SKA1 SDP Construction and Verification Plan

Document Number SKA-TEL-SDP-0000047
Document Type .... MPL
Revision 04
Author .... F. Graser, B Nikolic, P Alexander, J Coles, C Broekema
Date 2018-10-31
Document Classification .... Unrestricted
Status .... Released

Name	Designation	Affiliation	Signature	
Authored by:				
F. Graser	SDP Project Systems Engineer	SARAO	Ferdl Graser Ferdl Graser (Oct 29, 2018)	
			Date:	
Owned by:				
F. Graser	SDP Project Systems Engineer	SARAO	Ferdl Graser Ferdl Graser (Oct 29, 2018)	
			Date:	
Approved by:				
Paul Alexander	SDP Project Lead	University of Cambridge	Paul Alexander Paul Alexander (Oct 30, 2018)	
			Date:	
Released by:				
Paul Alexander	SDP Project Lead	University of Cambridge	Paul Alexander Paul Alexander (Oct 30, 2018)	
			Date:	

DOCUMENT HISTORY

Revision	Date Of Issue	Engineering Change Number	Comments
02C	2016-03-24		Prepared for SDP delta-PDR review
03	2016-07-21		Prepared for SDP delta-PDR closeout
03C	2018-04-23	ECP-SDP-180001	Prepared for M20 Pre-CDR review
04	2018-10-31	ECP-SDP-180002	Prepared for M21 CDR review SDP PRECDRs addressed: Major: SDPPRECDR-14 ‘’Test cases or procedures for the L2 requirements’’ SDPPRECDR-18 ‘’Milestones and Solution Intent’’ SDPPRECDR-19 ‘’Work Breakdown Structure’’ SDPPRECDR-20 ‘’Continuous Integration and Deployment proposals’’ SDPPRECDR-25 “Cross checking between the schedule and the Cost Model” SDPPRECDR-21 ‘’Release Management’’ SDPPRECDR-32 ‘’SDP system in the ITF’’ SDPPRECDR-34 ‘’Qualification of the SDP Operational System’’ SDPPRECDR-37 “Phasing of SDP expenditures” SDPPRECDR-53 ‘’Role of the System ITF in the qualification of the SDP’’ SDPPRECDR-78 ‘’SDP benchmark suite’’ SDPPRECDR-182 ‘’Spend profile’’ SDPPRECDR-232 ‘’Can commissioning and AIV support system cope with the load up to AA4?’’ SDPPRECDR-233 ‘’Additional software needed for commissioning and AIV support’’ SDPPRECDR-234 “Functionality versus scaling during deployment” SDPPRECDR-259 ‘’Operational System verification’’ Minor: SDPPRECDR-15 ‘’Applicable documents’’ SDPPRECDR-16 ‘’Hardware acceptance’’ SDPPRECDR-22 ‘’SDP 2nd operational deployment’’ SDPPRECDR-35 ‘’SDP Commissioning and AIV Support System as pre-production representation of SDP Operational System’’ SDPPRECDR-52 ‘’Reference for SKA System Team’’ SDPPRECDR-183 ‘Use of "SDP"’ SDPPRECDR-185 ‘’initial system sizing’’ SDPPRECDR-186 ‘’Networking’’ SDPPRECDR-231 ‘’Functionality available before AA-1’’ SDPPRECDR-253 “Qualities verification” SDPPRECDR-260 ‘’SAFe program team’’ SDPPRECDR-261 ‘’SAFe system team’’ SDPPRECDR-262 ‘’MVP definition’’ SDPPRECDR-268 ‘specify what ‘’early in construction phase’’ means’ SDPPRECDR-269 ‘’priority from commissioning plan’’ SDPPRECDR-270 ‘’Use of SDP in SRCs during construction phase’’ SDPPRECDR-271 ‘’additional domain feature team’’ SDPPRECDR-272 ‘’SDP Commissioning milestones’’ SDPPRECDR-336 ‘’Update List of Abbreviations / Explain Abbreviations at first use’’

			SDPPRECDR-337 ''Better explain difference between milestones in AIV support system''

DOCUMENT SOFTWARE

	Package	Version	Filename
Word processor	Google Docs		SKA-TEL-SKO-0000000-01_GenDocTemplate
Block diagrams			
Google docs Add-ons	Table of contents		Used for heading numbering.

ORGANISATION DETAILS

Name	SDP Consortium
Lead Organisation	The Chancellor, Masters and Scholars of the University of Cambridge The Old Schools Trinity Lane Cambridge CB1 ITN United Kingdom
Website	www.ska-sdp.org

Table of Contents

List of Abbreviations

AA4 / 3 ..n	Array Assembly 4 / 3 / ...
AIV	Assembly Integration Verification
ART	Agile Release Train
C0	Start of Construction
CI&D	Continuous Integration and Deployment
COTS	Commodity Off-The-Shelf
CPF	Central Processing Facility
CSP	Central Signal Processor
DD CAL	Direction Dependent Calibration
HPC	High Performance Computing
HTC	High Throughput Computing
ICD	Interface Control Document
ITF	Integration Test Facility
IVOA	International Virtual Observatory Alliance
M&C	Monitoring and Control
MFMS	Multi Frequency Multi Scale
MVP	Minimal Viable Product
PI	Program Increment (a SAFe construct)
PV	Performance Verification
QA	Quality Assurance
QE	Qualification Event
RPF	Remote Processing Facility
SaDT	Signal and Data Transport
SAFe	Scaled Agile Framework
SDP	Science Data Processor
SDHP	Science Data Handling and Processing
SKA	Square Kilometre Array
SKAO	Square Kilometre Array Office
SPC	Science Processing Centre
TBC	To Be Confirmed
TM	Telescope Manager
WBS	Work Breakdown Structure
WSJF	Weighted Shortest Job First

1. Introduction

The SKAO has adopted SAFe (in the Large Solution configuration) [RD01] for the construction of the software dominated elements of the SKA project, and this is essential for the construction and verification of SDP for the following reasons:

- Due to the nature of the SDP system within the context of the SKA system (see SDP Architecture Overview [AD04]) it is not feasible to mitigate all risks impacting the construction of SDP. A risk driven approach is therefore needed that allows the balancing of lifetime cost against performance. Close interaction with stakeholders (mainly SKAO) is required in order to trade-off performance against cost.
- There are many unknowns in the project and therefore an approach is needed that can respond to changes without significant impact to cost and schedule.

