ICT Project Guidance

Supporting Systems to Consider  
For Assisting Technical Delivery Stakeholders

Version:

0.6

Author:

Sky Sigal, Solution Architect

## Description

This document outlines the capabilities required by technical delivery stakeholders to deliver to project delivery stakeholders’ expectations over the service’s full service-lifespan.

## Synopsis

To setup practices and resources for the full service-lifespan of a system, Technical Delivery Stakeholders require having access to provisioned services specific to the task of delivering iterative improvements to services over the full lifespan of the service made available to end users.

## Contents

[Description 1](#_Toc149038662)

[Synopsis 1](#_Toc149038663)

[Contents 2](#_Toc149038664)

[Introduction 3](#_Toc149038665)

[Background 3](#_Toc149038666)

[Services 5](#_Toc149038667)

[Versions Control Code Management Service 5](#_Toc149038668)

[Code Types to Consider Managing 6](#_Toc149038669)

[Technical Document Management Service 8](#_Toc149038670)

[Technical Work Item Management Service 9](#_Toc149038671)

[Work Item Synchronisation 9](#_Toc149038672)

[Compilation, Packaging and Deployment Pipeline Service 10](#_Toc149038673)

[Cloud Infrastructure Services 11](#_Toc149038674)

[Application Lifecycle Management Suites 12](#_Toc149038675)

[Conclusion 12](#_Toc149038676)

[Appendices 13](#_Toc149038677)

[Appendix A - Document Information 13](#_Toc149038678)

[Versions 13](#_Toc149038679)

[Images 13](#_Toc149038680)

[Tables 13](#_Toc149038681)

[References 13](#_Toc149038682)

[Review Distribution 14](#_Toc149038683)

[Audience 14](#_Toc149038684)

[Structure 14](#_Toc149038685)

[Diagrams 14](#_Toc149038686)

[Terms 14](#_Toc149038687)

## Introduction



Figure : Service Supporting Services (HL)

In a related document (*ICT Project Guidance – Supporting Systems to Consider*), systems were categorised as being one of the following groups:

* End User Stakeholder Support Services
* Service Delivery Stakeholder Support Services
* Information Delivery Stakeholder Support Services
* Technical Delivery Stakeholder Support Services
* Service Support Services

This document outlines services to consider for assisting Technical Delivery stakeholders in their deliver to expectations of the Service Delivery stakeholders.

## Background

Technical Delivery stakeholders include technical analysts, developers, system supporters, operators, maintainers, quality assurance and accreditor specialists.

Business stakeholders (SMEs, Business Analysts, etc.) are excluded[[1]](#footnote-2).

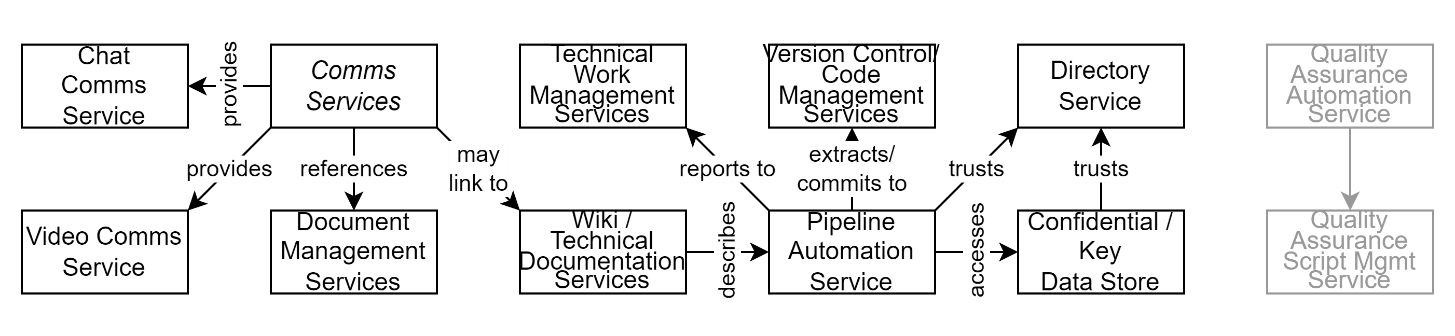
Technical delivery stakeholders require capabilities – and by extension, services and systems – that business stakeholders do not need.

