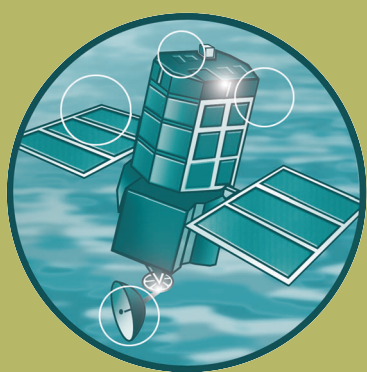


Joint Defra/EA Flood and Coastal Erosion
Risk Management R&D Programme

R&D Software Development Projects Guidance for Research Contractors

R&D Technical Report FD2121/TR2



Joint Defra/EA Flood and Coastal Erosion Risk
Management R&D Programme

R&D Software Development Projects - Guidance for Research Contractors

R&D Technical Report FD2121/TR2

Produced: September 2007

Author: Jon Wicks and Rob Honeywill, Halcrow

Statement of use

The report is intended to guide those involved in software development of joint Defra/EA Flood and Coastal Erosion Risk Management R&D modelling outputs.

Dissemination status

Internal: Released internally

External: Released to public domain

Keywords: software requirements

Research contractor: Halcrow Group Ltd, wicksjm@halcrow.com

Defra project officer: Prof. Edward Evans

Science Theme Manager: Dr. Suresh Surendran

Publishing organisation

Department for Environment, Food and Rural Affairs
Flood Management Division,
Ergon House,
Horseferry Road
London SW1P 2AL

Tel: 020 7238 3000

Fax: 020 7238 6187

www.defra.gov.uk/environ/fcd

© Crown copyright (Defra) 2007

Copyright in the typographical arrangement and design rests with the Crown. This publication (excluding the logo) may be reproduced free of charge in any format or medium provided that it is reproduced accurately and not used in a misleading context. The material must be acknowledged as Crown copyright with the title and source of the publication specified. The views expressed in this document are not necessarily those of Defra or the Environment Agency. Its officers, servants or agents accept no liability whatsoever for any loss or damage arising from the interpretation or use of the information, or reliance on views contained herein.

Published by the Department for Environment, Food and Rural Affairs (Sept 2007). Printed on material that contains a minimum of 100% recycled fibre for uncoated paper and 75% recycled fibre for coated paper.

PB No. 12794B

Acknowledgements: This document has benefited from the involvement of the following organisations: Environment Agency CIS, HR Wallingford, Wallingford Software, EdenVale Modelling Services, JBA and Wallingford HydroSolutions.

Executive Summary

Software products form important outputs from many Flood and Coastal Erosion Risk Management (FCERM) R&D projects. For these software products to be readily useable by operating authorities and their consultants, it is important that the software adheres to relevant software standards. The FD2121 project has developed guidance material to assist research contractors in understanding and conforming to the relevant standards.

The primary source of relevant standards for FCERM software is the Environment Agency's Corporate Information Services (CIS) 'Enterprise Architecture: Technical Reference Model' (TRM). Guidance material for R&D contractors has been developed from the TRM and from discussions with CIS staff and other industry experts. The guidance contained in this report will guide the R&D contractor towards producing conforming software and, importantly, will foster early and informed discussions between the R&D contractor and CIS.

The guidance is arranged in sections covering the pre contract award, post contract award and software deployment planning stages of an R&D project involving software development. The pre contract award guidance is designed to help gauge the appropriate level of CIS engagement in the project and identify any major software issues before award of contract. The post contract award guidance forms the bulk of the guidance and covers software and hardware platforms, database requirements, application architecture, development tools/languages, security and testing. The final guidance section covers implementation planning (deployment, support and maintenance and data storage issues).

This document provides an important and accessible source of guidance which, when followed, will help improve take up of R&D software outputs. The guidance document is only one element in improving take up of the software outputs of high-quality and appropriate research - on-going dialogue and planning will be required, throughout the development life cycle, involving the research contractor, CIS and Agency/Defra managers.

Contents

	Executive Summary	ii
1.	Introduction	1
1.1	Intended Audience	1
1.2	Background	1
1.3	The purpose of this document	1
1.4	How to use this document	1
1.5	Engaging with CIS	2
1.6	Project Lifecycle	3
2.	Pre Contract Award Guidance	5
3.	Post Contract Award Guidance	7
3.1	General Project Information	7
3.2	Agency Software Platforms	8
3.3	Hardware Platforms	8
3.4	Database Usage	11
3.5	Non Database Data	13
3.6	Application Architectural Compliance	14
3.7	Development Tools & Languages	16
3.8	Modular, Sustainable Development	17
3.9	Security (User & Data)	18
3.10	Testing and Acceptance	19
4.	Implementation Planning	23
4.1	Software Deployment	23
4.2	Transition to Support and Maintenance	23
4.3	Storage Requirements	24

Figures

- Figure 1 – Environment Agency Improvement Cycle
- Figure 2 – R&D Project Lifecycle
- Figure 3 – Agency Platforms Flowchart
- Figure 4 – Database Usage Flowchart
- Figure 5 – Application Architecture Flowchart
- Figure 6 – IT Project Delivery Process (V-Model)

Appendices

- A Agency Reference Documents
- B Application Types
- C Deployment
- D Guidance Notes

1. Introduction

1.1 Intended Audience

The intended audience for this document is contractors embarking on R&D projects for the Environment Agency which involve software development. The primary focus is for use on Flood and Coastal Erosion Risk Management (FCERM) R&D projects but the guidance may prove useful for other projects.

1.2 Background

One key deliverable from many FCERM R&D projects is software products. Historically these software deliverables have not been readily usable by all members of the anticipated user communities. A particular problem has been access to the software deliverables by Agency technical staff. A further issue has been that the software may not have been developed using 'best practice' approaches which may result in difficulties in taking the software forward after completion of the R&D project. This guidance document is an output from the FD2121 R&D project which helps address these problems by providing guidance on a common set of software requirements for FCERM R&D projects. The guidance has been developed with input from key stakeholders including the Agency's Corporate Information Services (CIS). A key element of the guidance is the need for on-going engagement between the research contractor and CIS.

1.3 The purpose of this document

The purpose of this document is to provide guidance for software development projects in order to satisfy the project, CIS and any non-Agency stakeholder requirements (the business requirements). In order to follow this guidance, contractors and CIS are likely to need to engage in discussions at several times through the project. The time taken, and the impact of meeting planning on timescales, should not be underestimated and should be factored in to project plans. Please note that the content of this document does not constitute an exhaustive list of questions regarding the proposed solution architecture, but addresses some important aspects of the development which will generally prompt further discussion.

1.4 How to use this document

The "Pre Contract Award" guidance (Chapter 2) should be completed before commencement of the software development project in order to gauge the level of CIS involvement required (and help identify any 'red flag' issues) early in the life of the project. Depending on the necessary level of CIS involvement identified, the Agency may wish to assign a strategy analyst and project architect to guide the contractor through the project. This will almost certainly be the case for any software to be installed on Agency systems.

The "Post Contract Award" guidance (Chapter 3) is for use during the project and will steer contractors towards 'good practice'. Use the associated decision

trees as a guide to your likely level of compliance with the guidance. It is suggested that you edit the appropriate tables in this document and submit and discuss with CIS (as early on in the project as is practical).