A basic understanding of the SAFe principles and terminology is required to correctly interpret this document. Refer to the SAFe 4.5 Introduction White Paper [RD05].

This SDP Construction Plan has been designed to minimise cost and risk. The SDP Cost Estimate [RD04] is based on it and its principles.

1.1. Scope

This document defines the plan to construct and verify the SDP systems (as defined in section 3) within the context of SAFe [RD01] and supersedes the previous version (delivered for pre-CDR). It takes into account deliverables and integration points with other telescope products as defined in the MID [AD05] and LOW [AD06] roll-out plans.

Continuous Integration and Deployment (CI&D) and Release Management are key aspects of the development, testing, deployment and release of software in the construction of the SDP system, but they are software engineering aspects and therefore not in the scope of this plan. The implementation of the SDP system will be done according to the SKA software engineering processes and standards described in the SKA Software Engineering Management Plan [AD02], the Fundamental SKA Software & Hardware Description Language Standards [AD01] and the SKA Software Verification and Test Plan [AD03].

The SDP Construction Plan is based on the following assumptions about the programme structure:

- All construction activities start at C0, which at the time of writing is defined as 30 March 2020, with no activities occurring between the SDP CDR and C0.
- The SKAO will be involved in the detailed technical management and risk management of the SDP programme.
- Continuity of skills is maintained from the pre-construction to the construction phase (loss of key skills will have significant cost and schedule implications that have not been taken into account in this plan).
- The procurement and contracting model is structured so that the required skills are available from suppliers that are prepared to work within the SAFe methodology and the required programme structure.
- A Program Increment (PI) cadance of 3 months.

Section 6 describes the resources required for the Science Data Handling and Processing (SDHP) Agile Release Train (ART) [RD16] that will implement the SDP system according to the SAFe implementation for SKA [AD02].

2. References

2.1. Applicable Documents

The following documents are applicable to the extent stated herein. In the event of conflict between the contents of the applicable documents and this document, **the applicable documents** shall take precedence.

AD01	SKA-TEL-SKO-0000661 "Fundamental SKA Software & Hardware Description Language Standards", Rev 2, 2016-11-24
AD02	SKA-TEL-SKO-0000828 SKA Software Engineering Management Plan, Rev A, 2017-11-01
AD03	SKA-TEL-SKO-0000990, SKA Software Verification & Test Plan
AD04	SKA-TEL-SDP-0000013 SDP Architecture Documentation, Rev 06
AD05	SKA-TEL-AIV-2410001 Roll-out Plan for SKA1_MID, Rev 06
AD06	SKA-TEL-AIV-4410001 Roll-out Plan for SKA1_LOW, Rev 06
AD07	SKA Solution Intent, https://confluence.skatelescope.org/display/SE/Solution+Intent

2.2. Reference Documents

The following documents are referenced in this document. In the event of conflict between the contents of the referenced documents and this document, **this document** shall take precedence.

RD01	http://www.scaledagileframework.com
RD02	SKA-TEL-SDP-0000052 SDP Risk Register, Rev 08 https://jira.ska-sdp.org/projects/SDPRISK
RD03	SKA-TEL-SDP-0000081 SDP Operations Plan, Rev 02
RD04	SKA-TEL-SDP-0000043 SDP Cost Model, Rev 04
RD05	SAFe 4.5 Introduction White Paper, https://www.scaledagile.com/resources/safe-whitepaper/
RD06	SAFe Program Level, http://www.scaledagileframework.com/program-level/
RD07	SAFe System Team, http://www.scaledagileframework.com/system-team/
RD08	SAFe Agile Teams, https://www.scaledagileframework.com/agile-teams/
RD09	SAFe Component and Feature Teams, https://www.scaledagileframework.com/features-and-components/
RD10	SAFe Features and Capabilities article, https://www.scaledagileframework.com/features-and-capabilities/
RD11	SAFe Enablers article, https://www.scaledagileframework.com/enablers/
RD12	SKA-TEL-SDP-0000046 SDP Costing Basis of Estimate, Rev 04
RD13	SAFe v4.6 Agile Testing article, https://v46.scaledagileframework.com/agile-testing/

RD14	SAFe Roadmap article, https://www.scaledagileframework.com/roadmap/
RD15	SKA-TEL-SKO-0000953 SDP PRE-CDR Panel Report, Rev01, 2018-07-12
RD16	SKA Work Breakdown Structure for the Science Data Handling and Processing (SDHP) Agile Release Train (ART), https://confluence.skatelescope.org/display/PPM/1.4.1+Science+Data+Handling+and+Processing
RD17	SKA-TEL-SDP-0000180, SKA1 SDP High Level Overview, Rev 01

3. Program Vision

Refer to the **SDP High Level Overview** [RD17] and the **SDP Architecture Reading Guide** in [AD04] for an overview of the SDP element and its stakeholders. These aspects are an integral part of the Program Vision for SDP, but are not repeated in this document.

The construction approach of the SDP software is driven by the following factors:

- Although performance, scale and other qualities are only required from Array Assembly (AA) 4 onwards (up to AA3 only 1% of the full scale performance is required), incremental verification of Non-functional Requirements (NFRs) is needed from C0 to AA4.
- The need to support commissioning, integration and testing of both telescopes. This is required early in the construction phase (ITF → AA2).
- Many uncertainties exist (detailed calibration strategies, future hardware performance, science priorities, etc.) with associated risk. Cost and risk need to be minimised as much as possible.
- The implementation of the majority of the functionality required for SDP is low risk (high maturity level) and therefore does not drive the construction approach, but we cannot preclude the need for new domain-specific functionality emerging during construction.

These driving factors have led to a construction approach where Product Variants of the SDP Operational System can be produced to satisfy use cases like telescope commissioning, early engagement and deployment at SKA Regional Centres (SRC). Producing Product Variants is enabled by the modifiability and portability quality attributes of the SDP Architecture. Requirements for Product Variants (in particular for the Commissioning and AIV support software) are not available yet and will be developed jointly with SKAO during bridging (a period of activity between element/system CDR close-out and C0) as per agreement with SKAO (refer to recommendation 10 in the SDP PreCDR Report Cover Note supplied with RD15] as well as the architecture of any other required product variants.

