**Configuration Management Review of Travis Perkins**

|  |  |
| --- | --- |
| Company/ Address: | Travis Perkins HQ Address |
| Workstream /Project: | Project Management |
| Document Reference: | Doc Ref |
| Role / Name: | Configuration Manager (Jamie Martin) |
| Address: | Salthouse Rd |
| Issue Date: |  |
| Authorisation: |  |
| Revision and Status | 0.1 (Draft) |

<Copyright data goes here>

**NATO UNCLASSIFIED**

Contents

[1 Purpose and Scope of the Document 5](#_Toc507408829)

[1.1 Scope 5](#_Toc507408830)

[1.2 Overview 5](#_Toc507408831)

[1.3 Maintenance of this Document 5](#_Toc507408832)

[1.4 Assumptions 6](#_Toc507408833)

[1.5 Definitions 6](#_Toc507408834)

[1.6 High Level Process 6](#_Toc507408835)

[1.7 Change Management 7](#_Toc507408836)

[1.8 Requirements 7](#_Toc507408837)

[1.8.1 Requirements Definition and Management 7](#_Toc507408838)

[2 Operational Configuration Management Review 7](#_Toc507408839)

[2.1 Service Asset Management and Configuration Management 7](#_Toc507408840)

[2.1.1 Licence Management 7](#_Toc507408841)

[2.1.2 Hardware Asset Management 8](#_Toc507408842)

[2.1.3 Client Build Management 8](#_Toc507408843)

[2.1.4 Configuration Management Database 9](#_Toc507408844)

[3 Development Facing Configuration Management 9](#_Toc507408845)

[3.1 Configuration Control Across Development Workstream 9](#_Toc507408846)

[3.1.1 Change Management 10](#_Toc507408847)

[3.2 Configuration Management 10](#_Toc507408848)

[3.2.1 Tools and Control 10](#_Toc507408849)

[3.2.2 Continuous Integration 11](#_Toc507408850)

[3.2.3 Environment Management 11](#_Toc507408851)

[3.2.4 Heritage Systems 11](#_Toc507408852)

[4 Project Configuration Management 12](#_Toc507408853)

[5 Recommendations and Gaps 12](#_Toc507408854)

[5.1 Recommendations 12](#_Toc507408855)

[5.1.1 Environmental Hosts 12](#_Toc507408856)

[5.1.2 Spreadsheet Driven Behaviour 13](#_Toc507408857)

[5.1.3 Greater interaction with Reihill house teams. 13](#_Toc507408858)

[5.1.4 Maintaining Service and Transition between R1 and R2 13](#_Toc507408859)

[5.2 Gaps 14](#_Toc507408860)

[5.2.1 Requirements Management 14](#_Toc507408861)

[5.2.2 Document Management 14](#_Toc507408862)

[5.2.3 Quality Assurance and Standards 15](#_Toc507408863)

[5.2.4 Centralised ‘Defect Management’. 15](#_Toc507408864)

Index of Figures and Tables

[Table 1.6-1 High Level Process 7](#_Toc507408865)

Document Control

Changes History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version Number | Date | Author | Change | Status (Draft / Authorised) |
| 0.1 | *20/02/2018* | *Jamie Martin* | Initial draft | Draft |
|  |  |  |  |  |

Authorised by

|  |  |  |
| --- | --- | --- |
| Name | Role | Date |
|  |  |  |
|  |  |  |

Distribution

|  |  |
| --- | --- |
| Status | Location |
| Electronic master (all other copies are uncontrolled) |  |
| Purchaser and Supplier personnel |  |
| Supplier personnel |  |

References

1. Concise Oxford English Dictionary, 11th edition
2. Managing Successful Projects with PRINCE2, 2009
3. ITIL Foundation Handbook, 2012
4. ITIL v3.0 Service Asset Configuration Management
5. ITIL v3.0 ITIL Guide to Incident Management
6. ITIL v3.0 ITIL Guide to Defect Management
7. ITIL v3.0 ITIL Guide to Release and Deployment Management
8. ITIL v3.0 ITIL Change Management