‘Implementation planning’ guidance (Chapter 4) is used to prepare a software development project for implementation, for example, covering the need for support and maintenance. Support, maintenance and training requirements are not considered in depth in this guidance document.

Appendix D provides additional guidance notes on a range of common issues.

As part of completing the yellow highlighted areas of the tables in this guidance document, it may be the case that the development in question will not be able to fully conform to Agency requirements. The Agency CIS project architect will be able to help you to adjust the architecture to be compliant or guide you through the additional procedures for seeking acceptance for your proposed architecture.

The Agency’s “Enterprise Architecture: Technical Reference Model” document should be considered the most up to date source of information wherever particular CIS standards are referred to, but this is not the only pertinent document that will need to be consulted (other key documents include those listed in Appendix A). This and other project specific documents can be obtained from the strategy analyst or project architect. Please note CIS Enterprise Architecture is constantly evolving and you will need to obtain the latest versions of any required documentation when embarking on a new project.

Where “non Agency users” are mentioned, this refers to possible non-Agency users of developed software, typically other operating authorities and consultants.

1.5 Engaging with CIS

It is ***very important*** to note that if the project will deliver software that will be deployed within the Environment Agency, it will be necessary to enter into a dialogue with CIS, via the strategy analyst and/or project architect, to fully review and understand what compliance to the CIS Enterprise Architecture specifically means to the project. Consideration and completion of this document in itself will not supply the required level of detail to confirm a satisfactory solution.

As projects pass the “Idea and Proposition” stages of the Improvement Cycle (see Figure 1 below), the key CIS contact is a strategy analyst. The strategy analyst may assign further staff to the project for subsequent stages, such as a CIS project architect.

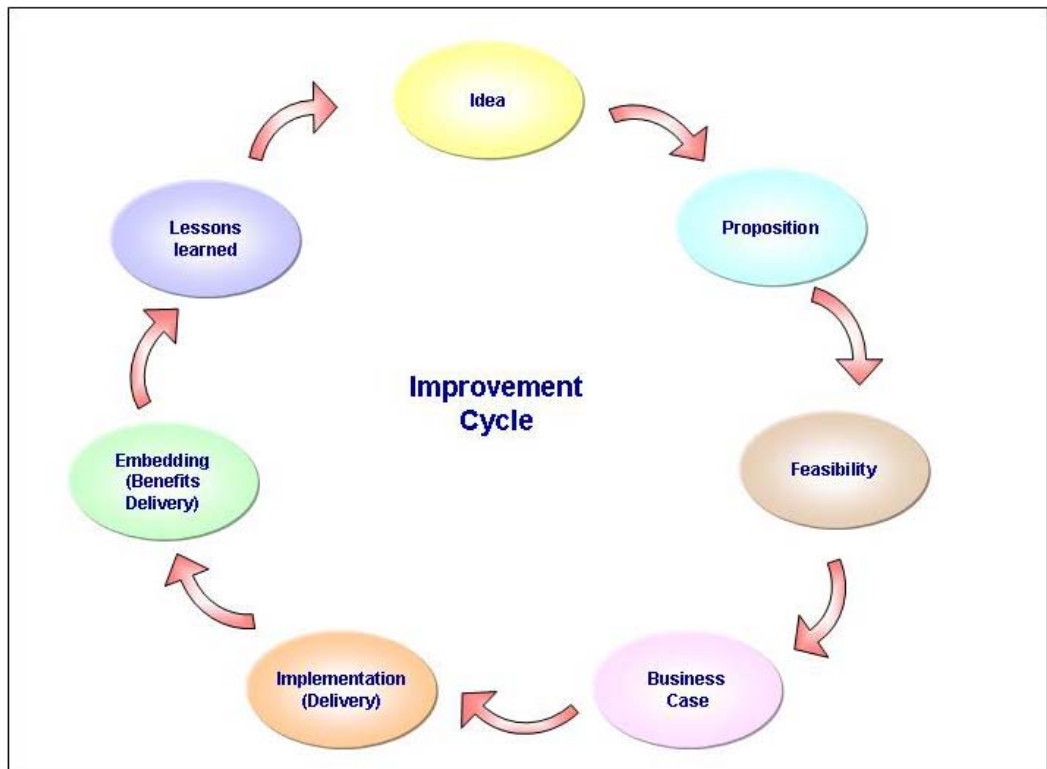


Figure 1 – Environment Agency Improvement Cycle

1.6 Project Lifecycle

The project lifecycle diagram shown in Figure 2 illustrates the main software development related tasks of an R&D project, showing the flow of tasks performed by the Agency and those to be performed by the supplier in order to facilitate a smooth development process. It does not necessarily cover the full lifecycle of the software, for example the delivery of support, maintenance and training are not covered.

