. They are in chronological order and are numbered but no specific time of observation is stated.

### **Observational notes from OT service creation session 14/11/03**

1. On logging in the users all pressed the return key after populating the username and password fields. Some then found the submit button on the form and pressed this, while others (half) where unable to find the submit button without help.
2. Users initially unable to identify scroll bar, then seem unable to use it, even though it functions in the same manner as that on a more familiar application such as word.

3. Users express difficulty with visual representation. All pages look the same, not easy to identify where they are. "Text links for performing key actions are difficult to see in surrounding text, not intuitive way to access key task sections." An example is the "new TSE" link, this is small and obscure (according to users), they express preference for clearly defined button.
4. Users also express problem with expanding the collapsed sections of editing pages, some do not understand, are not familiar with this type of representation, others cannot readily find the activation button.
5. users express difficulty locating user editable fields.
6. users express difficulty in grasping naming conventions
7. the users are showing an ability/desire to mentally link the service elements they are creating with a hypothetical real world situation and what they imagine a form representation to look like. They are discussing TSE in these terms amongst themselves.
8. users lose data as there is no prompt to save when exiting an editing page that has been updated by the user.
9. users beginning to debate the actual TSE content , even though this was agreed prior to this occasion, the creation process appears to be provoking more analysis of TSE structure and requirement.
10. Users can locate and use the action symbol buttons readily on successive pages (these are the "home", "back", "delete", "save", "confirm(tick)" and "select for editing(dot)").
11. users became confused by the value list aspect of a TSE they were creating. They began constructing a list of values that were analogous to individual TSEs and they assumed they would function and appear as such in the subsequent service. However , when the concept of the value list was explained beyond the explanation given in the user guide, they understood its function and were able to use it correctly in the construction of subsequent TSEs. Users Complain that user guide to not make this clear. Reluctance to use user guide before asking for assistance.
12. more data loss from failure to save element. Some confusion with function of red tick and save action.
13. users do not understand what is meant by "instantiate", did not find explanation in guide useful.
14. Recurrence of issue with value list.
15. group of six people have created 8 elements in one hour

16. Users find the displayed list of created elements is not easy to read, "confusing , messy and unclear". Difficult to locate a specific element to edit, this is getting worse as list grows.
17. problems with disallowed characters.
18. users appear to have developed an understanding of the concept of re-use of TSE within same service and how TSE groups can be used to structure this.
19. several users now expressing the opinion that visual display is "messy", difficult to navigate especially as more new elements are created.
20. Space provided for content in the editor pages is greater than space allocated to content in the list fields. Users having difficulty tailoring informative text to suit this.
21. Users comment that hover text providing information on functions of links and buttons would be very helpful.
22. understanding of the concept of KUs appears to have become apparent to users, they are now discussing what information may accompany TSE in the form of KU.
23. Users continuing to express wish for buttons rather than text links
24. Users began to create TSE groups without prompting. They expressed that they found it a logical step in the structuring of TSEs, expressed confidence in creating TSE group based solely on experience of creating TSEs. Express accord with use of similar visuals and techniques for editing and creating different elements.
25. Users observed that in editing pages the term "content" is used to refer to the field populated by a text description of the element, however, when elements are listed, this content is listed under the heading "description". Users expressed the view that this was confusing and had caused some problem in deciding what text to enter in the field.
26. The selection of TSE from the list of new ones is difficult
27. BUG there is a bug in the platform that causes no TSE ID to be listed in the associated TSE section of the TSE group editor.
28. More saving problems
29. One particular user (Jackie) is able to assist other users appears to have firm understanding of the platform.
30. the group appears to be forming a type of internal self help dynamic, pulling resources to overcome problems in understanding the platform and development process.

31. Platform seems to generate extra, spurious windows through use.
32. users appear to have enjoyed the session, are keen to continue at next session.