Keeping the domains separate follows best practice[[2]](#footnote-3) permits each group of stakeholders to optimise their delivery while not being exposed to domains they are less familiar with.

Adherence to the principle of separation of concerns between the two groups and the services they use, means that integration effort is required to keep both groups synchronised.

# Services

While Delivery Stakeholder Services are focused outwards towards project stakeholders, Technical Delivery Support Services are focused inwards to isolate other stakeholders from the precision required to deliver technical services[[3]](#footnote-4).



## Versions Control Code Management Service

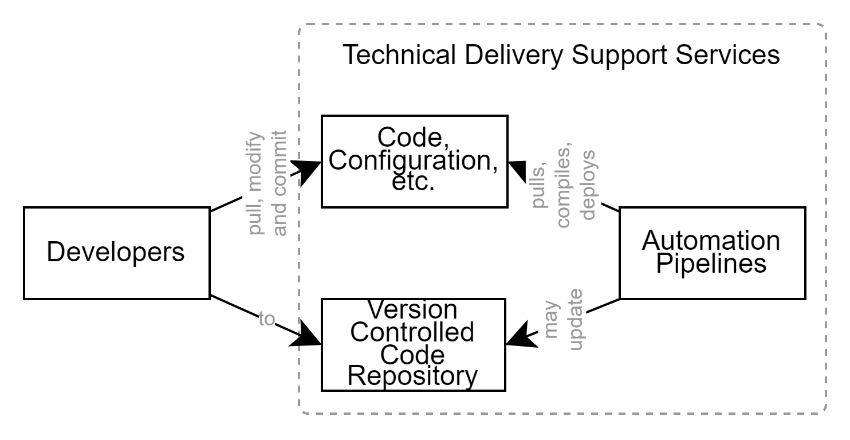


Figure : Version Control Code Management Service

The development of systems involves code.

To manage the code artefacts worked on by multiple technical specialists, a specialised service is used, called a version-controlled code repository that is primarily used to store and manage text (the code).

Note:  
while a versioned controlled code repository can also handle other forms of media (e.g., images, etc.) it is best practice is to not extend this to persisting compiled code packages.

The code repository technology most prevalent in the industry is the open source and distributed version control system called “Git”[[4]](#footnote-5).

### Code Types to Consider Managing

While it is easy to assume that the only code that requires developing and managing – and by extension, if one “buys before build” failure to deliver risk has been significantly reduced -- this assumption is very wrong, increasing the very same risk.

As outlined in more detail in *ICT Project Guidance – Code Types to Consider Managing*, delivering a system, and then managing it throughout its service lifespan, requires many forms of code.

For example, all the following, bar one, are common code deliverables, no matter whether buying or building the system itself:

* the deployment pipeline code,
* the infrastructure that it is deployed to needs to be created first into multiple environments,
* the database schema development instructions,
* the system code requires compilation into machine language,
* static testing of the code.

All of the above pieces of code are invoked before other pieces of code are used for packaging the system, deploying it, configuring its integrations to internal and external supporting services (data stores, identity providers, mail, cache, malware detection, rule engines, etc.), then provisioned with core reference data, before dynamic testing is done in-situ, before being provisioned with data backup restorations and/or legacy system data migrations.

Simply put, a poor understanding of the different types of code that require planning, designing, developing, implementing or delivering, puts a project at risk in the short term, and at risk of unnecessary costs over the long term, which in turn is used to justify why the iterative improvements, testing, and redeployments are avoided, putting the service at risk.