There are currently no explicit requirements to deploy SDP software or SDP variants at SRCs. However, SKAO have expressed a desire (through the pre-CDR OAR SDPPRECDR-270) for early engagement with SRCs and deployment of SDP software to SRCs which could benefit science commissioning of the telescopes by providing significant compute resources for commissioning.

Architecture refinement and detailed design will continue during construction through to operations and will produce the core documentation for SDP software and hardware.

In order to ensure a smooth transition from the construction phase to the operational phase of the project it is necessary that the SKA operations software team is directly involved with construction of the SDP Operational System towards the end of the construction phase. In order to mitigate the risk that the transition does not occur as planned (TSK-2112 [RD02]), it is expected that one of the domain feature teams will migrate to the SKA operations software team at the end of the construction phase to facilitate a smooth transition and retain the necessary skills and experience required within the SKA operations software team.

3.1. Software implementation approach

The SDP Operational System will satisfy the SDP L2 requirements. In order to minimise risk and cost, a minimal viable product (MVP) of the SDP Operational System will be established early in the construction phase (with the first year), and then functionality and performance will be added as driven by science value according to the SAFe Weighted Shortest Job First (WSJF) prioritisation model [RD01]. Development dependencies and artefacts such as the SKA1 Commissioning Plan (which is not written as of October 2018), AIV roll-out plan [AD05 & AD06] and others will be used by Solution Management to drive the prioritisation (WSJF) of SAFe Capabilities, Features and Enablers that will be implemented, and this is done at every Program Increment planning session (every 3 months). The construction of the SDP Operational System will also include set-based design and construction elements [RD01] for certain areas (like execution frameworks) in order to reduce risk in these particular areas. A major driver of the construction approach of the SDP Operational System is the incremental verification (by testing) of key performance and scaling qualities and this is evidenced in the milestones described in this document.

The SDP Module Decomposition and the dependencies between modules have specifically been designed to be practical to build within the constraints of the construction phase. Refer to section 5.2 in the SDP System-level Module Decomposition and Dependency View [AD04] for a detailed explanation of how this is achieved.

The majority of the functionality will be implemented and tested incrementally throughout the construction phase at Sprint (2 weeks) and at Program Increment (3 month) cadence. Apart from the incremental verification of functionality, end-to-end demonstration of functionality is also needed at modest scale early in the construction phase and this is represented by the "Functional verification at modest scale" milestone.

The Commissioning and AIV support software as well as SDP variants for testing/running at SRCs are Product Variants of the main SDP software with additional functionality and 3rd party software integrated as needed. The current plan, which is based on the detail given in the AIV roll-out plan, is that the Commissioning and AIV support software will satisfy the high level functionality (as specified in the AIV roll-out plan) needed to support the ITF QE, AA1 and AA2 integration milestones and that the SDP Operational software will support the integration milestones from AA3 onwards. The role of the SDP Commissioning and AIV Support is explained in more detail in the SDP Architecture Overview document [AD04]. Product Variants are enabled by the modifiability and portability quality attributes of the SDP Architecture and therefore the architectures of any product variants pose a low risk. Implementing the Commissioning and AIV support software product variant is included in the cost estimate of the SDP element, but not implementing any other Product Variants. Due to the modifiability and portability of the SDP architecture the cost of creating other product variants during construction will not be prohibitive (assuming the variant does not vary significantly from the standard SDP software) and therefore could be prioritised if required.

The SDP Architecture is documented in terms of Components, Connectors, Modules, Data Models, etc. [AD04], but these elements do not translate directly into development activities during the construction phase. The software development activities will be described in terms of SAFe Features and Enablers and these Features and Enablers will incrementally develop the functionality and architecture as specified and described by the SDP L2 requirements, SDP Architecture documentation and other artefacts (see Program Solution Intent below), according to the SAFe lean-agile methodology. Refer to the SAFe Features and Capabilities article [RD10] and the SAFe Enablers article [RD11] for detailed descriptions of how Features and Enablers are defined.

3.2. Hardware implementation approach

The SDP hardware will be acquired as procurements of COTS computer hardware in several phases:

- Two deployments of hardware per telescope, one at the ITF facility and another at the Central Processing Facility (CPF) on site. The deployment is approximately 0.1% (TBC) of the size of the full scale system. This is expected to be one rack of equipment for each deployment and the deployments are required in time for the ITF Qualification Event and AA1
- A deployment (per telescope) of hardware to the Science Processing Centre (SPC). The deployment is approximately 1% of the size of the full scale system. This deployment is needed for AA3 and can be integrated once the SaDT long distance links are available at the SPC.
- The full scale SDP hardware system is deployed at the SPC for AA4. Note that a full size hardware deployment for AA4 complies with the design baseline of SKA. The SKA deployment baseline is a phased approach where a portion of the full size hardware is deployed for AA4 followed by one or more additional deployments over a number of years to reach the full scale system. This enables the SKA to manage risks related to uncertainties in hardware costing - refer to the Phased Hardware Rollout section and Appendix A in the SDP Costing Basis of Estimate [RD12] for details.

The hardware deployment in the CPF on site (one rack) is only required for AA1 and AA2 and can be decommissioned once the hardware deployment for AA3 (in the SPC) has been integrated. This hardware deployment may however still be useful for the integration and commissioning (by AIV) of individual dishes or stations and therefore may remain on site until the end of AA4.

Although the procurement of the SDP hardware will be managed and conducted by the SKAO, the following input and support activities are required in order to support the evaluation and procurement of the hardware:

- Software suite for SKA to evaluate (benchmark) hardware. This is required for procurement of AA4 hardware and therefore is needed at least 12 months before AA4 SDP deployment (see section 10). Benchmarking software will be developed incrementally in order to allow for regular evaluation of vendor hardware throughout the construction phase.
- In order to maximise the scientific usefulness of the hardware investment, SDP construction will use the benchmarking results above to select the most suitable system that can be built from off the shelf components. For production. Considering the crucial nature of the SDP as

an integral part of the SKA telescope, the use of custom or unproven emerging technologies is a risk that must be carefully considered against the potential gains. Such components may be evaluated and de-risked in the SKA engineering centres and integrated in small numbers on an experimental basis in the inherently heterogeneous SDP when proven reliably superior to existing off the shelf solutions.

Refer to the SDP Costing Basis of Estimate [RD12] for further cost risk mitigation activities before and during the construction phase.