### **Observational notes from OT service creation session 26/11/03**

1. Users experienced more login difficulties. Most forgot their password/ID, most could not remember URL, all used return key instead of submit button.
2. Users keen to tackle creating a form element, all feel relatively confident with using the platform. All capable of creating TSEs and Groups
3. Users expressed disappointment that they are unable to view the visual representation of the form that they are creating during the development process. All agreed that this would make the process easier, more intuitive.
4. Users show understanding of need for a naming convention.
5. Use of the environment has stimulated discussion of, and some changes to planned service structure.
6. Users appear to understand value of grouping TSEs and the ability to reuse these groups. Have constructed form that uses three repetitions of the same group to gather slightly different information. Users state that they now realise that some TSEs could have been made more generic to facilitate this type of re-use.
7. Several occasions where users did or nearly did, press the logoff button instead of the save button. Users complain that these buttons are similar in appearance and located in too close proximity to each other.
8. Users state that the platform/user guide does not clearly indicate that only instantiated groups can be added to a Form.
9. Some name issues arise when selecting TSE and TSE groups from list to be included in Form. There is a bug in the software that prevents the name of the element from being shown, there is also a bug that prevents all available elements being displayed. This has been highlighted to Indra.
10. It is unclear in the platform what sections of an element have been save and which have not, or rather where one is required to press save or tick button. Users express desire for more prompts from system.
11. One particular user continues to show a sound understanding of the platform and the methodological approach.

12. Again users state dissatisfaction with lack of visual representation of form within the system. (IT specialist who will be creating, tagging the XHTML forms expressed point that he will need a domain expert present when doing this as he has know knowledge of the logic component and thus would not know which element goes where on the form. This highlights issues with the separation of the syntactic and semantic parts of the service development).
13. More issues with listing of available elements in form editing and group editing mode. (\*\*\*\*Bug fixed by Indra, however this has impacted severely on the design process which will now have to halt)
14. More confusion over the function of tick and save buttons
15. Issues with need to refresh view in order for more recent elements to be listed.
16. Trust issues, platform provides insufficient feedback to establish trust in the system carrying out the user's command.
17. Where dialogue boxes are used to give feed back (rare) they appear to be well accepted by, and understood by the users.
18. Navigation issues. Users often seem confused as to which portal they are in, and which they need to open to perform next task. More issues of finding the link to the next action (ie "new tse" link is small and obscure within the page. Much mouse wandering)
19. More debate on which groups to include in which form.
20. One particular user (the same as mentioned previously) user has grasped the use of yellow dot for editing, none of the other users have .
21. Users discussing more issues concerning the structure of their service, have realised that they need to create some new elements that had not been envisaged. Platform appears to assist in the understanding of the service process/logic.
22. Inconsistent layout between editing portal. The TSE editor differs from the KU editor in its field layout.
23. the issue of the bug mentioned earlier has brought the form/group development process to an end.
24. more navigation problems
25. Beginning KU construction.
26. Users showing understanding of taxonomy as K-map
27. Users begin creating basic KUs to be added to the more simple elements.

28. Tendency to discuss KUs purely in terms of online help for the developed service
29. not using KU to include process info as yet.
30. some KU properties do not appear to be intuitive to the users.

### **Observational notes from OT service creation session 05/12/03**

1. Users enter into discussion of content of KU to be attached to Postcode TSE.
2. Users asked about naming convention, had to be reminded of the format they had adopted in previous sessions.
3. Users now ask one particular co-user for help.
4. One user forgot to save before exiting edit portal for KU, all work lost.
5. Same user expressed need for a prompt to save on exit, other users concur.
6. Users showing difficulty with the language choice defaulting to Spanish in content/description fields.
7. Users requested help with "create new section" in KU editor. They seem to have forgotten how to use this section of the editor.
8. User almost pressed "logout" button instead of "save" this has happened before.
9. Users then expressed their dislike of the chosen button symbols, claiming that these are too similar and confusing.
10. The above led to a discussion about general aesthetics and layout. Users felt that the positioning of action buttons and links in diverse areas of the screen made it difficult to use. Could not find "prev" and "next" buttons on some pages. Expressed view that collecting action buttons together in one area of screen would be ideal.
11. Users requested explanation of function of the "description field" in the header section of the KU and the "content" field in the sections section.
12. Users proceed, unprompted, to add new KU to Taxonomy.
13. Users scan through existing KU to ascertain what new KUs to create.
14. Users again express confusion about "content" and "description" fields.
15. Users begin using the "link" feature unprompted. Intended to link external, relevant document to the KU, thus seeming to grasp the function of the link facility intuitively.

