Chapter 5
Technical Standardisation

If you want to build a ship, don′t drum up the men to gather wood, divide the work and give orders. Instead, teach them to yearn for the vast and endless sea.

Antoine de Saint-Exupéry

Technical standardisation is what sets out the enterprise-specific standards which frame the development trajectory for your IT landscape. Even if the landscape you have at present is a heterogeneous patchwork, you can gradually usher it toward your vision by defining which technical standards are to apply in projects and maintenance measures (see Fig. 5.1).

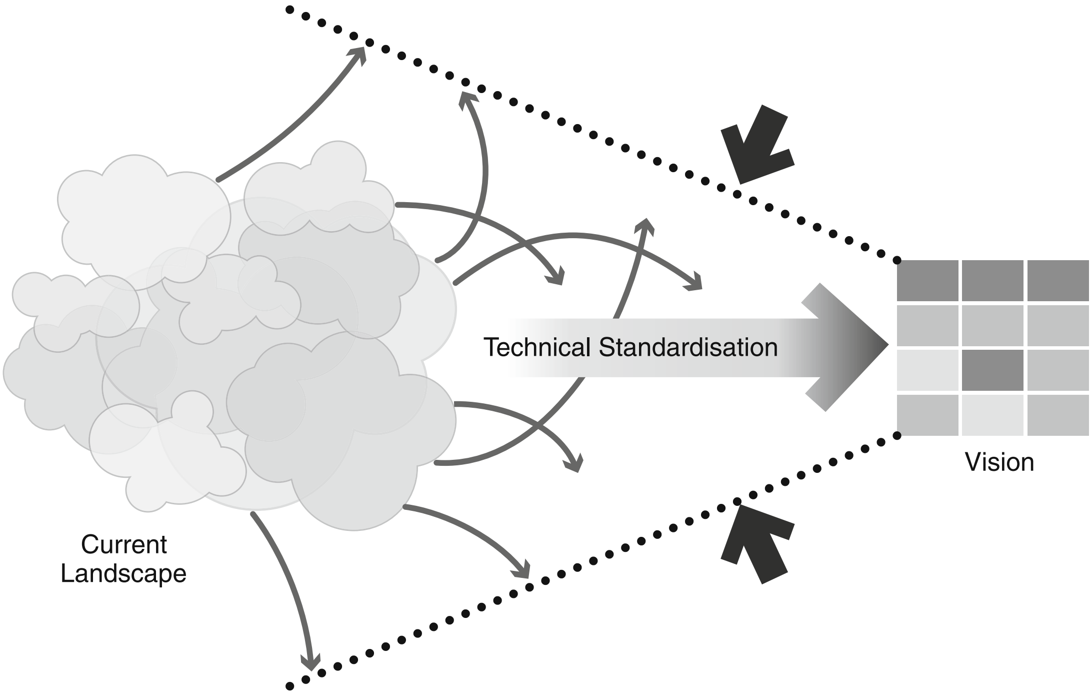

Fig. 5.1 Technical standardisation

I. Hanschke, *Strategic IT Management*, DOI 10.1007/978-3-642-05034-3_5,

Questions answered in this chapter:

- What contribution does technical standardisation make to strategic IT planning?
- Which objectives are being pursued?
- What are the components of a standardisation catalogue?
- Which processes are required, and how can technical standardisation be embedded in the organisation?
- Which technical standards are appropriate for you?
- In which incremental steps can you introduce technical standardisation?

5.1 Scope and Definition

What Contribution Does Technical Standardisation Make to Strategic IT Planning?

Like IT landscape management (see Chapter 4), technical standardisation is fundamental to strategic IT planning. In this process, you take the objectives defined in the IT strategy, and out of them derive a combination of technical standards which are then shaped and fine-tuned to provide the best possible backing for your current and future business requirements.

By staking out the "permissible" technologies, databases, middleware solutions and reference architectures, technical standards are a key input for IT landscape management (see Fig. 5.2). They are the imperatives governing implementations of the applications and interfaces and – particularly importantly – the design of future application landscapes.

If technical standardisation shows what things should be like, IT landscape management provides information on how things are, showing which technical building blocks really form part of applications, interfaces or infrastructures. This information on the use of building blocks is valuable input for technical standardisation.

Technical standards can also be defined for productive operation as well as for infrastructure, with many enterprises electing to scope out service strategies and service designs (see ITIL V3 [Buc07]) that include specific standards on which hardware, operating systems and network components to use. For more information, please refer to relevant literature, including [Buc07], [Joh07] and [itS08].

Fact file:

- Technical standardisation prescribes enterprise-specific technical standards for projects and productive operation.

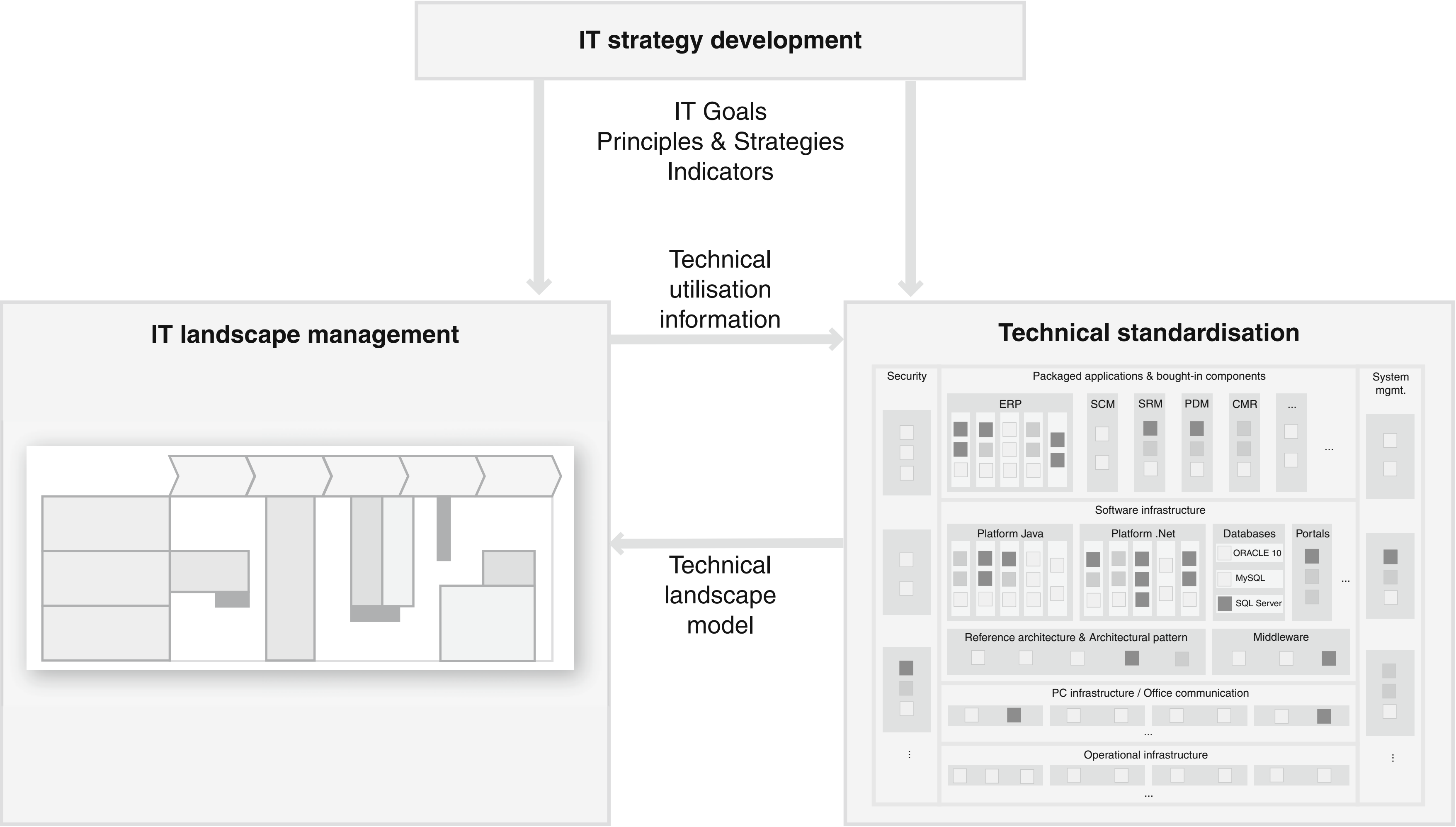
IT strategy development
IT Goals
Principles & Strategies
Indicators
Technical utilisation information
IT landscape management
Technical standardisation
Technical landscape model
Security
Packaged applications & bought-in components
System mgmt.
ERP
SCM
SRM
PDM
CMR
...
Software infrastructure
Platform Java
Platform .Net
Databases
ORACLE 10
MySQL
SQL Server
Portals
Reference architecture & Architectural pattern
Middleware
PC infrastructure / Office communication
Operational infrastructure

Fig. 5.2 How technical standardisation fits into strategic IT management

- IT landscape management provides a clear view on how and where technical standards have been applied in the IT landscape.
- Technical standardisation is operationalised by means of projects and maintenance measures.

5.2 Objectives of Technical Standardisation

The objectives an enterprise has in mind with technical standardisation will differ depending on the strategic positioning of IT in the organisation as a whole (see Sect. 2.3):

- **Cost savings** through economies of scale, with the opportunity of greater negotiating power when purchasing is centralised, and of more focused expertise as an outcome of technical standardisation and homogenising.
- **High technical quality** of applications and the operating environment, an outcome of reusing tried-and-tested technical components such as reference architectures, frameworks or third-party software within the enterprise.

  The technical quality depends both on where the system is in its lifecycle and on the degree to which quality requirements have been enacted concerning performance, security etc. The technical quality is important for creating sustainable systems.
- **Appropriate IT support** for business requirements, such as the agility of IT systems to adapt to evolving business requirements and external constraints or to fulfil security or compliance requirements.
- **Sustainability** of the IT landscape through explicit IT innovation management and strategic evolution of the technical standards; see also Sect. 5.4.

Fact file:

- The huge cost savings potential is in general a key incentive for embarking on technical standardisation.
- Having the right technical standards is vital to making the IT landscape robust enough and fit to underpin the enterprise’s business operations into the future.

Important:

The only route to accomplishing the named goals is by planning and controlling technical standardisation – and by planning and controlling how technical standards are utilised in the IT landscape.

See also Sect. 5.4.

5.3 Elements of a Standardisation Catalogue

The bare skeleton of the standardisation catalogue has to be fleshed out with details specific to your enterprise. Ultimately, the catalogue is nothing other than the technical landscape model (see Fig. 5.3), also termed "blueprint" in the following. The technical landscape model comprises two levels:

- Technical reference model: the modelling equivalent of a cabinet with drawers;
- Technical components – the elements which fill the cabinet and its drawers.

The "drawers" in the model are the architectural domains. These domains can be organised into a hierarchy, i.e. each drawer can be subdivided into further trays.

The technical components can in turn also consist of subcomponents (i.e. hierarchical relationships can exist within component structures). There can also be other associations connecting technical components. For example, a Java application server has a "uses" relationship with a JRE. Technical components can be versioned, making it possible to manage multiple release states of a Java application server (e.g. Version 1 and Version 1.1) in the same blueprint. These version states can be linked by a predecessor-successor relationship.

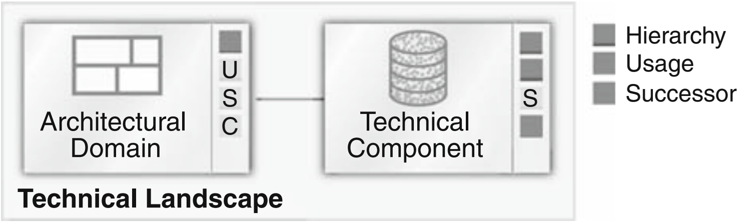

Fig. 5.3 Technical landscape

The blueprint can comprise the following categories of filler elements:

- **Technologies** (a kind of catch-all term here): Taken from the context of software engineering, third-party software or operation such as ".Net", "JEE", "SAP" or

"BS2000 host", these umbrella terms are a concise way of summarising both the technology target and the status quo.