3.3. Verification and testing approach

3.3.1. SDP Software

Verification and testing of SDP software (for the SDP Operational System and any Product Variants) will be done according to the SKA1 Software Verification and Test Plan [AD03]. Aspects of functional and non-functional (mainly performance and scaling qualities) verification that drive the construction approach of SDP are given here, but general principles and processes described in the SKA1 Software Verification and Test Plan [AD03] are not repeated.

The bulk of functional verification will be done incrementally through System Demos [RD01] at Sprint and PI cadence throughout the construction phase. Since System Demos verify features in relatively small batches, an additional milestone is planned to verify end-to-end functionality (of the SDP Operational System) at modest scale relatively early in the construction phase.

Achieving the performance and scaling qualities of the SDP Operational System will be challenging and therefore progress towards achieving these qualities needs to be regularly evaluated. Verification of the performance and scaling qualities must therefore be done incrementally and will require larger tests (either Enabler Tests or Exploratory Tests[1]) needing dedicated compute resource and a significant amount of time to set up and execute. Each test will typically be performed as one or more Enabler Tests or Exploratory Tests Stories by an Agile Team supported by the Systems Team. These tests may require a significant amount of compute resource and should SDP resource at the required scale not be available (since the hardware has not been deployed at that point), then these tests need to be executed using compute resource at other HPC facilities (for example in national labs facilities). Learning Milestones (see sections 5 and 7) are defined to demonstrate how the incremental verification of the performance and scaling qualities *could* be achieved. Performance Verification learning milestones are defined at a cadence of approximately 6 months as this cadence is suitable for measuring increments in the performance and scalability of the current in-process SDP software. The cadence of these Exploratory and Enabler Tests can be adjusted during the construction phase to meet the needs of the project and mitigate risk where necessary. It is expected that these learning milestones and their specific goals will change in response to the output of these milestones, changes to the intended Solution (see section 4) and continuous improvement efforts. Note that the Performance Verification learning milestones do not have any external dependencies or dependencies between themselves other than those resulting from their incremental nature.

The Performance Verification learning milestones are described in section 10. The first 5 Performance Verification learning milestones aim to demonstrate the performance and scaling

[1] Refer to Q3 and Q4 of the Agile Testing Matrix in the SAFe v4.6 Agile Testing article [RD13].

qualities of individual or groups of related Components in the SDP Architecture (see the SDP Component and Connector View [AD04]) and their order is determined mainly by the module dependencies (see the SDP System-level Module Decomposition and Dependency View [AD04]).

The next 5 Performance Verification learning milestones aim to demonstrate scaling and/or performance of particularly challenging aspects of the complete SDP system by running particular Workflows that exercise these aspects. This also verifies that these Workflows can achieve the required scale or performance. The order of these 5 learning milestones is determined by the module dependencies, progressive workflow complexity and to support early scientific operation of the telescopes.

The final Performance Verification learning milestone is a demonstration of the performance at full-scale with a full feature set and is a demonstration that SDP is achieving the performance that is required.

The operations related qualities of the SDP Operational System can be verified by early engagement and involvement with the SKA Operations Team as part of a soft transition into operations.

The verification of SDP L2 requirements occurs incrementally through the verification of the Features, Enablers, Stories and System Qualities Tests that trace to these requirements according to the SKA1 Software Verification and Test Plan [ADxx].

3.3.2. SDP Hardware

The verification of hardware performance will occur incrementally through the use of hardware benchmark suites (which will be developed by SDP and used by both SDP and hardware vendors) and by later performance verification milestones. The delivery of the benchmark suites for use by the hardware vendors as part of the procurement process is included as a milestone (see section 7) but development of the benchmark suite will occur incrementally to support evaluation of new hardware released by vendors. The benchmarking and verification of hardware performance will occur at the cadence that SKA has access to new hardware and this is mainly determined by the cadence at which vendors release new hardware to the market. Although this activity is continuous, its cadence is driven by external factors and is not expected to occur during every Program Increment.

Acceptance testing of SDP hardware will start during the construction phase and continue during the operational phase of the SKA Observatory (since hardware will be refreshed on a regular basis throughout the lifetime of the Observatory). This is not a continuous activity and only done once for each hardware procurement and therefore it is planned to occur 3 times during the construction phase and many times during the operational phase of the SKA Observatory. Acceptance testing of hardware during the construction phase and the operational phase will follow the same industry standard processes and procedures for large COTS hardware procurements and this is described in the SDP Operations Plan [RD03].

4. Program Solution Intent

Solution intent is the repository for storing, managing, and communicating the knowledge of current and intended Solution behavior. Where required, this includes documented, fixed, and variable *specifications and designs; reference to applicable standards, system models, functional and nonfunctional tests; and traceability.* [RD01]

The following Solution Intent, within the context of the SDP, has been identified and will form part of the SKA Solution Intent [AD07]:

- **Specifications:** L2 requirements (including Architecturally Significant Requirements) and ICDs
- **Design:** Architecture documentation (views & beyond) and its supporting documentation, analysis, models and any other relevant design documentation.
- **Tests:** Functional & performance verification/testing documentation.

5. Milestone Schedule

This plan covers an extended planning horizon from C0 in 2020 to the end of construction in 2026 and therefore (according to SAFe guidance for long term planning) the schedule shown in Figure 1 below contains key milestones and critical delivery dates. See section 10 for the description of each milestone.

The milestones for the construction of SDP are defined following the SAFe [RD01] principles. SAFe defines 3 types of milestones:

- **Program Increment** (PI) milestones which are used to objectively evaluate progress towards the technical or business hypothesis. These occur at PI cadence.
- **Fixed-date** milestones are those driven by external events, third-party deliverables, external constraints, etc. These are distinct from the development cadence.
- **Learning** milestones which demonstrate evidence of the viability of the current in-process solution.

PI and Learning milestones are meant to provide objective evidence of working systems and are therefore ideal for the incremental risk-driven verification of the performance and scaling qualities of the SDP Operational System as well as the evaluation of alternative design options (set-based design [RD01]). The specific date at which these milestones occur or the specific outcome that is achieved is not important, but rather the value added by the objective measurement of a working system.

Note that the milestones shown here *do not* represent reviews, sign-offs or formal baselines, but rather the incremental build-up of content and the demonstration of working and tested systems. This is one of the SAFe lean-agile principles [RD01] and is different to a sequential, phase-gate development model and therefore needs to be kept in mind.