Definitions

| Term | Meaning |
| --- | --- |
|  |  |

Acronyms

|  |  |
| --- | --- |
| Acronym | Meaning |
| ITIL | Information Technology Infrastructure Library – A IT Standard |

# Purpose and Scope of the Document

This document will serve to present a ‘high level’ overview of aspects of configuration management (CM) disciplines, across the Travis Perkins project, specifically focusing on the Momentum project. It will not focus on specific concepts of configuration management but will deal with related disciplines outside of the definition of ITIL CM where necessary, or the author deems useful to express how application of knowledge and CM principles can be utilised to greater effect within the project.

## Scope

This document serves as the basis for recommendation to Travis Perkins Momentum project about how Configuration Management exists, and how it can be expanded across projects and workstreams to achieve a greater level of operational functionality and quality assurance. This will not specifically be limited to just configuration management in terms of control of configuration controls within development / release lifecycles, but also wider in terms of project management controls, milestone deliverables and project controls.

This document will assume that the reader is familiar with terminology relating to change and configuration management, such as Configuration items (CI), Configuration Management Systems (CMS), Configuration Management Databases (CMDB), Definitive Media Libraries (DML) as well as wider terminology such Service Integration and Management (SIAM), Development Operations (DevOPs), Continuous Development and Continuous Integration.

Throughout this document, CI will refer specifically to Configuration Items.

## Overview

Travis Perkins exists as a series of brands positioned within the supply of materials primarily associated with construction. The Momentum project aims to rationalise its business processes and IT capacity across its brands through a series of releases over the period of several years; moving primarily from a physical architectural and virtual machine-based system of aging bespoke, heritage applications towards a more rationalised estate of cloud-based applications.

At the centre of this rationalisation is the deployment of a third-party system from Infor, referred to as M3, which will phase out heritage applications and services over the life of the momentum project.

The first rollout will be Release 1, which is currently within the development lifecycle. This will affect the BSS Brand which is primarily focused on plumbing, pipeline and heating solutions into trade customers.

## Maintenance of this Document

This document will be presented as a report, rather than traditional document, produced on behalf of Te Kalliste Ltd, under sub-contract via HCL, to Travis Perkins. It will be issued to HCL, the primary subcontractor for initial review, as well as to the client, Travis Perkins.

## Assumptions

This document presumes basic principles of Configuration Management, drawn from ITIL definitions. It assumes familiarity with ITIL Foundations (3).

A distinction drawn will be drawn throughout this document in relation to development configuration management, operational configuration management and project configuration management (see section 1.5 below).

Each of these aspects of configuration management will be dealt within in their own section of the document.

It is also assumed that the reader is familiar with common IT Terms. However, these will be listed in the definitions below (prior to approval).

## Definitions

**Developmental Configuration Management** relates to the use of configuration management disciplines within a pre-production environment, specifically towards the production of a deployed and operational baseline.

**Operational Configuration Management** willrelate to the use of configuration management disciplines in the support and functionality of an operational, deployed baseline.

**Project Configuration Management** relates to the deployment of configuration control outside of traditional development asset, hardware and software lifecycles to the management, support and fulfilment of wider project deliverables defined by milestones.

## High Level Process

|  |  |  |
| --- | --- | --- |
| High Level Step | Description | Configuration Management |
| Define | Requirements Definition | Development |
| Design | Production of an approved design specification based on requirements | Development |
| Development | Production of applications based on design specifications | Development |
| Integration | The connection of interdependent developed applications into an end to end system to fulfil business requirements | Development |
| Verification | Testing Phases utilised to determine that requirements have been fulfilled. | Development |
| Release | The deployment of the integrated application suite, in line with deployment requirements. | Development and Operational |
| Live | Management and support of the live service to fulfil defined requirements. | Operational |
| Support | Ensure that service support requirements are maintained and achieved. This may trigger new development streams to maintain the existing service (such as patches, based on defined requirements for support). | Operational and trigger of Development |
| Redundancy | Ensure the removal of existing controlled services and applications does not affect existing requirements. | Operational |