- **Reference architectures and architectural patterns** (see also [Aie04], [Aie05], [Bas03], [Foe03], [Star08] and [Vog05]): Reference architectures and architectural patterns serve as solution templates either for a complete application category – such as templates for the rollout of standard software across multiple sites, or reference architectures for web-based JEE applications – or for a specific area, e.g. a data access tier.

It can also be useful to distinguish between reference architectures or architectural patterns which are tied to particular technologies and products, and those which are not. In practice, though, with the few reference architectures usually in use in a single company, this distinction tends to be unnecessary.

- **Third-party IT products**: These are software and hardware solutions which are sourced on the general market as "off-the-shelf" products with no enterprise-specific customising.

  Examples are:

  - Packaged business software such as SAP or Siebel, or bought-in components for specific functionality such as OCR
  - PC infrastructure and office communication products such as word processing, groupware and fax solutions, DMS or CMS
  - Runtime environments, including application or web servers such as Tomcat or JBoss
  - Databases such as ORACLE or SQL Server
  - Middleware such as MQSeries or CORBA
  - Security components such as firewalls or virus scanners
  - Hardware and network infrastructure such as servers and network components

- **Technical building blocks** that can be used to build applications: Technical building blockThese can be own-build frameworks to cover specific aspects of an application (e.g. security or logging) or third-party products such as workflow engines or rule systems.
- **Tools** for software development and system management

  - Software development-related tools such as development environments or test tools also belong in this category, as do version and configuration management tools, build and deployment tools
  - System management tools are tools for running applications, e.g. for system administration, monitoring and software distribution

The following sections describe architectural domains and technical components in greater detail.

5.3.1 Architectural Domains

A separate architectural domain is earmarked in the technical reference model (TRM) for each functional area (e.g. databases) requiring standardisation. With these domains representing the "drawers" in the cabinet-of-drawers model, you can locate the right sort of technical components for the context you are working in simply by looking in the appropriate drawer.

The following information is required to document an architectural domain:

- **Name**: The unique identification of the architectural domain, e.g. "software infrastructure"
- **Description**: Description of the content of the architectural domain

  Example: the architectural domain "software infrastructure" contains all the technical components for developing custom software.
- **Hierarchy**: Description of the hierarchical structure of the architectural domain; in other words, this defines which drawers and trays the architectural domain consists of, and in which drawer or tray the domain itself is located.

Many enterprises elect to structure architectural domains to reflect their own architecture conventions, using "tiers" and "layers" to organise a domain into suitable units. Tiers are the vertical subdivisions and structure an application in terms of logical functionality. Layers, on the other hand, are horizontal subdivisions. Each tier and layer has clearly demarcated responsibilities. Here are some examples of tiers:

- **The presentation tier** is responsible for composing and visualising data and other content.
- **The business tier** provides business functionality, e.g. as business services.
- **The data tier** delivers data to the application either directly from the database or indirectly from other interfaced applications.

In many cases, these tiers are complemented by integration tiers for front-end (e.g. via portals) and back-end integration (e.g. via EAI).

Figure 5.4 illustrates how an application might be organised into layers and tiers. Only two layers are used here: "functionality" and "technology", though in practice both would be subdivided into further layers. The technology layer would be organised into software infrastructure and hardware and network infrastructure layers; "functionality" would consist of "IT-functional" structuring units such as core business processes or functional blocks, to reflect the structures most appropriate for the enterprise. As well as the layers and tiers, other aspects which cut right across an application – system management, cross-cutting concerns or security, for example – can be used as architectural domains [Hor02], [IEE00], [Sch04] and [Sch07].

The technical reference model is presented in chart form using cluster diagrams (see Sect. 3.3.2). Figure 5.5 illustrates what a technical reference model could look like for a particular enterprise.

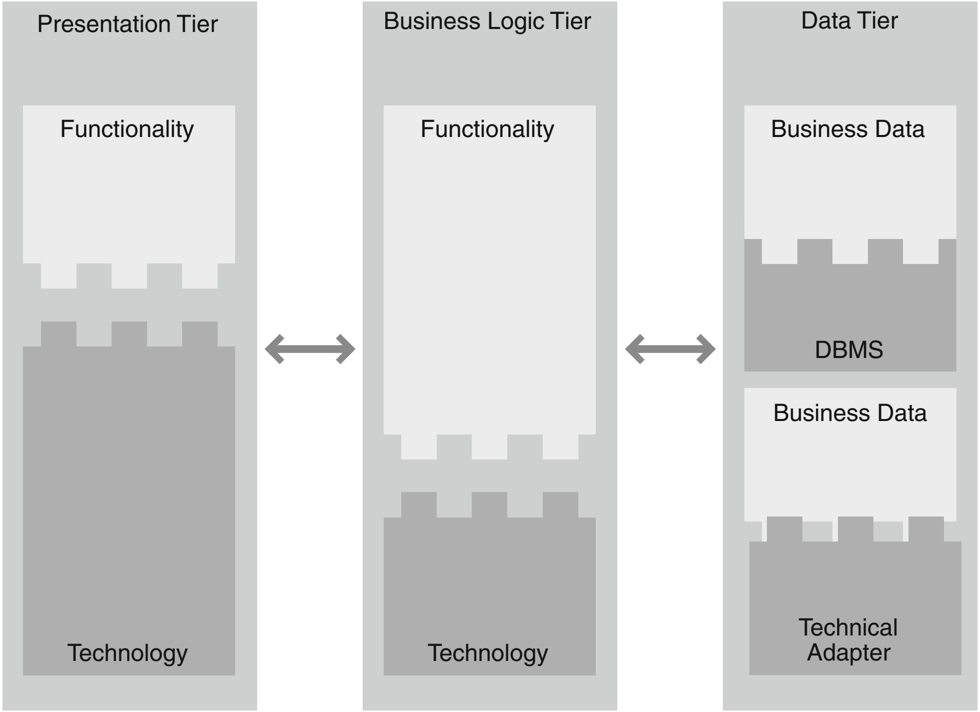

Fig. 5.4 Example layers and tiers

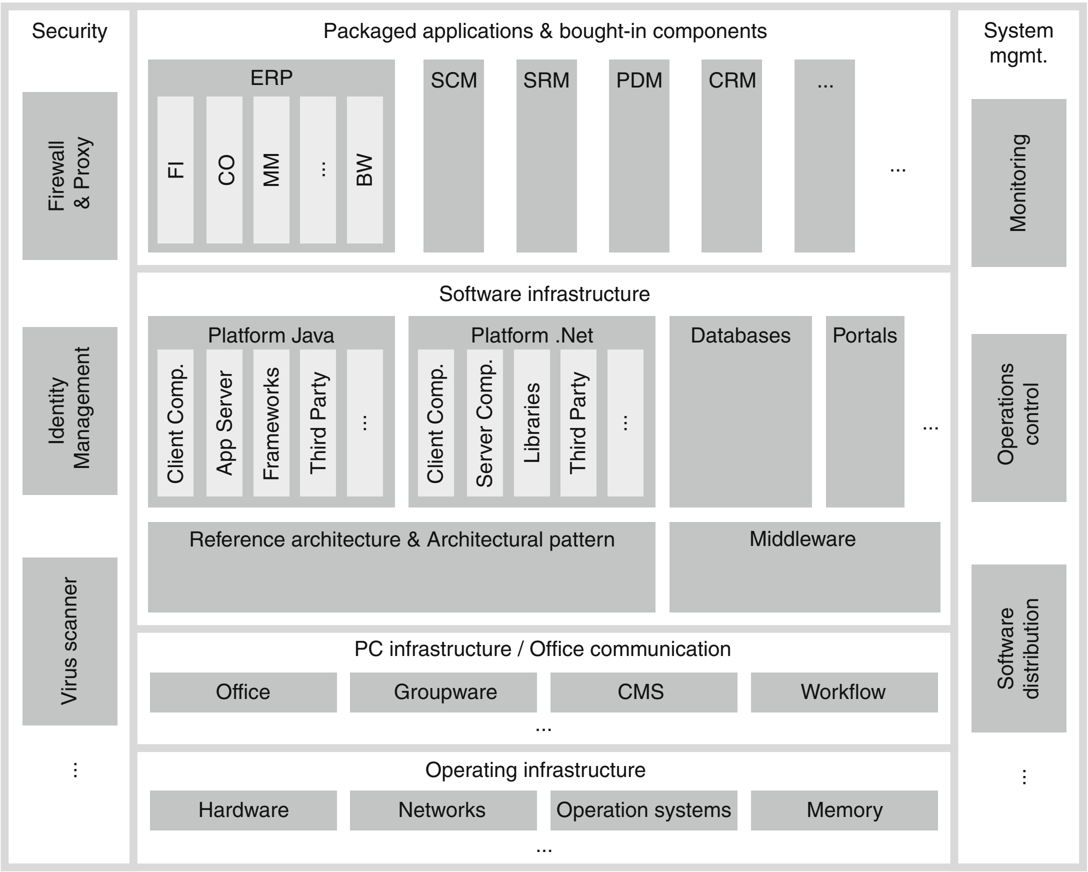

Fig. 5.5 Reference model for a blueprint: an example

In this example, the subdivisions are largely within the individual layers, with only "system management" and "security" cutting right across the model. The functionality is located on the top layer with "packaged applications and bought-in components". This architectural domain is subdivided into units such as "ERP", "SCM" and "SRM", reflecting the functional units with which IT operates.

The technology layer in this example is subdivided into software infrastructure, "PC infrastructure/office communication", and operating infrastructure. The software infrastructure comprises all the technical components for custom software solutions. The components are routinely grouped into platforms with compatible technologies, named after the key technology. One example is the Java JSF platform, which can be viewed as an entity. Platforms can, however, also be bundled into packages on code level, a means of ensuring that only components which really fit together will be used together. This approach helps drive down the costs of provisioning, support and consulting, also simplifying utilisation of the components in IT landscape management. Having clear categorisations in the model renders transparent the connections and interdependencies such as those existing between complementary technologies, substitution and competing technologies.

As suggested in the figure, the platforms can be refined as necessary into additional substructures. However, this is only a valuable exercise if there are enough technical components of sufficient variety to warrant more detailed grouping.

Useful hints:

- The architectural domains you define should have enterprise-wide validity.
- Choose names for the architectural domains which are easy to understand and make clear what the domain does. This makes it a lot easier for people to locate what they need.
- The structure you choose for the blueprint should reflect the structure you have decided upon for the architecture, e.g. layers and tiers.
- Where possible, group compatible components into logical platforms.

  A few suggestions: you can bundle components into software packages and thus head off dependency errors, reduce testing effort and make it easier to utilise the building blocks in IT landscape management processes.
- When defining the architectural domains you intend to use, you should make use of accepted standards, at least regarding accepted taxonomy. One commonly used standard is the TOGAF Technical Reference Model [TOG08].

  Elaborate the selected standard such that the resulting model is a reasonably accurate reflection of the scenario in your enterprise. Essentially, take care to use only the "drawers" which are genuinely relevant for your context.

  The functional and software infrastructure drawers in the TOGAF TRM are often not enough. Design these to fit your objectives.

5.3.2 Technical Components

Technical components are the elements which populate the technical reference model. For ease of use in landscape modelling each component must be adequately documented. Accordingly, each component definition must include the following information. Optional elements are enclosed in square brackets [].

