1. **Introduction**

This brief paper presents the minimum standard to be addressed by nationally funded software development projects or local projects with potential for, or aspiration of, national use. This paper defines:

- A minimum standard for all new developments of software for use in NHS Scotland.
- A platform to share best practice across NHS Scotland.
- A vehicle to help reduce duplication of effort and 'reinventing wheels' between separate developments.

It is important to appreciate that this standard is a minimum and should not constrain projects that aspire to more formal standards such asTick IT[^1] rather it should be seen as step on the path towards such formal standards.

The paper has been reviewed by representatives of a number of national and local development projects, has been distributed widely in the NHS Scotland IM&T community and tested with a number of projects.

The standard is approved by the eHealth Programme, Infrastructure Steering Group, and all new national and strategic developments are required to meet, or better, the standard. In addition the standard is available for local developments and it is hoped that the standards will be increasing the norm for all developments in NHS Scotland.

**Why is it needed?**

There are a number of problems that application of a minimum standard would help alleviate:

1. Individual clinician enthusiasts have often driven the development of software for use in the health service. While this has proved very effective for generating and proving clinical IM&T innovation it has made adoption by other clinicians problematic. Sharing practical experience on how more general clinician 'buy-in' has been achieved would help.

   A successful example is the SCI Diabetes Collaboration. A clinically owned and led nationally funded project to develop a hospital diabetes clinic system (SCI DC Clinical) and a health board area wide diabetes shared clinical record to underpin diabetes managed clinical networking (SCI DC Network) for national use. SCI DC has taken two successful local systems (Lanarkshire Diabetes and DARTS) and created systems with national presence, acceptance and more importantly use.

[^1]: [www.tickit.org](http://www.tickit.org)
2. Developments are often focused on short-term rather than long-term deliverables. That is, documentation, support and maintenance are often given a low priority leading to high cost of ownership and problematic re-use elsewhere.

Adherence to appropriate minimum standards could minimise this. While this is an upfront cost there is considerable downstream saving from this approach. It may still be appropriate for local projects not to use this approach but this would be conscious local decision.

An example of the problem is SCI DC. Even though the two initial products had been operating locally for some time, significant rework, new process and extra documentation were required before they could be viably implemented and supported in another areas. Implementation was delayed by many months. After the extra effort the products are now capable of supporting the clinical aspirations for the products and are being rolled out.

3. With each new development the project manager, or developer, decides what approach, process and tools will be used often from a blank sheet of paper. If tested this is usually after the event as criticism. Much could be done at the outset to support the project to benefit from experience elsewhere.

For example, an NHS Lothian team developing a web based cancer system asked what level of documentation was needed and what if anything could be used as a start point. Unfortunately there was no stock answer and that team have had to extend effort starting from a blank sheet of paper. This in part has been a driver for producing this document.

4. Developments are often done in isolation from the wider IM&T community leading to duplication of effort. This could be avoided by providing a mechanism to ensure national infrastructure and standard ‘givens’ are known.

An example of this is the development of multiple ‘point-to-point’ lab system interfaces for a product when the necessary interfaces were already available using the SCI Store as a hub.

5. Another symptom of the isolation of the typical development silos is the lack of professional support between projects. Any standard by itself won’t break down these silos, but the discipline of external inspection to check adherence to standards should ensure that project teams are actively encouraged to share of best practice and support individual development.

A recent example of this is the newly established cancer systems developer forum, created to provide a vehicle for service and commercial cancer clinical system development projects to exchange ideas and experience and share effort in resolving issues.

6. Developments, particularly those that are user led, often focus on functionality at the expense of non-functional requirements. Thus key factors such as scalability, security and performance are often not designed and built in.

