A No-BS look at moving your .NET apps to Azure

Brisbane Azure User Group - 13 April 2022

Codify Pty Ltd
Level 3/133 Leichhardt St
Spring Hill, QLD 4000 Australia

ACN: 074 168 148
ABN: 67 074 168 148

Introduction

1.1 About this document

Thanks for attending the Brisbane Azure Usergroup and we hope you found the content useful!

This document is an extract from one of our statement of work templates and presents a number of talking points to help you create a holistic view of your application's maturity and what it will take to get to the cloud.

1.2 How to use this document

The document is broken down into the following criteria:

- Business Ownership Criteria – These help you understand where your organisation sits with regard to the overall ownership of the application. Do you have SME's on hand who understand the business logic? Do you have appropriate support structures in place? What is the long-term outlook for the application?
- Technical Analysis Criteria – These help you talk through the technical quality and maturity of the application while also keeping one eye on the high-level issues that might make the adoption of platform-as-a-service offerings more challenging.
- Custom Application Criteria – Do you own the application source code, build and deploy it? These points explore how well you're positioned in terms of both the application's internal architectural plumbing and also the DevOps processes that manage the software development lifecycle.

Sit down as a team and talk through each of these points and see how your application measures up against the criteria in the document. You can create an action plan out of the undesirable artefacts that you identify.

1.2.1 Business Ownership Criteria

	Desirable Artefacts	**Undesirable Artefacts**
Internal Subject Matter Expert Has the customer clearly identified the internal subject matter expert?	▶ The customer has clearly identified the internal subject matter expert for application use cases. ▶ The customer provides this resource with dedicated time to act as the application SME.	▶ No resource is appointed.
Application Support Has the customer allocated internal resources or an external supplier for application support?	▶ The customer has a well developed support process for supporting application use cases.	▶ No resource is appointed.
Azure Resource Identification Have the Azure resources for the application been defined?	▶ There is a clearly defined consumption plan for the application. This should normally be accomplished as a part of Managed Azure onboarding.	▶ No Azure resource plan has been defined.
Strategic Outlook Has the application's strategic outlook been defined in terms of the Gartner six R model?	▶ There is a clearly defined future strategy for the application. ▶ Each proposed R option should specify both the plan and timeframe.	▶ No six R model has been defined.

1.2.2 Business Continuity and DevOps Maturity Criteria

	Desirable Artefacts	**Undesirable Artefacts**
Business Continuity Plan Has the customer clearly defined the business continuity plan for the application?	▶ A clearly defined backup and recovery plan exists. ▶ RPO and RTOs have been defined.	▶ No business continuity plan exists.
Change Management / Maintenance Has the customer formally defined a change management process?	▶ A change management and approval process exists. ▶ Maintenance windows are defined and agreed to by the application's users.	▶ No change management / maintenance process exists.

Release Management Has the customer defined a process for the release of new versions of the application as a regular occurrence?	▶ A release management process exists.	▶ No release management process exists.

1.2.3 Technical Analysis Criteria

	Desirable Artefacts	**Undesirable Artefacts**
Architectural Resilience / Regular modes of failure Does the application have regular modes of failure? Is the application's architecture generally resilient?	▶ High application reliability. ▶ No regular preventative maintenance (reboot to contain memory leaks) required. ▶ No culture of accepting recurrent failures.	▶ Regular or intermittent failures. ▶ Failures accepted as business as usual. ▶ No understanding of root causes.
Performance Characteristics Does the application perform acceptably for its user base?	▶ Acceptable performance for all users. ▶ Acceptable performance achieved within a reasonable cost envelope.	▶ Regular performance issues. ▶ Acceptable performance obtained through onerous infrastructure expenditure.
Monitoring Does the application have any run-state monitoring in place?	▶ The application is instrumented for elemental monitoring. ▶ The application is instrumented for deep inspection (AppInsights, etc) with searchable historical logs. ▶ Features with significant revenue impact are monitored with synthetic tests.	▶ No monitoring exists.
Thick Clients Is the application a thick client (in part of whole)?	▶ The application does not contain a thick client. ▶ No deployment to end user devices is required.	▶ The application is a thick client and required extensive testing, packaging and end user deployment as a part of release management.
Integration Interfaces Are the application's integration interfaces modern and maintainable (i.e. modern REST/JSON interfaces or messages dispatched by a service bus	▶ Modern integration patterns are followed. ▶ Interfaces use REST/JSON calling conventions.	▶ Integration is performed by loosely structured files such as CSVs

