SDP CDR PANEL REPORT

Document Number	SKA-TEL-SKO-0001013
Document Type	REP
Revision	01
Author	SDP CDR Panel
Date	2019-01-31
Document Classification	Unrestricted
Status	Released

Name	Designation	Affiliation	Signature	
Authored by:				
Maurizio Miccolis on behalf of SDP CDR Panel	SDP CDR Panel Secretary	SKAO	Maurizio MM	
			Date:	2019-02-08
Owned by:				
Nick Rees	SDP CDR Panel Chair	SKAO		
			Date:	2019-02-08
Approved by:				
Nick Rees	SDP CDR Panel Chair	SKAO		
			Date:	2019-02-08
Released by:				
Nick Rees	SDP CDR Panel Chair	SKAO		
			Date:	2019-02-08

DOCUMENT HISTORY

Revision	Date Of Issue	Engineering Change Number	Comments
A	2019-01-18	-	First draft release for internal review
01	2018-01-XX		Final version

DOCUMENT SOFTWARE

	Package	Version	Filename
Word processor	GoogleDocs	Jan 2019	SKA-TEL-SKO-0001013 SDP CDR Final Report
Block diagrams			
Other			

ORGANISATION DETAILS

Name	SKA Organisation
Registered Address	Jodrell Bank Observatory Lower Withington Macclesfield Cheshire SK11 9DL United Kingdom Registered in England & Wales Company Number: 07881918
Fax.	+44 (0)161 306 9600
Website	www.skatelescope.org

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AAn	Array Assembly n
ARL	Algorithmic Reference Library
ATAM	Architecture Tradeoff Analysis Method
CASA	Common Astronomy Software Applications
CDR	Critical Design Review
CSP	Central Signal Processor
ICD	Interface Control Document
ITF	Integration and Test Facility
LFAA	Low-Frequency Aperture Array
OAR	Observation-Action Register
PBS	Product Breakdown Structure
PDR	Preliminary Design Review
SAFe	Scaled Agile Framework
SB	Scheduling Block
SBI	Scheduling Block Instance
SDP	Science Data Processor
SEI	Software Engineering Institute
SIP	SDP Integration Prototype
SKA	Square Kilometre Array
SKAO	SKA Project Office
TM	Telescope Manager

1 Introduction

1.1 Purpose of the document

This document is the Panel Report on the SDP CDR process. It provides a record of the meeting, including:

- an overall summary of the panel's opinion on the status of the design
- a discussion of the Observations raised during the review of the documentation,
- a description and summary of the ATAM process used to review the SDP architectural quality
- the conclusions reached by the panel, and
- a recommendation to the SKA Programme Board as the correct action to take as a result of these recommendations.

1.2 Scope of the document

The scope of this document is to report on the panels findings as a result of the SDP CDR process. The process included two face to face meetings between the panel and the consortia:

- An informal meeting to discuss the SDP prototyping efforts held in Cambridge on 22nd and 23rd November 2018.
- The formal CDR meeting at the SKAO Headquarters between the 14th and the 18th of January 2019.

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.

[AD1]

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.

[RD1] SKA-TEL-SKO-0000041 - The Organisation and conduct of SKA Element Design Reviews
[RD2] SKA-TEL-SKO-0000652 - Critical Design Review Definition
[RD3] SKA-TEL-SKO-0000961 - SDP Critical Design Review Plan

3 Review Summary

Note: This summary has been prepared by the external panel members and is written from their perspective as external to the project. However, their viewpoints have been endorsed by the remaining members of the panel.

The Review Committee commends and thanks the SKA SDP team for their excellent preparation and support of the Critical Design Review. The material under review and supporting documents embody an impressive increase in the level of maturity of the SDP architecture since PDR and pre-CDR. The SDP Architecture seems to support requirements well with no obvious gaps. The prototyping has been extensive and highly informative. The Construction Plan reflected an excellent depth of analysis in terms of incremental and agile development, scheduling to address dependencies and SDP system complexities, as well as the integration of SDP into SKA overall. The Basis of Estimate in the plan seems appropriate and as complete as it can be at this stage of development.

We further want to commend the team on their depth of knowledge of the SDP architecture and openness in addressing our questions and issues. There was virtually no occasion where the team did not immediately and completely answer our questions; this was particularly notable in the discussion of ATAM Scenarios. They were able to provide both high level and detailed answers as necessary. The team worked seamlessly together and they are clearly comfortable with their individual roles and with each other.

A number of issues that would traditionally be addressed in CDR have been carried forward to the bridging activity (or later) and the project must ensure that these are addressed, and reviewed by technical and programmatic experts as appropriate. Each of these has been described below in our findings, comments, and recommendations. Some of these are not strictly SDP gaps, but system level gaps that have impact on SDP scope or design (e.g. Security, SKA Common, SRCs).