Table 1.6-1 High Level Process

## Change Management

All functions of control relating to configuration items are dependent on effective change management strategies. As such, this document will review the change management process, as well as individual systems of change management utilised within the development project (such as defect management (DM) and technical change requests (TCR) as they apply to CIs. See section 3.1.1 below

## Requirements

### Requirements Definition and Management

There appears to be a lack of coherence over the management and availability of requirements within Momentum. There is no centralised ownership of requirements which should be reviewed as delivered at each stage in the service delivery lifecycle. Requirements define what is produced, whether its fit for purpose and define any effective test strategy.

Requirements should drive design, development, integration, verification + QA as well as service management in deployment.

# Operational Configuration Management Review

The following section of this document will overview the existing operation configuration management services functioning within Travis Perkins. These are primarily based in Reigate House at present, and are outside of the current Momentum auspice, as they relate to the ongoing Travis Perkins operational live estates and Brands.

## Service Asset Management and Configuration Management

Several individuals have been identified as working within areas commonly associated with operational management in terms of Service Asset Configuration Management (SACM). These are based within Reihill house for Travis Perkins.

It will be necessary for Momentum to engage with these teams to roll out Release 1 about any CI deployed into the managed estate.

### Licence Management

Nick Ingram. Service assets are rolled out with the Snow licence management application as part of their build. This software reports back on licence usage for Travis Perkins, on assets other than 2,500 thin clients, that are rebuilt daily (which would be problematic for determining licence management, and not cost effective).

The Snow client installed in each build reports back in real time details of licence usage for reporting purposes.

### Hardware Asset Management

Umesh Dave. Not currently interviewed, as not in at the time of discussion.

All assets deployed into the environment are asset tagged and recorded in the hardware asset database in LOGIT (ServiceNow). Hardware Asset Management is very closely aligned with the CMDB (see 2.1.4 below). Follow up will be scheduled.

### Client Build Management

David Golding within the Testing and Config team, fulfils a service configuration management role in the maintenance and support of client machines. Within this team process exists for control of the client estate using Microsoft Service Centre Configuration Manager, managed by Michael Kennedy, to secure builds of clients and deploy Group Policy Updates (GPO) across the client environment. These builds will be necessary for utilisation of testing within Momentum to fabricate clients for verification stages.

The existing process is robust and objective driven, with a good grasp on controlling and maintaining baseline builds across the estate.

This team have made excellent progress in the production of standardised builds and maintaining these builds in line with group policy updates – as well as mapping these updates across the builds.

#### Concerns

The naming convention utilised for clients and servers, prevents specific identification as to what the asset is being utilised for. Rationalisation of these to identify their role (developer, manager etc) would allow a greater control of GPO and build deployments based around specific functionality. This process of naming new client assets is outside of the team’s influence.

Limited resources. Due to the size of the team there are resource limitations on how quickly the team could respond to needs generated by Momentum and more specifically changes necessary for Momentum within the client estate. An example being AD Single Sign On, proposed by Momentum – as this conflicts with the Auto-Login that is deployed in the base roll out at client onboarding. This would require the generation of a new GPO change to remove autologin. To achieve such fulfilment, achieving this without impacting the existing responsibilities of the team would be challenging; as such its vital that this team is able to impact any changes affecting the client estate early to provide an accurate estimate of the turnaround of work.

Use of business machines for daily junk usage, such as where a developer will install applications for testing or general usage, onto a client build machine. This can create issues when it comes to automated deployment of builds and GPOs, if not suitably sanitised. Junk usage should be conducted either on a separated drive share, or a machine specifically designated for that purpose, that can be periodically rebuilt without the threat of loss to existing work.

Non-Client Machines within the environment. Individual assets, such as laptops sourced from outside of a corporate build are a risk to the control of the environment. In Momentum, there exists several personally owned devices that connect to the systems – this presents a potential risk to the existing estates.