A recent example of this is the Scottish Birth Record system produced by ISD. While an exemplary prototype with considerable user buy in, the technical ‘underpinnings’ require attention to support long term usage across Scotland.
In summary it is hoped that the publication and adherence to these standards will ensure that NHS Scotland gets the most ‘bang for its buck’ by:

- Saving money - through avoiding waste, reinventing wheels and reducing the overall cost of ownership.
- Improving quality - by sharing best practice.
- Reducing time - by avoiding rework and barriers to uptake.
- Maximising national clinical benefit - by minimising ‘not invented here’.

**How will the standards be used?**

All new nationally funded developments would need to demonstrate how all the elements of the standards would be met and maintained during the development, and any divergence justified. The standards are intended to be supportive rather than directive and any issues would be worked through with the SEHD IM&T Technical Lead\(^2\), for example divergence agreed, proposal ‘beefed-up’ or standards updated.

Continued conformance with the standards would be monitored during the development. This could take many forms (from informal peer review through to formal membership of project boards, etc) and would be mutually agreed. Important here is that the ongoing monitoring would be two way, to support the development but also to improve the standards to share best practice.

Existing applications being considered for national adoption would have to demonstrate (retrospectively) how the various elements of the standard were met. This would be reviewed and appropriate actions (e.g. rework) agreed.

No standard set of documentation is proposed, this is deliberate to avoid a ‘tick box’ approach that could lead to documents produced for the sake of the document standards rather than the sake of the project. However, to encourage the sharing of best practice it is intended to establish a forum for the sharing of examples of documentation\(^3\). Thus a start up project would be able to browse a number of PIDs (and other documentation) produced by other NHS Scotland Projects and select appropriate comparable ones to help production of their own PID. This would help projects decide if a one or a hundred pager best fits their needs.

A number of projects have requested a standard template to aid production of responses. This has been produced and is available\(^4\) where this would help, but its use is not required, but the information it contains is.

**Fit for purpose?**

Important here is meeting the ‘spirit rather than the letter’ of the standards. This is not intended to be a bureaucratic process that is paper heavy, rather just enough information, just in time, for what’s needed.

\(^2\) If this approach takes off it will become a significant task requiring additional resource. Early thoughts of how this could be provided include; some form of external consultancy (a la RFA accreditation), establishment of an NHS Scotland Technical Assurance Group drawn from the NHS Scotland IM&T community, or some other from of other peer review.

\(^3\) Most likely as part of a technical architecture site on SHOW

\(^4\) Response template to demonstrate compliance with minimum standards for the development of NHS Scotland software, version 1.0, 17/06/04
Key is that developments are ‘fit for purpose’. Thus the rigour needed to for a ‘stand-alone’ ‘throw-away’ local research package will differ to the rigour needed for an operational clinical network system feeding national audit analysis. However, all developments should be able to demonstrate that all the elements have been considered and addressed accordingly.

Thus the simple ‘stand-alone’ project, mentioned above, would need to consider the acceptability to users, but this could be demonstrated by a paragraph defining the local user stakeholders and why this is all that is required. The bigger clinical network system would require much more extensive process to ensure that the needs of all clinicians in the network have been captured. This is likely to need comprehensive documentation to define the arrangements.

There will be no hard and fast rules for the level of rigour needed to demonstrate all the elements have been addressed. This will need to be considered and addressed on a ‘case-by-case’ basis, as we build experience in this new approach.

2. **Acceptability to users**

**The problem**

“Developing software is very easy - developing software that will be used by clinicians is near impossible.”

The reasons for this are complex, but put simply reflect the difficulty in getting adequate levels of informed clinical input and ownership in software developments. This reflects lack of time on the side of the clinicians, but also a lack of recognition of the issue by developers. It is telling that many of the most successful software products in NHS Scotland have originated from software developed part time by enthusiast clinicians. An example of this is GPASS the most used clinical system in NHS Scotland.

Development of software by, or led by, enthusiasts is not without its problems. Significant here is the difficulty of getting the system used by clinicians elsewhere where clinical working practice is different. Often this can only be addressed by building in local configuration. This is typically identified after the event leading to re-work and delayed implementation.

