## What architects do

As architects we must act as a bridge between the business and technology, representing and protecting the interests of each stakeholder, often mediating between the two, but allowing the business to drive. This means keeping the following interest in balance.

* functional & quality interests
* cost containment interests
* ease-of-administration interests
* ease-of-learning
* ease-of-maintenance interests

Part of the challenge of letting the business drive is providing enough quality information about the ongoing system development effort back into the business to support good business decision making.

## Goal

The ultimate goal of an architect is creating a production system while optimizing over the architectural significant requirements

# Architecturally significant requirements

Architecturally significant requirements are used in [system design](https://en.wikipedia.org/wiki/Software_design) to drive and justify [decisions](https://en.wikipedia.org/wiki/Architectural_decision); if not satisfied properly, they contribute to the accumulation of [technical debt](https://en.wikipedia.org/wiki/Technical_debt). They are often non-functional requirements and are often described by quality attributes (the -ilities).

# a common set of quality attributes exist, ISO/IEC 9126:

## [Functionality](https://en.wikipedia.org/wiki/Functional_requirement)

## [Reliability](https://en.wikipedia.org/wiki/Reliability_engineering)

## Usability

## Efficiency

## Maintainability

* Portability

Note: ISO/IEC 9126 is een wat ouder model: nieuwe model: ISO/IEC 25010:2011

## function requirements

As an architect we are the bridge between business and technology, it is our responsibility to convert the Architectural requirement to functional requirements. The functional requirements can go into a devops team.

## Feedback

Team feedback converted back to business

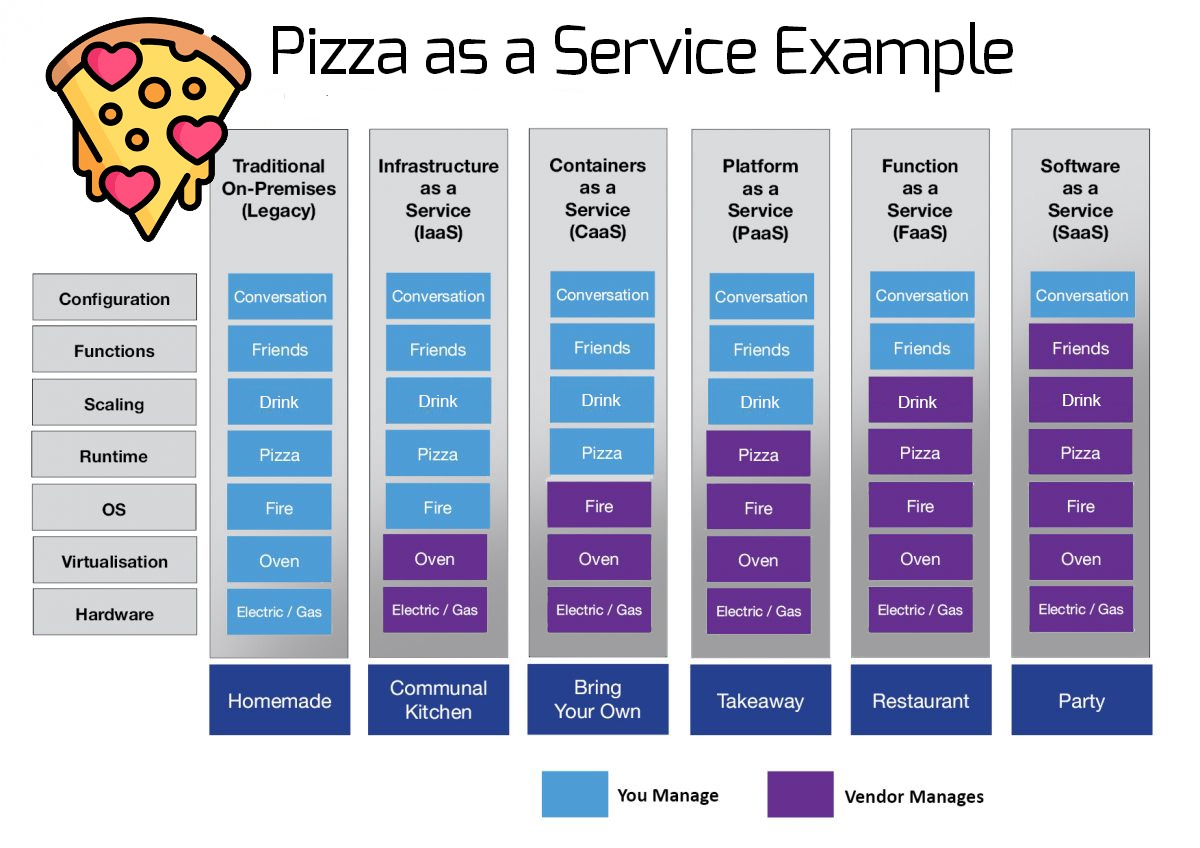
# Team

Admins

Developers

## Dependency

Dependencies on other teams can be usefull or painfull



# Systems

Monitoring nagios/libenms -> Prometheus (organizational common)

Icinga -> checkmk (organizational common)

Provisioing forman -> maas (makkelijker, (organizational common)

Configuration management chef -> ansible (organizational common)

Deployment ansible

Continues integration bamboo (organizational enforced)

Testing selenium

Code review sonarcube

Build gradle, maven, go-build, CMake

Software repository bitbucket

Documentation confluence -> SPHINX