as opposed to CSV files exchanged via FTP).		▶ Integration is done via file drops or FTP copies.
Authentication Strategy What is the application's authentication strategy and how maintainable is it?	▶ The application uses existing corporate authentication mechanisms and controls.	▶ The application uses custom authentication mechanisms and controls.
User Lifecycle Management How are end user accounts created, modified and deleted.	▶ The application onboards users using existing corporate authentication mechanisms and controls.	▶ The application requires that end users are added to an internal database.
Business as usual manual operations activities What manual activities are required to keep the application operating correctly?	▶ The application is self-sustaining and does not require any manual operations, hacks or work around to continue operation.	▶ The application requires hand-holding to operate correctly. For example, a service bus might generate dead letters that require human intervention for resolution.
Environmental access and agility How readily accessible is the application's hosting environment.	▶ Codify staff may have direct access to all environments.	▶ No direct access to the application's environments is permitted.
Desired Environment Stages Have the desired environment stages (dev, test, prod etc) been defined and realised?	▶ The customer's vision for necessary environment stages is currently met and working.	▶ The customer's current vision for environment stages is not met.
Service Account Management Is there a process for the management of service accounts?	▶ The list of service accounts is defined and passwords are retained in a credential vault.	▶ Service accounts are unknown and/or credentials are lost.
Certificate Management Is there a process for the management of certificates?	▶ Certificates are stored in a credential vault. ▶ The corresponding private key is stored in a credential vault. ▶ The necessary secrets to decrypt the key pair are stored in a credential vault. ▶ The deployment of certificates is actively tracked.	▶ There is no comprehensive key management strategy.

	▶ The expiration date of certificates is actively tracked.	

1.2.4 Custom Application Criteria

	Desirable Artefacts	Undesirable Artefacts
Architectural Correctness Does the application follow any obvious antipatterns.	▶ The application architecture is sensible. ▶ Services run as services rather than desktop applications. The application is built with mainstream and well supported frameworks.	▶ 'Clunky' architectural artefacts such as needing to log into servers to start jobs. Use of esoteric frameworks or languages.
Legacy Frameworks Is the application significantly dependent on legacy frameworks? Examples include .NET WebForms, DotNetNuke, etc.	▶ The application uses modern frameworks that do not impose a burden greater than their benefit. ▶ The application uses frameworks that are no longer supported or is built in terms of an old version of a current framework that would require significant remedial activity uplift to the latest version.	▶ The application is heavily dependent on legacy frameworks and these present an obstacle to new builds.
Data Model Quality What evidence is there of quality in the underlying data model?	▶ The application uses mainstream database technology. ▶ Evidence of database normalisation. ▶ Use of natural keys. ▶ Foreign key relationships are enforced. ▶ The schema is naturally self-documenting and comprehensible.	▶ The application uses custom data files. ▶ The schema is opaque/difficult to understand either through manual design or by being an ORM repository.
Data Access Strategy What is the application's data access strategy?	▶ The application uses a highly deterministic data access strategy.	▶ The data access strategy is not deterministic due to the use of an ORM framework or similar.

Build Server / DevOps Tooling What are the current / desired hardware, software and cloud service requirements for mature DevOps practices. This criteria applies to custom applications only.	▶ A build server is available to perform DevOps functions against all of the application's environments. ▶ All necessary licenses are available.	▶ A build automation server is not permitted or allowed due to network access restrictions. ▶ The necessary tools and licenses are not available / cannot be acquired.
Capability for developmental changes How capable is capable is the customer of making changes to the application?	▶ Resources available in house or with a partner with a good working relationship. ▶ The source base is well understood and the customer or partner organisation is comfortable with making changes.	▶ No capability exists to make source level changes. ▶ No organisational knowledge of the application's inner workings exists.
Source Code Ownership and Access Does the customer legally own the source code for the application in order to be able to derive new works?	▶ The customer owns the source code in its entirety and has ready access to it.	▶ All or part of the source code is unavailable and no access to it can be negotiated.
Build Automation Are application builds automated? Does the build process provide success and failure metrics?	▶ Builds happen automatically on check in. ▶ The DevOps tooling provides reports on success and failure.	▶ Builds are not automated.
Build Unification Does the application build process produce a single build that is usable on all environment stages?	▶ A single build is created for all environments.	▶ Bespoke builds are required for each environment.
Build / Promotions across desired environment stages Does the build automation process allow for the promotion of a single build through all environment stages?	▶ The DevOps tooling provides a way to promote releases from one stage to the next. ▶ Appropriate workflow and approval processes exist.	▶ New builds are required for each stage. ▶ Deployments are manual.

CODIFY

Contact us

Codify Pty Ltd
Level 3/133 Leichhardt St
Spring Hill, QLD 4000 Australia

hello@codify.com
https://www.codify.com/

GPO Box 864
Brisbane, QLD 4001

ACN: 074 168 148
ABN: 67 074 168 148

This work is subject to copyright. Apart from any use permitted under the Copyright Act 1968, no part may be reproduced by any process without written permission from the publisher.