- **Name**: Name of the technical component, e.g. "ORACLE"
- **[Release no]**: Release number of the technical component, e.g. "ORACLE 9.2"

  Together, the name and release number serve as the unique identifier of a technical component.
- **Description**: Description of the technical component, e.g. "ORACLE database system"
- **Position in architectural domains**: The architectural domains into which this technical component has been placed
- **[Planning status]**: The planning status of the technical component: This can be "as-is", "planned" and "to-be".

  - **"as-is"** is the current valid status of the component in the technical landscape model. In other words technical components with planning status "as-is" can be used in projects.
  - **"planned"** is the status attributed to technical components which are being planned or whose development is already underway, e.g. customised framework versions in the process of being developed, or third-party IT products for which purchase negotiations are in progress.
  - "**to-be"** describes the target vision of technical standardisation e.g. "SAP" and "Microsoft" in the "logistics" business segment.

- **[Compliance with standards]**: The degree of compliance with standards indicates whether or not the technical component has been set as an enterprise-specific standard. You might choose to have settings here such as "compliant", "conditionally compliant" and "non-compliant"
- **[Release status[1]]**: The release status indicates whether and under what conditions the technical component is available for inclusion in applications, interfaces and operating infrastructure units. The settings here can include: "unrestricted", "restricted to condition",[2] "individual release[3] and "not released"
- **[Status in lifecycle]**: This allows conclusions to be drawn on questions such as the stability of the component. The possible settings can include: "prototype", "pilot", "in production", "legacy"[4] and "decommissioned"

[1] Also termed standardisation status.

[2] Conditions should be formulated in text form, e.g. "For use in sales support business segment only".

[3] Released only in exceptional cases; separate authorisation is required in each case.

[4] Also termed "phase-out".

- **[Utilisation period]**: The period over which the technical component is used. For components with "as-is" status, this will be the period of time over which the component can be included in applications, interfaces and operating infrastructure units. For "planned" or "to-be" components, this is a statement on the planned period of use.
- **[Hierarchy]**: An indication of what the technical component consists of, e.g. "Application Server" consists of "Web Container" and "EJB Container".
- **[Interdependencies]**: A description of component interdependencies, e.g. "Application Server X" needs "JRE Version Y". The information here serves as the basis for later dependency analysis.
- **[Successor]**: A link with the chronological predecessor or successor of the technical component in question, e.g. successor release of a third-party IT product.
- **Guidance for use**: You must provide guidance on how each technical component is to be used. Requirements and dependencies – such as what is required in order to install an application – have to be stated; users also need guidelines for configuration, programming examples for frameworks and instructions for integrating into portals and migrating to new versions.

This core data can be enriched by additional information specific to your enterprise. For example, you might elect to categorise technical components, specify manufacturer details or component "owners" in your enterprise (see also Sect. 4.3.1).

The more optional information is used, the greater the overall complexity. To give an example, if you are making use of the planning status, period of use and the release status, you set up the following dependencies:

- If the planning status is "planned" or "to-be", the period of use must lie in the future.
- If the planning status is "planned" or "to-be", the release status must be "not released", because the technical components may not at the present time be included in applications or other entities.

Useful hints:

- You should include either compliance with standards or the release status in the definition of a technical component – but not both.
- Use the status in the lifecycle merely as an additional information field. Making it mandatory will drive up complexity unnecessarily in both use and maintenance.

- You can combine the planning status with the lifecycle status, using settings such as “to-be”, “planned”, “in-development”, “prototype”, “pilot”, “in-production”, “legacy” and “decommissioned”.

  In such a case, though, the use of the technical components will have to be made contingent on their status.
- The period of use can be defined for each status in the lifecycle. However, since the meaning of the period of use can be different for each status, and periods can also overlap, you are setting yourself up for a huge increase in complexity. This makes use of the component difficult and also entails considerable maintenance and update effort.

When you are first embarking on technical standardisation, consider which information is likely to be the most appropriate way for your enterprise to describe the technical components. Bear in mind that certain information is essential if you want to effectively manage how technical components are used and integrated into the landscape models.

The following summary of the optional components provides valuable ground rules for your first venture into technical standardisation.

Getting started: ground rules for technical standardisation:

- **Your blueprint should include only as-is components which can be used directly.**

  If you are going to document to-be and planned components, you will need to stake out development trajectories clearly, also in terms of interaction with other aspects. Considerably more effort will have to be channelled into update and maintenance of technical landscape models.
- The **release number** is essential to highlight any dependencies between versions.

  Not all technical components need a release number – for example, programming languages or general technology terms such as SAP. However, be sure to specify release numbers for reference architectures, architectural components, third-party IT products and technical building blocks.
- The **release status** or alternatively **compliance with standards** is necessary to show the degree of standardisation.
- By specifying the **period of use**, you can show when particular components are going to be phased out (valuable for projects).
- By describing the **use**, you can ensure that dependencies between technical components show through in the blueprint. This is key in order to identify the technical flashpoints where there is a need for action.

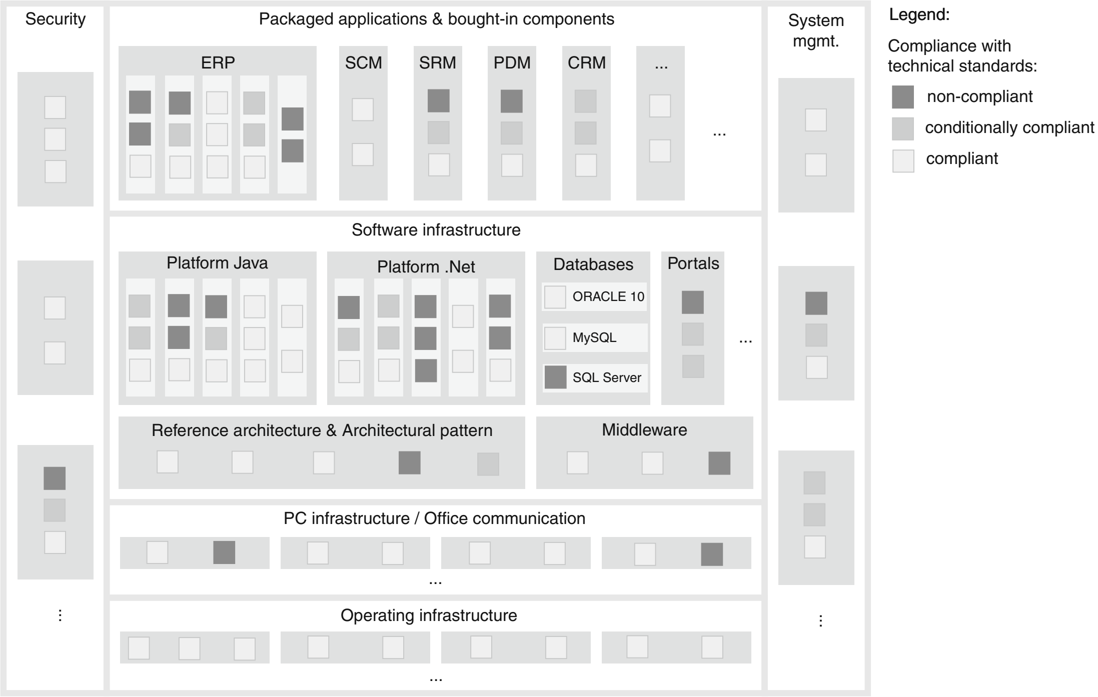

Fig. 5.6 Blueprint with contents (Cluster diagram)

The blueprint is generally presented in graphical form in a cluster diagram or a technical landscape diagram. Figure 5.6 shows the reference model presented in Fig. 5.5, enriched with technical components – for example, the “databases” drawer is shown containing the database systems ORACLE 10, MySQL and SQL Server.

Figure 5.6 provides additional information on the release status of the blueprint elements. Accordingly, the diagram shows at a glance which technical components can be used in projects and, if so, under what conditions.

Figure 5.7 presents the populated blueprint as a technical landscape diagram. Each of the technical components is placed to reflect its affiliation to architectural domains and tiers.

When a blueprint is presented in a technical landscape diagram, the *x*-axis generally shows the architectural domains (subdivided into sections as appropriate). What you choose to show along the *y*-axis depends on what correlations you wish to investigate. Common choices are locations,[5] release status, degree of standardisation, or responsibilities.

As in a cluster diagram, there are various ways to differentiate technical components in the technical landscape diagram. Components can be colour-coded, for instance, or have different border styles to indicate information such as the release status or degree of compliance with standards, also aspects such as locations, responsibilities, technical status, security level or cost.

[5] For example when there are different technical standards at different enterprise locations.

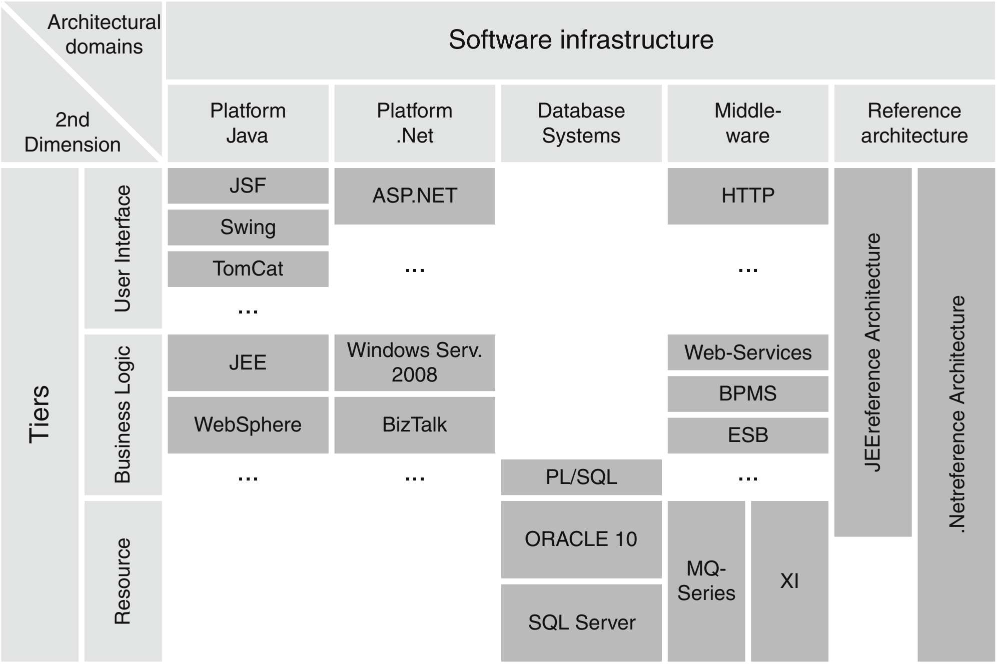

Fig. 5.7 Blueprint with contents (technical landscape diagram)

Technical landscape diagrams can typically also serve to show the utilisation of components in applications, interfaces or operating infrastructures. Unlike the blueprint in Fig. 5.8, "applications" are used here as the second axis. For each application, the diagram shows a row with the application's technical realisation,