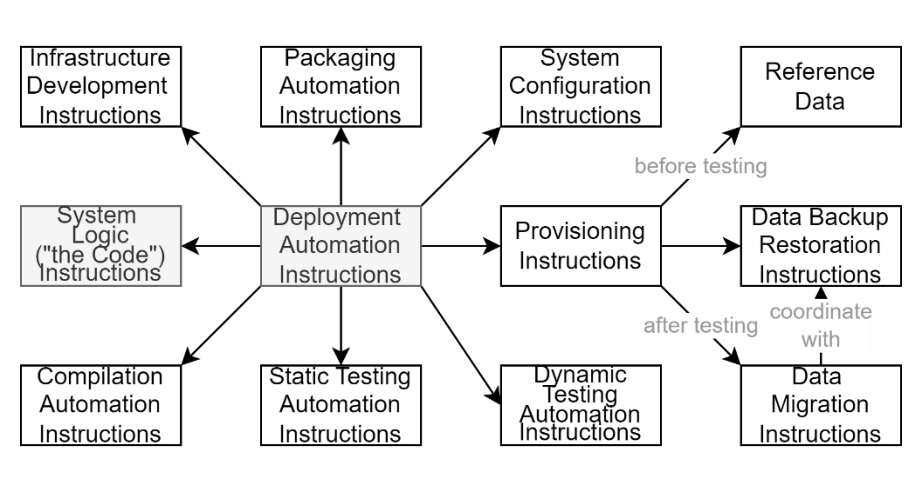


Figure : Code Types

Note:  
Distributed version code systems are by their very design useful, but insecure.   
Best practice is to protect them[[5]](#footnote-6) from users including Integration Credentials.

Wiki Information Store

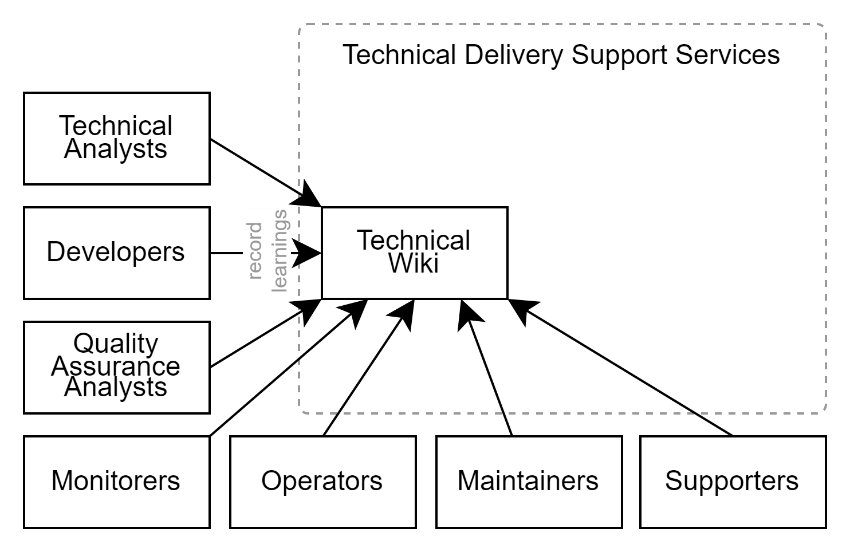


Figure : Technical Wiki Information Store

Wikis permit the development of linked resources that can be presented hierarchically.

While Service Delivery stakeholders may have a wiki for managing information they are developing, it is recommended to develop a second wiki – the Technical Delivery Wiki – to keep the two domains separate.

The reasons for this include:

* the kinds of information developed by developers is technical and distinct from the domain of delivery stakeholders,
* it permits sharing technical information with technical vendors and consultants without inadvertently sharing financial or other confidential information that may be in the service delivery stakeholder’s wiki.
* It permits using a different technology that that used by the service delivery stakeholder’s choice for wiki. For example, service delivery stakeholders may prefer Confluence due to its easy of use, whereas technical stakeholders may prefer Azure DevOps’ wiki due to its preference for Markdown and integration with other services in the Azure DevOps Application Lifecycle Management (ALM) suite of services.

Note:  
Well-known examples might include Confluence, Azure DevOps wiki, etc.