In a nutshell the problem is: IT specialists who develop software have problems developing systems that fit clinicians’ needs; enthusiast clinicians who develop software tend to develop for their own personal use, which does not translate well to the needs of others.

**The solution**

Development must include an appropriate level of user involvement and ownership. All successful IT developments, are sponsored by people at the top of the business, senior stakeholders who will get rewarded if the software goes in successfully but conversely who will have to answer for it if things go wrong. For successful development in NHS Scotland clinicians and other stakeholders will need to fill this role.

The key driver is to ensure that users own requirement definition, help manage change requirements and own acceptance of the system. Elements to be considered and addressed include:
Identification of all potential user 'stakeholders' - local and national interested parties and if appropriate formal representation of clinical bodies (e.g. SGMC, Royal Colleges, SCIMP, SIGN, etc.)

Involvement of users in the management of the project. Specific examples should include, managing risks/issu's and business case production.

Process for user involvement in requirements gathering, review agreement and tracking/controlling change during a development.

Involvement of users in the development using an appropriate methodology (e.g. JAD meetings, business modelling, document review, prototyping, etc.)

Process for stakeholder acceptance.

Communications strategy to inform, and involve, the wider user/stakeholder community with the development. To ensure realistic expectations can be set and maintained.

Provision of user groups.

How demonstrated

Most of these elements are basic common sense, and no specific documentation is likely to be required, though some methodologies (e.g. DSDM\(^5\), PRINCE2\(^6\), etc.) could provide this, for example to help particularly large or demanding developments.

A single pager, with pointers to other standard project documentation, could cover that each element has been considered and addressed. Likely project documentation could include; PID, testing strategy, communications strategy, user group ToR, etc.

3. **Project Management**

The problem

"Software developments never deliver what was wanted when it was wanted."

Plans start wrong - This partly reflects the difficulty in estimating for jobs never attempted before, but also that the only constant is change. For example, changed user requirements and expectations. Most service developments fail to appreciate this.

Management of change is not allowed for. Plans do not include processes to catch and suitably handle change as early as possible.

The solution

There are two elements to the solution. The first is to start with appropriate initial plans. Key here is the 80:20 rule of optimum return for optimum effort. For example, sensible consideration and presentation of; roles/responsibilities, products, constraints, risks, assumptions, dependences and critical path are of more use than detailed Gantt charts.

The second is to ensure that change is captured and addressed so that the plan can evolve to meet the needs of the stakeholders. That is, the development must be appropriately managed.

---

\(^5\) [www.dsdm.org](http://www.dsdm.org)

\(^6\) [www.ogc.gov.uk/prince](http://www.ogc.gov.uk/prince)
While the management standard for all projects in the NHSScotland is PRINCE2, full implementation of the methodology would be overkill and wasteful for all but the very largest development projects. However, PRINCE2 provides a very useful and well-crafted bag of tools. As a minimum developments must be able to demonstrate an adequate level of:

- Planning including justification for the resource and other estimates.
- Consideration of the full life of the product. I.e. not just the development, see section 4.
- Consideration of the business change aspects of the project. No project is just technical and the softer or people factors must be addressed.
- Project management including an appropriate project board, project managers and project support.
- Project assurance including the quality plan and stakeholder review process.
- Risk and Issue identification management and resolution.
- Project tracking including checkpoint and other milestone meetings; highlight reports, exception reporting, the monitoring of effort against targets and monitoring delivery against critical success factors.

**How demonstrated**

The appropriate level of documentation, needed to demonstrate that these elements have been properly addressed, will vary with the size of the project. However, as a minimum it should consist of a project initiation document defining how the elements are addressed, a 'living' project plan, a similarly 'living' risk/issue register(s), highlight reports, and records of relevant project management meetings.

