	Process & Organisation	Technology	Quality assurance	Deployment routines	Configuration management	Build & CI
Advanced	**Self organised and cross functional.** The team can solve any task. Releases continuously. Continuous improvement. DevOps!	**Loosely coupled architecture.** It is easy to replace technology for the benefit of something better (Branch by abstraction).	**All testing automated.** Almost all testing is automated, also for non-functional requirements. Testing of infrastructure code. Health monitoring for applications and environments and proactive handling of problems.	**One-click deploy.** Everybody (including the customer) can deploy with one click. 0-downtime deploy. Feedback on database performance and deployment for each release.	**Infrastructure as code.** Fully automated provisioning and validation of environments. Orchestration of environments.	**Build/deploy pipeline.** Same binary is deployed to all environments. Continuous improvement and automation of repeating tasks. Optimised for rapid feedback and visualisation of integration problems.
Intermediate	**Pull-based process.** Measurement and reduction of cycle time. Continuous focus on process improvement. Always production ready code.	**Simple technology.** In depth knowledge about each technology; why it is used and how it works. All of the used technologies are easy to configure and script against. Technology that makes it simple to roll back and forth between database versions.	**Automated functional tests.** Automated acceptance- and system tests. Tests are written as part of requirements specification. All stakeholders specify tests.	**Automated deploy.** Same process for deploy to all environments. Feature toggling to switch on/off functionality in production. Release and rollback is tested. Database migration and rollback is automated and tested for each deploy. Database performance is monitored and optimised.	**Application configuration control.** All application configuration in version control. The application is configured in one place.	**Continuous integration** Continuous integration of source code to mainline. All changes (code, configuration, environments, etc.) triggers the feedback mechanisms. Artifact repository. Reuse of scripts and tools. Generation of reports for the build. Builds that fail are fixed immidiately.
Baseline	**Agile 101.** Improved communication with business. Releases after each iteration. Developers have access to production logs.	**Best-of-breed.** Chooses technology stack based on what is best for each purpose. Chooses preferably Open Source. Avoids products that causes vendor lock-in.	**Automated technical tests.** Automated unit- and integration tests.	**Repeatable deploy.** Documented and partially automated deploy. Database changes are scripted and versioned.	**Dependency control.** Dependencies and libraries are defined in version control.	**CI-server** Automation of builds/tests on CI server. Can recreate builds from source code.
Big challenges	**Silo organisation.** People who in some way depend on each others work are not sitting together. Infrequent releases. Developers do not have access to production logs.	**"Enterprise" infrastructure.** Technology that can only be configured via a GUI. Large "enterprise" suites claiming they can solve all your problems.	**Manual testing.** Test department. Testing towards the end, not continuously. Developers do not test.	**Manual deploy.** Deployments require many manual steps. Manual and unversioned database migrations.	**Manual configuration.** Manual configuration in each environment and on each server.	**Manual routines** Manual routines for builds. Lack of artifact repository. Lack of reports.