4 Detailed Review Evaluation

4.1 Review Structure

The review was held in two major phases:

1. A “documentation review” phase based on reading, understanding, and finally raising Observations against the SDP documentation package (30 documents). This part was aimed to understand and assess the suitability and risks associated with SDP design, before entering into the System Critical Design Review process, and eventually the SKA Construction. Scenarios that exercised the SDP architecture for gaps, risks, and qualities were also devised in this initial phase. Reviewers logged Observations in a Jira-based Observation Action Register (OAR).
2. A face-to-face meeting, divided in two parts: first, an analysis of the suitability of SDP software architecture to meet the needs of its stakeholders, conducted using the Software Engineering Institute (SEI) Architecture Tradeoff Analysis Method process, and based on the previously generated scenarios; and second, discussion of major Observations made against the documentation pack.

There was also a separate meeting held in November 2018 to discuss and demonstrate the prototyping that was done as part of the development.

4.2 ATAM

This section presents the results of the architecture evaluation of the SKA Science Data Processor (SDP) software architecture, which took place at SKA Headquarters, Jodrell Bank, on January 15-16 2019, as part of the Critical Design Review (CDR) for the SDP Element. The evaluation was performed by the SKA Office under the guidance of a facilitator, and followed a tailored version of the SEI Architecture Tradeoff Analysis Method® (ATAM®) process, which is a method for evaluating a software system's architectural decisions in light of desired system quality attributes.

Nearly all internal and external reviewers had previously received ATAM training, and where they hadn't, custom training was conducted.

The ATAM is a stakeholder-centric method—the evaluation team facilitates a process that helps stakeholders ask the right questions about the architecture. The process uses *scenarios* to represent stakeholder interests and understand quality attribute requirements. A scenario is a concise description of how the system, operating in a particular environment or context, should respond to a specific stimulus.

A planning meeting in November 2018 derived early ideas of scenarios to focus on. On January 14, just before CDR, the review panel met to finalize new scenarios to use. The focus was on scenarios that addressed risk themes from the pre-CDR report, as well as outstanding concerns. As a result, the panel identified 24 scenarios. These were consolidated to eliminate redundancy, and then prioritized along two dimensions: SKA Office prioritized scenarios according to importance to achieving mission goals, and SDP architects prioritized according to architecture impact and difficulty to realize the scenario. Based on this prioritization, thirteen scenarios were analyzed. The detail of the scenarios are in Section 6.3.

Analysis of the scenarios consisted of having the SDP architecture team present the details of how the architecture allowed or did not allow the system to achieve the desired response to the stimulus. During the analysis, the evaluation team identified and captured *risks* (potentially problematic architecture decisions), *non-risks* (good architecture decisions), *tradeoffs* (architecture decisions that promote one or more quality attributes at the expense of other quality attributes), and *sensitivity points* (a property of a system element that is critical for achieving the desired quality attribute response).

The following artifacts provide more details about the evaluation:

- Presentation with more details about the review process: https://docs.google.com/spreadsheets/d/1yACQPZWOwC6cfFeiLQAOmp548juQQm_eQ4FXAUxlFVk/edit#gid=769702257
- The full list of scenarios generated, with priorities: https://docs.google.com/spreadsheets/d/1-rBhCxTLbOfFwxDuoqRo6YzzJCjfXPGcpNKMrJvOjCM/edit#gid=0
- Detailed findings and notes taken during the scenario analysis sessions, which provide context and document the discussion leading to each of the risks: https://docs.google.com/document/d/1m659NBUYvjYPg4isb6qx6MvvNPrCM7aSFzF2s2tPtPs/edit#heading=h.8mffl3numu3i

4.2.1 Non-Risks

In the ATAM process, non-risks represent good architecture decisions. In the SKA context, these are especially important to explicitly capture since software detailed design and development is deferred to the Construction Phase. The identified non-risks are:

NR9.1	Slow delivery will not impact observing
NR6.1	The combination of logical data islands and preparation step allows for redistribution of data prior to processing
NR23.1	The data storage abstraction is sufficiently flexible
NR1.1	The prototyping effort is giving a path through an incremental development plan. It is documented in the Integration prototyping document
NR12.1	Significant expertise within LOFAR and MWA
NR16.1	A stripped down version of SDP can be deployed easily on a laptop

4.2.2 Risk Themes

The evaluation initially identified 24 risks—potentially problematic architecture decisions that may result in the system not achieving the desired quality attribute responses. These risks are grouped into 9 risk themes, with each risk assigned to one or more of the themes in the following subsections. The detail of the risks is shown on Section 6.4. Since the ATAM process is a sampling process some aspects of the Pre-CDR review didn't show up in the sampling done for the CDR. A specific area in question was the TM-CDR interface. However, a later section in this report comments on all Pre-CDR recommendations, and so these themes have not been lost.