4. **Cost of Ownership**

**The problem**

“The excitement of development risks producing systems which are difficult or expensive to implement and run.”

Software is often developed for the convenience of the developers and without cognisance of the full lifecycle of the product. This makes implementation of the software difficult and costly and imposes unnecessary overheads on its operation.

Examples include: inappropriate dependencies on software products/versions, recurring licence costs, incompatibility with existing IT platforms, lack of automated installations, etc. Thus, it is easy and cheap to develop quickly using the latest tools. However, it becomes prohibitively expensive to implement this software when installation requires upgrades to the latest version of Windows (etc.), installation of a new database, extensive re configuration of the PC, installation of required components resulting in additional licence cost and slow complicated on-site installation.

**The solution**

The total cost of ownership across the whole life of a product must be considered at the outset when the technical architecture of the project is being defined.

The individual project must consider and balance the benefits and overall cost of each component. The overall costs must take account of the potential uptake of the product. It must be appreciated that overall costs of ownership will provide a significant constraint on
architecture and style of development. This is unlikely to fit comfortably with the enthusiasm of developers to use the latest innovation. Project management must be alive to this issue and ensure that is highlighted, considered and appropriate compromises reached. External liaison and review with the National Technical Lead and other national projects is likely to help here.

**How demonstrated**

Documentation will be required to demonstrate that overall cost of ownership has been considered as part of selection of the development and technical environments. This should include an analysis of the benefits and costs of each element. The latter should be based on realistic projections of the scale of uptake and life expectancy of the product.

In addition there should be a process to ensure that technical decisions that might impact on the overall cost of ownership are escalated as appropriate. That is, not left to individual developers or small teams operating in isolation.

The appropriate level of documentation will, again, reflect the size of the project. As a minimum the PID should include an analysis of the technical architecture. At the other extreme a formal analysis of options including business cases may be required.

For large projects a suitable named technical architect may be appropriate.

### 5. Software production

**The problem**

“*Software development is not as professional or as efficient as it could be.*”

As stated earlier “developing software is very easy, developing software that will be used by clinicians is near impossible”. This in part reflects the approach to developing software in the service, typical of this approach is:

- Use of relatively inexperienced staff - through lack of funding and training.
- Use of contract staff – through restriction to capital funding.
- Isolated teams leading to lack of shared experience and repeated mistakes.
- Focus on development rather than the full lifecycle of the product.
- Poor requirements capture, analysis and design.
- Lack of, or inappropriate use of, mainstream software development methods, tools and languages.

The upshot of this is waste of effort. On the other hand there are significant benefits to the approach not least the energy and enthusiasm delivering real clinical IM&T innovation. The trick is to address the former while retaining the latter.

**The solution**

Application of all the other solutions outlined in this paper would go a long way to addressing the problem, but there are some specific additional elements relevant to the production of software. These are:
• The selection of the most appropriate development approach / life cycle, for example traditional waterfall or one of the rapid application development techniques\(^7\).
• Section on the most appropriate technical architecture and development tools (choice of operating system, database, etc.)
• Definition and application of appropriate techniques for requirements gathering, business analysis, design and coding
• Definition and application of appropriate local production processes for example covering; coding standards, style guides, data definitions, etc.
• To build upon success and avoid repeating mistakes, through investigation of what has already been done and liaison with colleagues in other development teams.

NHS Scotland would maintain a list of preferred development; methods, tools and languages. An initial list would form part of the forthcoming ‘technical architecture of IM&T in NHS Scotland’\(^8\).

The preferred list would be the first choice for new developments except where divergence could be justified. This would allow experience in the service to be built up and maintained rather than diluted as new developments become constrained to uncommon tools (e.g. the latest flavour of 4GL, or object database) from limited (often one commercial) sources.

**How demonstrated**

There is no simple solution here as it is not the intention to stifle or constrain the most creative and personal elements of software development. However, projects should be able to demonstrate that each aspect described above has been considered and a suitably ‘fit for purpose’ solution provided.