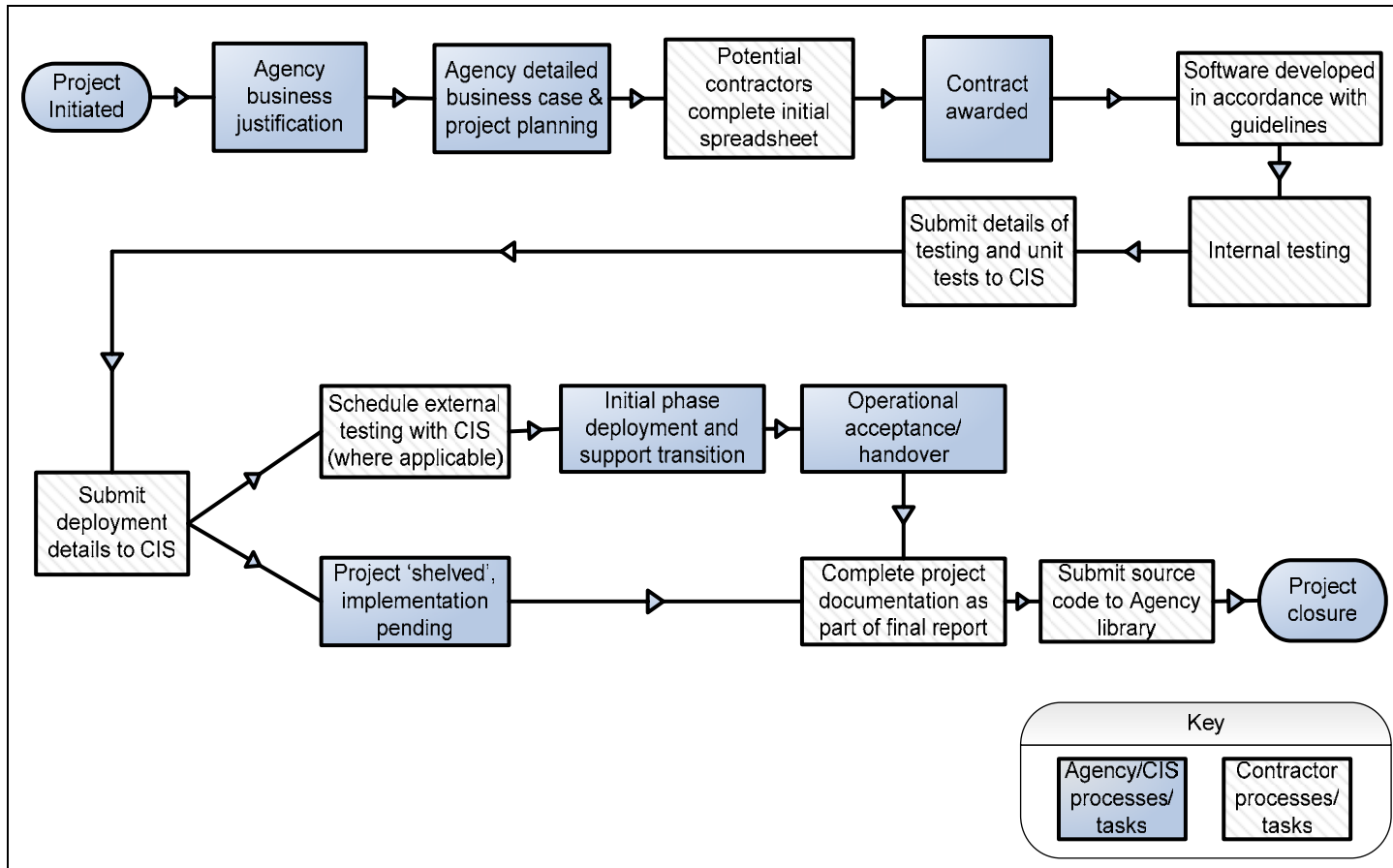


Figure 2 – R&D Project Lifecycle

2 Pre Contract Award Guidance

Either the potential research contractor or the Agency/Defra manager/officer should complete the 'Rating' column of the "Pre Contract Award" table provided below. The completed table will assist the Agency in defining the likely level of CIS involvement in the project and will foster a common understanding of likely key software issues before the start of the contract. During the lifetime of the project, this table can be referred to in order to confirm whether the project is being developed in accordance with the initial expectations.

This table should not be considered "full and complete" as there will almost certainly be project specific items which will need to be considered on a case by case basis. These items can be appended to the base of the completed table.

It is recommended that reference is made to Appendix B to ascertain where your project lies within the range of different project types. In addition, the Agency Flood Risk Management Modelling Strategy provides best practice advice on a range of related areas such as data, standards and procurement and access to software.

Pre Contract Award Table

Development Issue	Low	Medium	High	Rating L/M/H or N/A	Comments
End user type	No end users	Non Agency users	Agency users		Agency users are of the most concern to CIS and require the most involvement by them.
Development type	Updating existing software	New software reusing existing modules	Brand new software		Write conforming software to satisfy CIS standards, provide strong justification for any exceptions.
Total no. of users/ concurrent users	Single	Tens	Thousands		Large scale deployment on Agency systems will almost certainly require appointment of a Project Architect.
General support and handover requirements	CIS involvement not expected	Some CIS involvement required	Significant CIS involvement required		It is important to involve CIS as early as possible in planning the transition of the software to support and maintenance. This will give them the opportunity to spot potential problems and plan ahead adequately.
Physical location of users	Single location	Localised locations	Widespread locations		Installations at multiple physical locations require more effort by CIS. Where it is known this will be the case, CIS should be informed as early as possible.

Development Issue	Low	Medium	High	Rating L/M/H or N/A	Comments
Database usage	No database	Lightweight database (no separate install or driver installation)	Corporate / Enterprise database		If the developed software is to be run at both Agency and non-Agency sites then if possible develop for the standard Agency database. If this is not possible or will harm uptake by non-Agency parties write database agnostic software which will run on both the standard Agency databases and those in use by non-Agency parties (a recommended approach in general).
Likely network traffic generated by application	None - the application will be "stand alone".	Light bandwidth usage likely (e.g. 20kb avg / 100kb burst)	Heavy bandwidth usage likely (e.g. 100kb avg / 5MB burst)		If the software is to be put onto Agency machines it is important CIS are informed as early as possible about application bandwidth requirements.
User interface type	No client-side component	Browser	"Rich client"		"Rich client" software does not follow the Agency CIS standards. Justification and discussion with CIS will be required.
Predicted regularity of software updates requiring code changes	Never	Bi-annually	More than monthly		A high level of update frequency will normally be unacceptable to CIS. You should talk to CIS at the earliest opportunity if this will be necessary.
Status of product before R&D work begins	Industry standard product	R&D work based on established concepts	Concept or pre-concept		As a general assumption software based on a new "concept" will have a higher risk of failure/changing architecture.
Security implications	No data sensitivity / No user security issues	Some data sensitivity / Some user security issues	High data sensitivity/ Complex user security issues (e.g. external access required)		Issues to consider include system access by internal / external / standalone users (from intranet, Web etc.) as well as information security from a data sensitivity perspective.

You should consider IPR (background and foreground) issues (methods, compiled software, source code and necessary data) providing information below explaining how licensing issues will be dealt with both during the R&D and to enable use of the outputs by end users.

3 Post Contract Award Guidance

This section of the guidance is designed to raise awareness of key issues affecting compliance with CIS Enterprise Architecture. Please note that your responses in this section (recorded in the yellow fields of the subsequent tables) are intended to serve as a “launch point” for the dialogue between the contractor and CIS regarding Enterprise Architecture compliance. The outcome of these further discussions should be reflected in the normal project documentation (e.g. requirements specification or solution architecture) and be subject to normal sign-off.

3.1 General Project Information

Project Ref	
Project Title	
Contact name	
Company	
Tel	
Email	
Target Audience	
No. of Agency users (approx) if applicable	
Project overview	
Architectural Diagram	
Where appropriate, insert below a diagram showing a high level architectural overview of the proposed system. This should show how the proposed solution is made up in terms of hardware and software and the links between the components of the system.	
Other relevant information	

3.2 Agency Software Platforms

Develop software to run harmoniously on existing Agency systems (hardware/network/software) and non-Agency systems to maximize user acceptance.

Software developed to be run on Agency systems needs to run on the platforms the Agency already uses (or will have at the time of delivery) and can support (for a synopsis of Agency platforms, correct at the time of writing see the latest, "Enterprise Architecture: Technical Reference Model").		
Response		Rationale
Will run	<input type="checkbox"/>	
Will not run	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	
Software to be run on non-Agency machines should be written to maximize uptake of the software by these 3 rd parties, i.e. write the software to run on the most commonly used platforms.		
Response		Rationale
Implemented	<input type="checkbox"/>	
Not Implemented	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	
Software to be run on both Agency machines and non-Agency machines should marry the requirements of the two in the best way possible. This issue must be discussed with CIS.		
Response		Rationale
Plan in place	<input type="checkbox"/>	
Not considered	<input type="checkbox"/>	
Agreed with CIS	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	

If the timescale of development is very short the information in this document and the current "Enterprise Architecture: Technical Reference Model" can be considered up to date and should be your main reference. On larger timescale developments (over 6 months) discuss possible changing Agency platforms with CIS.