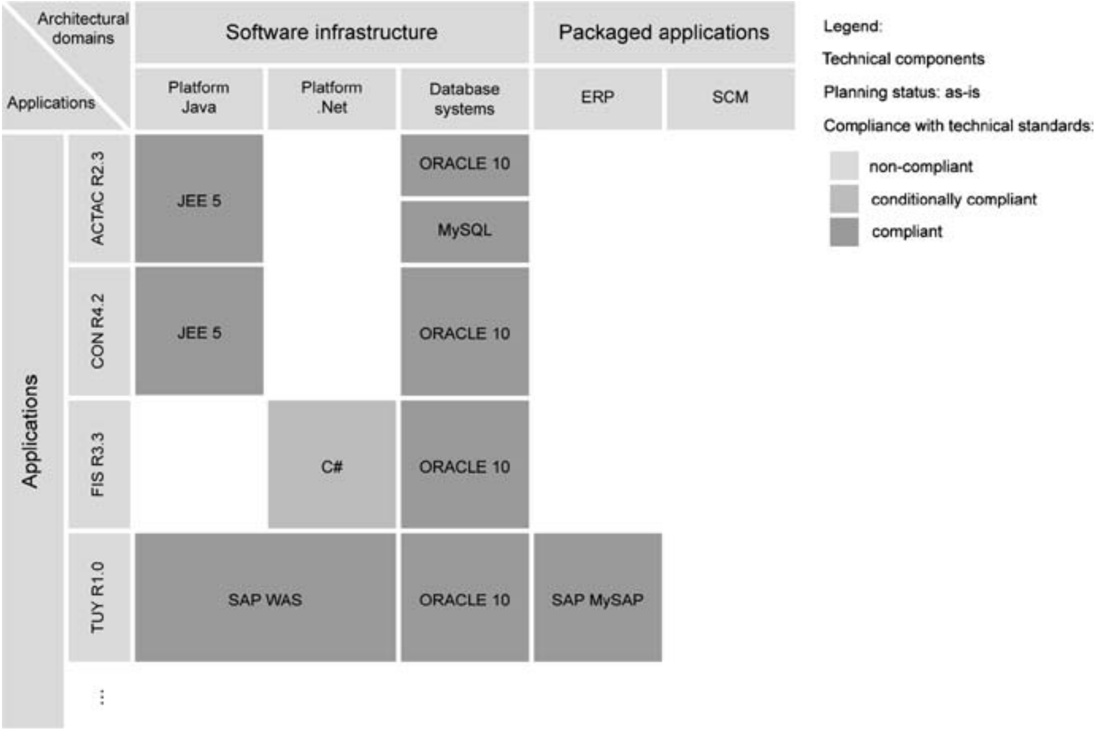

Fig. 5.8 Utilisation of technical components (landscape diagram)

sorted by architectural domains. To pick an example from the diagram, application ACTAC R2.3 uses the technical components “JEE 5” from the architectural domain “Platform Java” and databases “ORACLE 10” and “MySQL” for its technical implementation. The diagram can also present other information – the example here shows “compliance with technical standards”.

Fact file:

- Each technical component should be placed in as few architectural domains as possible, ideally just one.
- When you first start working with technical standardisation, it is advisable to limit the number of information fields which needs to be gathered for each technical component. A few pointers are provided earlier in this section.

  By keeping things simple, you can get technical standardisation moving quickly and effectively, and also keep a firm lid on effort for initial and ongoing maintenance.
- Cluster diagrams or technical landscape diagrams are used to present a blueprint in graphical form.

  Technical landscape diagrams use architectural domains as their primary axis. What you choose for the second axis (and how you sort and arrange clusters in cluster diagrams) depends on what you wish to investigate with the diagram. You can choose to emphasise (e.g. through colour coding or border styles) aspects such as the release status or whatever is relevant for you.
- You will generally use a landscape diagram to present component utilisation (with architectural domains along the first axis and applications along the second).

5.4 Technical Standardisation Processes

This section describes the processes which are applied in technical standardisation. They include:

- **Maintaining, providing and communicating the blueprint**: This means making sure that the blueprint – the technical reference model and the filler elements – are published in up-to-date form, including the release status (see Sect. 5.4.1).

- **IT innovation management**: If you scan the market on an ongoing basis, you will be able to identify potentially relevant technological innovations and trends while they are still on the horizon, and to appraise them in terms of maturity, associated risks and practical feasibility (see Sect. 5.4.2).
- **Strategic evolution of technical standards**: This involves designing the to-be blueprint and deriving appropriate standardisation to reflect the requirements arising from IT landscape management and the operational side of the business. Technical innovations are also channelled into the evolution of technical standards (see Sect. 5.4.3).
- **Enacting standardisation**: This is all about creating or modifying technical components in accordance with strategic objectives and the specific requirements of your enterprise (see Sect. 5.4.4).
- **Directing compliance with technical standards**: Ensuring that the standards, once defined, are actually applied (see Sect. 5.4.5).

5.4.1 Maintaining, Providing and Communicating the Blueprint

Maintaining the blueprint has to be an ongoing process. Whenever a technical component is added or modified, you have to update and re-publish the blueprint. One option is to publish on the intranet, which you can also use as a platform for providing additional guidance for technical standards or as a resource for downloading software.

Important:

- **Communicate new and modified technical standards**: People can only apply technical standards if they know they exist. Take steps to ensure the blueprint is always up to date and that all relevant stakeholders are aware of its existence.
- **Make the blueprint simple to use**: The technical blueprint must be easily accessible, and the technical standards must be easy to locate.
- **Guidance for use**: Only if you actually provide guidance on how the standards are to be used – checklists, for instance, or a use-case concept – will you have any chance of making sure technical components are used for their intended purposes. At the very least, you should make sure there is a link to documentation or to the installation packages provided with the technical components.
- **Keep things tidy**: Also part of maintenance is the continuously checking if standards changed. Any standards which are no longer relevant should be clearly marked for decommissioning. Otherwise you will find technical standards extremely difficult to manage and maintain.

5.4.2 IT Innovation Management

The future relevance of technical standards – i.e. whether they are going to remain valid for an appropriate length of time – is a key aspect you will need to consider in the evolution of your blueprint. IT innovation management makes an important contribution here (see Fig. 5.9). Make it a principle to scan environments on an ongoing basis in order to identify and investigate trends, and be better able to appraise technological developments and the lifecycles of the technical standards you are using. This will help "future-proof" the IT in your company, enabling you to manage fast-paced technological change and ever shorter lifecycles of IT products.

Key elements of IT innovation management:

- **Technology scanning**: Keep your radar on at all times! This will help you perceive fledgling trends and technologies and new third-party IT products emerging onto the market, and to keep close track on their development.

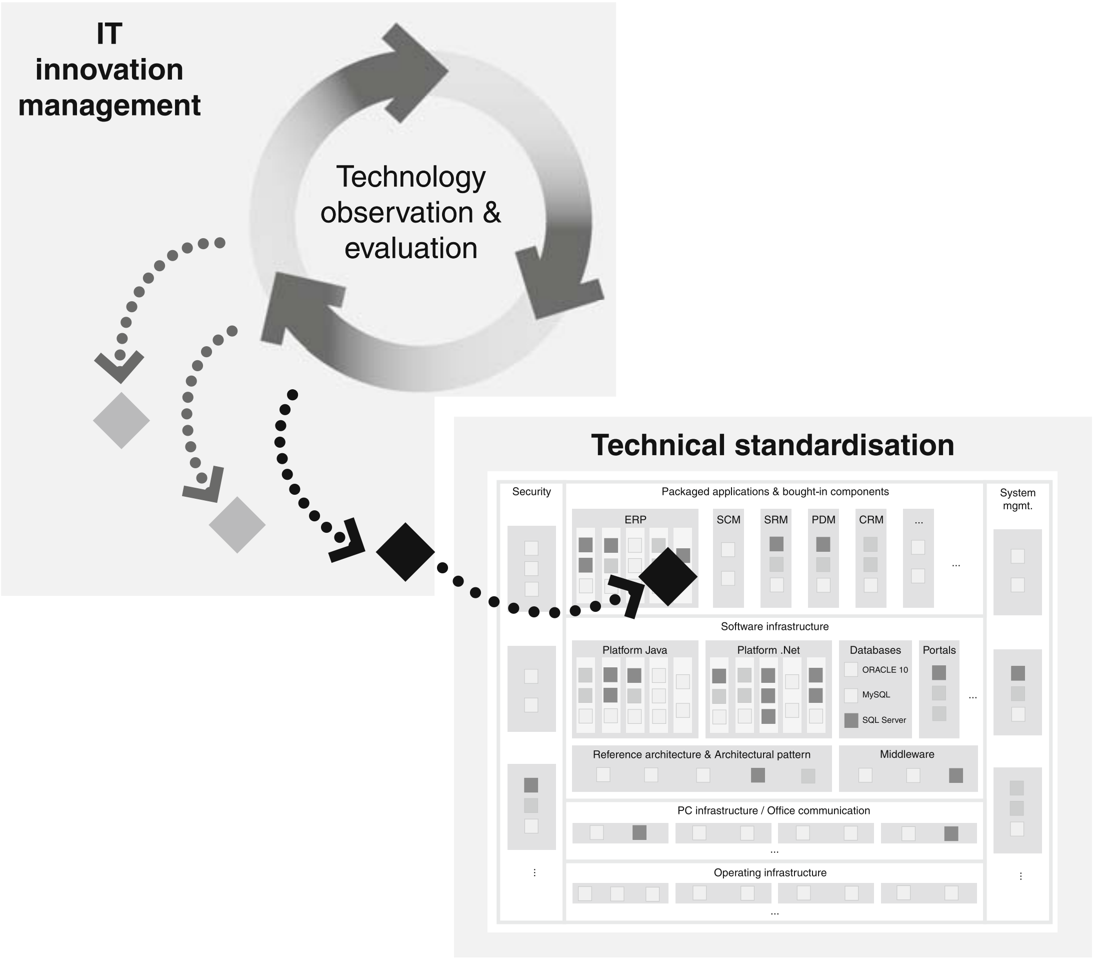
IT innovation management
Technology observation & evaluation
Technical standardisation
Security
Packaged applications & bought-in components
System mgmt.
ERP
SCM
SRM
PDM
CRM
...
Software infrastructure
Platform Java
Platform .Net
Databases
ORACLE 10
MySQL
SQL Server
Portals
Reference architecture & Architectural pattern
Middleware
PC infrastructure / Office communication
Operating infrastructure

Fig. 5.9 IT innovation management

Since there are technological changes going on all the time (and in considerable numbers), you might have to scan a wide range of potential areas and topics.

- **Rough assessment of technologies**: Analyse the technological maturity and the potential of the likely-looking IT innovations in order to narrow down the scope somewhat. You can base your technology evaluation on either general or enterprise-specific criteria, e.g. available expertise in the market or company, or availability of solutions (IT or otherwise) and references. Making a rough estimate of the likely lifecycle is also paramount, particularly when you are looking into third-party IT products. In general, you will also make use of analyst reports, such as how a product is positioned in the Gartner Hype Cycle, to filter out some of the innovations.
- **Identify IT innovations which are relevant for the enterprise**: To identify the relevant IT innovations, appraise the technological maturity, the potential and the possible time of introduction in your enterprise, and weigh all this up in terms of opportunities and risks for the company. You can use a range of criteria – for example, to appraise maturity you can conduct trials with the technology or third-party IT products and obtain some hard, practical evidence. You might also like to consult expert opinion, either people in your own enterprise or outside consultants. Appraise the maturity and robustness of each IT innovation you consider relevant.

The time of introduction of an innovation in your enterprise depends particularly on your enterprise's attitude toward innovation. Of the categories shown here (see [Rog95], [MIT03]), which best fits your enterprise?

- **Innovators**: Fond of experimenting, innovators are among the first to engage with a new technology, despite all its risks and the lack of a practical track record. It is these people, with their enthusiasm, who spark off a new trend.
- **Early adopters**: are aficionados of technology and are quick to take up an innovation once an initial level of maturity has been reached and there are opportunity paths charted out. Despite potential early glitches, the early adopters see the new technology as an opportunity to carve out competitive advantage. They are the first real customers for the providers of the new technology.
- **Early majority**: customers have to be persuaded by the innovators and early adopters that there is a case for the technology. They draw on the experience of innovators and early adopters to reduce their own risks. Once the first positive reports are in and clear benefit can be identified, they jump on board before the technology has really picked up pace. As a rule, the early majority are looking to achieve benefits which are measurable in financial terms (faster, cheaper or better).
- **Late majority**: These customers adopt the technology only when the innovation has taken firm root and is already in widespread use. This group is more

conservative in outlook. They do not come on board until the new technology has been tested and has a proven track record. The risk is minimal and the benefit secure.