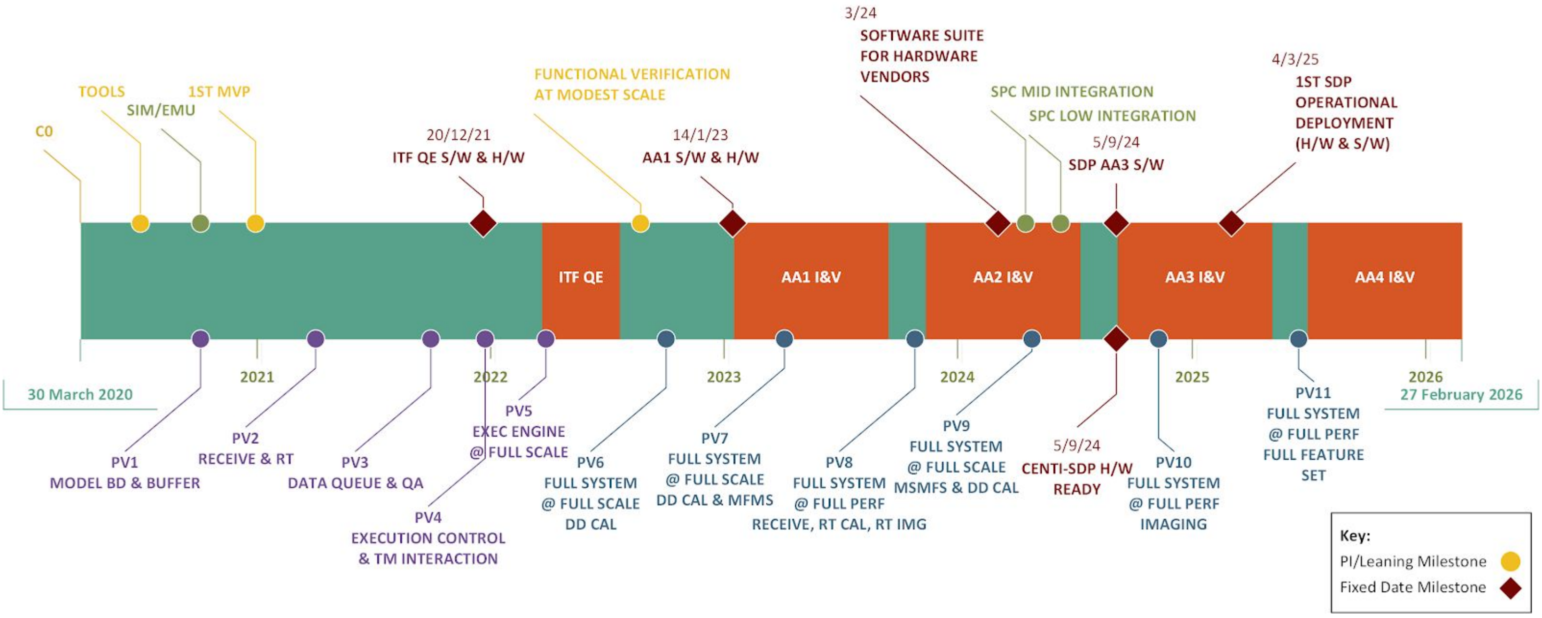

Figure 1: Milestone schedule for the SDP Commissioning and AIV support system and the SDP Operational System

The high level Milestone schedule shown in section 5 represents the high level Roadmap or Solution Roadmap [RD14] for SDP. The Solution Roadmap covers the long term planning horizon beyond 3 PIs. The PI Roadmap consists of a committed plan for the current PI and offers a forecast of the deliverables and milestones for the next two to three PIs. It is important to note that a forecast does not represent a commitment.

The Solution Roadmap and Solution Intent both inform the development of the SAFe PI Roadmap [RD01]. The PI Roadmap is developed during PI Planning where the teams commit to meeting the PI Objectives for the next PI. The first Roadmap will however be forecasted before the first PI Planning takes place (at the start of the construction phase) in order to facilitate planning across Agile Release Trains (ARTs), and will be developed closer to the start of Construction (C0).

6. Resourcing

The SDP system will be implemented by the Science Data Handling and Processing (SDHP) Agile Release Train (ART) which delivers value as part of the SKA Solution Train. This section describes the resources needed for the SDHP ART (Program Team and Agile Teams) as well as SDP specific resources required for the SKA Systems Team in order to support the use of SAFe in the context of SKA.

The resources required (in terms of people and number of teams) for the SAFe teams shown in this section are taken from the SDP Cost Estimate [RD04]. The team roles and sizes follow SAFe’s recommendation. The resource numbers in this section are given in terms of full-time people dedicated to their particular team.

6.1. SDHP ART Program Team

The SDHP ART Program Team is based on the roles defined for the SAFe Program Level which is defined in [RD06].

The SDHP ART Program Team has the responsibility to coordinate the SDHP ART, help the ART and its Agile Teams align to the common SKA mission and provide the necessary governance. Table 1 below defines the roles, expertise and people required for the SDHP ART Program Team.

SAFe Role	**Responsibilities or expertise required**	**Resource required (persons)**
SDHP Release Train Engineer	Servant leader and coach for the SDHP ART responsible for facilitating the major events and processes and assisting teams in delivering value.	1
SDHP Architecture & System Engineering	A cross-discipline team that applies systems thinking, evolves the architecture (hardware and software) for the system, defines requirements and interfaces, validates technology assumptions and evaluates alternatives.	5.5
SDHP Product Management	Content authority for the Program Backlog, responsible for defining and prioritising system features, developing the Program Vision and Roadmap, and participates in validation. *Note the lead SDHP Product Management role is resourced by the SKAO and not included in the SDP cost estimate.*	1 *+ 1 SKAO*

Table 1: Roles, responsibilities and resources required for the SDP Program Team.

6.2. SKA System Team

The SKA System Team [AD02] is a special Agile Team on the Solution Train that is responsible for building the development infrastructure (including continuous integration), integrating assets from Agile Teams and supporting end-to-end testing at both Solution and Program (ART) levels. The SKA System Team works closely with the Agile Teams (in all ARTs) and responsibilities are shared between the System Team and Agile teams to enable effective solution development and maximise ART velocity. The SKA Systems Team is based on the SAFe Systems Team (defined in [RD07]) and defined in the SKA Software Management Plan [AD02].