For the development lifecycle, technical architecture and development tools this would consist of a definition of the approach taken together with the rationale for its selection. I.e. the business case for rejecting any ‘first choice’ tools, languages, etc. This could range from a one pager to a comprehensive feasibility/options appraisal for each aspect. For the ‘nitty-gritty’ of the development (the specific processes for requirements gathering, business analysis, design, coding, etc.) more detailed documentation would be needed, but since this is needed for any development this should not be onerous. Thus a single pager, with pointers to the project working documents would more than suffice.

It would be essential to demonstrate widespread consultation to ensure previous positive and negative lessons learnt by others were being built on. The newly established NHS Scotland Developers Forum\(^11\) and the IM&T CIG meetings would be a useful vehicle here.

### 6. **Testing**

**The problem**

“*Software does not work; it does not meet the users requirements and is full of bugs.*”

\(^7\) It is important to recognise that no single mainstream development approach has been found to meet the needs of the health service. Thus there is a need to consider objectively what process or combination of processes is likely to be most effective for each project.

\(^8\) A separate paper is being produced to define the NHS Scotland technical architecture. This is likely to define strategic and preferred tools, where the former will be mandated and the latter actively encouraged.
There are two aspects of the problem. First, software is often released with significant numbers of hitherto unknown major issues (both faults and new features). This has lead to delays, increased cost and loss of confidence by users. The reason for this is multifactorial, not (as often perceived) just a lack of professionalism of the developers. Factors include:

- Lack of understanding of the requirements – if not known how can they be tested?
- Lack of understanding how the software will be used – for example, which are the most used, most important, parts of a system?
- Testing to prove what works rather than what does not.
- Testing tacked on to the end of developments often with the approach of ‘beta testing will catch everything’.

Second, managers and users often expect that software will be fault free. Such unrealistic expectations lead to frustration, guilt and anger when the inevitable occurs and the production software fails.

The upshot of both is software that is, or is perceived to be, not fit for purpose – often articulated by clinicians as ‘clinical risk’.

The solution

All products and supporting components should be tested to an appropriate level and this level agreed with the clinical and other stakeholders.

The level that is appropriate will vary and some applications will require specific formal certification.\(^9\) For most others, testing should be about risk reduction and making things as good as possible given the resources available. As often the case in IT in the NHS while appearing technical the problem and the solution is really about ‘people issues’. As such is difficult to address. However the following is proposed:

1. At the outset purpose of testing should be defined and agreed by stakeholders. It is suggested that the purpose of testing should be:

   - To check conformance to requirements\(^{10}\) – for this the requirements must be ‘testable’. So for each requirement, the developer and stakeholder must state how it will be possible to verify that that requirement has been met.
   - To minimise rework, by identifying errors soon after they occur to allow quick corrective action to reduce the chances of adversely affecting later development work.
   - To demonstrate acceptability and usability of the system – the system is reliable and without usability issues.

2. The expected outcome of testing should be defined and agreed. That is, what level/severity of faults would be acceptable in which parts of the system? What is acceptable will differ greatly for a robotic drug dispensing system and a stand-alone audit system.

---

\(^9\) E.g. BS7925 or the CE marking regime
\(^{10}\) It should be remembered that the requirements are a living part of a project. Thus conformance should take account of changes to the requirements during life of the project.
3. Once an appropriate purpose and outcome to testing have been identified, the strategy for testing should be defined and agreed. The test strategy would cover:

- How requirements would be tested.
- The testing environment, including any datasets and tools to be used.
- Where testing fits in the development lifecycle. This is likely to include most, if not all: of unit, integration, system, user acceptance, performance, stress, volume, and security testing.
- Regression testing – to check that things still work. That is, to ensure that change has not introduce an issue elsewhere.
- The scope of testing, what will be tested – it is impractical that the whole system can be tested to the same level. Thus where will effort be concentrated? For example; the most used elements, the biggest risk elements, those elements with the most change, etc.
- How testing will be done and who will do it.
- How issues will be; caught, monitored, prioritised and addressed – including the issue tracking tool and process for involving stakeholders.
- How clinical and other users will be bought into the testing process across the development lifecycle. Important here is to ensure that the local Caldicot Guardian and DPA Data Controller are involved to ensure appropriate arrangements are included to protect patient confidentiality and sensitive material during testing.