- **Laggards**: The laggards are more likely to look to the past and traditional values for their guidance. Often, any change is precipitated only after enormous pressure from outside.

The right adopter category for your enterprise will depend on your enterprise's strategic direction and its culture. If your company is striving for technological leadership, it will usually be an innovator, early adopter or at least part of the early majority. If it is committed to cost leadership, the company will usually be among the late majority or a laggard.

IT manufacturers tend for the most part to pursue early-adopter strategies. There are ramp-up programmes in the SAP field, for example, which have good effort-to-benefit ratios. Reduced licensing and maintenance fees, and training and coaching, make early adoption choices far easier for the customers.

The need to identify relevant IT innovations can easily be submerged by day-to-day business, so special groups such as think tanks and "innovation spearheads" can be instituted to ensure space and time is set aside. Another source of information is quite simply other people: just think of IT communities, consultants and universities.

It is essential to keep permanent track on how well your IT innovation management is working. Keep tabs on the number of issues you are scanning, or on what feedback you are receiving from markets. There is no point trying to number-crunch your way through innovation management. Unfortunately, there are no statistics testifying to a connection between research & development effort and corporate success. R&D capital is venture capital.

Irrespective of this, all the IT innovations you identify must be put under scrutiny: are they a good fit with your needs? Are they stable enough? As you go about technical standardisation, you need innovations which will be a good basis for technical implementation of your applications, interfaces or elements of your operating infrastructure. To appraise innovations, you should use the same criteria as for the strategic evolution of technical standards (see Sect. 5.4.3).

Fact file:

- By permanently scanning and monitoring emerging technological developments and keeping tabs on the lifecycle of the technical standards you are using, you can ensure the continuity of business operations.
- IT innovation management delivers key input for strategic evolution of technical standards.

- Deciding on whether a technology is future-sustainable often comes down to “gut feeling”. Experienced software architects and tried-and-tested methodology help distil this gut feeling into fact-based findings.

Ultimately, only time will tell whether the assumptions underpinning your strategic IT planning turn out to be the right ones. However, you can help build in as much certainty as possible by careful evaluation, analysis, design and testing.

5.4.3 Strategic Evolution of Technical Standards

Over time, blueprints will need updating to keep them in line with business requirements and IT goals. The technical reference model, or rather the architectural domains, have to be brought into line with the changed situation, as do the technical components that fill the model. The following questions are of value:

- Which of the existing technical standards are still appropriate, robust and future-sustainable?
- For which technical trends and innovations are new technical standards to be produced for the enterprise? Which existing technical standards are to be phased out in their stead?
- IT landscape management and operational-level project work often reveal areas requiring action or improvement. For which of these action or improvement points are new technical standards to be developed or existing standards modified?

Technical standardisation requirements can be pulled from a variety of sources:

- **Information from IT landscape management: which technical standards are actually being used?**
  The use of technical standards – and any nonstandard technical implementation – shows through clearly in IT landscape management. IT landscape management also highlights particular points requiring action for improvement (e.g. to eradicate technical redundancy) and these action or improvement points serve as input for the technical standardisation process. IT landscape management can also be used to derive the requirement profile for technical components, e.g. in terms of flexibility or SLAs.
- **Operational-level sources**: Technical standardisation requirements can come from grassroots-level activities such as projects, maintenance measures or operation. For example, it might emerge in a compliance project that the enterprise requires some form of standardisation in its archiving solutions, or an own-build

software project might show up a need for technical standardisation for OR[6] mapping.

The standardisation requirements and strategic objectives are channelled into drafting proposals for the to-be blueprint. Figure 5.10 illustrates the result, showing in which architectural domains technical standards are to be produced, modified and phased out.

As shown in Fig. 5.11, the evolution of the technical landscape is an ongoing process. It entails defining the to-be blueprint, the planned blueprint and carrying out the interventions that will move the blueprint toward the plan. The to-be blueprint, which presents a vision of what the blueprint will look like some time in the future, is refined in an iterative analysis & design process which takes account of standardisation requirements.

The to-be blueprint can consist of elements such as reference architectures or architectural components, e.g. "Web 2.0" applications for the relatively new technology "Web 2.0" or also just be a more generalised statement of a need for third-party IT products for archiving solutions.

Once the to-be blueprint has been approved by the relevant decision-making entity (see Sect. 5.5.2), the next step is iterative development of the planned blueprint. Appropriate standardisation measures are derived, evaluated and prioritised for the new standards and for standards which are to undergo modification. The

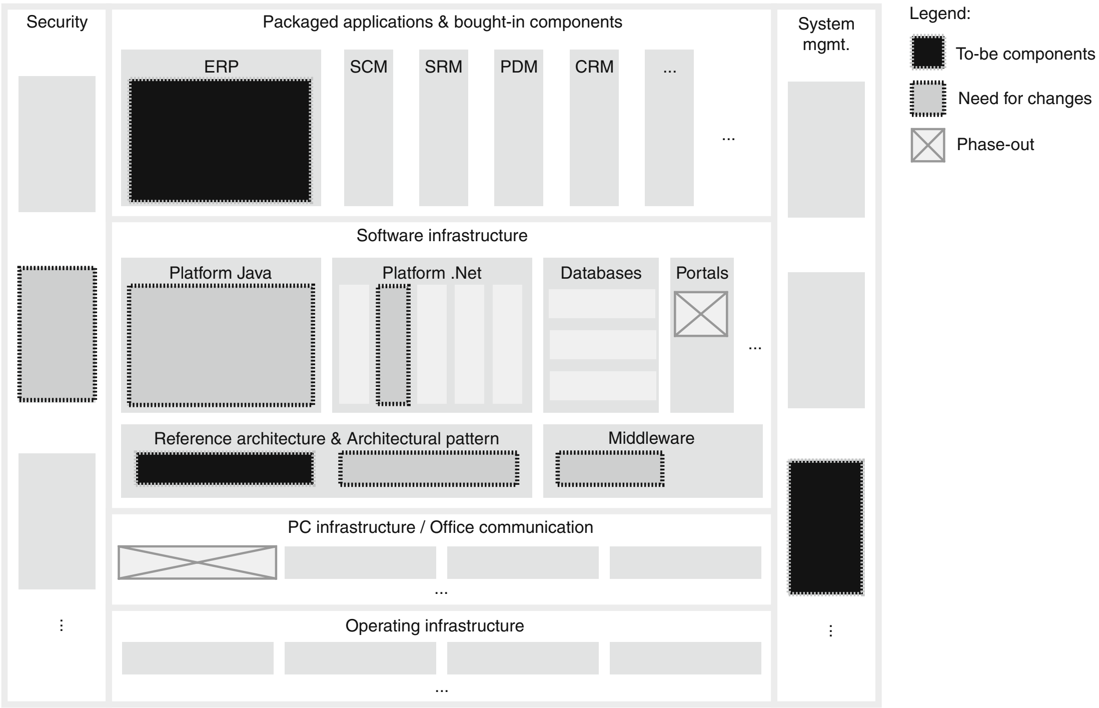

Fig. 5.10 Need for changes in the blueprint

[6] *O*bject *R*elational.

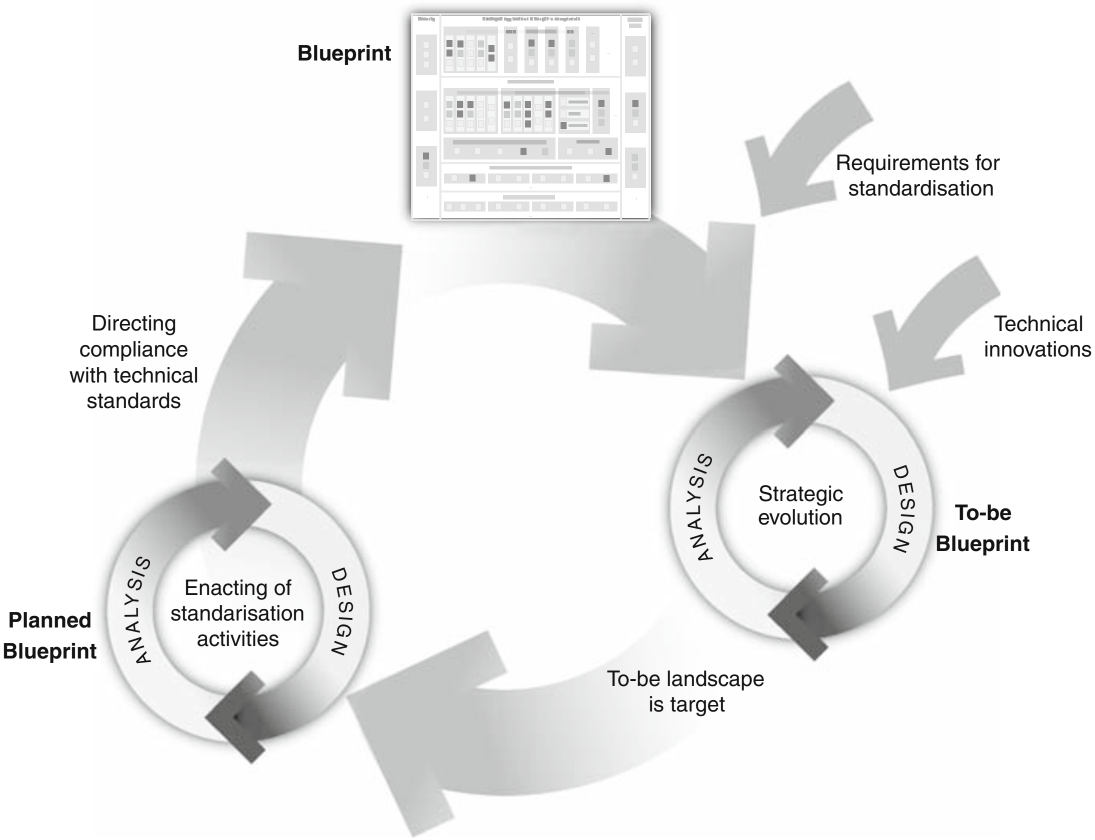
Blueprint
Requirements for standardisation
Technical innovations
Directing compliance with technical standards
ANALYSIS
Strategic evolution
DESIGN
To-be Blueprint
ANALYSIS
Enacting of standarisation activities
DESIGN
Planned Blueprint
To-be landscape is target

Fig. 5.11 Blueprint evolution process

process of enacting standardisation, i.e. putting the planned blueprint into action, keeps the blueprint in a state of permanent evolution.

Your appraisal of standardisation measures will of course consider cost,[7] benefit and operational urgency, but be sure also to apply criteria such as maturity level, consistency with strategy, and operational risk. Once standardisation measures have been be given the go-ahead by the relevant decision-making board, you can proceed with enacting them.

Fact file:

- Alongside technological trends identified in IT innovation management and standardisation requirements emerging from operational-level business, information from IT landscape management – details on how standards are actually in use – is the key input for the strategic evolution of the blueprint.

[7] One-off costs for producing the technical standard, including items such as licensing costs plus ongoing cost such as for maintenance, training or support.

- To evolve the blueprint, you produce a to-be landscape on the basis of strategic and operative requirements and derive standardisation measures from this.
- All standardisation requests should be appraised at least in terms of their maturity level, consistency with strategy, operational risk, cost and benefit – but you can of course use additional criteria if you wish.