This section describes the SDP specific resources required for the SKA Systems Team. Although these resources are not part of the SDHP ART, they are critical to the implementation and integration of the SDP system and therefore included in the SDP Cost Estimate.

Table 2 below defines the responsibilities of the SDP resources required for the SKA Systems Team.

Resource required: 6-7 people

Responsibilities:

- **Building Development Infrastructure:** Create and maintain infrastructure including continuous integration, automated builds, automated verification testing. Create and maintain platforms and environments for system and solution demonstration, quality assurance, performance testing, etc.
- **System Integration & Demos:** Participate in planning meetings and backlog refinement to define integration and test Capabilities and Features. Perform automated and manual integration of work from ARTs and Suppliers for both the MID and LOW Telescopes. Support daily activities of other teams as required. Help stage System and Solution demos.
- **End-to-end and NFR testing:** Collaborate with the Agile Teams to create test scenarios and test suites. Support manual and automated tests for new Features and Stories, and System Qualities Tests (to verify Nonfunctional Requirements). Assist in identifying system shortfalls and bottlenecks.
- **Quality Assessment and Release:** Validating that the solution meets the Solution Intent and participating in the release management process.

6.3. SDP Agile Teams

The SDP Agile Teams are based on the SAFe Agile Teams defined in [RD08] and SAFe Feature and Component Teams described in [RD09].

Each SDP Agile team has the responsibility to perform *all* of the following functions while delivering value at every iteration [RD01]:

- Estimates the size and complexity of the work;
- Determines the technical design in their area of concern, within the architectural guidelines;
- Commits to the work it can accomplish in an iteration or Program Increment (PI) timebox;
- Implements the functionality;
- Tests the functionality;
- Deploys the functionality to staging and production;
- Supports and/or builds the automation necessary to build the continuous delivery pipeline;
- Continuously improves their process.

The SDP Agile Teams incorporate the following roles (see [RD01] for more detail):

- **Dev Team** - software developers and testers, engineers, and other dedicated specialists required to complete a vertical slice of functionality;
- **Scrum Master** - servant leader and coach of the agile team;
- **Product Owner** - serves as the customer proxy and is responsible for defining stories and prioritizing the team backlog.

The organisation of the SDP Agile teams is based on the SDP Module Decomposition. To ensure highest feature throughput, SAFe generally recommends a mix of perhaps 75-80% feature teams and 20-25% component teams. For SDP this recommended mix of feature and component teams is difficult to achieve due to the high degree of specialisation required to implement certain SDP modules. However, the mix of SDP agile teams is still biased towards feature teams in order to maximise velocity and efficiency and minimise dependencies between SDP agile teams.

The SDP agile teams have been organised in 3 types of agile teams: non-domain feature, component, and domain feature teams. The module decomposition diagram in Figure 2 has been colour-coded to indicate expertise required by the different types of SDP agile teams.

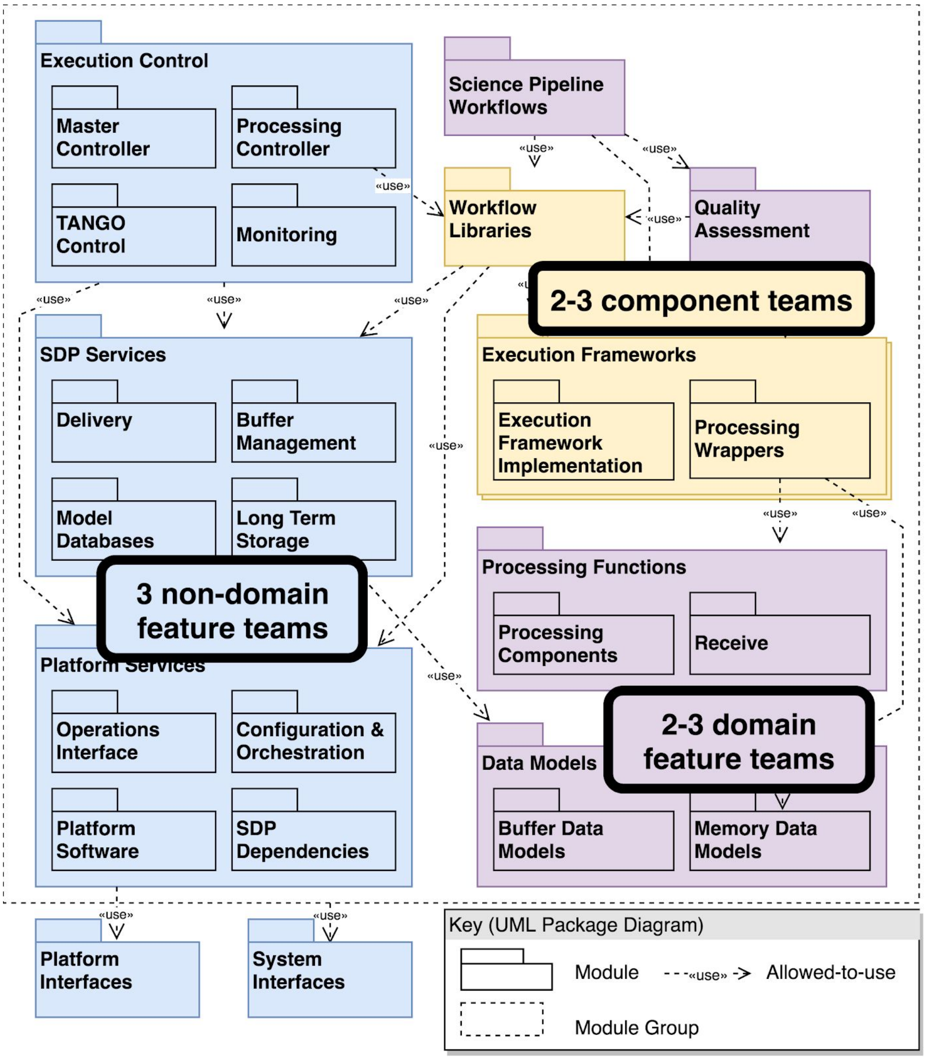

Figure 2: The SDP module decomposition diagram showing the allocation of the 3 types of SDP Agile Teams to modules.

Note that Figure 2 is not intended to show the extent of the software development of the SDHP ART or imply that specific Agile Teams have a responsibility (or exclusivity) to develop specific software modules. The SDHP Agile Teams would also develop other software modules not shown in Figure 2, like SDP Product Variants, SDP Resource Model, etc.