16. Another user almost logged out instead of saving work.
17. Discussion of KU shows that users have good understanding of how KU can assist users.
18. The exercise has prompted discussion of viability of some previously created TSEs
19. More debate on KU content
20. Discussion on importance of well-chosen KU content.
21. Users access taxonomy regularly
22. Users state that taxonomy has provided meaningful location for the current KU.
23. Users show understanding of taxonomy structure and are able to navigate to nodes deep within structure and attach KU there.
24. I had to enforce meaning of some taxonomy node terms, users intended to derive a different meaning than that originally intended.
25. Users thought that the end user would be able to access the taxonomy.
26. Users working for period of several minutes without any need for assistance.
27. User managed to remove a link entry that had been erroneously created, without assistance.
28. Users navigating taxonomy, discussion as to where to locate new KU, users decide to locate at high level node in order to cover all lower level nodes.
29. Users discuss use of instantiation to make KU linked only to very specific instance of TSE.
30. Users create KU with link to both URL and a local document.
31. User again navigating and using taxonomy.

**Session date: 24 November**

## SMARTGOV FIELD NOTES

Angus

CEC Social Work Dept training suite



Participants identified by initials:

AM Andy McKay Hubbard, Napier

OT Occupational Therapist (x5)

The first issue noted was that the OT's had not observed the naming convention for TSEs and conversely the system had not enforced any naming convention. Before the session AM had changed the names to conform to the convention and had asked the CEC IT support to upload a new version of the database. However this had not been done, which meant that no further work could be done on the TSE's in the session. The session was therefore used to ask the OT's for feedback on the training materials that had been provided in the form of D8.2. The feedback was recorded using questionnaires, and the OT's elaborated on these with verbal comments. A copy of D8.2 was passed around so that it was available for reference.

Comments from the OT's were:-

We wouldn't have known where to start without you (AM) there, and we couldn't have picked it up from that (D8.2).

We don't use that kind of terminology- it's not for layman's use – it depends on understanding IT terminology and it needs a glossary.

It's too big, too dry.

It's not structured logically, for example we need to do the TSEs first.

Diagrams are too small

I couldn't find the relevant bits, but once you find them it's quite easy to do it.

An overall understanding of how the form fits together is needed to understand what you need to do.

5 December 03

Training suite, CEC Housing Dept.

Participants identified by initials:

CL Occupational Therapist (OT), CEC  
 LB ^---^  
 MD ^---^  
 MK ^---^  
 JS ^---^  
 AM Andy McKay Hubbard, Napier (facilitator/ observer)  
 NA Nick Adams, Napier (facilitator/ observer)  
 AW Angus Whyte, Napier (observer)

The OT's had the previous week finished creating the TSE's and had created KU's for the address and name fields.

The first issue that arose was that the system had been updated for a bug fix, but the updated version had not been installed by CEC IT Support. This meant that it would not be possible to look at TSE groups or create a form.

The OTs sat in two groups, each group around one computer, to collaborate on the next steps. The comments noted below are intended to highlight ease of use and learning issues. Note that some of the OT questions were addressed to colleagues in the course of working together and were not requests for help to the facilitators. He latter are specifically identified.