3.3 Hardware Platforms

Software developed to be run on Agency machines must be developed to run on existing hardware platforms at the Agency.		
Response		Rationale
Will run	<input type="checkbox"/>	
Will not run	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	

Any network bandwidth usage by the software must be communicated to CIS, including details of average and burst activity (see Appendix D.4 for further information).		
Response		Rationale
Communicated	<input type="checkbox"/>	
Not communicated	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	
(Server) Processor usage should also be communicated to CIS (see Appendix D.4 for further information).		
Response		Rationale
Communicated	<input type="checkbox"/>	
Not communicated	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	
The need for any peripherals will need to be agreed with CIS.		
Response		Rationale
Agreed with CIS	<input type="checkbox"/>	
No agreement	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	

See Figure 3 – Agency Platforms Flowchart for more guidance.

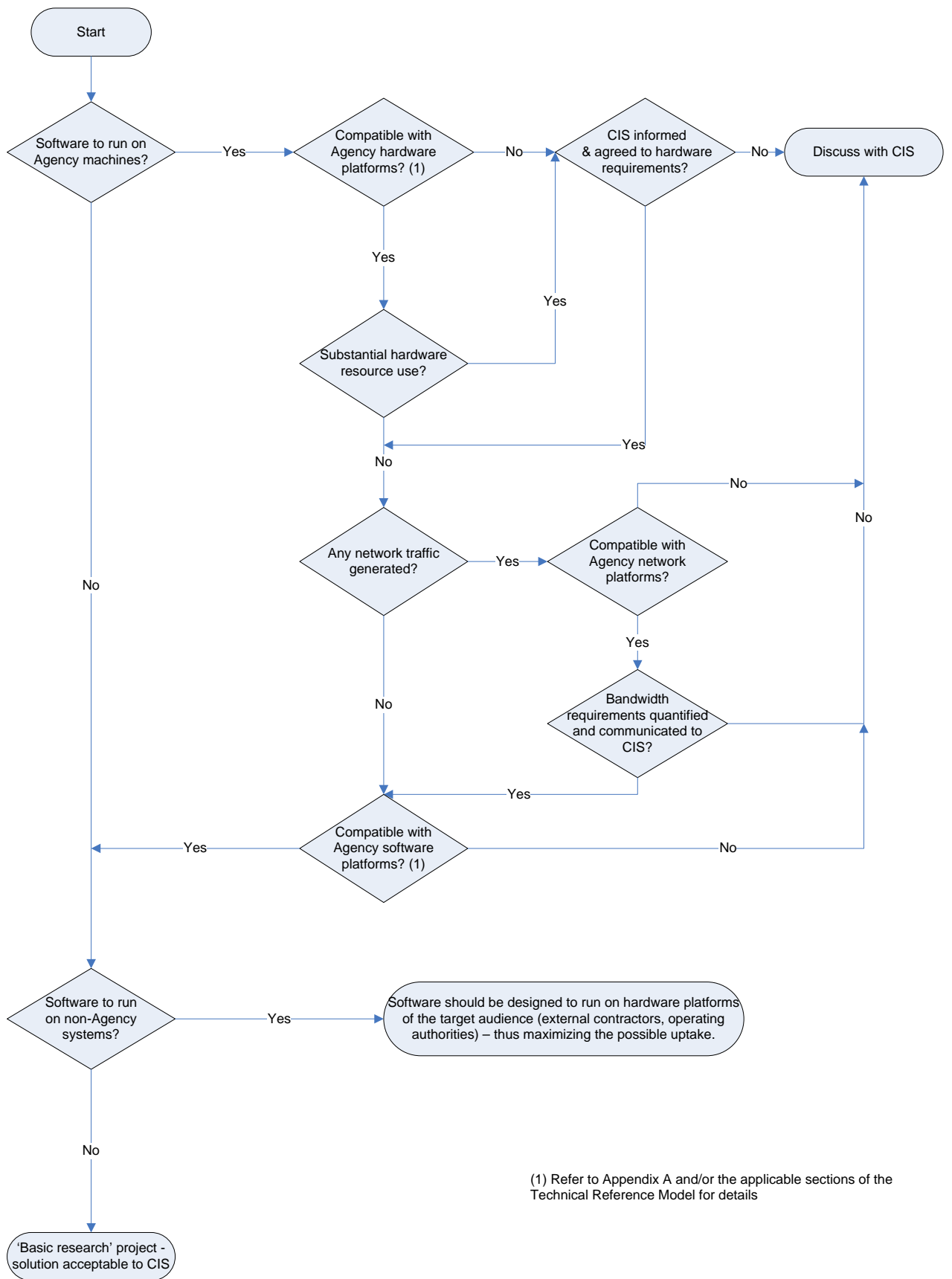


Figure 3 – Agency Platforms Flowchart

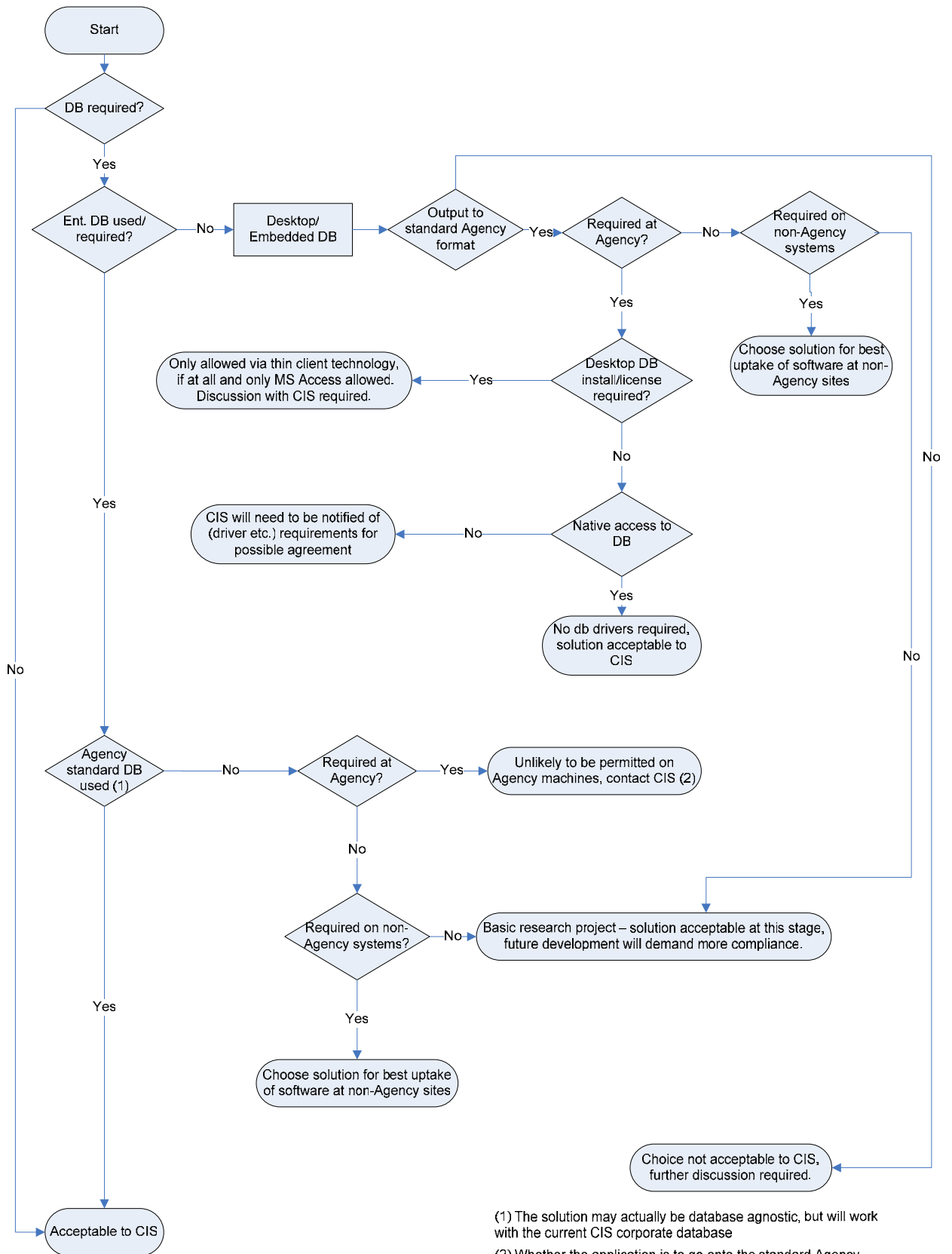
3.4 Database Usage

Develop “Enterprise” database based software to run on the Agency standard database. Develop “Desktop database” software in a way that doesn’t require client installs and is not locked to a proprietary format.

Database based solutions must run on the standard Agency database (currently Oracle) if the program is to be run at the Agency. Databases other than the standard enterprise database will not be allowed onto Agency systems.		
Response		Rationale
Will run	<input type="checkbox"/>	
Will not run	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	
If the developed software is to be run at both Agency and non-Agency sites then if possible develop for the standard Agency database. If this is not possible or will harm uptake by the non-Agency users then write database agnostic software which will run on both the standard Agency databases and those in use by the non-Agency entities (a recommended approach in general).		
Response		Rationale
Agency standard	<input type="checkbox"/>	
DB Agnostic	<input type="checkbox"/>	
Other	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	
Where software is to be developed for use at the Agency and “enterprise” databases are inappropriate for the task, desktop/embedded databases may be required. In these cases native access from within the application would be required, with no application or client installs on Agency desktop PCs. It is also required that output to a non-proprietary format (e.g. XML) is easily available from the database.		
Response		Rationale
Will comply	<input type="checkbox"/>	
Will not comply	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	

If the software is only to run at non-Agency sites, it is still preferable that it works with the Agency standard database.

See Figure 4 – Database Usage Flowchart for more guidance.



(1) The solution may actually be database agnostic, but will work with the current CIS corporate database