## Technical Document Management Service

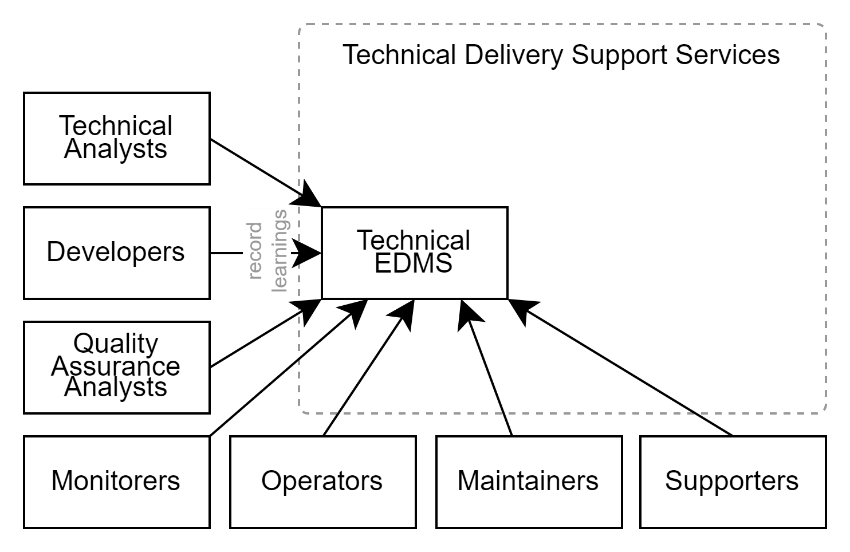


Figure : Technical EDMS

Technical Stakeholders develop documents. Examples include Stakeholder, System (Functional and Quality), and Transitional Requirements, Solution Architecture Descriptions (SADs), Application Support Guides, Delivery, Operations and Backup and Restoration Manuals, etc.

If these documents are for internal only it is possible to use a web-based solution (e.g.: the technical wiki) but if the documents are to be published externally, it is recommended that the documents created are branded and styled as per organisation expectations and managed via a service capable of performing full-lifecycle information management (e.g.: Word on SharePoint within O365).

The EDMS used can be separate from or the same EDMS as the one used by Service Delivery Stakeholders.

Note:  
If using two different EDMS systems, or areas, some processes are required to be developed to synchronize the sharing of some documents between the two different electronic document management services.

## Technical Work Item Management Service

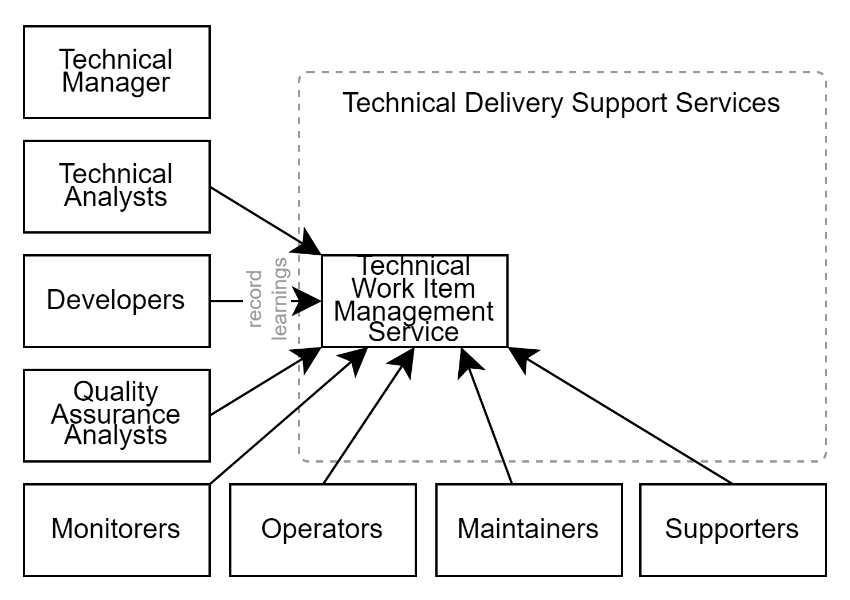


Figure : Technical Work Item Management Service

A technical work item management service is required to coordinate the efforts of technical delivery stakeholders.

The use of different Work Item Management Service than the one used by business service delivery stakeholders is recommended:

* Permits a different number of States
* Permits a simpler view for non-technical business stakeholders
* Permits working towards automation of state changes on the technical work item management service, while remaining mostly manual in the service used by service delivery stakeholders.
* Protects technical delivery from inadvertent disruption by service delivery stakeholders moving things around without understanding the potential impact.

