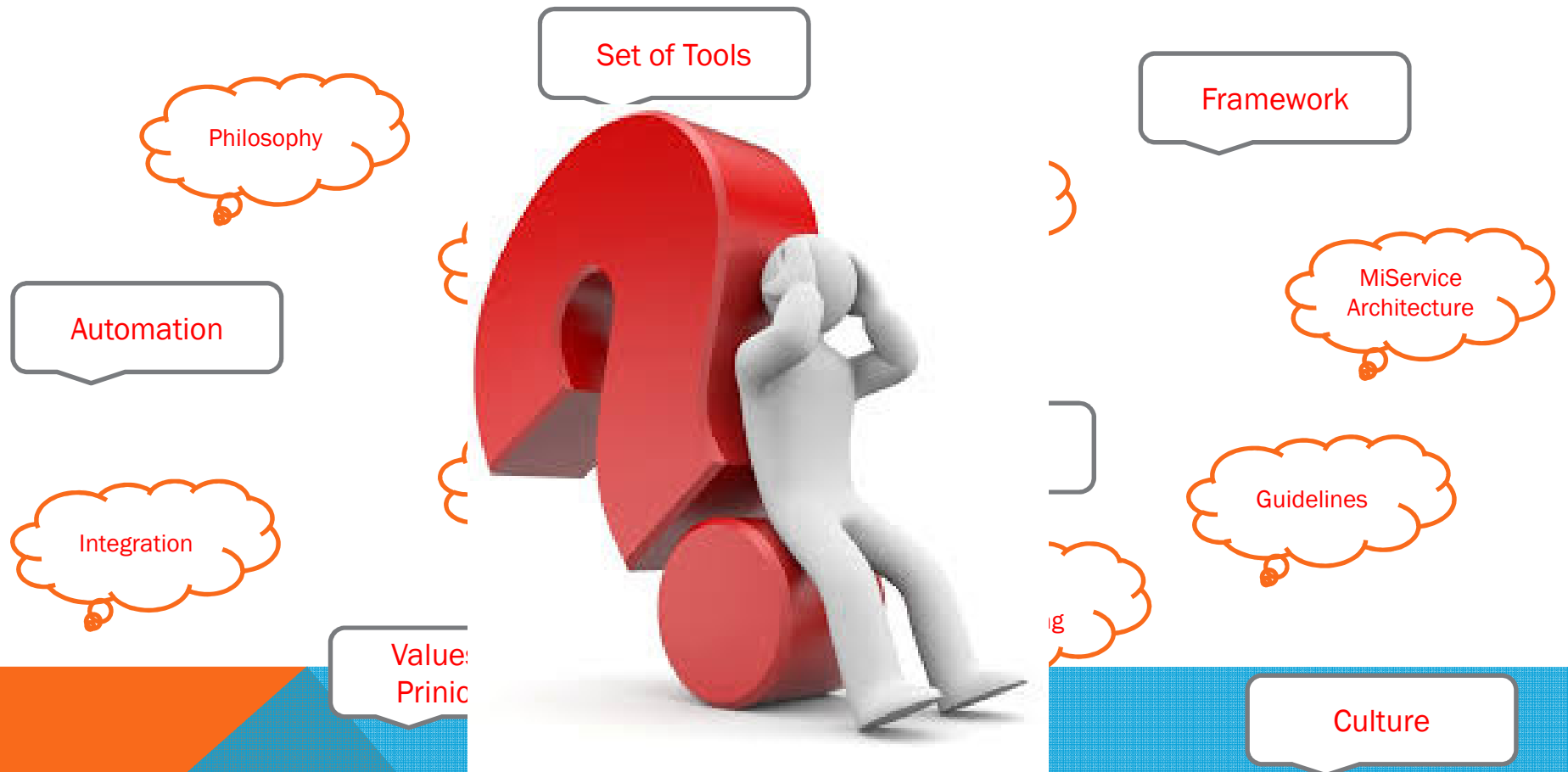