#### Software Resources

During interview and discussion, David Golding raised a desire for the use of several specific applications that would benefit the management and deployment of GPOs.

* Flexaria Corporate Software Inspector. This application ‘patrols’ an estate and interrogates clients to discover existing vulnerabilities, dating software updates, prioritising vulnerabilities and produces of risk assessment (RAG) based prioritisation for resolution of vulnerabilities.
* Nomad. Currently deploying a build to a client utilising an image effectively rebuilds the machine entirely and affects bandwidth availability. Nomad rationalises the bandwidth impact greatly, and preserves the existing data and information on the client during a migration.

### Configuration Management Database

Pascal Cosnuau. The existing CMDB is based in LogIT (ServiceNow). This functionality is conducted by a discovery sweep run every five days, supported by imports for assets outside of discovery scope. Whilst several third-party assets are outside of full interrogation, this encompasses around 3m TP assets.

There are several limitations relating to the CMDB. Sweeps are conducted every five days this creates a problem for assets that are not connected to the estate at that specific time, most specifically laptops (about 60% of laptops are captured in a sweep). Additionally, some aged infrastructure is not suitable for discovery, and cannot be interrogated. Similarly, thin clients and some fixed build assets such as tills, are not captured within the CMDB.

Third party assets such as phones are subject to discovery but not interrogation, resulting in a limited knowledge base for the asset within the CMDB.

# Development Facing Configuration Management

This section of the document deals with aspects relating to configuration management, and development.

## Configuration Control Across Development Workstream

This section of the document will review the use of Configuration Management, Configuration Management Tools and Processes within the journey through to the production environment.

This is primarily implemented through a DevOps process utilising controlled builds, based on process of Continuous Development and Integration. With environments being refreshed under schedule every Monday and Wednesday, from three master configuration builds into which promoted CI changes are merged (and deployed); based on required schedules.

These customer builds represent what would be called environments under non-cloud-based utilisation, representing traditional development, integration and verification environments.

Each change to these customers is based on functional requirements, approved request for change and issue / defect resolutions.

There are four existing environments, which can be built to a series of specified configurations, based on the master build.

All change within the development and existing BAU systems are managed under the auspice of Change Management (Simon Doyle).

### Change Management

Simon Doyle – A robust and coherent Change Management team exists consisting of three people. Service Now is utilised as a change management tool, and processes are utilised to drive and manage all change within the Travis Perkins estate, including configuration changes. Whilst this has some connectivity and utilisation of the CMDB it is limited by the shortfall in how Discovery covers the estate (i.e. limited solely to connectivity to the asset, not the associated content of the asset).

Processes for change relating to Infor are ‘black box’ concepts, common to third party suppliers, in that change is initiated and deployed under change management but governed between these stages within Infor.

Processes for change management with third party suppliers, momentum and Infor are in development. Existing processes are flexible enough to allow tailoring of these requirements into the existing change management process.

CAB meets once a week, on a Thursday – However CAB is not required as a mechanism to approve change, only to escalate specific issues and discussion relating to changes.

#### Concerns

Decommissioning Workstream required – Due to the extensive cross brand integrations within the existing estate, all decommissioning must be tied into change request and approval processes to avoid risks to the functionality of BAU.

Degree of CMDB depth requires change management team to have deeper understanding of each change request than would normally be expected.

Gap – Release Management only extends its ownership into Pre-Prod. Momentum and TP demonstrate a lack of concept of how to maintain and support R1 BAU post cut over. This also extends to the period between R1 and R2. There needs to be planning about how to roll R1 out into the Brands in scope for R2, as well as the deployment of R2 for both R1 Scope (BSS) and R2 Brands without impacting existing service capability.

## Configuration Management

### Tools and Control

There is an emphasis on the utilisation of spreadsheets as methods of controlling information within the Momentum project (see 5.1.2 below), creating a reliance on manual systems which do not have a reasonable level of auditability. I would be surprised if this achieved or approached reasonable degrees of ISO, ITSM, ITIL or CMMI standards relating to management of configuration data or deliverables.

