

Architecture Principles

From Matchi Wiki

Contents

- 1 Overview
- 2 Core architecture principles
 - 2.1 Simplicity
 - 2.2 Repeatable
 - 2.3 Agile
 - 2.4 Supportable
 - 2.5 Cost effective
 - 2.6 Secure

Overview

The architecture principles provide a guide for architecting a sound solution. A solution should *strive* to comply with all of Matchi's architecture principles. The architecture principles are not mutually exclusive and on in certain cases they may conflict with each other in the context of the business requirements and the final implementation. It is therefore not possible to adhere completely to all these principles. Solutions will, however be scrutinized against these principles as a guide to solution quality.

Core architecture principles

Simplicity

IT solutions must be:

- Understandable
- Simple to use
- Simple to deploy
- Simple to operate
- Simple to support

Rationale:

- Solutions designed in this way will be quicker and more cost effective for IT to implement, operate, maintain, change and support them. If solutions are easy for users to understand then training will be optimised and user errors should be minimised. More often then not, the simpler solution tends to also address the business requirement that it attempts to address better than the complex solution.

Implications:

- The introduction of an IT solution must simplify business operations and fulfil a stated business capability.
- Use as few components as possible to meet business's need.
- Always look to reduce duplication of capabilities.
- Architecture designs must describe where complexity is being introduced and what alternatives were considered. Necessary complexity must be hidden from the user and this may require the deployment of

additional IT systems and services.

- Applications must have a clear purpose and well-defined scope. Applications should be designed to have modular functionality in order to make it easy to modify and re-use. Be especially careful when adding new functionality to existing applications.
- The user interface must present our employees and customers with application front ends that have a common look and feel. They should be intuitive to use, guiding the user to complete the business process.
- IT documents must use clear language and avoid the assumption of technical or domain specific knowledge.
- IT systems and capabilities must be consistently named.

Repeatable

IT solutions must be re-usable and support the same business processes across all sub-businesses.

Rationale:

To enable business growth, IT solutions must be designed to be re-usable and repeatable wherever possible across the sub-businesses, in order to minimise development, deployment effort and training costs.

Implications:

- Common processes and systems must be established wherever possible. A business and technical operating model is slowly evolving and the patterns that this establishes should be applied to new business and IT systems.
- All solutions must be standards based and *open* industry standards must be used where possible.
- Service Oriented Architecture approach: Requirements should be met by building or using existing *services* and consuming them in applications, rather than by building monolithic applications.
- Service orientation give future flexibility and allows a better response to new requirements.
- Service orientation lowers the cost in the long term as it facilitates re-use.

Agile

IT solutions must be flexible and scalable so as to enable future business growth and process change.

Rationale:

- The rapid proliferation of business initiatives results in new IT requirements. IT needs to be able to anticipate and respond to these new requirements quickly and in an agile way in order to support the business strategy. We therefore need to architect solution designs to be flexible and scalable without requiring architectural change.
- It is simpler and more cost effective to build scalability into solutions at the start.

Implications:

- IT solutions must be designed to be capable of being deployed rapidly in new countries.
- IT solutions must be capable of being re-used without the need for architectural change.
- Do not use UK specific volume estimates when designing new solutions. IT solutions must be capable of being scaled, for example up to larger numbers of users or transactions. Artificial constraints (e.g. number of stores, products, promotions, customers, etc.) must not be built into solutions.
- IT solutions must be capable of being both vertically and horizontally scaled.
- IT solutions must be capable of being deployed in different ways, e.g. centralised or distributed deployments in order to be flexible to the specific needs of different markets.

Supportable

IT solutions must be capable of being deployed, operated and supported remotely.

Rationale:

- Well-supported applications provide a reliable automation framework for the business to operate in. Easy-to-support application reduce the operation and maintenance costs of . applications, and also helps in managing a central support operation.

Implications:

- Supportability must be built into solution design as standard. All applications, services and infrastructure must be designed with appropriate resilience, be fault tolerant, be recoverable, be able to be monitored and be able to emit activity alerts to a standard event and monitoring service.
- Use standard technical infrastructure building blocks to reduce variation in the IT estate simplifies infrastructure support, lowers maintenance costs and increases overall reliability of infrastructure..

Cost effective

IT solutions must be designed with total cost of ownership in mind over the entire life cycle of the solution

Rationale:

- The investment in IT solutions must consider the overall long-term cost of hosting, supporting, and decommissioning the solution, not just the short-term costs of development and deployment..

Implications:

- Business needs drives IT developments, which means that a validated business case must exist before architectural design begins.
- Before starting any development or purchasing of services, Total Cost of Ownership (TCO) analysis should be carried out for all solutions and fed back to the business after a solution has been designed.
- Group deals with existing vendors must be leveraged where possible. Check for existing contracts and take them into account, together with the relevant longer term vendor-roadmaps when considering alternative solutions.
- Architecture designs must always consider the overall IT systems portfolio and include any resulting decommissioning of systems.
- Leverage existing repeatable technical building blocks to reduce costs.
- Use one shared and central infrastructure to support multiple systems.

Secure

IT solutions must be designed to be appropriately secure.

Rationale:

- We must protect our corporate information and assets appropriately to meet regulatory, legal and customer needs. We must protect against unauthorised access and both intentional and accidental modification. UK law and regulations provide strong rules on protecting confidential information. It is essential that we are fully compliant with all requirements while looking forward to ensure future compliance.

Implications:

- Security requirements must be integrated into the entire architecture process.
- Architecture designs must reflect policies to minimise improper use of data and security violations.
- Security must be applied consistently and compliance able to be monitored.
- Information security controls need to be clearly defined so security and risk are balanced and understood.

- Auditing requirements must be considered in all architecture designs where appropriate.

Retrieved from "http://wiki.matchi.info/index.php?title=Architecture_Principles&oldid=1265"

- This page was last modified on 14 June 2016, at 13:50.
- Content is available under Creative Commons Attribution unless otherwise noted.