 (2) Whether the application is to go onto the standard Agency desktop or a small group of Agency machines, it is unlikely a corporate database other than the Agency standard database will be allowed.

Figure 4 – Database Usage Flowchart

3.5 Non Database Data

Do not create new proprietary data formats; store ancillary data, such as program settings, using XML file formats (see Appendix D.1 for more details).

Software developed should not write to / read from its own proprietary format; in general XML should be used for new formats. The only justification for creating proprietary formats in extreme cases might be due to performance issues, but this would have to be agreed with CIS beforehand. Where binary formats are proposed the Agency would expect to receive documentation as to the format of these and also expect some ability to handle/produce XML input/output. It is acceptable to use the de facto “standard” file formats that the Agency itself uses for things such as GIS systems.		
Response		Rationale
No proprietary formats	<input type="checkbox"/>	
Proprietary formats	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	
Where ancillary data (program settings etc.) is required to be stored you are expected to use XML as the format for this data.		
Response		Rationale
XML used	<input type="checkbox"/>	
XML NOT used	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	

3.6 Application Architectural Compliance

Develop applications using an n-Tier, server side logic, thin client browser based approach, wherever this can satisfy the project requirements

New software developments to run on Agency machines should follow the Agency standard application architecture - an n-Tier approach, utilising a “business logic” server side in conjunction with a browser based thin client. Where this approach cannot satisfy the project requirements you will need to agree an alternate strategy with CIS, strong justification will be required (see Appendix D.6 for one possible alternative - Citrix).		
Response		Rationale
Will comply	<input type="checkbox"/>	
Will not comply	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	
Software developed to run both at non-Agency sites and on Agency machines should follow the above Agency application architecture wherever possible. If this is not practical for non-Agency entities then a dual interface approach (using the same basic code base) is preferred, e.g. a rich client application at non-Agency sites and standard Agency application architecture for Agency machines.		
Response		Rationale
Agency standard	<input type="checkbox"/>	
Dual interface	<input type="checkbox"/>	
Other	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	

The Agency would still prefer software which will not run on Agency machines to follow the Agency application architecture, but this is not mandatory given appropriate justification.

Where updating an existing architecturally non-compliant program, the Agency would encourage a migration to a compliant state.

See Figure 5 – Application Architecture Flowchart for more guidance.