5.4.4 Enacting Standardisation

Enacting standardisationThe blueprint is progressed by enacting standardisation measures which have been approved for go-ahead. Standardisation measures include activities such as:

- **Developing reference architectures or architectural patterns**: e.g. creating a reference architecture for JSF applications
- **Evaluating third-party IT products**: This entails determining what products are available on the market, and appraising them in terms of how they fit your company's requirements. A shortlist is then assembled and a recommendation given for one of the shortlisted products.

  Products routinely selected in this way are DMS, CMS or workflow engines.
- **Creating technical building blocks**: Technical building blocks are either developed from scratch, or adapted and configured to meet your company's requirements, e.g. enterprise-specific frameworks for business transactions or auditing
- **Provision of migration guidelines**: When your enterprise elects to phase out technical standards which are already installed in applications or interfaces, you will have to provide guidance on migration – e.g. how to update to successor components. You might choose to develop a complete migration concept, complemented by specific migration scripts. Such guidance is particularly important when you are introducing new releases of technical components.

Standardisation is often enacted as part of projects under the stewardship of technical specialists. For activities involving third-party products, the enterprise may be aided by product vendors.

There must be a dedicated release process in place to check the technical quality of all new or modified technical components, sign off on the components, define their release status and publish the components in the blueprint. Over time, this sequence of actions moves the blueprint gradually toward the to-be status.

Work does not stop once a new technical standard has been included in the blueprint. Rather, you have to keep close tabs on its fitness for purpose, cost and benefit, taking care to obtain feedback from users. After all, the benefit promised in the standardisation request might fail to materialise! If you find there is little or no demand for a technical component, if you feel it is unsustainable or its cost-benefit ratio leaves a lot to be desired, strip it out of the blueprint – quickly. Should the component already be in productive operation, you will have to find an alternative and scope out a migration path.

Fact file:

- Keep a close eye on the fitness of purpose, robustness and future sustainability of all new or modified technical components.
- It is by keeping a constant check on robustness and future sustainability, cost and benefit of each component, and the frequency of its use, that you can effectively steer the evolutionary development of the blueprint. A good way of obtaining such information is to invite feedback from projects or users.

5.4.5 Directing Compliance with Technical Standards

Even the best technical standards are no use if they are not applied! A proactive stance is what works best here, ensuring standards are actually carried forward into practice and enabling your enterprise to attain the goals it wishes to accomplish through its standardisation efforts (see Sect. 5.2). What matters here is that people actually adhere to the technical standards in their projects and maintenance measures and in routine productive operation. You can direct compliance with:

- Strategic controlling, using metrics such as the degree of standardisation or standards compliance as key criteria for investment decisions (see Sect. 6.3).
- Embedding technical architecture management into your organisational structures. This builds more effectiveness into the process of selecting and applying technical standards and enforcing their correct use (see Sect. 5.5).

Important:

- It will be impossible to direct compliance unless there is a possibility to directly influence the technical design stage of projects.
- To make standardisation enforceable, the overarching policy has to stake out precisely how much latitude people are allowed in decisions – there should be clearly demarcated trajectories which people cannot break out of without good reason.

5.5 Organisational Structures

The effectiveness of technical standardisation is due in no small part to how well it is embedded in the organisation. The following aspects are important:

- Clear roles and responsibilities in technical standardisation
- Integration into strategic and operational level IT processes, notably project management
- Decision-making entities and processes for
  - Strategic blueprint planning (to-be landscape and standardisation) and
  - IT investment decisions, including defining project portfolios and releases with due consideration of technical standards
- Maturity level of IT in terms of its technical standardisation

5.5.1 Roles and Responsibilities

A separate role with clear rights and remits is essential to move the technical standardisation processes forward. This role is termed “IT architect” in the following.

The IT architect is tasked with providing and evolving the technical standards, and for guiding and advising on their use. The architect also has the job of quality-assuring any technical standards developed by other parties, and of signing off on them.

Whether technical standardisation accomplishes its goals ultimately stands and falls with the people tasked with implementing it. Accordingly, IT architects have to have a persuasive, compelling manner – and be as convincing as the prescribed technical standards themselves!

The skill profile of an IT architect is daunting – but it does take finely-honed capabilities to accomplish the goals of technical standardisation.

- **Highly experienced employee, in the company for many years, with an excellent grasp of software architectures and technology and extensive project and business administration experience**: IT architects must be able to give prompt, realistic appraisals of new technologies and trends. As such, they need to be in touch with how things work in practice, and have experienced the issues that typically confront people in projects and productive operation – for example, by working for a spell as a software architect. Further details on the role of the software architect are provided in [Star08] and [Vog05].
- **Ability to grasp new concepts quickly and find workable solutions; good conceptual thinking skills**: To convince others that a particular technical standard, be it a reference architecture or third-party IT product, is the right way to go, the IT architect will need to place it in the context of its later use, e.g. in a project, and show how the standard will lead to requirements being fulfilled.

- **High level of commitment, good communication skills, and ability to present a compelling case**: Particularly in the first few years after introduction, technical standardisation has an uphill struggle. The IT architect has to persevere, reinforcing the case to application managers and project leaders and showing what value the standards deliver in specific contexts.

Fact file:

- A specific role, the IT architect, is tasked with delivering enterprise standards and ensuring their continual development over time.
- The skill profile of the IT architect is described in vivid terms in [Nie05]: "IT architects must work with their heads in the clouds and their feet on the ground".

5.5.2 Entities, Boards and Integration into Processes in IT and Decision-Making

In order to be effective, technical standards genuinely have to be applied in practice – and this means having authoritative boards in the enterprise and integrating technical standardisation into operational-level and strategic IT management and decision processes.

The impact of IT on investment decisions is fundamental. When project portfolio managers come to appraise project requests, they also need to include evaluation criteria and metrics to measure the degree of standardisation or deviations from technical standards, and channel these findings into their decisions.

Blueprint Board

Investment decisions also have to be taken on standardisation efforts themselves. There should be a dedicated board in place to define the to-be technical blueprint, appraise what each standardisation measure is about, and prioritise them. This board is termed the "blueprint board". In practice, blueprint boards can exist both with and without authority[8] to take decisions on investments. In addition, the blueprint board must ensure technical standards are monitored continually with regard to their appropriateness, robustness, future sustainability, cost and benefit.

The members of the blueprint board are generally the CIO and IT managers in charge of IT operation, application engineering and possibly also infrastructure provisioning. Other members – key to the board's composition – are experienced IT architects.

[8] Whatever authority the board has, its scope for taking investment decisions will usually be substantially restricted.

Project Support

Alongside the quality and benefit of the technical standards, the key driver of success is the guidance and coaching provided to projects to help them select and use standards and embed them into project management and commissioning processes.

Without qualified project support, there will be no way of actually ensuring that technical standards are being used to the purpose intended. Therefore, even while projects are at the high-level draft stage, appropriate technical components should be selected from the blueprint. IT architects must be proactive in delivering advice and guidance to project teams. The IT architect can either be involved in the project itself and play the part of the project's software architect, or take an external role as a coach or sparring partner.

Whether you take a light touch with project support or aim for more intensive involvement depends on the complexity or strategic significance of the project in question. Be sure always to provide appropriate support to all major projects!

Quality Assurance Measures

Once the conceptual IT design has been finalised, if not earlier, instigate quality assurance measures such as reviews or handover & acceptance assessments to verify that projects are consistent with standards and use the appropriate technical components to the purposes intended.

Reviews and acceptance assessments must at least verify proper use of the components at each of the major milestones. To build in the required pressure and motivation, this practice should be embedded in project management and also in maintenance and operating processes.

At the very latest before the software goes into productive operation, reviews must be conducted to verify compliance with the technical parameters drawn up in the conceptual IT design, notably that the agreed technical standards have been used to the purposes originally intended.

Fact file:

- Include the degree of standardisation (or noncompliance) as appraisal criteria to be investigated when making investment decisions.
- Work with the blueprint board to establish an entity to take decisions regarding evolutionary development of the blueprint.
- Be sure to provide qualified project support.
- Quality-assure the conceptual IT design and sign off on the implementation at least before systems go into productive operation.

5.5.3 Maturity of IT in Terms of Technical Standardisation

The way you organise structures and processes, and what approaches you take to technical standardisation, must fit with the current maturity level in your enterprise. To determine the status quo, you can use the following maturity model (see Fig. 5.12), which draws on project experience:

- **Entrance**: IT peripherals are standardised, and the resulting economies of scale and concentration of expertise deliver enormous cost savings.
- **"Black-box" standardisation**: Technologies, third-party IT products and tools are defined as mandatory objectives for projects, maintenance measures and operation.
- **"White-box" standardisation**: "White-box" standardisation has the objective of enhancing the technical quality of applications.

Each level of standardisation builds on the preceding one. The levels are described in greater detail below.

Entrance

The huge cost savings potential tends to be the primary motivation for embarking on technical standardisation – which begins with commodity IT products

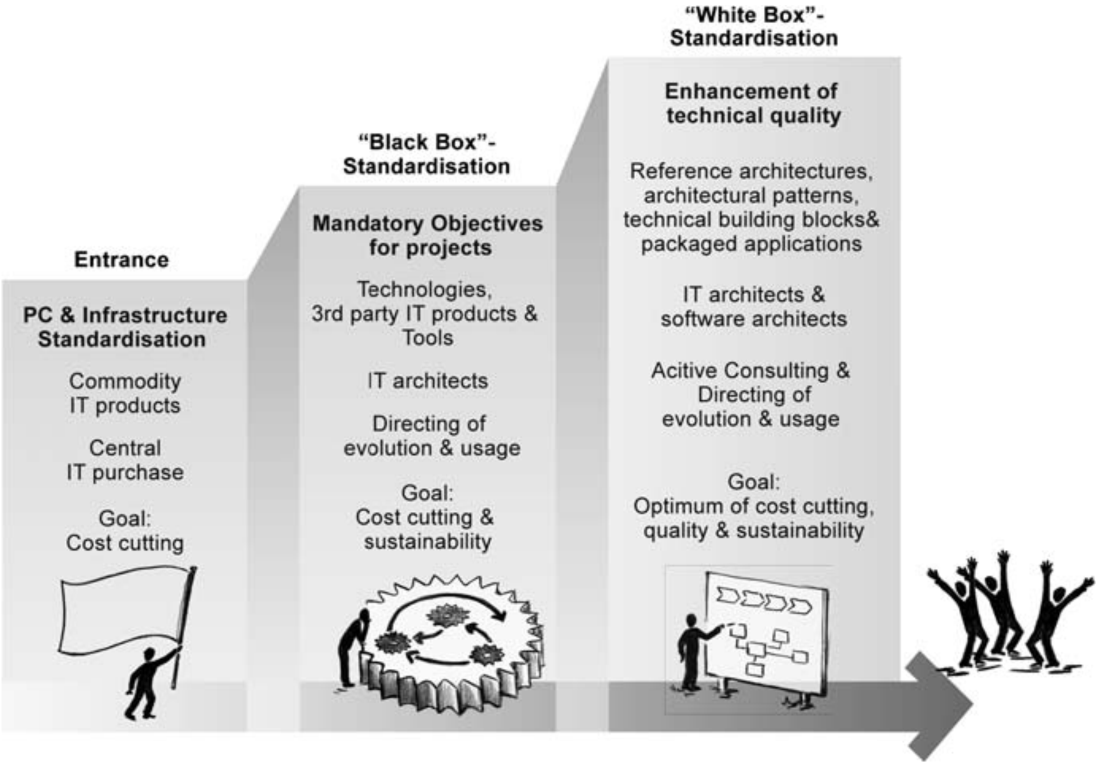
"White Box"-
Standardisation
Enhancement of technical quality
Reference architectures, architectural patterns, technical building blocks& packaged applications
IT architects & software architects
Acitive Consulting & Directing of evolution & usage
Goal: Optimum of cost cutting, quality & sustainability
"Black Box"-
Standardisation
Mandatory Objectives for projects
Technologies, 3rd party IT products & Tools
IT architects
Directing of evolution & usage
Goal: Cost cutting & sustainability
Entrance
PC & Infrastructure Standardisation
Commodity IT products
Central IT purchase
Goal: Cost cutting

Fig. 5.12 Maturity levels in technical standardisation

such as PC infrastructure and office communication, also operating infrastructure. The enterprise can achieve economies of scale by standardising these peripherals. Standardised workstations and operating platforms simplify procurement, provisioning and operation, enabling major cost savings to be achieved with relatively little effort. A key lever for implementing technical standardisation at this level is that it enables IT procurement to be centralised (e.g. with technical buyers). There is no need for specific roles, boards or processes.