6.3.1. SDP non-domain Feature teams (blue)

Resource required: 3 teams (7±2 people per team)

Based on the SDP module decomposition, these feature teams will require the following expertise:

- System operation;
- Observation/processing planning and resource management;
- Distributed systems, IVOA services, science data models;
- Science pipeline operation;
- Data preservation;

- Sky model and telescope state/model;
- Data lifecycle management;
- Real-time telescope operation;
- Cloud or Distributed Computing;
- Data Centre Administration;
- Data streaming;
- Log and health metrics aggregation;
- Operating systems, standard libraries;
- Cluster file or object storage systems;
- Logging systems, debugging;
- HPC and HTC.

6.3.2. SDP Component teams (yellow)

Resource required: 2 larger or 3 smaller teams (7±2 people per team).
These component teams will be focussed on implementing Execution Frameworks and will require the following expertise:

- Data flow implementation;
- Data flow kernel integration.

6.3.3. SDP domain feature teams (purple)

Resource required: 2 - 3 teams (7±2 people per team) from the start of the construction phase with an additional team that starts working during the second half of the construction phase. One team is expected to migrate to the SKA operations software team at the end of the construction phase.
The domain feature teams will focus on implementing features requiring domain knowledge and will require the following expertise:

- Radio astronomy algorithms;
- Radio astronomy data models;
- Radio astronomy workflows (pipelines).

6.4. Resource Profile

The resource profile shown in Figure 3 is based on the following:

- Implementation dependencies in the SDP Architecture (as shown in the Module Decomposition View) are taken into account in order to avoid implementation bottlenecks that would reduce the velocity of the SDHP ART.
- The Telescope rollout ([AD05] and [AD06]) schedule and integration activities have been taken into account, but these mainly impact the System Team.
- This resource profile is a coarse grained estimate appropriate to the planning horizon (long term) of this construction plan. The PI Roadmap (which contains a forecast for next 2-3 PIs) will determine the actual required resource profile on a continuous PI-by-PI basis.

Rationale for the resource profile of each team:

- The Program Team is expected to the fully staffed from the start to the end of the construction period.

- The System Team will start slightly smaller and then increase in size after approximately 2 years when the bulk of the telescope integration activities start.
- In the first 2 years many non-domain Features and Enablers will be needed in order to support the rollout of domain Features and therefore the non-domain feature teams will need to ramp up quickly in the first 2-3 Program Increments (PIs). After year 2 the non-domain feature teams can reduce slightly to implement the remaining non-domain features and enablers. In the final year of construction most non-domain features and enablers should be complete and therefore the non-domain feature teams can be reduced further.
- The component teams (due to their nature) will have dependencies on the output of the other agile teams and in particular on the non-domain feature teams during the first year. Therefore the components teams should start smaller and ramp up after the first year to avoid dependencies causing development bottlenecks. In the final year of construction most of the work of the component teams should be complete and the component teams can start to ramp down. Note that the resource profile of the component teams is dependant on emergent Execution Framework technologies.
- Domain feature teams are needed from the start of construction, but due to their dependence on non-domain Feature and Enablers, the domain feature teams can be ramped up from 2 to 3 teams after the first 2 years.

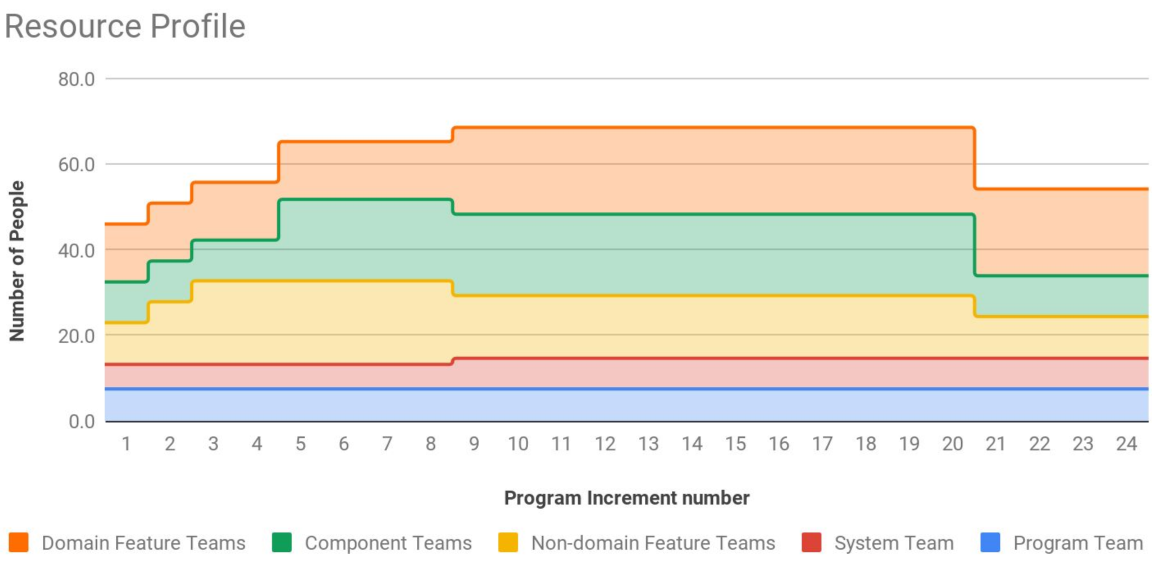

Figure 3: The resource profile in each team shown as the number of full-time people per Program Increment starting from C0 until the end of Construction.

7. Milestone details (Solution Roadmap)

Note that the dates given for Program Increment (PI) and Learning Milestones are approximate dates only and are meant as a guideline for PI Planning sessions where the Roadmap is developed in detail.