**How demonstrated**

Developments must include appropriate testing to ensure that each element is addressed. The detail content of the documentation needed to demonstrate this would vary with the size of the project and the lifecycle. However, as a minimum should address each of the following:

- The test strategy, detailing the purpose of testing and the expected outcome as well as the testing environment, processes and methods to be used.
- A living definition of the requirements to be addressed and criteria (agreed with stakeholders) by which delivery will be assessed.
- Test plan – ‘what will be done when’. This may form part of the overall project plan
- Test cases – the IEEE definition of test case is “A set of inputs, execution preconditions, and expected outcomes developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.”
- Test scripts – manual or automated scripts for the execution of the test cases. These should include any data or underlying conditions required.
- Test results report – a report on the outcome of testing.
- Non-functional testing to ensure appropriate performance, scalability, security, etc.
- A log of all issues and fault raised and the action taken. This may as sophisticated as a system such as Test Track Pro or as simple as a spreadsheet. To avoid reinventing wheels prior to developing a bespoke system or adopting a commercial product advice should be sought, for example through the NHS Scotland Developer Forum\textsuperscript{11}.

\textsuperscript{11} [http://ikonboard.nhsscotland.com/](http://ikonboard.nhsscotland.com/)
7. 'Collateral' Documentation

The problem

“There is never enough useful documentation to support the system and its users once the developers have done.”

Software development projects typically only deliver software, often with little or no, meaningful, helpful, supporting documentation – ‘collateral’. This makes: implementation, use, support, maintenance, and ongoing development of the system more difficult and expensive than it need be.

Developers like writing software and do it in preference to anything else, often with little thought given to the unfortunates that are to use the software or maintain it in the future. Project managers reinforce this by not giving sufficient prominence to ‘collateral’ as project deliverables.

The solution

To ensure that at the outset software development projects identify all document products to be delivered and build in sufficient resource to ensure their delivery.

The type and level of documentation will reflect the scope of the development but as a minimum all the following headings should be considered and addressed:

- User documentation - including on-line help, user manuals, product overviews, etc.
- Training materials.
- Release letters for each release.
- Installation/implementation materials – including pre-requisites, installation scripts, etc.
- Technical documentation for any external interfaces, outputs, and inputs.
- Technical documentation to facilitate support of the system – including data models, system configuration, FAQ, etc.
- Technical documentation to facilitate maintenance and further development of the system – including data model/dictionary, designs, commented code, test strategy, etc.

How demonstrated

The development should be able to demonstrate that the need for collateral has been investigated and sufficiently ‘fit for purpose’ documentation is a project deliverable. The form and level of the documentation to be produced will vary with the scale of the project but each of the above elements should be addressed to appropriate quality and depth.

8. Configuration Management

The problem

“There is no control or audit of changes made to software, making support and maintenance both difficult and expensive.”

Configuration management (CM) here refers to the process for managing versions of the project’s products, to make sure that the content of a particular product is known and that changes to the product can only be made in a controlled manner.
There are several problems associated with inadequate CM. Examples include:

- Content of products (database, software documentation, etc.) being lost through subsequent change - resulting in wasted effort. This is a particular problem for teams working on a product.
- Local variation (e.g. system configuration, software patches, local data files, etc.) being undocumented – resulting in problematic (and unduly expensive) support, maintenance and upgrade.
- Difficulty (some times impossibility) of replicating built versions of software – resulting in extra cost and problematic business continuity.
- Product version free for all (e.g. different developers working to different versions of designs or requirements, or uses with out of step documentation.)
- Difficulty in parallel development of an operational system. For example providing patches and other incremental updates while at the same time developing the next major version of a product. Without appropriate CM, at worst such development is impossible, at best effort is wasted through duplicating changes to both versions.