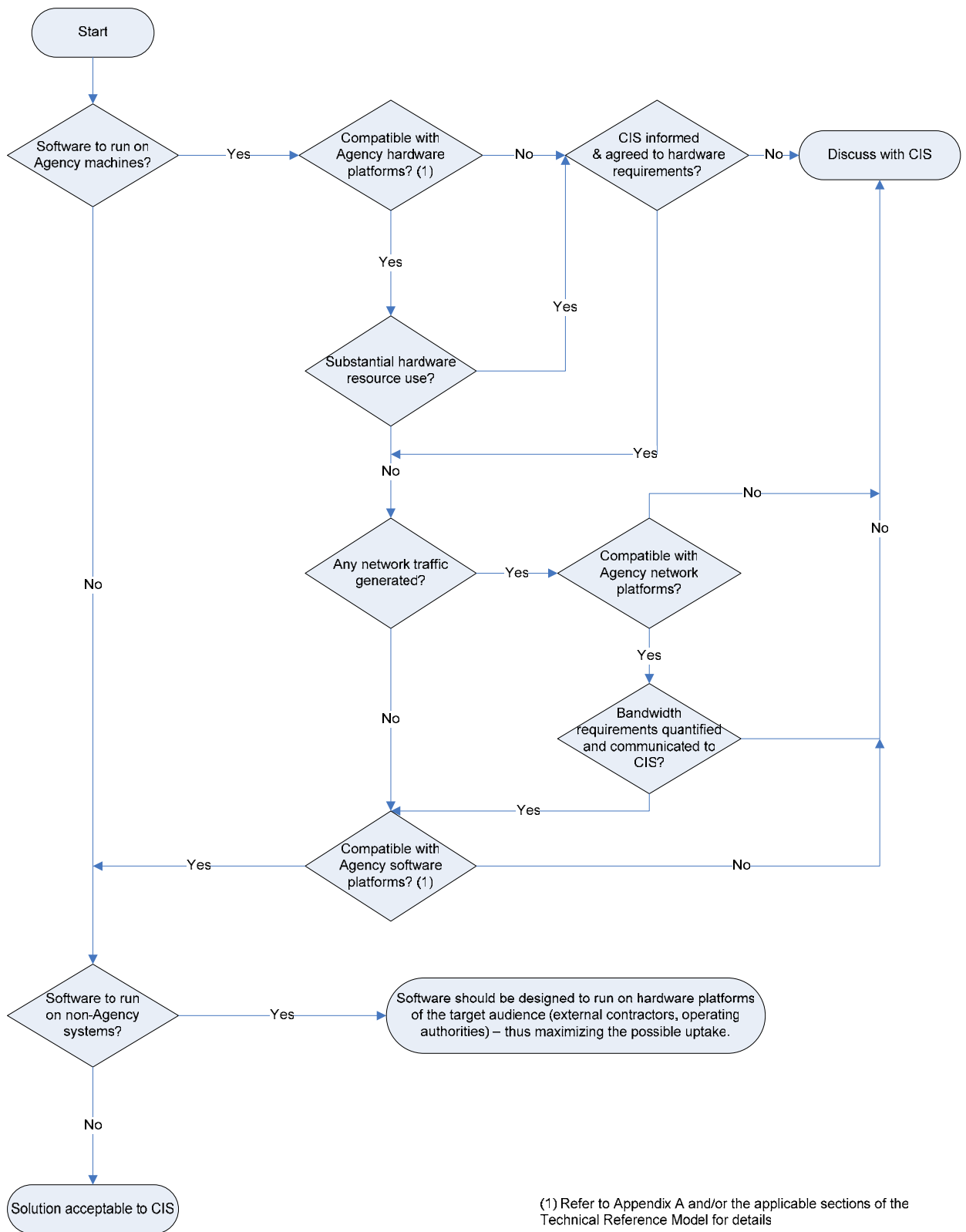


Figure 5 – Application Architecture Flowchart

3.7 Development Tools & Languages

Write software using Agency standard development tools.

The applicable development tool standards can be found in the latest, “Enterprise Architecture: Technical Reference Model” Agency document. In line with the application architecture requirements, the Agency standards for Enterprise scale applications is component based, n-Tier, using an application server.

When considering the development tool guidelines below, indicate where the following items are part of the justification for exceptions (research contractors should bear in mind that the overriding concerns for the Agency are **total cost of ownership** of the software over its lifetime and whether the software satisfies **essential business requirements**):

- The program will use an existing code base written in another language (consider migration and/or making “callable” from Agency standard tools)
- Cost implications of implementing and/or developing using Agency standard tools (e.g. software license costs)
- Negative impact on functionality of software due to use of standard tools (e.g. non-interoperability with 3rd party libraries)
- Performance considerations
- Impracticality of and resistance to installation at non Agency sites
- Agency standard tools will not deliver software satisfying project requirements

One of these in isolation may not be justification for exceptions, so communicate all that apply as well as any other mitigating circumstances you believe to be relevant.

Ideally, all new development should take place in the standard development language (currently Java). Any deviation from this requires justification. If the software is not to run on Agency machines then justification will be easier.		
Response		Rationale
Will comply	<input type="checkbox"/>	
Will not comply	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	
	<input type="checkbox"/>	
Where the exact CIS standards cannot be met, you should provide justification. Note that an architecturally compliant solution (i.e. thin client browser based) is preferable to a strict adherence to specific tools.		
Response		Rationale
Standards fully met	<input type="checkbox"/>	
Architectural compliance	<input type="checkbox"/>	
Other	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	

Where the Agency standard development language cannot satisfy the project requirements, for example modelling applications, then the use of a different language could be justified – the Agency standard for modelling applications is currently C++.		
Response		Rationale
Agency standard	<input type="checkbox"/>	
Non standard	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	

Where a non-compliant existing application is being updated, the Agency would prefer a migration to standard development tools.

Where there is the need for some development in “legacy” languages the Agency requires that this is done in a sustainable way – as an example Fortran functionality could be “wrapped in Java” using XML, controlled by XML inputs/outputs or put into a documented DLL callable by other languages (preference in that order) to mitigate the risk to the Agency of developing in that language. If this cannot be done for any reason then a discussion with CIS will be required.

At the time of writing, the Agency policy regarding applications written for the .NET framework is to allow “commercial off the shelf” packages to be installed, but not bespoke development to be undertaken (see Appendix D.2 for more information). Justification for bespoke .NET development must address total cost of ownership issues as well as technical issues to achieve essential business requirements.

3.8 Modular, Sustainable Development

Develop modular, easily extensible and reusable software to obtain maximum value from the Agency’s investment.

Contractors should develop their software in as modular a fashion as possible, using loosely coupled functions/methods probably via Object Oriented development.		
Response		Rationale
Done/will do	<input type="checkbox"/>	
Not appropriate	<input type="checkbox"/>	
Not considered	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	
Develop software so that user interfaces are decoupled from program logic as much as possible.		
Response		Rationale
Done/will do	<input type="checkbox"/>	
Not appropriate	<input type="checkbox"/>	
Not considered	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	

The use of design patterns and other modern programming techniques should be considered. Use techniques such as inheritance and encapsulation appropriately and to their best advantage.		
Response		Rationale
Done/will do	<input type="checkbox"/>	
Not appropriate	<input type="checkbox"/>	
Not considered	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	
Developed software may contain functionality that itself will be useful for reuse in other software perhaps by another contractor or the Agency itself. You should make this as easy to achieve as possible and should endeavour to make it possible regardless of development environment.		
Response		Rationale
Done/will do	<input type="checkbox"/>	
Not appropriate	<input type="checkbox"/>	
Not considered	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	
Achieve maximum interoperability by following the CIS standards along with various methods such as creating “wrappers” around software, separating code into libraries/componentisation, open communication and data exchange via SOAP and XML.		
Response		Rationale
Done/will do	<input type="checkbox"/>	
Not appropriate	<input type="checkbox"/>	
Not considered	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	
External interfaces and available functionality should be clearly documented.		
Response		Rationale
Done/will do	<input type="checkbox"/>	
Not appropriate	<input type="checkbox"/>	
Not considered	<input type="checkbox"/>	
N/A	<input type="checkbox"/>	

You should consider the use of coding standards and provide details and/or references to these below.		

3.9 Security (User & Data)

Only implement application level security where absolutely necessary.

It is recommended developers refrain from implementing application level security unless absolutely necessary, especially software which is to be run on

Agency systems. Any application that does this will need to be agreed with CIS. Acceptable reasons for implementing application level security are to limit access to applications where there is high processor usage which needs to be managed and where use needs to be limited to authorised users. Any securing of the system should be communicated (e.g. “the program will use standard Windows XP security”).

It should be borne in mind that a browser based application running on Agency systems is potentially accessible to all users whether valid or not - without any user level security.

If the software is to store any information pertinent to the Data Protection Act, inform CIS of what data is to be stored and how it is to be stored. This is very important. However, please note, there are other aspects of regulatory and legal compliance – guidance should be sought and will be given by the Information Management Unit (IMU) via the CIS Project/Enterprise Architect.

A recommendation when developing (server based) software is to have a management front end accessible through a browser client to perform administration.

The Agency expects that applications will not be locked down on the desktop for any reason after installation (for example, some applications require you to connect to the internet to verify the license).

It is important to explain the nature of the data that will be used or created by the program, along with how that data will be managed within the system.

Comments

3.10 Testing and Acceptance

Plan testing from the beginning of the project and follow the Agency testing model.

The Agency asks that contractors use the “V” model of testing (see Figure 6, “*IT Project Delivery Process (V-Model)*” below for an understanding of how this fits into the Agency development lifecycle).