"Black-Box" Standardisation

Black-box standardisation expands technical standardisation to include technologies, third-party IT products and tools. Specific technologies – for instance SAP, JEE and .Net – create the technical cornerstones for evolutionary development of the application landscape and operation. Third-party IT products and tools are black-box components which can be used unchanged as self-contained units. Technologies, third-party IT products and tools are specified as mandatory objectives for projects and maintenance measures and for planning the operating infrastructure.

Black-box standardisation entails integrating the standardisation processes into IT processes and decision-making, establishing a board for technical standardisation (see Sect. 5.5.1), and grooming qualified IT architects (see Sect. 5.5.2) to guide the people in charge of projects and maintenance. Before new or modified applications go into productive operation – if not before – tests must be conducted to verify that technical standards have genuinely been applied.

Useful hint:

While you are moving ahead on black-box standardisation, you should actively drive technical homogenisation. It is only by retiring the non-standardised technical components that you can lessen the heterogeneity and achieve sustained cost savings.

"White-Box" Standardisation

White-box standardisation takes things a step further, also standardising reference architectures and architectural patterns, technical building blocks and packaged applications such as Siebel. White-box standardisation moves beyond slotting ready-made technical components into landscapes; it also sets out rules for the inner structure of applications, at least to some extent, in order to raise the applications' technical quality.

In developing a set of practice-tested white-box standards for projects, maintenance and operation, the IT architects take into account the technical innovations relevant for the enterprise (see Sect. 5.4.4).

By staking out particular reference architectures and architectural patterns, you are defining the software architecture of applications on the conceptual level, and giving people a set of practice-tested solution patterns for tasks such as data access, and particularly also for implementing non-functional requirements such as performance or security.

Technical building blocks such as framework components can be used for custom application systems. Off-the-shelf business software will as a rule be mandatory for particular purposes in the to-be blueprint. The introduction of packaged software can be made substantially easier by tools such as configuration templates (see also [Vog05]).

To ensure white-box standards are applied for the proper purpose, IT architects must be able to provide qualified guidance, particularly to projects. The IT architects should either take on the role of software architects themselves in the projects, or support the projects' own software architects.

Important:

- The appropriateness and the cost-benefit ratio of technical standards are what decides whether a technical standard will really remain in lasting use.
- Whether the white-box standards are applied as intended is largely determined by how simple it is to use them, and by the quality of the guidance provided by the IT architects. See also [Star08] and [Vog05].
- IT architects must have a high skill level and project experience in implementing standardisation measures. Train IT architects if you do not already have such highly qualified software architects on your team.

Drafting out the white-box standards to a high level of quality, and above all keeping them up to date, will generally be a lot of work. However, standardisation is not an end in itself – so be sure to keep a vigilant eye on the costs for providing and maintaining standards and ensure they do not outweigh the benefit.

Fact file:

- The entry point into technical standardisation is generally via the commodity IT products, with centralised IT procurement aimed at achieving cost savings.

No IT architects are required for this. Processes are simple to get underway, and this level of standardisation is easy to embed in the organisation.

- After commodity IT products, standardise the technologies, third-party IT products and tools, and present them as mandatory technical standards for projects and maintenance measures. This will help keep your IT landscape aligned with strategic considerations.

  While all this is in progress, actively drive forward the homogenisation of technical building blocks as a means to achieving sustained cost savings.
- Following black-box standardisation, you can, if appropriate, tackle the white-box standards.

  Since development of the standards takes so much time and effort, the white-box standardisation level may not necessarily be the right way forward for your enterprise. Please weigh up the costs and benefits carefully and keep a permanent check on them as the landscape develops.

Dependency Between Objectives and Degrees of Maturity

Not all objectives (see Sect. 5.2) can be attained at all levels of maturity. See also Table 5.1.

The "cost savings" objective can be achieved at all levels of maturity, with a different focus at each level. It is simple to standardise IT peripherals and operating infrastructure, and this also has substantial impact on costs. Particularly by standardising peripheral equipment, you can achieve enormous savings with relatively little effort.

Sustained reduction of operating and maintenance costs is achievable only by homogenisation. By "tidying up" server[9] or software elements or entire systems, you can cull the legacy elements. And you can also reduce the proliferation of maintenance contracts, push down integration costs, simplify workflows and concentrate expertise.

Actually putting homogenisation efforts into practice is no simple matter, in many cases progressing no further than good intentions. One of the reasons is that it means making changes to existing systems. As a rule, the effort then clearly outweighs the benefit. Plus, the benefits – in the form of reduced maintenance costs or faster time-to-system – tend to be long-term and difficult to quantify. As a result, all that happens is the gradual levelling via update measures (projects and maintenance) that would be taking place anyway, whereby effort and benefit have to be compared for each measure. For example, retiring a VAX operating system platform can drag on for many years. It takes perseverance and rigour to see it through!

[9] Such as homogenisation of operating locations by merging data centers.

Table 5.1 Achievable objectives in correlation with degrees of maturity

	Degrees of maturity		
	Entrance	"Black-Box"	"White-Box"
Cost savings			
Use of economies of scale and central negotiating power in procurement	√	√	√
Concentration of expertise	√	√	√
High technical quality			
Re-use of practice-tested black-box components		√	√
Standardisation of white-box standards			√
Appropriate IT support, e.g. flexibility			
Reference architectures and architectural patterns which support the principle of service and component orientation			√
Standard middleware and interfaces/API solutions such as an Enterprise Service Bus		√	√
Future security			
IT innovation management		(√)	(√)
Strategic planning and direction of technical blueprints, and use of building blocks in the landscape		(√)	(√)

Huge benefits are achievable by standardising and homogenising applications. However, this is also the greatest challenge. There must be genuine business reasons driving the move to standardisation. For instance, some enterprises, having investigated business process re-engineering or the likes, might be keen to standardise the IT support of their business processes or business functions. As a rule, implementing technical standardisation often goes hand-in-hand with the introduction of packaged software such as SAP, largely because this is an established way forward ("you can't make the wrong decision!"). Packaged software sets standards in terminology and in business processes. The process standards used in the software[10] can be rolled out across the enterprise.

Enforcing such standards is immeasurably easier than creating your own, because the former, being defined externally, are not changeable. At the same time, this can be a problem if functionality is inappropriate or the structures in packaged software are not really a good fit for the enterprise, meaning substantial modifications have to be made. Changes to packaged software – quite apart from the upfront effort – make life difficult when you move to a new release or install bug

[10] Possibly customized for your enterprise.

fixes. In other words, before standardising on applications, you should make a thorough investigation of the business needs to decide whether a standard solution or a custom-build is more appropriate to map the uniform to-be business processes and functions.

Homogenisation is in any case an ongoing task. Each technical standard which has reached the end of its lifecycle must be marked for retirement. This helps stem the heterogeneity which would otherwise inevitably grow with every new technical innovation.

Useful hints:

- Use the business requirements to make your case for gradually homogenising the IT landscape.
- Explicitly carry out lifecycle management for your technical standards and mark the ones which are likely candidates to retire.
- Common in many enterprises is a range of retirement projects which are "90% complete". This is best avoided!

  Make sure your retirement projects undergo regular controlling. This ensures they can actually be brought to an end and deliver the envisaged cost savings.
- Standardising or homogenising on application level should always be motivated by business reasons. Genuine business benefit is the only justification for the huge effort involved.

To achieve the aims of improving quality or providing appropriate IT support, you need to extend technical standardisation to include basic technologies or even application or operational architectures. For both these objectives, you should prudently select at least technologies, third-party IT products and tools, design sustainable reference architectures and architectural patterns as necessary and ensure there are high-quality technical building blocks available. Standardising at this level engages far more effort than just standardising peripheral equipment.

You can achieve business manoeuvrability by installing flexible, service-oriented and component-oriented architectures, and standard middleware solutions such as an Enterprise Service Bus.

If you are looking to increase the technical quality or build flexibility into an outmoded application landscape which has proliferated over time, you may seriously have to consider complete replacement of at least parts of your application landscape. There is no immediately obvious business benefit to this course of action, and the reduction of high maintenance and operation costs (on systems that have been "maintained to death") is the only real savings potential you can name. Even so, the rollout costs often exceed the savings. Nor do the danger of inflexibility, or the unreliability of the IT, add more credence to the case. These arguments are often not enough to win the right stakeholders over. A quantifiable business benefit has to be found!

Anticipatory scanning of technological innovation, which is part of the IT innovation management process (see also Sect. 5.4.4), is key to the objective of "future-proofing" the IT landscape – as are strategic planning and control of the landscape and of which building blocks are used. The future sustainability of the IT landscape is largely dependent on the core applications and their operating environment, i.e. it is determined by high-level technical standards and as such correlates with the higher levels of maturity.

Important:

Be sure to set realistic goals.
A goal is only realistic if it is consistent with the current or targeted maturity level.

5.6 Guidelines for Technical Standardisation in Practice

Under the stewardship of the CIO, the enterprise team makes a first pass at developing the blueprint, organisation and processes for technical standardisation in line with the IT goals and current standardisation requirements. This initial version is then adjusted constantly to bring it into line with changed environmental parameters. As a rule, the initial version is drafted as part of IT strategy development (see Sect. 2.5) in a series of workshops over a period of several months, possibly with different participants depending on the subject matter under discussion.

Important:

Be sure to involve at least one representative of the executive board, and all IT managers for all core IT areas.

This section provides a guideline for developing technical standardisation to fit your company. Introducing technical standardisation essentially comprises four steps, as shown in Fig. 5.13:

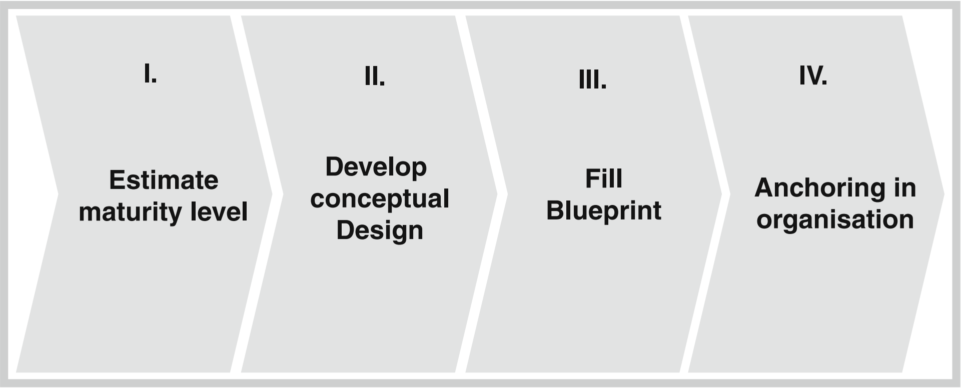

Fig. 5.13 Four steps to introducing technical standardisation

- **I. Estimate the maturity level of your IT in terms of technical standardisation**: Work out where you stand right now regarding technical standardisation.
- **II. Develop conceptual design for technical standardisation**: Define your objectives, the technical reference model, the organisational processes, and tool support.
- **III. Fill the blueprint**: Define the initial blueprint, filling it with the technical building blocks which are already in use. Determine the to-be landscape and the standardisation you will need to introduce in order to get there.
- **IV. Anchoring technical standardisation in the organisation**: Establish the role of the IT architect, and the blueprint board as an authority in your enterprise. Integrate the technical standardisation processes into the IT and decision-making flows and initiate the change process.