OT comments and actions	Observer comments (AW)
JS How do we do this, associate the KU with TSEs? How to describe JS types in a description for the KU, saves it and opens the properties.	AM began by reminding the OT's to use the naming convention for KU's

<p>JS I'm completely lost... don't understand KUs at all.          You need a reminder of what you did last.          JS reads manual section 'core of a KU'          JS (to AM) What do you put in the 'comments'          AM You don't need to... though you need something in 'content'          JS adds some comments about equipment delivery, clicks back, but then goes back to the KU to add a link to a postal service website giving postcodes,          JS I don't understand how it fits together          The previous description is not visible and the OT's wonder aloud what has happened to it. They go back to add a section.          (To AM)'Section visibility' – what is that?          AM says that this defines who gets to see the section.</p>	
<p>A message box appears, stating "Explorer User Prompt; script prompt...."          JS to AM) What does that mean?</p>	<p>It is clear from the screen that there are now two sections, one with 'content' and another with 'link' but the OTs cannot do what they want: to associate the link with the content, without re-typing it in one section and deleting the other. They can do this without asking for help however.          (MODERATE)</p>
<p>The Taxonomy Retrieval window is open          JS to Am) How do you go back? There's no back button          AM) use the 'prev' link</p>	<p>Message not understandable, not regarded as serious.          (MODERATE)</p>
<p>Taxonomy- associating KU          JS cannot find "postcode" but can see "4<sup>th</sup> line of</p>	<p>The prev link is not highly visible, nor is its</p>

<p>address”</p> <p>JS I don't know what 'instantiated' means</p> <p>AW (re finding the postcode) to do that you needed to know that the postcode was the 4<sup>th</sup> line of the address?</p> <p>JS Yes</p> <p>AW So if you didn't have a form you were already familiar with you would need to know or decide the layout first of all?</p> <p>JS Yes it would be better to have a flowchart so you can visualise what you get at the end...and a glossary of all these terms.</p> <p>JS goes to add a new section</p> <p>JS You're presented with all these boxes but you don't know what to put because you don't know where it's going to be seen.</p> <p>JS and colleagues decide to add a section for the 'attention leaflet' and discuss how to describe it.</p> <p>JS to AM) How do you add the actual attention leaflet file?</p> <p>NA explains how to browse the network and add the file</p>	<p>function obvious since it is nowhere near the 'next link. (MODERATE).</p> <hr/> <p>The 'section visibility' functionality is not understood – may be a more fundamental design issue? (MODERATE)</p> <p>Quickly understood and likely they could have worked this out. (MODERATE)</p>
<p>Taxonomy – finding terms</p> <p>JS browses through the list of terms (service provision, transport provision... etc.)</p> <p>JS to AM): Never seen these, don't know where they've come from. What is the 'need'?</p> <p>AM) When we designed...</p> <p>JS) It's what <u>we</u> think? But would that mean anything to Access?</p> <p>JS and AM discuss other terms...</p>	<p>The purpose of the taxonomy was not difficult for the OT's to grasp when explained.</p>

<p>AW) would it make sense to see whose ideas about the categories the taxonomy represents?</p> <p>JS) I'm not that sure what a taxonomy is...</p>	<p>However they showed little interest in it when told it has no direct effect on what the end-user sees.</p> <p>The taxonomy design assumes that, once a term has been defined acceptably for one service worker (for example) then that definition is meaningful and usable for any purpose and all other service workers? If so, this seems likely to be problematic.</p>
--	--

### 9.4 Cost-benefit analysis questionnaire

User Profile

Company/Organization: .....

Job Title: .....

Age range (please check one):

18 - 24

25 - 34

35 - 45

over 45

Sex (please check one):

Male

Female

Name of software being evaluated:

.....

Describe briefly the sorts of task you can carry out with this software:

.....

.....

.....

Have you received any training with this software (please check one):

None

1 day

2 - 3 days

more than 3 days

Did you find the training adequate (please check one):

Yes

No

any further comments on the training?

.....

.....

.....

For how long have you been using computers in your work (please check one):

- less than 6 months
- between 6 months and a year
- 1 - 3 years
- 3 years or more

#### Reduction of development and deployment time

The SmartGov platform reduced (in average) the development and deployment time for **simple** (one-screen) electronic services:

- Less than 10%
- Between 10% and 30%
- Between 30% and 50%
- More than 50%

The SmartGov platform reduced (in average) the development and deployment time for **composite** (many-screens) electronic services:

- Less than 10%
- Between 10% and 30%
- Between 30% and 50%
- More than 50%

How much time costs in average to develop a service by traditional IT means?