Note:  
if there is still a desire to use a single system for both groups, ensure one develop distinct views for each party, hiding technical aspects from business stakeholders, it is recommended to protect end users and stakeholders from the complexity of delivering technical systems as well as the ability to disrupt work inadvertently.

### Work Item Synchronisation

Best practice is for the second (technical delivery stakeholders) system to update the status of work items managed with the first (service delivery stakeholders) system.

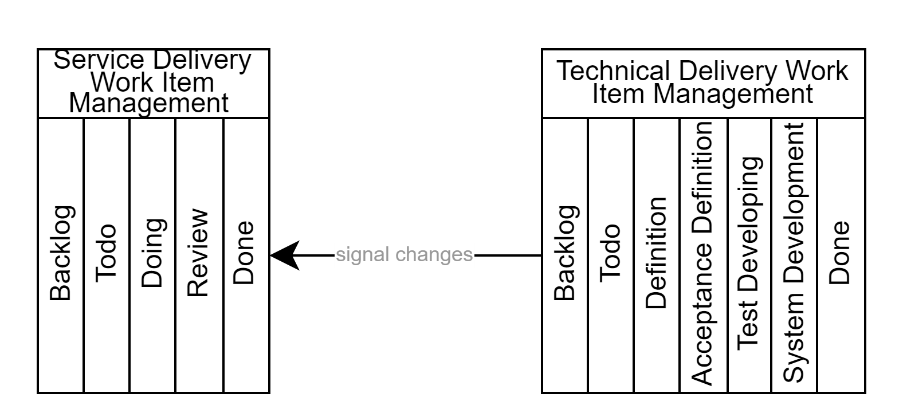


Figure : Synchronisation of Work Item Management Services

Note:  
Integration between two work item management services requires non-trivial specialised expertise, time and cost to be allocated starting from nearer the beginning of the project and ongoing from there. Hence -- in the majority of cases -- the integration is left as a ongoing manual exercise.

## Compilation, Packaging and Deployment Pipeline Service

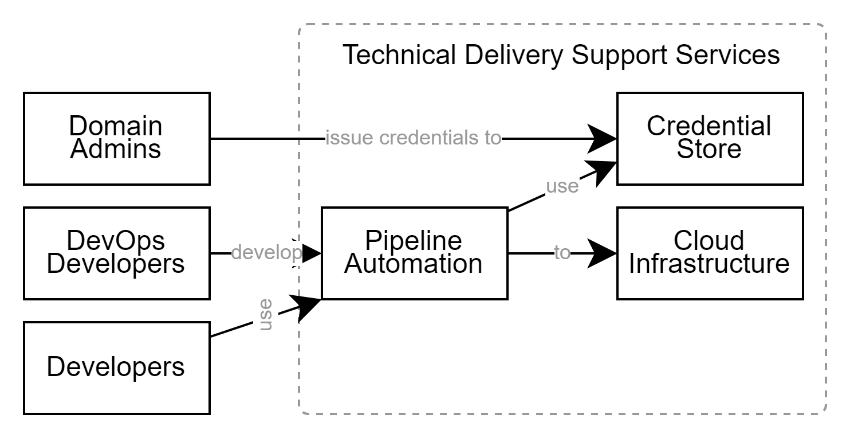


Figure : Pipeline Automation Services

Code is compiled, packaged deployed to target infrastructure environments, quality tested and hydrated with information using automation.

The instructions to do this are referred to as a “deployment pipeline”, which is run on a delivery automation service.

Integration is required between the delivery automation service, the source code repository, and target infrastructure.

Pipelines are used to orchestrate one or more common tasks listed below:

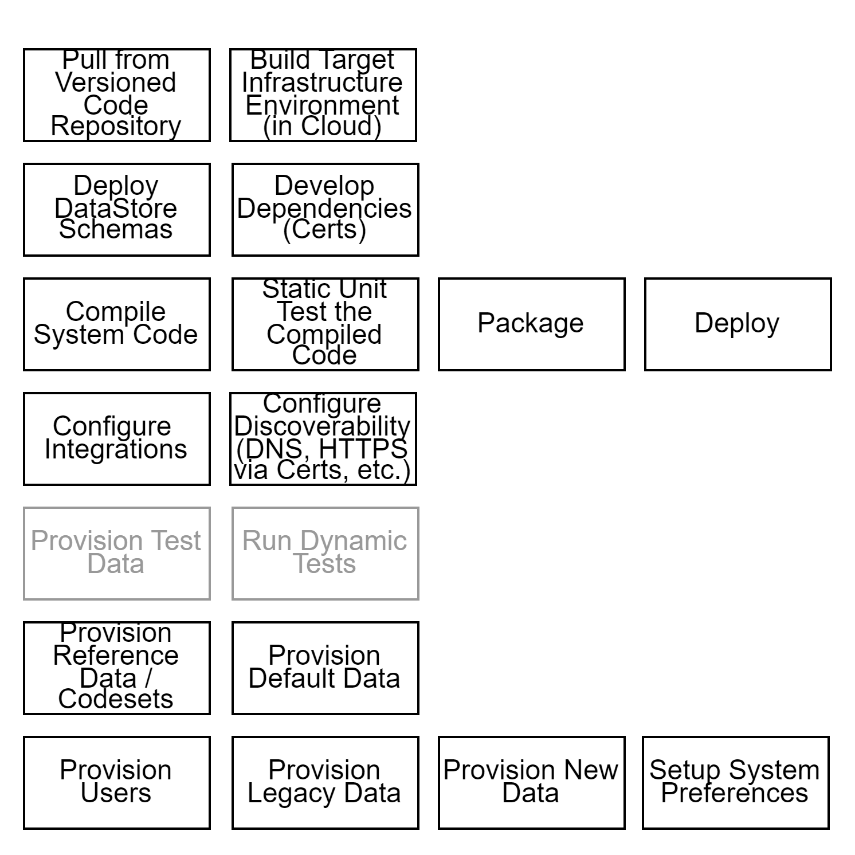


Figure : Common Pipeline Tasks

## Cloud Infrastructure Services

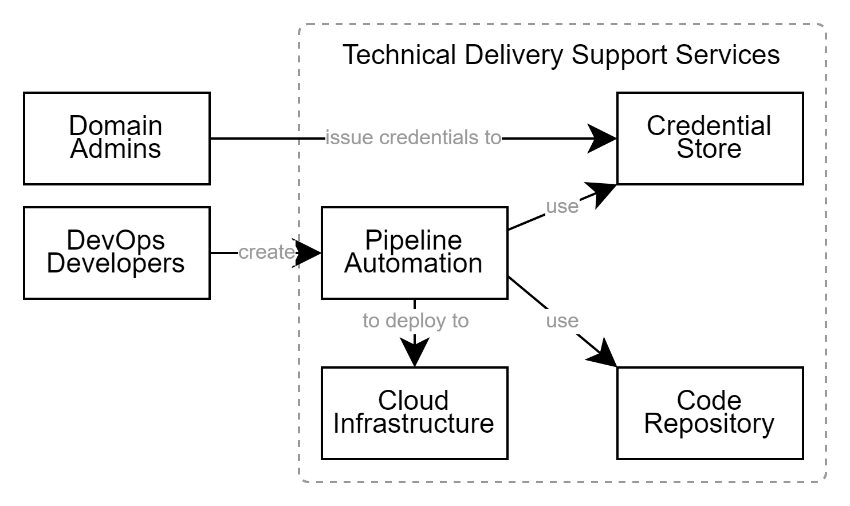


Figure : Cloud Infrastructure Services

Current best practice is to purchase SaaS based solutions, and failing that develop custom light PaaS services that are hosted “in the cloud”, on platforms that are managed by cloud infrastructure providers.

Infrastructure as Code (IaC) instructions are developed by DevOps developers to build multiple environments (BT, DT, ST, UT, PP, TR, PR, etc.) on cloud infrastructure.