These steps are described in greater detail in the following.

I. Estimate the Maturity Level

The maturity model described in Sect. 5.5.3 is used to estimate the maturity level. To determine your current level of maturity in terms of technical standardisation, it helps to answer the following questions:

- What is already technically standardised today?
- Who is in charge of producing the technical standards?
- How is technical standardisation enforced?
- How and by whom is technical standardisation monitored?

Below is a table for each level of maturity which you can use to answer the questions. Determine your current maturity level, and the level you would like to work towards. A maturity level is considered attained when all aspects mentioned in the table are covered (Tables 5.2, 5.3 and 5.4).

Important:

- Set a realistic goal for the maturity level you are aspiring to.
- Do not have too many irons in the fire at once!
- A level is only realistically attainable once the predecessor level is anchored and demonstrable benefit has been delivered.

Table 5.2 Entrance

Maturity level: Entrance	
What is already technically standardised today?	*Commodity IT products* *PC infrastructure and office communication (standardisation of peripheral equipment)*: Standardisation of items such as PCs and notebooks, PC workstation software such as office products and communication platforms *Operating infrastructure*: Homogenisation and standardisation on hardware and network level in operation, e.g. standardisation of servers and network components
How is technical standardisation enforced?	Centralised IT procurement
Who is in charge of producing the technical standards?	IT managers
How and by whom is technical standardisation monitored?	No specific roles, boards or processes are necessary

II. Conceptual Design

Define a reference model for the blueprint, organisational structures and processes in accordance with your enterprise's specific objectives, also factoring the current and target maturity level into your decision. The following activities are required here:

- **Select the enterprise-specific objectives**: From the set of possible objectives (see Sect. 5.2), select those which are relevant for you, or elaborate them to fit your requirements.

  Make sure they fit with the level of maturity you are aiming at. See also Table 5.1.
- **Define the reference model**: Next, bearing in mind your selected objectives and the standardisation requirements, define the "drawers" and "trays" in the model (considering the principles and strategies of the IT strategy, see Sect. 2.4), and refer the outcomes back to your initial stakeholder group.

Table 5.3 “Black-box” standardisation

Maturity level: “Black-box” standardisation	
What is already technically standardised today?	*Commodity IT products* *Technologies*: General technology objectives, e.g. JEE or .Net *Third-party IT products*: Standardisation of third-party IT products in application engineering and in operation, e.g. runtime environments, database systems or middleware *Software and system management tools*: Defining unified tools for software engineering and system operation, e.g. software development or monitoring environments
How is technical standardisation enforced?	Mandatory objectives for projects and maintenance measures, and for planning the operating infrastructure; set by a technical standardisation board
Who is in charge of producing the technical standards?	IT architect
How and by whom is technical standardisation monitored?	Involvement of IT architects in rolling out projects Reviews and handover assessments, integrated into IT processes Permanent monitoring of cost to-benefit ratio

Table 5.4 “White-box” standardisation

Maturity level: “White-box” standardisation	
What is already technically standardised today?	*Commodity IT products* *Technologies, third-party IT products and tools* *Reference architectures and architectural patterns* (see also Sect. 5.3) e.g. SOA reference architecture as vision for further development of applications *Technical components* (see also Sect. 5.3): Standardisation of technical components in the context of custom development of applications, e.g. frameworks for security aspects, troubleshooting or monitoring *Packaged business software*: Standardisation and homogenisation of packaged business software in terms of business support, e.g. a single predefined application for a specific business function
How is technical standardisation enforced?	Mandatory objectives for projects and maintenance measures, and for planning the operating infrastructure; set by a technical standardisation board
Who is in charge of producing the technical standards?	IT architect, possibly collaborating with software architects
How and by whom is technical standardisation monitored?	Involvement of IT architects in rolling out projects Qualified consulting and guidance Reviews and handover assessments, integrated into IT processes Permanent monitoring of cost to-benefit ratio

Useful hints:

- The potential standardisation points you identify when analysing the maturity level must each be categorisable into an architectural domain.
- In choosing architectural domains, stick to those which are consistent with the maturity level you are aiming for. Try to keep the number down, and use no more than eight main structural elements.
- Each architectural domain should be filled with at least one technical building block.

  Remove the domains to which you are unable to assign building blocks.
- Make use of best practices and standards such as TOGAF TRM, as described in Sect. 5.3.1, to define your architectural domains.

  Go through all the architectural domains in the selected standard (e.g. TOGAF TRM) or in the best-practice approach, and deliberately ask whether there is a requirement for standardisation in this context.

- **Define processes, organisation and tool support**: Define processes, organisational structures and tool support in accordance with the maturity level you are aiming for and the objectives you have selected. See also Sects. 5.4 and 5.5.

III. Filling the Blueprint

Once you have decided on the drawers and trays for which technical standards are to be offered, you have to fill them – by assigning each drawer and tray initial contents.

Proceed as follows:

- **Determine which technical building blocks are currently in use (status quo)**: If there is no documentation available giving an adequately up-to-date picture, you will have to stock-take the technical components which are currently in use. You can use information from IT landscape management, findings of earlier inventory processes or items such as purchase lists for third-party IT products.

  The existing technical components are possible candidates for inclusion in the blueprint. Evaluate whether you wish to include these in the blueprint as pre-existing de-facto standards.
- **Design the to-be blueprint (see 5.4.3)**: Design this in accordance with your IT goals, principles and strategies, and what you require in terms of standardisation. You have to identify potential to-be components for each architectural domain. Describe these components in relatively general terms such as “embedded workflow component for JEE applications”.

Evaluate the importance of each to-be component on the basis of your standardisation requirements and information from IT landscape management. You can then make a selection from the to-be components. Consolidate the proposals and integrate them into the reference model you have already created.

Submit the to-be blueprint to the blueprint board if existing, otherwise to the CIO or head of IT for a decision (see 5.5).

- **Derive your standardisation measures**: Using IT landscape management and operational-level business requirements as input, you now derive appropriate standardisation measures to move the blueprint from its present status (as-is) toward your planned status. Elaborate and assess the measures you develop. This is the basis on which the blueprint board or project portfolio board will take an investment decision and thus move the blueprint to the next stage of development (see also Sect. 5.5.2).

IV. Embedding in the Organisation

"Isn't it nice, when things just work" describes succinctly the ultimate target status of technical standardisation. The benefits of technical standardisation far outstrip the costs – at which point (and not before) technical standardisation will be embedded in your organisation.

Keep a close eye on the appropriateness, robustness, future sustainability, cost and benefit of all technical standards, and how often each is used. You can then be sure that the prescribed technical standards are actually delivering on their promised benefit!

When calculating effort, however, do not just consider the initial work of preparing and publishing standards. Prior to unrestricted release of such standards for reference architectures and architectural patterns, third-party IT products and technical components for projects, there has to be clear evidence that they are fit for the purpose and up to the job. Trial-run a technical standard in at least one project or in all relevant application cases. The time taken for such trials must also be considered in your cost-benefit estimate, as does all maintenance and advisory work.

It will generally be impossible to fully quantify the cost and benefit of technical standardisation. Nonetheless, you should try to determine potential costs and benefits and at least make a rough estimate of figures.

Useful hint:

Deliberating the cost and benefit aspects at least helps you work out the decision case. If you are unable to find enough arguments in favour, there is in general not enough need to go ahead with the standardisation!

Before including a new technical standard in the blueprint, you should answer the following questions:

- What costs will arise for producing and providing the technical standards?
  What one-off costs can we expect for the development of technical standards, including licensing costs and effort for producing guidelines on use?
  What one-off costs can we expect for piloting the technical standard?
  What costs will arise in project(s), e.g. for learning or customising? Are there any risks we need to consider?
  What costs[11] or what savings can we expect in operation per year or per project?
  Are there any other costs (such as for coaching, consulting or training) to consider?
- What benefit will ensue when the technical standard is used in productive operation, in various project types or in a specific project, possibly depending on the number of use cases or the number of entities in the data model?
  Which of the following benefit aspects are relevant, and how should these be evaluated for the technical standard?

  – Business benefit, e.g. by simplifying business processes or paving the way to new business models
  – Shorter time-to-system
  – Higher flexibility
  – Greater productivity through efficiency gains (e.g. fewer employees)
  – Higher quality through practice-tested technologies and architectures
  – Cost savings through fewer licences, less training or less maintenance (owing to reduced system complexity)
  – Use of synergy effects as an outcome of reducing technical diversity, e.g. economies of scale through more clout in purchasing, more efficient operation, and concentration of expertise.

Important:

- Each technical standard and each recommended technical innovation has to be demonstrably appropriate, fit for purpose and sustainable into the future.
- Stick to the technical standards that are important and essential for your enterprise. Each standard you introduce brings with it extra work for producing guidance on use, provisioning (particularly updates and maintenance) and guidance for projects.
- Each technical standard must have a clear reference to business requirements and a clearly defined benefit.

  A particular challenge is providing for flexibility in applications, which tends to involve major one-off and ongoing effort. Only make provisions for flexibility where it is genuinely required.

[11] For example maintenance contracts.

It is essential to have technical standards which are suitable, fit for purpose and future-sustainable if you are going to embed standardisation in your organisation. Apart from making sure the standards themselves measure up, you also need to complete the following activities:

- Identify highly qualified IT and software architects, or train them if you have no one with suitable skill profiles on your team (for details on the skill profile, see Sect. 5.5.1).

  You need to do this if you are working toward the maturity levels of black-box or white-box standardisation.
- Establish technical standardisation processes, and integrate them into your IT and decision-making workflows (see Sect. 5.5.2).
- Put the tool support in place and communicate this appropriately to relevant stakeholders.
- Appropriate tool support in line with the conceptual design (see Sect. 5.4.1) helps reduce the scope of communication measures to get technical standardisation underway. Announcements, training and project guidance are essential to get the new technical standards really established.
- Offer guidance and coaching, and ensure these services are a fully integrated part of IT processes.

  Ultimately, you need the active involvement of IT architects, at least in key projects, to ensure compliance with technical standards. Proper guidance and coaching, plus reviews and handover & signoff assessments integrated into IT processes, are the only real way to ensure technical standards are being applied properly (see Sect. 5.5.2).
- Get standardisation metrics established in decision-making processes. The boards with enough clout to drive through decisions are the groups which must enforce compliance with the technical standards. These boards must also use the degree of standardisation (or noncompliance with technical standards) as criteria for appraising proposals and taking investment decisions.

Technical standardisation prerequisites for success:

- You have to strike a balance between the competing goals of flexibility, economy and innovation.

  Flexibility: the basis for responding quickly to trends and changed requirements.

  Economy: planned and executed IT projects have to make economic sense.

  Innovation: the ability to make new business products and processes possible and optimise existing ones impacts substantially on the competitive strength of any company.

- Make sure you have the backing of your executive board.

  Win board members over by making your case on the basis of business benefit (see Sect. 5.2), and be sure to keep up the dialogue. You are then more likely to have the board's backing when it comes to enforcing technical standards.
- Produce high-quality, appropriate technical standards and processes, and train qualified IT architects and software architects with the right skill profiles (see Sect. 5.5.1) and outstanding communication abilities.
- The change process is a long road and you need to be persuasive– and gather up and convince all relevant stakeholders along the way.