A significant issue is that the benefits of appropriate configuration systems are achieved in the medium term - the support, maintenance and re development stage of a project, while all the costs fall at the short term. CM needs to be built in at the start of a development.

**The solution**

All software, and ideally other products and components (e.g. testing products, documentation etc.) should be under formal CM control. The scope and sophistication of the CM product will reflect the number of developers, the development/release/support process and the complexity of the project but it is very likely that some form of CM package would be needed.

**How demonstrated**

Projects should define a CM policy covering the approach, the scope, tool to be used, and working practice/process. There should be a nominated Configuration Manager responsible for the policy and ensuring that it is implemented. Except for large (double figures of developers) projects the role is likely to be part time.

There are numerous CM products, including freeware, products bundled with operating systems/databases and sophisticated dedicated products. A product frequently used by large national projects is PVCS. Prior to adopting a CM product advice should be sought, for example through the NHS Scotland Developer Forum

**9. Conformance with National Standards and Cornerstones**

**The problem**

“We complain of not having enough resource, then we waste it reinventing wheels.”

NHS Scotland provides a range of national products and infrastructure. Recently SEHD have announced that some of these form fundamental foundations, or cornerstones, of eHealth and Integrated Care Record in Scotland. In addition there are a number of NHS Scotland and NHS standards, for example for patient identification, clinical communication, clinical audit, treatment statistics, etc.
Where these cornerstones and standards are not used but instead duplicated by local developments there are serious problems of wasted effort (for development, support and hosting) and incompatibility (preventing exchange of information between local systems, national and other local systems).

Unfortunately the benefits of reuse of national standards and cornerstones are not immediately apparent. Thus, it is seductive for a project to do its own local thing rather than reuse a national thing. For example, it is quicker and easier for a local system to develop its own patient register, staff directory, and referral content rather than inter work with the local patient demographic master (SCI Store), National Staff directory (Health net) or use the NHS Scotland referral XML standard. However, without using these national standards the ability for the local system to contribute to the integrated care record for the patient is severely restricted, if possible at all.

In summary the problem is waste of the already limited finance and resource for IM&T in NHS Scotland and delay in attaining the vision for IM&T to support the care of patients.

**The solution**

The solution appears obvious; developments must reuse and not duplicate national cornerstones and standards. However, to date there has been little clear steer on the existence and use of these standards. This has now been recognised with development for publication of a technical architecture of IM&T in NHS Scotland\(^8\). New developments should ensure they fit with the mandated elements of the architecture. These include:

- National infrastructure including the managed national network/telecommunication service; CHI as unique patient index, national directory for authentication/access control across NHS Scotland, SCI Store as consistent repository across all NHS Boards and SCI Gateway as the remote transactions mechanism.
- Components of national programmes (e.g. GPASS, SCI DC, ETP, SBR, etc.)
- National standards (e.g. ISD definitions and terms, ISD XML message/data standards, ISD coding/terming standards, appropriate national clinical minimum data sets, etc.) These national standards are developed in collaboration with the service then supported and maintained by the Data Intelligence Group, ISD. In due course many of these national standards will be published in the Data Dictionary.

Divergence from standards, which may be appropriate for local pragmatic reasons, would require discussion and agreement.

**How demonstrated**

Documentation required to support this would include: description of the fit with national; infrastructure, programmes and standards; consideration of how best to address any gaps; justification for any divergence from standards and agreement by the relevant stakeholders that this is acceptable. This is likely to form part of the PID or any initial feasibility work and checked by ongoing review of the technical architecture and requirements.