Non - Functional Requirements

	Jira Story	Definition of Done	Story Owner	Priority	Implemenation
Product Management	L4D/Service Description	The documentation must include: - Description of service Basics and options - SLA's and KPI's - Costing and price model - Consumption model (including ITSM)	Product Manager	1. Critical	Action: To be discussed with Product Management on how to track progress of the stories
	Business Case	For a service it is needed to check the full business case, including how it was built, rational, assumptions, costing and funding.		1. Critical	
	Ready to Sell	Ready to sell announcement published.		1. Critical	
	Service Termination Plan	If the new service is replacing another service the termination plan of old service must be available.		2. Medium	
	Risk Register	Must include: - List of risks identified during development that might have an impact on the delivery of the service for the customer. - Mitigation plan - Action Owners estimated time for mitigation		1. Critical	
	Prepare Costing	Product Manager to provide: - Quotation, list of cost divers, activities/KPI's and assumptions based on which MAIA calculations will be created; - Mapping between all cost drivers and WU's; - CES Management approval.		1. Critical	
	Customer User Manual (e.g. portal, presentation)	Service Customer User Manual (pr on line guide) must be available and presented, including a description on how to use the portal and the service specific SSR's.		2. Medium	
	Security and Compliance Integration	The Security Functional requirements to be included in AHA. (Product Management)			
Finance	Creation of the MAIA model	- Creation of the Maia Model - Validated outocmes by the Product Manager; - Cost levels split into HW/SW, FTE and hosting.	Financial Architect	1. Critical	Action: To be discussed with Finance on how to track progress of the stories
	Source and procedures for volume measurement (metering file from infrasructure)	- Volume measurement (volume metric collection) based on all cost drivers; - Daily upload to CloudDB.	Technical Lead Architect	1. Critical	
	Nessie Setup (Global WU ID's and corresponded Activity types)	- Creation of the necessary definitions in FIT and Nessie based on the MAIA input; - Global FIT template to be filled in by BOS team for the WU creation; - Creation of WU's in Nessie by Global FIT team; - Activity types and the secondary cost elements to be driven by Global Finance.	Global FIT and Finance	1. Critical	
	Mapping volume measurement to WU's	- Creation of the WU instruction document; - Once the volume metering collection is in place present a clear mapping between the meeting ile and WU's.	Technical Lead Architect and Financial Architect	1. Critical	
	FIT (csv) created	Note: FIT file will be created once the MAIA model is fully approved. - For the services using ChargeDB: Define WU's and input files mapping in ChargeDB; As a result FIT file should be presented as evidence (from CAT environment). This is an automatic process and is done once a month. (it should be done during TOS in the test environment) - For the services NOT using ChargeDB. The template for FIT should be presented as evidence. This is manual process.	Financial Architect	1. Critical	
	Input file for FIT (csv) is created	Point applicable only for the services NOT using ChargeDB. The template for FIT should be presented as evidence. This is manual process.	Technical Lead Architect	1. Critical	
	KPI's for Productivity and Utilization	Clear list of KPI's and the implementation in relation with the assumptions made in the MAIA model.	Technical Lead Architect	2. Medium	
	WU Cost rates	Based on the Wus cost rates should be calculated amd approved for all RBU and GDC.	Financial Architect	1. Critical	
Operations Process Compliancy	Define Development Methodology	Properly documented development methodology in line with company requirements and objectives.	Technical Lead Architect, Operations Team	1. Critical	
	Organizational Setup	The organizational setup must have the following acceptance criteria: - Right skills embedded in the team; - Trainines and formation activities for the team members.	Business Owner, Product Owner, Service Responsible Manager	2. Medium	
	High Level Designs	High Level Designs will document the product and service designs: - HLD Technology describing the product setup, including Security; - HLD Service describing the service setup (ITSM).	Technical Lead Architect	1. Critical	
	Configure Version Control System (VCS)	Version control system (VCS) must be in place with the proper configuration settings enabled. All code changes must be logged (who, what and when) and the log data must be maintained for a long period of time.	Technical Lead Architect, Operations Team	1. Critical	
	Define Branch Policy	Branch policy must be documented and followed.	Technical Lead Architect, Operations Team	1. Critical	
	Define and verify coding standards	- Documented coding standards which contains guidelines for the use of a programming language, programming style, practices and methods. - The code should be automatically tested for adherence to the defined coding standards.	Technical Lead Architect, Operations Team	1. Critical	
	Define test approach for unit and component testing	- Documented test approach for unit and component testing including the requirements on test coverage and required level of minimal test coverage per code module.		1. Critical	
	Perform unit and component testing	- Perform unit and component testing and log the results in the VCS.	Technical Lead Architect, Operations Team	1. Critical	
	Define code peer review guideline	- Documented peer review guideline describing the required testing, documentation of findings, merging process and VCS logging.		1. Critical	
	Peer review is performed and findings are documented	- A peer review is performed and documentation of the findings is realized. The merge action is performed after successfully passing the peer review.	Technical Lead Architect, Operations Team	1. Critical	