- Less than 2 hours
- One day
- One week
- More than one week

What was the improvement in connecting legacy systems to online electronic services:

- Less than 10%
- Between 10% and 30%
- Between 30% and 50%
- More than 50%

Lowering of skills of personnel for creating services

What are the required skills for someone to use the SmartGov Platform?

	1	2	3	4	5	6	7	8	9			0
Low										High	NA	

How many actions for service development/deployment can be performed by low-skills staff?

	1	2	3	4	5	6	7	8	9			0
Low										High	NA	

What would be the reduction of costs in your organization if you could use low-skills stuff?

- Less than 10%
- Between 10% and 30%
- Between 30% and 50%
- More than 50%

Accelerating testing process

The SmartGov platform reduced (in average) the debugging and testing time for electronic services:

- Less than 10%
- Between 10% and 30%
- Between 30% and 50%
- More than 50%

Reduction of maintenance costs

How easy is for you to maintain electronic transaction services that were created by using the SmartGov platform?

	1	2	3	4	5	6	7	8	9			0
Very difficult										Very easy	NA	

The SmartGov platform reduced (in average) the time for maintenance of electronic services by:

- Less than 10%



- Between 10% and 30%
- Between 30% and 50%
- More than 50%

How much does it cost (annually) for your organization to maintain an electronic service (compared to the development/deployment costs of the service)?

- Less than 10%
- Between 10% and 30%
- Between 30% and 50%
- More than 50%

Reusability

It was very easy to re-use knowledge and components stored in the SmartGov platform:

	1	2	3	4	5	6	7	8	9			0
Strongly disagree										Strongly agree	NA	

In how many cases did you re-use knowledge and components stored in the SmartGov platform?

	1	2	3	4	5	6	7	9			0
Very rare									Very often	NA	

Improvement of working conditions

Did the platform facilitate you in the service development and maintenance process?

	1	2	3	4	5	6	7	8	9			0
Strongly disagree										Strongly agree	NA	

What was the value of the SmartGov platform in helping you organize the electronic service life-cycle?

	1	2	3	4	5	6	7	8	9			0
Very low										Very high	NA	

Improved services to the citizen

How stable were the deployed services?

	1	2	3	4	5	6	7	8	9			0
Unstable										Stable	NA	

What was the overall impression of your end-users from using electronic services provided through the SmartGov platform?

	1	2	3	4	5	6	7	8	9			0
Frustrating										Satisfying	NA	

## 9.5 CEC resources for Smartgov

Roles	Hours				Total	Unit Cost <sup>13</sup>	Total Cost
	Design <sup>9</sup>	Development <sup>10</sup>	Deployment <sup>11</sup>	Training <sup>12</sup>			
Management	40	10	10	6	66	£40	£2,640
Domain Experts	250	50	20	24	344	£33	£11,467
IT Staff	35	10	10	0	55	£100	£5,500
System Admin	10	15	15	4	44	£33	£1,467
Users			40	40	80	£33	£2,667
<b>Total</b>	<b>335</b>	<b>85</b>	<b>95</b>	<b>74</b>	<b>589</b>		<b>£23,740</b>
<b>IT Expenditure<sup>14</sup></b>							
Hardware							£4,834
Software							£5,565
					<b>Grand Total</b>		<b>£34,139</b>

<sup>9</sup> Hours designing the service through the life-cycle, including process mapping.

<sup>10</sup> Hours spent inputting service design into the Smartgov development environment, e.g TSEs and Forms

<sup>11</sup> Hours spent actually deploying the service in a live environment (estimated)

<sup>12</sup> Hours on training the various users on the Smartgov application

<sup>13</sup> Unit costs is based on £ per day per FTE.

<sup>14</sup> Cost of procuring and installing all Smartgov components.