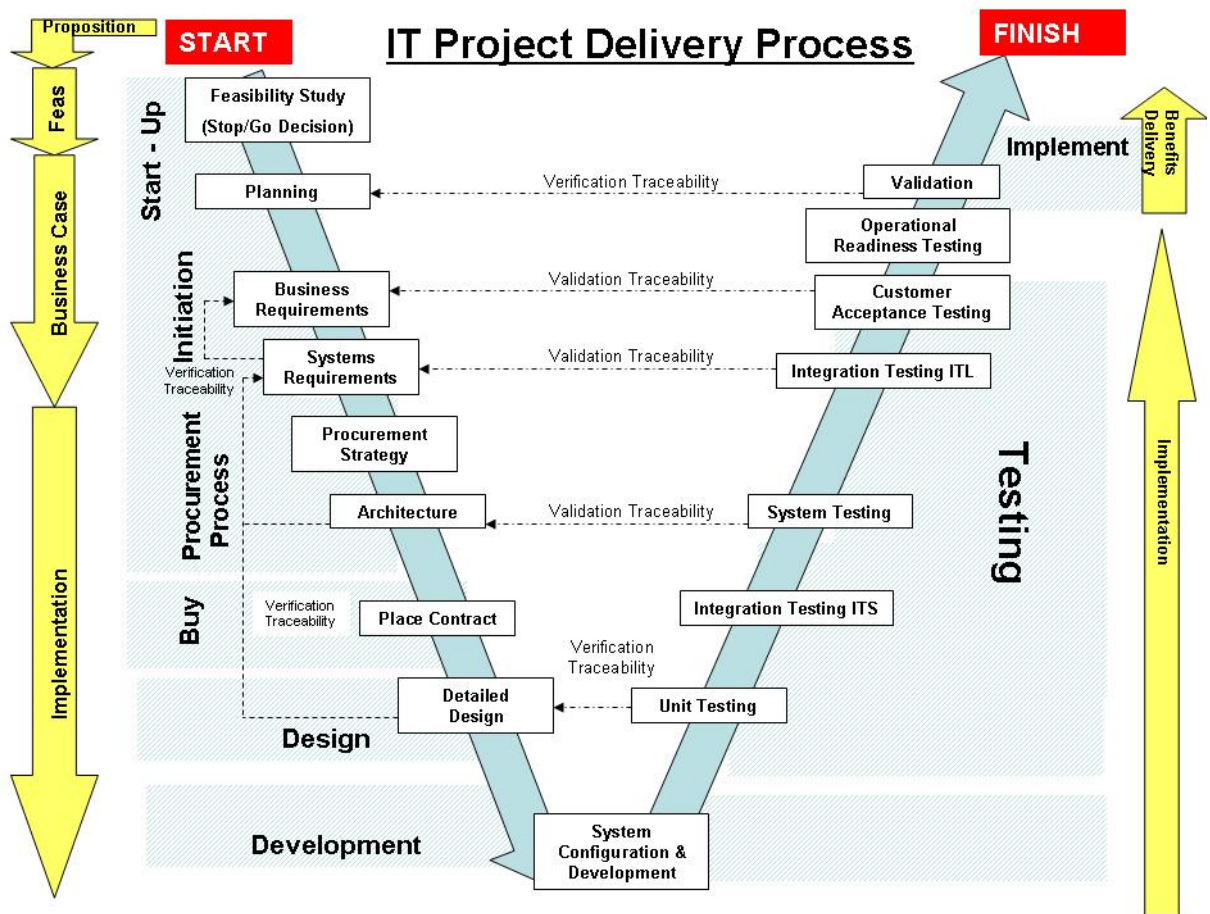


Figure 6 – IT Project Delivery Process (V-Model)

Agreeing the testing and acceptance strategy will form part of the agreement when a new project is begun and should be done before any test cases are written. It is important to identify the contractual significance of acceptance within the strategy. Testing and acceptance is a conversation with CIS. Where software is to be installed on Agency PCs, the functional specification should include a test strategy. This should adopt the current Agency principles, which will be provided by the project architect.

The contractor testing process should provide evidence of the tests run and results. In particular the Agency would like to see unit and system integration tests.

Testing methods other than the 'V' model would need justification and to form part of the project agreement. Software updates need to go through proper pre-production testing and the testing plan should be run against them.

Where software is to be installed on Agency machines, CIS need to be given ample notice in order for them to set aside time to test the application. The complexity of the application and the type of architecture will have a bearing on the time and effort required.

The contractor is responsible for (or will be involved in) the following areas of testing:

3.10.1 Unit Testing

The objective of Unit Testing is to ensure that reliable program units are produced that meet their requirements and to identify errors in program logic. Typically, the developer who coded the unit will design and run a series of tests to verify that the unit meets its requirements. Each unit should be tested individually and in isolation by exercising its inputs and observing its outputs or behaviour. There are widely used tools (unit testing frameworks) available to assist in this task such as JUnit for Java and NUnit for .NET languages.

3.10.2 Integration testing

Components of code are assembled into sub-systems and linked to form a complete system. The objective is to test the relationship and links between individual units of code.

3.10.3 System Testing (including FAT – Factory Acceptance Testing)

System Testing incorporates both functional and non functional testing. Functional system testing is focused on testing the system based on what it is supposed to do as defined in the functional requirements. Non functional system testing looks at those aspects that are important yet not directly related to what functions the system performs. Non-functional requirements are just as important as functional requirements and in all cases it is vital that these are tested prior to the launch of the system. It includes aspects like performance and security which are considered vital for today's web based applications.

3.10.4 System Test Process (involvement by 3rd Party Suppliers)

To ensure quality, prior to delivery to the Agency, the 3rd Party supplier will be required to demonstrate compliance by hosting Acceptance Testing on their site, this will in effect be a pre-User Acceptance Test known by the Agency as Factory Acceptance Testing (FAT). This process will have defined, agreed acceptance criteria and will be subject to a Test Readiness Review Meeting prior to deployment on any Agency infrastructure. In addition, the 3rd party supplier will assist in the installation of the new application on the agency site. This will be done as part of the Acceptance Testing stage and will again require the 3rd party supplier to show compliance on the Agency's infrastructure.

3.10.5 Site Acceptance Testing (SAT)

SAT will be used as a quality check to ensure that when the application is installed onto the Agency infrastructure that it functions correctly with no critical errors. The 3rd Party supplier will be on site to assist and ensure that the application, specifically the server code (where applicable), can be deployed correctly. The 3rd Party supplier will then conduct a subset of their System Tests to prove that the application can function without any critical errors. If possible,

at this stage the opportunity should be taken for the 3rd party to forward any test assets such as test scripts that may be re-usable by the Agency.

3.10.6 User Acceptance Testing (UAT)

UAT will take place on the Agency infrastructure and will confirm that the system meets its business requirements.

When dealing with 3rd parties, the Test Management Process should also be carefully considered. The Agency Test Manager/Coordinator will decide if the project should produce a combined Test Strategy taking into account all testing stages or if separate strategies should be produced, one by the Agency and one by the 3rd party. This will also apply to Test plans and approach documents.

Comments

4 Implementation Planning

Follow the Agency deployment procedures, determine who is responsible for support and enable a smooth implementation with no “nasty surprises”.