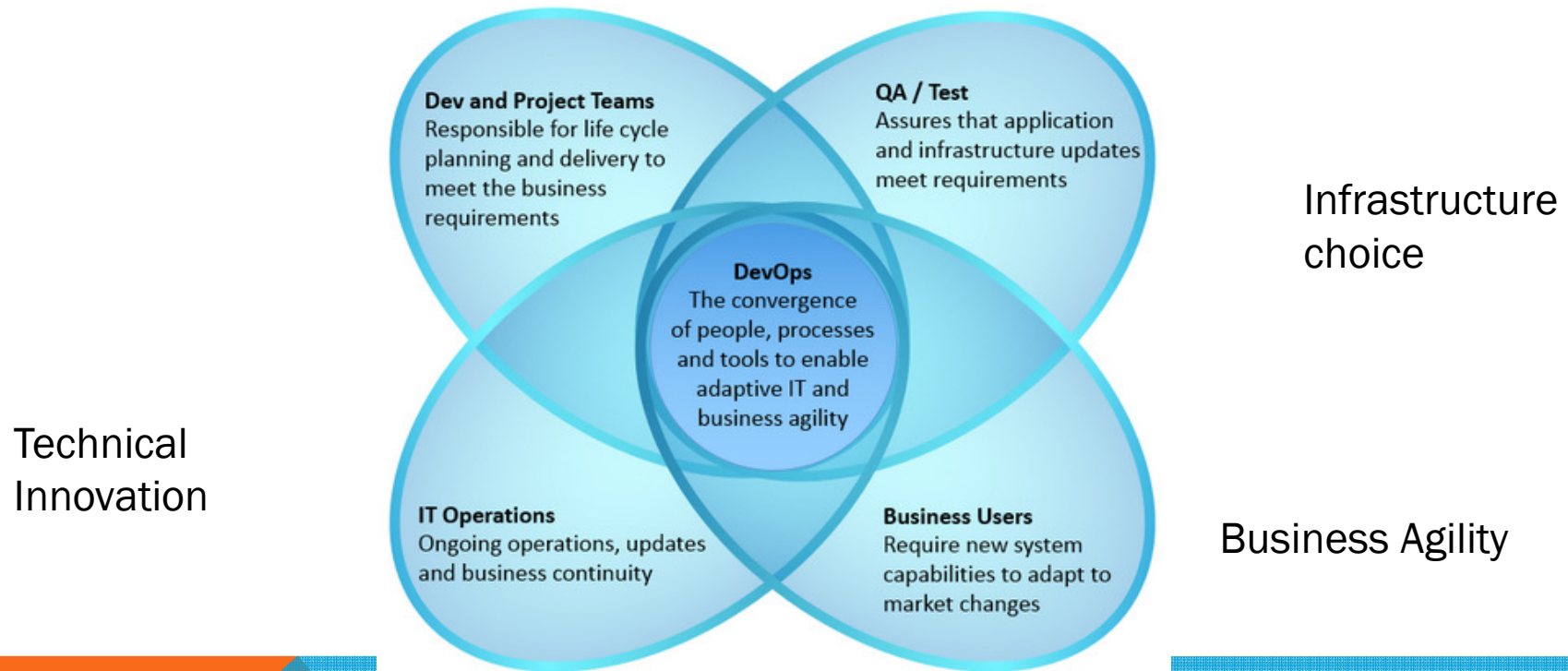
What is DevOps

Refresher Supplementary