## Application Lifecycle Management Suites

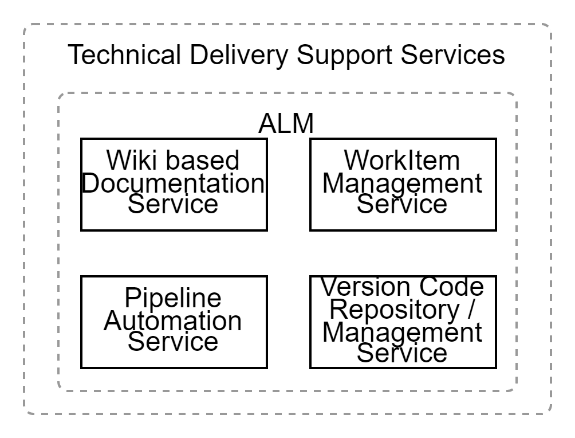


Figure : Application Lifecycle Management (ALM) Services

The integration between a code repository service, a pipeline service and work item management service -- all produced by different vendors -- is time consuming, while being needed right at the start of a project.

For this reason, there is a movement towards using Application Lifecycle Management Suites (ALMs) that have already done much of the work of integrating them.

ALMs generally provide some form of lightweight wiki-based documentation service, a code repository service, a pipeline automation service, and a work item management service focused on technical delivery.

Note:  
GitLab and Azure DevOps can all be classed as ALMs, and to some degree so can GitHub Enterprise.

## Conclusion

Being aware of the services that will be requested by technical delivery services permits project managers being better prepared to put them in place so that technical delivery services can deliver to service provider stakeholders faster and at higher quality.

Appendices

Appendix A - Document Information

### Versions

0.4: Initial Draft completed.

0.5: Added Issues/Request Management Service.

0.6: Added reference to ICT project Guidance – Code Types to Consider Managing

### Images

[Figure 1: Service Supporting Services (HL) 3](#_Toc148701720)

[Figure 2: Version Control Code Management Service 5](#_Toc148701721)

[Figure 3: Code Types 6](#_Toc148701722)

[Figure 4: Technical Wiki Information Store 7](#_Toc148701723)

[Figure 5: Technical EDMS 8](#_Toc148701724)

[Figure 6: Technical Work Item Management Service 9](#_Toc148701725)

[Figure 7: Synchronisation of Work Item Management Services 10](#_Toc148701726)

[Figure 8: Pipeline Automation Services 10](#_Toc148701727)

[Figure 9: Common Pipeline Tasks 11](#_Toc148701728)

[Figure 10: Cloud Infrastructure Services 11](#_Toc148701729)

[Figure 11: Application Lifecycle Management (ALM) Services 12](#_Toc148701730)

### Tables

### References

*ICT Project Guidance – Supporting Systems to Consider*

*ICT Project Guidance – Supporting Systems to Consider for User Stakeholders*

*ICT Project Guidance – Supporting Systems to Consider for Business Stakeholders*

*ICT Project Guidance – Code Types to Consider Managing*

**There are no sources in the current document.**

### Review Distribution

The document was distributed for review as below:

|  |  |
| --- | --- |
| Identity | Notes |
| Sandy Britain, Enterprise Architect |  |
| Matt Duguid, DevOps Specialist |  |
|  |  |

### Audience

The document is technical in nature, but parts are expected to be read and/or validated by a non-technical audience.

### Structure

Where possible, the document structure is guided by either ISO-\* standards or best practice.

### Diagrams

Diagrams are developed for a wide audience. Unless specifically for a technical audience, where the use of industry standard diagram types (ArchiMate, UML, C4), is appropriate, diagrams are developed as simple “box & line” monochrome diagrams.

### Terms

Refer to the project’s Glossary.

##### IT

: acronym for Information, using Technology to automate and facilitate its management.

##### ICT

: acronym for Information & Communication Technology, the domain of defining Information elements and using technology to automate their communication between entities. IT is a subset of ICT.

1. See ICT Project Guidance – Supporting Systems to Consider for Business Stakeholders. [↑](#footnote-ref-2)
2. Follows Separation of Concerns organisation principles [↑](#footnote-ref-3)
3. Consider Technical Services as ‘Factory Floor’ services. [↑](#footnote-ref-4)
4. The technology “git” is distinct from a well-known website called “GitHub” that is but one user of the technology. [↑](#footnote-ref-5)
5. Using branch check-in validation automation to, for example, pattern-scan code being committed. [↑](#footnote-ref-6)