	Test approach for infrastructure and application testing	Test (if possible automatically) infrastructure and application code. Validate the rules set by the team on failing the build, based on documented minimal requirements	Technical Lead Architect, Operations Team	1. Critical	
	Security and Compliance Integration	NonFunctional Security Requirements to be provided in advance by John Janse. Summary and evidence that all NFR's for Security & Compliance are integrated to be provided (provide complete Security & Compliance checklist).	Technical Lead Architect, Operations Team	1. Critical	
	Vulnerability scanning	The following automated vulnerability scans should be performed (not all of it is common practice yet): - software vulnerability scanning; - third-party (open-source) component/library scanning for known vulnerabilities and licensing issues; - code dependency scanning for (weak) dependencies; - operating system baseline scanning; - static code analysis (conformance to defined rulesets and security testing)	Technical Lead Architect, Operations Team	1. Critical	
	Technical Security Specification test	Provide the testplan which includes relevant security tests and result. This must include (automated) testing and also testing of NFR Security & Compliance requirements.	Technical Lead Architect, Operations Team	1. Critical	
	Boarding New Customer Runbook	The Overall onboarding runbook including the SNOW onboarding must be available and tested.	Technical Lead Architect	1. Critical	
	Define Integration test plans (tests, methods, frequency, tools)	The delivery process is started by commencing a set of tests to be run on the whole code base on a production like evironment. An integration test plan must be created specifying which tests, test methods,frequency and test tools apply for the given change, including the resolution method to apply. Where applicable a generic test plan related to the complete solution may apply instead of a test plan per change.	Technical Lead Architect, Operations Team	1. Critical	
	Setup Test Environment (representing production)	Create a test environment that is commensurate with the enterprise environment (i.e., production-like). However, to comply with privacy laws and regulations appropriate rules should be established for test data that comprises senstive data, e.g. rules that specify for which types of personal data the test data sets should be anonymized (de-identified).	Technical Lead Architect, Operations Team	1. Critical	
	Perform testing in the test environment (representing production): - unit testing - component integration testing - acceptance testing (feature testing) - E2E testing - ITSM integration testing	A register or log is maintained for test findings that need to be resolved. Tracking is performed in such a way that team members can easily follow the resolution of these findings to ensure safe delivery. Acceptance of test findings that need to be resolved is discussed within the team. When resolution is not possible in the short term, acceptance of the test finding and moving to production without resolution has to be done by the product owner, where needed in consultation with affected stakeholders.	Technical Lead Architect, Operations Team	1. Critical	
	Perform Security scanning	Based on the identified test approach, proper security scans on the finished code (static code test) are performed in a timely manner (e.g. vulnerability scanning, code dependency, penetration testing). Exceptions are documented, prioritized and followed up.	Technical Lead Architect, Operations Team	1. Critical	
	Change management process, CI/CD process defined for deployment	Documented process for the deployment (type of change, tool/VCS log, test) - Deployment Management Process	Deployment and Release Manager	1. Critical	
ITSM Integration	Monitoring implementation	Rules and tresholds implemented in the monitoring tool;	Technical Lead Architect	1. Critical	
	Event, Incident and Problem Management	Categories created for event, incident, problem tickets; Ticketing integration (SLA, CI, Support group).	Process Implementation Architect	1. Critical	
	Change Management	For SSR's/Standard changes (including internal standard Changes): - Defined list of SSR's or Standard Changes; - Proper documentation (CIP, WI, CAB Approval); - Tooling integration: Service Catalog and ticketing; For Non-Standard Changes: - Ticketing integration according with the process requirements defined in the Operations Manual.	Process Implementation Architect	1. Critical	
	Configuration Management	- Data Model containing the relathionship between the CI's and the type of relationship; - Implemented CMDB update: automated or manual; - CMDB reporting (generated from tooling).	Process Implementation Architect	1. Critical	
	Capacity Management	- The capacity management parameters, thresholds and reporting shold be defined ad implemented; - Capacity Management reporting (generated from tooling)	Technical Lead Architect	2. Medium	
Operational Requirements	Reporting setup according with the L4D	Reporting is implemented according with the L4D; Work instruction on how to generate the reporting is created.	Technical Lead Architect	2. Medium	
	Service Continuity Management Implementation	The implementation on availability compliant to the service (what availability are we delivering?) description must be documented in the designs. And a service continuity plan must be available.	Technical Lead Architect	1. Critical	
	Production Plan	Service specific activities must be defined in order to have the service up and running. The production plan is service specific and it must be compliant with the process described in the Cloud Operations Manual.	Process Implementation Architect	2. Medium *activities to be defined but WI can be delivered later	
	OLA's & Third-Party Agreements	Operational Level Agreement (OLA) is prepared, with the involved non-cloud units	Process Implementation Architect	1. Critical	
	Service RACI and service contact sheet	Service roles are defined and RACI is constructed based on the roles defined for the service. The RACI will include also external teams involved in the delivery of the service.	Process Implementation Architect	1. Critical	