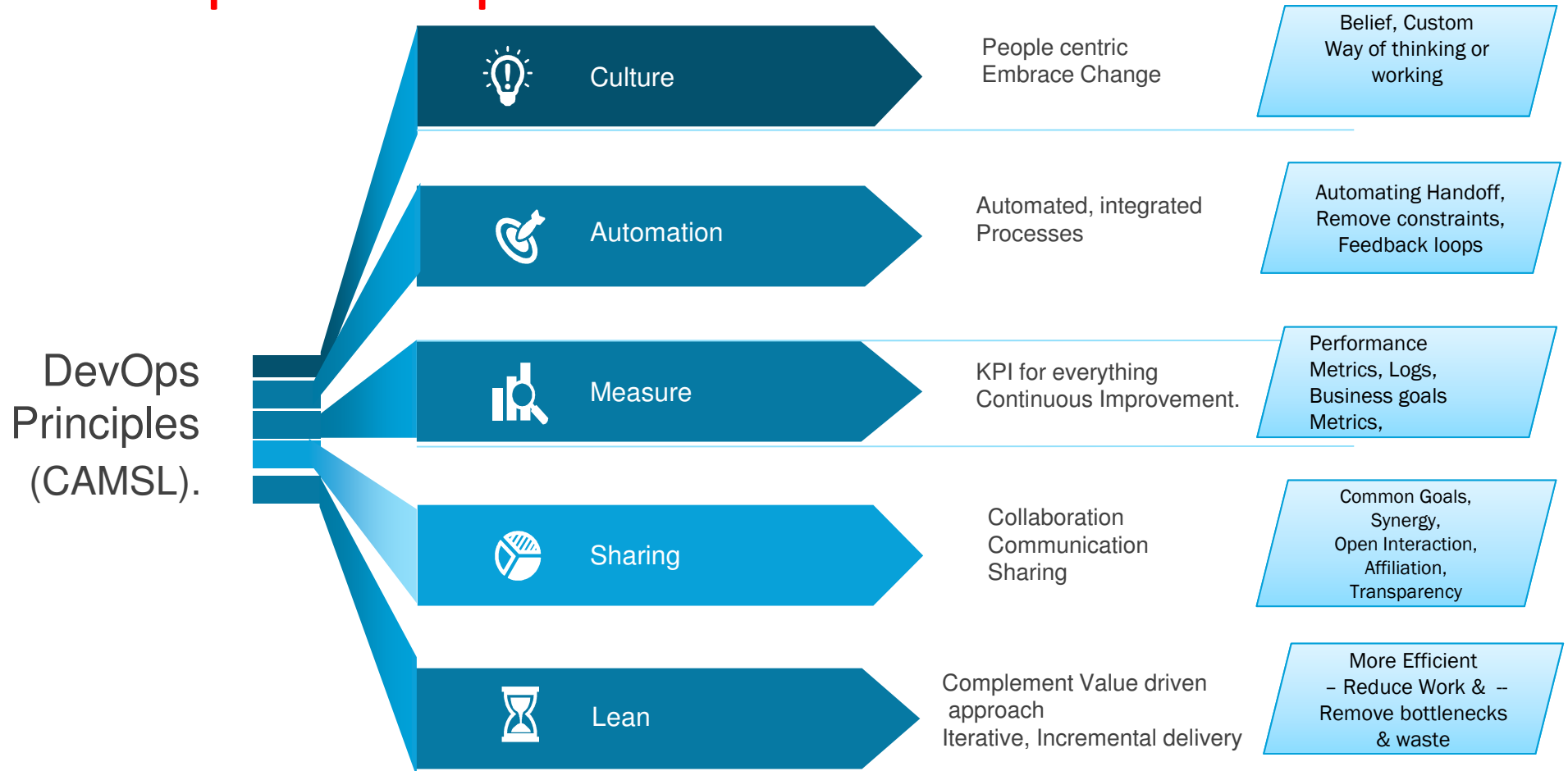
WHAT IS DEVOPS ?