As for the Pre-CDR all risks identified in the ATAM have been added to the SDP CDR Jira project (see https://www.google.com/url?q=https://jira.skatelescope.org/issues/?filter%3D12506&sa=D&ust=1548843758561000&usg=AFQjCNHmJmhZZS86hHIx2o-71S0aCzCn2A) and will be triaged and addressed as part of CDR closeout.

4.2.2.1 System-SDP boundary

The ATAM identified 9 risks that pertained to the division between system and SDP responsibilities. There are certain areas where the SKA System itself has undefined design that might impact the SDP. For example, since there is not yet a System-level security model or process, SDP itself cannot finalize a security model. Similarly, the SKA Common software development environment is properly managed by System, yet this will be an important aspect of the SDP. This impacts all areas of the SDP software architecture.

4.2.2.2 Documentation

This risk theme captures 11 risks caused by documentation gaps in the SDP CDR documentation.. For example, how exactly the RPC protocol works, or the new prototyping insights from SDP DaLiuGe experiments, exist in SDP collective knowledge, but have not been captured for SKA itself. This impacts all areas of the SDP software architecture.

4.2.2.3 Security and Data Integrity

One scenario, R29, was specifically created to test security documentation. This resulted in 6 identified risks. In general, SDP's architecture does not inhibit security practices, but no explicit security approach has been selected. This risk is related to the system-level risk theme, since security is not solely an SDP responsibility. The second aspect of this is the concern about long-term data integrity, i.e., bitrot, bit-flips at SKA data volumes. The data integrity challenges at this scale are unknown. This impacts security, integrity of the SDP software architecture.

4.2.2.4 Portability

This theme captures 2 risks that the architecture, and more specifically the software of SDP, will be ported to SRCs, but this mechanism has not been elaborated. Similarly, parts of SDP will need to be made available to developers for local testing of workflows and processing. This impacts the testability, portability and modularity of the SDP software architecture.

4.2.2.5 Testing

The ATAM found 4 risks around testability. Being able to simulate and test critical aspects of the SDP is crucial. However, testing will have certain gaps, in particular with respect to the very large data volumes. Simulations in prototyping have used emulations of the actual CSP hardware, and the performance and behavior of the actual hardware, e.g., FPGAs, will be different. SDP behaviors at the full SKA data volumes is necessarily unclear. This impacts testability of the SDP software architecture.

4.2.2.6 Monitoring

There were 4 identified monitoring risks. There is a monitoring component in the SDP architecture, but the specifics of what needs monitoring, when it should be monitored, how those logs are stored, etc., are unspecified. For example, if a rogue Execution engine or workflow occurs, are there profiles of well-behaved workflows that will support detection of this problem? The current documentation does not make this explicit. This impacts performance and monitorability of the SDP software architecture.

4.2.2.7 Implementation

The Implementation theme captures 10 risks, most generally a belief that the SDP architecture is a fairly complete, yet conceptual architecture. There is a need to begin implementation in order to make the architecture concrete, to begin understanding technology selection and its potential impact on the architecture. For example, how will data access happen in practice? This impacts performance and buildability of the SDP software architecture.

4.2.2.8 Complexity

The SDP architecture is complex. However, it is not clear in the architecture documentation whether the architecture is complex because the problem requires that complexity (i.e., is itself complex), or whether there is a risk of over-architecting. This resulted in 2 risks. This is particularly concerning in identifying aspects of the architecture that could be de-scoped. Another risk is of hidden coupling, made more likely by the complexity (e.g., when two logically isolated components share a common 3rd party library). This impacts modularity and buildability of the SDP software architecture.

4.2.2.9 Recovery

The SDP will have crashes and failures. 5 risks identified a gap in how recovery would be managed. For instance, if a link to an SRC fails, there is no way to directly dump or restore data to SDP. Reloading Processing Blocks from TM on restart is not captured in the ICD. This impacts recoverability and fault-tolerance of the SDP software architecture.

4.3 Scientific summary

4.3.1 Performance

The SDP architecture appears to be sufficiently robust and flexible enough to meet all of the principal science requirements. There remain some areas of concern, however. The most important of these is direction-dependent calibration. This is a particular concern for SKA-LOW, where the combination of a rapidly-varying, anisoplanantic ionosphere and potential uncertainties in the station beams makes the problem extremely complex, but also affect SKA-MID (e.g. in the context of pointing self-calibration). The recent development of the Model Partition Calibration workflow as documented in SDP Memo 97 (made available at the review) is a good step forward, but scaling to large datasets has not been addressed and is not yet reflected in the performance model. It will not only be necessary to import and evaluate algorithms developed in the community (particularly for the precursor arrays) but also to ensure that their implementation in SDP is adequately supported (Section 3.1.5).