Tool use appears to be defined locally, rather than centrally, resulting in a non-rationalised approach to tools and licensing. This can create additional expense, where workstreams are utilising different tools that can achieve the same results. An example of this is issue / defect reporting – In which Infor, Integration, Testing are all using different approaches. Rationalising this to a common tools approach creates a stronger negotiating point for licences, reduced support costs and a lack of need to integrate separate tools with each other. It also allows rationalising of support and reporting capabilities into single points, rather than having reporting located within each workstream.

### Continuous Integration

The DevOps solution provides the basis of continuous integration on refreshed environments. This utilises a range of ‘usual candidate tools’ such as SVN for Source Control, Git Hub, Sonatype Nexus etc allowing refreshed environments to be stood up. Environment stand up includes smoke testing.

This is managed under Change Management.

### Environment Management

Mike Beckett – An environments team exists for Universe. This needs to be utilised by Momentum. The environments team manage the cloud tenants. All environments are tracked status wise via ServiceNow and built through the automated Jenkins process utilised across the DevOps solution.

Prior to release to team usage all environments are smoke tested according to the customer defined integrations. Defects discovered in Environments are reported in HP ALM.

Environment status tracking tool exists for Universe, developed by Mike Beckett.

Environment provision and builds are managed under Change Management.

#### Concerns

Reliance on Infor for set up of users for environments – Infor should not be involved in the set-up of environments other than their usage in terms of development and integration. This is a result of a lack of sharing of documentation and training from the Infor team of knowledges required by the environments team.

Reliance on Spreadsheets – A heavy dependency, seen across Momentum.

### Heritage Systems

Rosemary Bell - Heritage system exists as a series of bespoke applications primarily developed in house. These have been deployed into the universe system, under change control procedures. Existing as many application packages connecting to a single database.

Change Management is organised by Simon Doyle, utilising a system defined on SharePoint.

A system of Continuous Integration is in place, and a release tool exists that is a bespoke application. Back up exists as a mirror copy of the database and system.

Exists on physical and virtual machines.

System is not under configuration control via a CMS in the typical sense (management of individual application is the result of a legacy of the IT.

#### Concerns

Lack of Infor Version control and environment limits creates a scenario in which defects discovered in test would potentially be deployed without going through the test lifecycle.

Concern raised that the Infor solution does not contain capability to deliver the same kind of flexibility regarding sharing across brands that occurs within Universe. Example is tool hire, within universe the return of hired equipment can be delivered within brand (i.e. Something hired in BSS can be returned to most TP brand sites). How is that system being reproduced within M3.

Continuity of Universe / Heritage and maintenance of service given that existing systems will need to change, that are outside of R1 Scope. Such changes will need to be reviewed in relation to the impact on the M3 Solution of R1.

Shared fulfilment – Customer fulfilment of orders is drawn across brands – how does this function within M3, especially if one of the brands is in scope and one isn’t – Is the design solution sufficient to deliver this capability.

Decommissioning Workstream required. Due to the historical cross pollination of brands and services and the historical usage of sharing across brands, decommissioning of an Application or Service or Brand requires investigation.

Lack of third party knowledge and publication from Infor. Supporting M3 without specialist Infor knowledge being rolled out into TP IT staff. Shared Knowledge Base doesn’t exist.

# Project Configuration Management

Project Configuration management represents treating specific project milestone deliverables as Configuration Items, treating the approach to completing key project stages as being defined by set requirements for entry, and deliverables that must be produced for exit from that stage. For example, to enter System Integration Testing, then criteria should be fulfilled for System Integration Testing to begin that is not based on the technical infrastructure and provision of customers on the cloud tenant. An example might be that a System Integration Test Strategy has been approved, Test Scenarios agreed, Test Scripts documented, that testers have been arranged to the schedule and so on.

By establishing a series of ‘review gates’ for entry and exit to a milestone should also be utilised to determine progress on deliverables for later gates has progressed to an approved level (for example during the SIT gate, it might be a requirement that planning for UAT has begun).