DevOps is a blend of Culture with people, tools and processes keeping high focus on Collaboration, Communication and Integration among various stakeholders in the software development and delivery



DevOps Principles



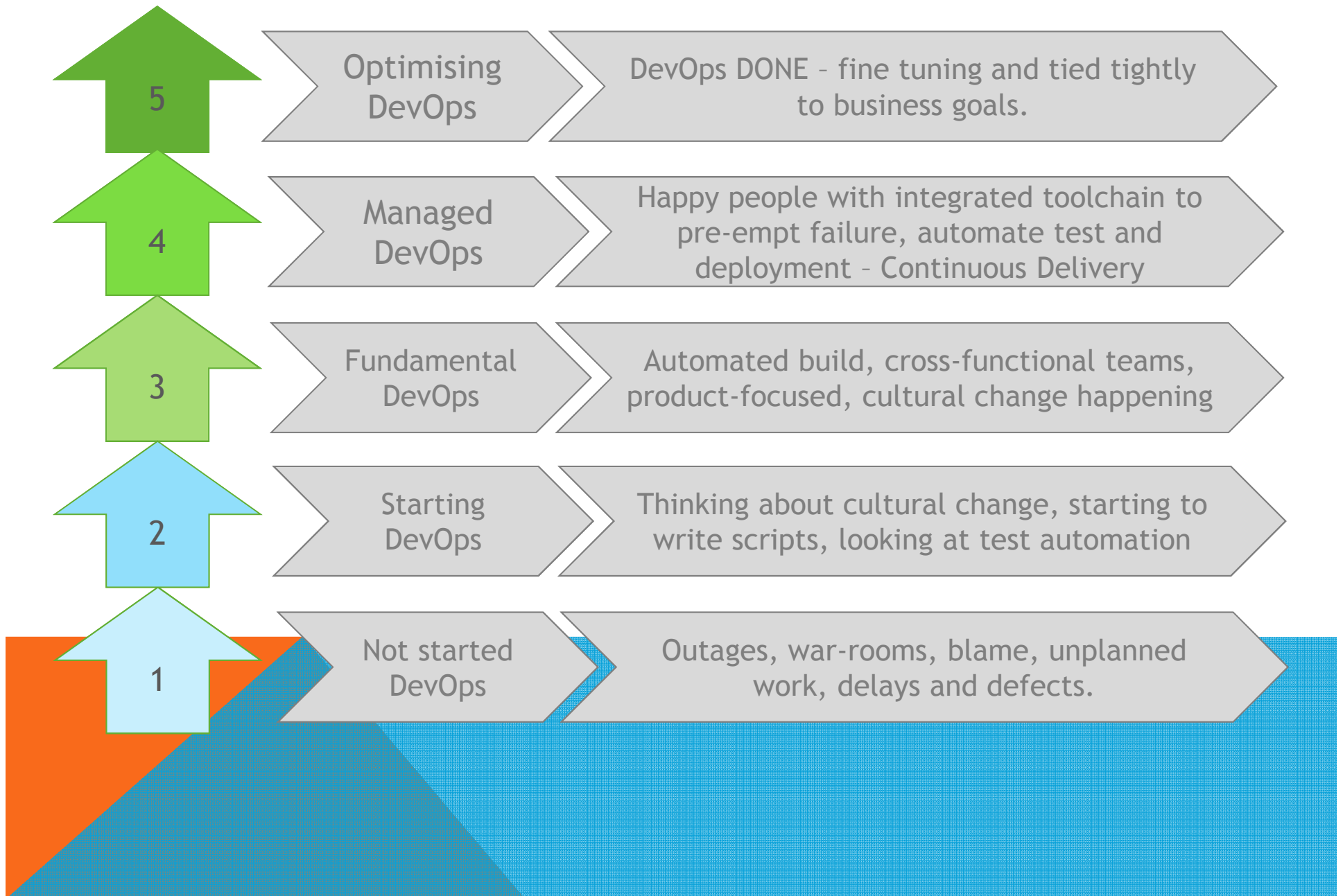
Requirements Management Maturity

Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
Undocumented	Written Requirements	Organized	Structured	Traced	Integrated
Requirements collected in conversations and not documented or shared.	Documented and shared, regular collaboration between teams, backup and restore enabled. No standardised format or shared repository.	Formatted consistently in an agreed way, stored and secured. Version tracked and easily accessible to those with rights in a single, shared repository.	Types (e.g. functional/non-functional) are specified. Attributes and prioritization is applied. Querying and filtering is possible. Using a recognised standard format such as a use case.	Determine and track requirements relationships, has a hierarchy of requirements: user needs, features and use cases. Coverage analysis reports implemented. Using a recognised format such as a user story and formalised acceptance criteria.	Requirements management process supported by a tool fully integrated with software development environment: used directly in design, development, change tracking, testing and PM.

Release and Deployment Management Maturity

Level 1	Level 2	Level 3	Level 4	Level 5
Manual	Packaged	Scripted	Complex	On Demand
Bespoke, unpractised process. Authorization and sign off incidental. Roll back via back up copy or scripts.	Some packages (e.g. MSIs) and scripts. Release authorization considered.	Multiple scripts allowing automation. Can deploy to multiple parts of route to live. Possible roll back through redeployment. Some version control.	Can deploy composite applications. Role based security available. Multiple platforms services through single interface.	Push button deployments when code is ready – continuous delivery and deployment achieved. Full auditability and compliance.

DevOps Maturity Levels



Gartner CD Maturity Scale

Practice	Build management and continuous integration	Environments and deployment	Release management and compliance	Testing	Data management	Configuration management
Level 3 - Optimizing: Focus on process improvement	Teams regularly meet to discuss integration problems and resolve them with automation, faster feedback, and better visibility.	All environments managed effectively. Provisioning fully automated. Virtualization used if applicable.	Operations and delivery teams regularly collaborate to manage risks and reduce cycle time.	Production rollbacks rare. Defects found and fixed immediately.	Release to release feedback loop of database performance and deployment process.	Regular validation that CM policy supports effective collaboration, rapid development, and auditable change management processes.