4.1 Software Deployment

For software to be installed on Agency machines, the contractor must follow the Agency standards for software deployment/install – see Appendix C for more detail.

Suppliers of proposed systems must provide documented support for the application integration task to the standardised Agency desktop.

Where desktop software is to be installed on standard Agency desktops, CIS expect the contractor to detail the impact on the workstation, including the following:

- Assurance that the installation/application makes no changes to or deletions of protected operating system files.
- A list of dependent components (e.g. Active X controls, DLLs, drivers etc.)
- A list of Dynamic Link Library (DLL) and Application Programming Interface (API) calls.
- A list of known changes to registry keys

Comments

4.2 Transition to Support and Maintenance

Before implementation, the decision needs to be made as to who performs the support function – the supplier, CIS or both. In general software which is not to be run on Agency machines will be the responsibility of the original contractor.

The service level required for the support must be set as must any need for business lead users, database administrators etc. Further to this, developers will need to complete the Agency document: “Service Support Requirements Brief” for projects that are to be handed over to the Agency. It is preferred this takes place as early as possible so that CIS can plan adequately ahead and spot any potential problems. Once the software goes into production, the document contents will generally be translated into a ‘Service Level Agreement’.

Specific supporting documents may need to be provided (such as Entity Relationship diagrams) as agreed with the Agency Project Manager.

Code storage and version control of the software is the responsibility of the contractor. However, where the intellectual property of the software belongs to

the Agency then the source code of distributed production versions should be submitted to the Agency library.

Who will support the application?		
Response		Rationale
Agency CIS	[]	
Contractor	[]	
Other	[]	
Not yet known	[]	

4.3 Storage Requirements

As part of planning for implementation on Agency machines, physical storage requirements must be evaluated and communicated to CIS. The principal areas to consider are:

- Frequency of backup – how often the program data needs to be backed up
- Recovery time – the speed of turnaround required if the program/data needs to be restored from backup
- Amount of storage required – how much storage the program and data will require now and how much will it grow in the future.

Storage requirements	
Frequency of backup required	
Recovery time	
Amount of storage required (now)	
Amount of storage required (future)	

Appendix A Agency Reference Documents

Please note this is not an exhaustive list of applicable documents. CIS (your strategy analyst or project architect) will be able to provide you with the most up to date versions of the items below and/or any other relevant documentation. The Agency Enterprise Architecture is constantly evolving and it is important that you obtain the most recent version of any documents.

Enterprise Architecture: Technical Reference Model (Version 2.0 DRAFT3 - 6/12/05) – Peter Wintle, Enterprise Architecture

Enterprise Architecture Approach for the Environment Agency (PowerPoint presentation – 1st February 2006) – Ash Dattani

Environment Agency Testing Method (Version 1.0 Final – 15/06/2005)

Technical Delivery Process Document (Version 1.3 - 23/08/2005)

Overarching IS Principles (Version 8 – 5/5/2005)

Service and Support Requirements Brief (SSRB v1.5)

Appendix B Application Types

The Agency recognises that various types of software will be developed under the FCERM R&D programme. The type of software project being developed will have an effect on the application of the CIS “Enterprise Architecture: Technical Reference Model” standards and how rigorously they need to be followed. The software types can be separated into the following categories, though there may be some overlap and some applications may change category over time.

1. Basic Research & Development projects where the focus is on development of new methods that are “far from market” and the project specification does not require that the application is put on Agency systems. This will often be a “proof of concept” piece of software, which is likely to require much further work before becoming production software. As such, there may be scope for the “Enterprise Architecture: Technical Reference Model” to be applied less stringently. Obviously there are still good practices to be followed, which if followed will ease future transition to a fully-developed state.
2. Software developed primarily for non-Agency users, where the specification states it is not required on standard Agency systems. This type of software needs to take account of the varying systems in place at the operating authorities (and their consultants) when referencing the “Enterprise Architecture: Technical Reference Model” standards. For instance it is not practical to expect all consultants to install, run and maintain the standard Agency corporate database.
3. Software developed both for other operating authorities/external consultant use and for use on standard Agency systems. This type of software needs to balance the requirements of the Agency with the practicalities of external consultants and other operating authority users being able to install and use the software.
4. Software developed solely to run on standard Agency systems. Any deviations from the “Enterprise Architecture: Technical Reference Model” will need strong justification.
5. Public facing systems (i.e. systems exposed to external use by the general public) have their own special requirements and costs. They are beyond the scope of this project and are not covered further.

Appendix C Deployment

Note: all references to particular software and current standards in this appendix can be verified by referring to the current “Enterprise Architecture: Technical Reference Model” which can be considered the most up to date source for this information and will contain further information on deployment..

In general, all workstation executable application code will be stored on file servers. The only code to reside on desktop workstations will be basic operating system components and some shared supporting code. Suppliers of proposed systems must acknowledge, and comply with, this approach.

Most sites with 100+ users have at least one file server per site. However with the lifting of the previously imposed 100GB server limit, this trend is being reversed. The server and storage architecture is under review.

Whilst licence strings and similar techniques are acceptable as a means of enabling the identity and source of a specific software installation, suppliers must not use any techniques to restrict access to applications.

Appendix D Guidance Notes

Below are some further notes on common issues that have arisen from use and review of the guidance document.

D.1 Proprietary formats

Where the guidance discusses the use of proprietary data formats, it is principally referring to those created by the developer specifically for the program being written. It is not suggesting that you should not use the de facto “standard” file formats that the Agency itself uses for things such as GIS systems.

D.2 .NET Policy

At the time of writing, the Agency policy regarding applications written for the .NET framework is to allow “commercial off the shelf” packages to be installed, but not bespoke development to be undertaken. With appropriate justification the use of .NET development tools may be acceptable. Where development is agreed to be undertaken in .NET, the Agency would prefer development to be undertaken using version 2 of the .NET framework.

D.3 Adherence to the “Enterprise Architecture: Technical Reference Model” document

Whilst complete adherence to the “Enterprise Architecture: Technical Reference Model” document is desirable, it is acknowledged that this may not be the appropriate course of action for a particular project. Completion of the guidance to explain the reasons behind an alternative path of development will help CIS gain an insight to the overall development strategy and help smooth the review/approval process.

D.4 Communicating “network bandwidth usage” and “server processor usage”

It is noted that communicating network bandwidth usage and “server processor usage” is difficult to do and depends on many factors, such as number of concurrent users, functionality used, dataset size etc. It is often particularly hard to specify exact figures before the development is complete and even then is hard to measure. However any information that can be provided to CIS in this regard is highly valuable.

D.5 Gradual uptake of R&D software outputs

Many projects begin with a very minor software output that can evolve into a large scale complex system that is eventually implemented at the Agency. In these cases the guidance should be completed to address the currently planned software outputs, noting possible future expansion. This process should help the developer gain an insight into a suitable long term strategy as well as helping CIS spot any possible future problems.

D.6 Alternative thin client strategy (using Citrix)

Where the Agency standard “browser based, thin client” approach is not appropriate for the development it is worth considering using CITRIX for Agency sites. This may be a practical alternative and should be discussed with CIS. However, it should be noted that this may have significant implications for hardware and network sizing.

PB 12794 B

**Ergon House
Horseferry Road
London SW1P 2AL**

www.defra.gov.uk