Failure to achieve an entry or exit criteria should either result in the determination of actions that either need to be completed (caveats) which will be reviewed at the next ‘gate’ or a return to activities within that milestone (for example, criteria for exiting System Integration Testing might that no critical or major defects are present. The inclusion of such, would require that those defects be resolved, retested and regression tested prior to exit being permitted).

By establishing a series of ‘review gates’ for entry and exit to a milestone should also be utilised to determine progress on deliverables for later gates has progressed to an approved level (for example during the SIT gate, it might be a requirement that planning for UAT has begun). Typically, these should be based around requirements drawn down from activities that must be in place for ‘Cut Over’ to pass its go / no go.

Evidence should be demonstrated and placed under configuration control for each deliverable forming a baseline, which then change control protocols are required to affect new revisions (ensuring that a change to an approved process or procedure or report etc, is shared to all parties).

# Recommendations and Gaps

## Recommendations

### Environmental Hosts

There are insufficient hosts / tenants to support Release 1, once it completes system integration testing (as the SIT tenant will be repurposed for UAT). In, order to accurately test the resolution of defects from SIT onwards, will either require them being deployed without System Integration Testing (presenting a major risk to UAT testing), or the tenant being maintained for SIT, impacting project time scales.

Expectations around the support period necessary for Release 1 may be unrealistic, as most systems tend to require more than a ‘three months’ support and patching period.

Whilst this shortage can be mitigated to some extent during the development of Release 1 (as the Prod environment will not be in use, until Pre-Prod is cut over) it will be a ‘very significant challenge’ in terms of supporting Release 1 once it enters live service, when there will be pressures for environments from both the M3 and Universe sources.

It isn’t reasonable to freeze all change for Universe / Heritage systems for the time periods involved in the Momentum lifecycle (as its likely that changes will be required for Brands outside of BSS, at roughly the usual scale and waiting for M3 may not be reasonable for business needs).

### Spreadsheet Driven Behaviour

Across Momentum there seems to be a reliance on the use of Spreadsheets as a means of tracking projects and reporting rather than real time data driven project management tools and reporting. This generates a problem in terms of communication and a reliance on focal points of knowledge. Given the size of the TP IT in terms of staff, this produces a lot of different ‘versions of the truth’, at least as it relates to a specific team.

Use of tools such as Jira would reduce this reliance, as well as providing a basis for real time reporting and dashboards across the project. Whilst designed specifically for Agile Development and issue tracking, Jira can become a very effective tool for communication any lifecycle-based process within a project and deliver real time control over task assignment and communication.

Within the development community the use of Jira is widespread and has been utilised as a means of communicating between teams, and across teams. The additional use of Confluence with Jira to build Wiki’s has provided a much more effective means of knowledge-based sharing than the use of a tool like googledocs or SharePoint can provide.

Modelling any process within Jira would provide a far more robust, and easy to use system for tracking activities and project status, that is immediately available to anyone. Furthermore, adoption of such a system would also present a more efficient basis on which centralised reporting throughout the programme could be driven.

This can be achieved either through the creation of ‘one source’ dashboard reports, or individual or shared filter sets that can drive online queries, or automated excel based reporting (requiring that the report need only be defined once).

### Greater interaction with Reihill house teams.

TP has a lot of talent across Momentum, but there seems to be a tendency for decision making to be based more exclusively within Momentum, without close regard to staff outside of its more direct management (such as the teams based in Reihill house). Whilst the involvement of these teams will be less significant in terms of the development of Release 1 in terms of deploying, monitoring and support of the estate these teams will be essential and vital.

### Maintaining Service and Transition between R1 and R2

None of the discussion or workshops have dealt with, or really mentioned how R1 for BSS will be maintained once live, who will support it and how (including how knowledge necessary for the support teams will be shared). In terms of the planning for R2, there doesn’t seem to be reasonable regard paid to how R1 will become available to the R2 Scope Brands or how R2 will become available to BSS (from R1 Scope). There is a considerable scope of work between R1 and R2 that needs to be considered, especially as during the period of R2 there should be an expectation that the R1 Baseline will change due to BAU requirements.