Level 2 - Quantitatively managed: Process measured and controlled	Build metrics gathered, made visible, and acted on. Builds are not left broken.	Orchestrated deployments managed. Release and rollback processes tested.	Environment and application health monitored and proactively managed. Cycle time monitored.	Quality metrics and trends tracked. Non functional requirements defined and measured.	Database upgrades and rollbacks tested with every deployment. Database performance monitored and optimized.	Developers check in to mainline at least once a day. Branching only used for releases.
Level 1 - Consistent: Automated processes applied across whole application lifecycle	Automated build and test cycle every time a change is committed. Dependencies managed. Re-use of scripts and tools.	Fully automated, self-service push-button process for deploying software. Same process to deploy to every environment.	Change management and approvals processes defined and enforced. Regulatory and compliance conditions met.	Automated unit and acceptance tests, the latter written with testers. Testing part of development process.	Database changes performed automatically as part of deployment process.	Libraries and dependencies managed. Version control usage policies determined by change management process.
Level 0 – Repeatable: Process documented and partly automated	Regular automated build and testing. Any build can be re-created from source control using automated process.	Automated deployment to some environments. Creation of new environments is cheap. All configuration externalized / versioned	Painful and infrequent, but reliable, releases. Limited traceability from requirements to release.	Automated tests written as part of story development.	Changes to databases done with automated scripts versioned with application.	Version control in use for everything required to recreate software: source code, configuration, build and deploy scripts, data migrations.
Level -1 – Regressive: processes unrepeatable, poorly controlled, and reactive	Manual processes for building software. No management of artifacts and reports.	Manual process for deploying software. Environment-specific binaries. Environments provisioned manually.	Infrequent and unreliable releases.	Manual testing after development.	Data migrations unversioned and performed manually.	Version control either not used, or check-ins happen infrequently.

Figure 15.1 *Maturity model*

DevOps Landscape



07/22/08