Milestone	**Key Engineering Goals**	**Milestone type**	**Date**
Tools	SAFe & tools training, testing of tools. DevOps and test platforms in place. SAFe management processes in place and roles filled.	Program Increment	Q3 2020
Simulators/ emulators	Provide a SDP-TM interface simulator for early testing by TM Receive visibilities	Program Increment	Q4 2020
Performance Verification 1	Demonstration of Model Databases and Buffer component scalability. This is an early demonstration of scalability of two critical long-lead time components which are key to the performance of the complete system.	Learning: sub-system scalability	Q4 2020
First Minimal Viable Product	This would be the first meaningful Solution Level integration of SDP software with software from other ARTs. Goals: - Implementation of the high-level architecture - External interfaces - Evaluation & correction of existing code base (prototypes)	Program Increment	Q1 2021
Performance Verification 2	Demonstration of Receive & Real-time processing component performance. This needs to be an early sub-system performance demonstration (against a model, given the hardware requirements) since this sub-system is needed for commissioning quite early on.	Learning: sub-system performance	Q2 2021
Performance Verification 3	Demonstration of Data Queue and Quality Assessment component scalability. These are grouped together since they are expected to share the same implementation.	Learning: sub-system scalability	Q3 2021
Performance Verification 4	Performance and functional test of Execution Control and TM interaction.	Learning: sub-system performance	Q4 2021
ITF QE product	Certain external interfaces:	Fixed Date	20/12/2021

handover	- Basic TM M&C interface (TANGO) - CSP visibility interface (SPEAD) Functionality supported at AA1 scale: - Visibility data Ingest (receive & pre-processing) - Generate Ingest QA metric data - Store data on disk in suitable format (i.e. MS) for processing with existing (3rd party) tools - Process data with existing (3rd party) tools for ITF signal displays (subset of commissioning displays). Milli-SDP (0.1% of the size of the final system) hardware deployment at ITF		
Performance Verification 5	Demonstration of execution engine(s) performance at full system scale, with a focused test that tests scheduling, task-startup, memory management and data transfer overheads. This will demonstrate execution engine performance at full scale, against a model of how fast the execution should take given the available hardware. As a sub-system test, it will not use real computational components, or other sub-systems such as the buffer.	Learning: sub-system performance	Q1 2022
Functional verification at modest scale	End-to-end processing from data block to calibrated image data with simplified calibration & imaging requirements.	Learning: System functionality	Q2 2022
Performance Verification 6	Test (or investigate) the architectural decomposition of a hierarchical scalable system. Test to see if the scalability within data islands and between data islands is as required for scalability to the full system. Use a workflow that exercises different communication requirements, e.g. Direction-Dependant calibration features.	Learning: Complete System & Scalability	Q3 2022
AA1 product handover	Provides complete commissioning and ITF/AIV support functionality including functionality required to support AA2. Certain external interfaces: - Basic TM M&C interface (TANGO) - CSP visibility interface (SPEAD) In addition to the functionality described in the ITF QE product handover milestone, the following functionality is supported AA2 scale: - Visibility data Ingest (receive & pre-processing)	Fixed Date	14/01/2023

	- Generate Ingest QA metric data
	- Basic real-time calibration
	- Store data on disk in suitable format (i.e. MS) for processing with existing (3rd party) tools
	- Process data with existing (3rd party) tools for:
	  - commissioning signal displays
	  - basic imaging
	  - basic time series analysis
	
	Milli-SDP hardware deployment in CPF (Mid) and RPF (Low).		

Performance Verification 7	Testing scalability of interactions between SDP performance components with a workflow including at least DD CAL and MFMS features. The focus of this milestone will be the efficiency of interaction between the execution engine, model databases, data queues and the batch processing.	Learning: Complete System & Scalability	Q2 2023
Performance Verification 8	Demonstration of whole system performance, at the full required scale, when running the ingest, real-time calibration and fast imaging pipelines. This is a separate (and earlier) milestone to (PV10) since these features have quite different requirements to (PV10) and are likely to be used at their full scale earlier in the scientific operations of the telescope. Although a system test, this milestone mostly tests the performance of the real-time processing components.	Learning: Complete System & Performance	Q4 2023
Software suite for hardware vendors	Software suite for SKA to evaluate (benchmark) hardware. Required for procurement of AA4 hardware therefore software needed 12 months before AA4 SDP deployment.	Fixed Date	03/2024
Performance Verification 9	Demonstration of whole system linear (or near linear) scalability up to full required scale with a complex feature set, e.g. MFMS synthesis with DD calibration. This is the scalability (but not performance) precursor to milestone PV10.	Learning: Complete System & Scalability	Q2 2024
SPC MID integration	Integration of SDP system with solution level once the long distance SaDT links (from site to SPC) are available (AA2). This would be the first time SDP systems are not running at ITF or on site, but in the SPC. This is ahead of SDP	Learning	Q2 2024

	hardware deployment for AA3 at the SPC. Examine issues with respect to messaging, packet loss, etc.		
SPC LOW integration	Integration of SDP system with solution level once the long distance SaDT links (from site to SPC) are available (AA2). This would the first time SDP systems are not running at ITF or on site, but in the SPC. This is ahead of SDP hardware deployment for AA3 at the SPC. Examine issues with respect to messaging, packet loss, etc.	Learning	Q2 2024
SDP AA3 s/w product handover	Sufficient functionality to perform real-time calibration and basic imaging workloads in the context of a system controlled by TM with appropriate control, monitoring and reporting. Minimal set of functionality for the AA3 release of the SDP Operational System to work at AA3 scale.	Fixed Date	5/9/2024
Centi-SDP H/W ready (AA3)	Integrate Centi-SDP hardware deployment (1% of the size of the final system at the MID & LOW SPC) at solution level & integrate SDP software with centi-SDP hardware before AA3.	Fixed Date	5/9/2024
Performance Verification 10	Demonstration of performance of whole SDP system at the required scale and performance level when running a spectral line imaging pipeline. This is a natural precursor to PV11 below and while this will not test the intricacies of data movements and scheduling of the most complex pipelines it will be a good whole system throughput test. Although this tests the complete system, the workflow chosen will mostly exercise the Buffer component and the efficiency of the computational tasks. The stress on the execution engine will be small.	Learning: Complete System & Performance	Q4 2024
SDP 1st operational deployment. (AA4 product handover)	Integrate SDP hardware deployment (at the MID & LOW SPC) at solution level & integrate SDP software with SDP hardware for AA4. Full software and hardware functionality and performance available at both SPCs.	Fixed Date	4/3/2025
Performance Verification 11	Demonstration of SDP whole-system performance at the required scale and performance level and with a full feature set. E.g.: full-scale MFMS synthesis with DD calibration stages; full-scale fast-imaging and	Learning: Complete System & Performance	Q3 2025

	calibration test. This is the final performance milestone: essentially a complete demonstration that SDP is achieving the performance that is required (or expected).		