#### Decommissioning.

Throughout the transition period of Momentum, Heritage applications, hardware, software and assets will need to be decommissioned. As there is no direct replacement in M3 for Heritage, and a continued basis of need relying and cross brand pollination of data and applications, on the Universe system, that process is more complicated that simply stating that’s a service replaced by M3 can simply be retired. Rather each act of decommissioning needs to ascertain the impact on BAU within the non-scoped estates as well as within the scoped estates needs to occur.

This should be considered as a workstream and will need to occur within a very tight frame of change control. Failure to do so, presents a major risk to BAU.

#### Metric Driven Success

Definitions of success of a project should be defined in terms of recordable metrics (as part of the requirements). As far as the author is aware, definitions of success are defined in terms of subjective statements. Requirements and objectives need to be defined as more specific a term as possible.

## Gaps

### Requirements Management

There is a distinct gap in terms of how requirements are managed and shared across Momentum. It becomes a challenge to attempt to define any strategy without an explicit basis in requirements definition. The High-Level Process in 1.61.6 above, outlines how all processes and stages of a lifecycle are driven by requirements at every stage.

There should be, for every program, at its heart a catalogue of Requirements, that have been approved and clearly defined, that is managed under change and configuration control. This should form the basis for every process, strategy and plan defined by Momentum, and its third party.

### Document Management

Systems of document management within momentum seem to be very immature and not clearly defined. The basis of this seems to be around sharing documents either via Googledocs or through emailing documents between teams. Formalised control and production of a central document library (or a tool like Confluence) can remove much of the dependency on searching specifically for documents. A central, well-structured library of documentation, under version control, prevents the necessity for sharing documents and storing them locally. Systems where finding documents can be difficult, tends to result in behaviours where documents are stored locally by individuals, resulting in conflicts when new versions are issued. In an ideal world, these should only be published PDFs with the document control system and document manager maintaining source documents, under revision control.

The project does tend to share google links to specific documents with invitations, which mitigates the risk of people storing documents locally.

### Quality Assurance and Standards

There is no centralised responsibility for enforcing, auditing and maintaining of Quality standards within the Momentum Project, or within Travis Perkins. It is assumed that the project operates to specified standards and methodologies, but there is no evidence that has been presentable of what these standards are, or the degree to which they are conformed to. Usually a project will have a defined ISO standard, and a set of stated methodologies that will be utilised in addition to corporate policies.

Without Quality Assurance, it is impossible to accurately determine whether Momentum is compliant or non-compliant, or even in breach, either on a workstream basis or wider.

It has not been possible to determine if development standards are being measured in terms of quality either, such as applications such as SonarQube (which measures quality of code) etc.

### Centralised ‘Defect Management’.

Whilst a defect manager exists within Momentum, this role is specifically located regarding Test, with issues and bugs being managed within specific workstreams. By centralising this function, the capacity to accurately identify the impact of a bug or issue, prior to test is restricted to how the discovering workstream defines it. A minor severity bug to integration or development, could have catastrophic consequences in terms of System Integration Testing.

Centralising all issue, bug and defect reporting into a single tool, with a set manager, can rationalise this into a more efficient system of control and communication. This role isn’t specifically interested in the technical aspects by which an issue or defect is fixed, but in the prioritisation, control, management, communication and planning around defects through their lifecycle.

Whilst this is less significant in terms of a project in its development phase, it becomes more important as a project approaches and enters live service and support. It also functions to produce a greater knowledge of the state of a release. In terms of supporting Release 1 the capacity to track defects, known issues, bugs, vulnerabilities that need fixing across a project wide basis is more advantageous through a single source, than within four or five different workstreams.

Notably as well, when R2 is live, there should be an expectation of support being required for those Release 2 Scoped Brands in terms of Release 1.