Some areas need additional attention to ensure that requirements are clearly specified and that a detailed design will meet all of the performance criteria. Single-dish observing needs some elaboration, although data rates are low. Real-time calibration also needs more attention in order to make sure that all latency requirements are met. In neither case are there any obvious architectural implications.

4.3.2 Commissioning

The SDP preCDR established that further work was needed during bridging in order to clarify the precise requirements for the SDP commissioning support system. The basic concept of ingesting data from CSP and writing it out in measurement-set format for later analysis by existing packages will certainly meet most of the initial commissioning needs (ITF and AA1), but development of real-time calibration (e.g. recording of pointing data for MID and complex gain solutions for beam-forming on both arrays) still needs to be addressed. Provision of real-time diagnostic displays should also be covered.

The Panel was concerned that there might be a problem towards the end of AA2 in coping with the data volume using the commissioning support system (hardware and/or software) at the same time as testing the operational system for the first time. Discussion at the CDR suggested that early deployment of the operational system could mitigate this risk.

The Panel pointed out that premature automation of workflows might cause problems during commissioning, for example by causing raw or intermediate data products to be deleted before processing has been validated (Section 3.1.6). The recommendation here is to maintain flexibility, with the option of keeping data products and additional diagnostics to support debugging.

As noted earlier (Section 3.1.1), one reason for early deployment of SDP at the SRC's would be to provide additional hardware resources for commissioning. This will be particularly important in the later stages when there is a continuing need to reprocess visibility data as workflows evolve.

4.4 Prototyping demonstration

On 22-23 November 2018, a face-to-face meeting was held between consortia members and a subset of the panel (Nick Rees, Robert Laing, Marco Caiazzo and Antonio Chrysostomou) to discuss and demonstrate the prototyping work that underpins the SDP architectural development. From the panel's perspective this gave a valuable in a number of areas including gaining an insight to the work that has been done and also meeting many of the developers who have not been represented in the major reviews. Our chief observations on this were:

1. The team is to be commended on the level of prototyping work. By and large it appeared to be appropriate for this stage in the project life-cycle.
2. There was a commendable range of backgrounds in the room - from computer science to mathematical to astronomical domain expertise. They appeared to work well together and we believe that this multidisciplinary approach will be effective during construction.
3. The prototyping included vertical (e.g. specific implementations of gridding algorithms) and horizontal (system level Integration) prototyping, as well as prototyping of specific areas (e.g. DASK and DALiuGE execution engines).
4. The SIP and ARL prototyping had been used to iterate and develop some major interfaces and this was to be commended. However, documentation of the SIP interfaces was lacking, so this should be done to avoid the knowledge being lost.
5. the ARL serves an important function in defining the reference implementation, but the kernels will almost certainly have to be implemented in some performance oriented language (e.g. C/C++/CUDA) which may be hardware-specific. The panel was therefore concerned that interfaces to ARL processing routines might have to change significantly if they were re-implemented for optimal performance, thus losing some of the advantages identified in item 4.
6. Both DASK and DALiuGE execution engines appeared to have a significant amount of infrastructure to diagnose running pipelines, and this will be important for any final system.
7. The vertical prototyping highlighted that there are tradeoffs between bandwidth and computation that can make a large difference in performance between different versions of the same algorithm. There will be a significant amount of work in this area to improve performance when the hardware architecture is more mature.
8. It will be important to define some reference, representative datasets for performance comparisons: at the moment it is difficult to compare benchmarks.
9. More attention needs to be paid to prototyping workflows in areas where significant recent algorithm development has taken place, as there may be architectural implications. The recent work on direction-dependent calibration is a good start.

4.5 Observations

During the documentation review period 337 Observations were raised against the 30 documents delivered (to simplify the review process the architecture document has been handled according to its 11 individual components).

- 105 Major
- 190 Minor
- 42 Typos

About 70% of the observations were discussed and the solutions agreed before the CDR meeting.

Among the remaining ~100, 75 were selected for discussion at the face-to-face review, based on the criteria that they were either:

- A major issue that had not had a conclusion previously agreed with the reviewer who made the observation or
- An issue that had been explicitly marked by the reviewer or the consortium for further discussion.

These were then further sub-categorised, based on the document the observation was made about, into four categories for reviewing during the face-to-face sessions:

- Prototyping documents (11 observations)
- Architecture documents (34 observations)
- Programmatic documents (18 observations)
- Other documents (12 observations)

The panel and the team agreed a resolution to all the tickets that were discussed at the meeting.

This left a small number of minor and typo observations that were not discussed at the meeting and still had no conclusion agreed. Nevertheless it has been agreed with the SDP Consortium, that their proposed solutions are going to be agreed by the 25th of January.

At the time of writing (23 Jan 2019, 5 days after the meeting), this amounted to 14 observations, 4 of which had conclusions proposed by the consortium but not yet agreed by the observation reporter.

After the completion of the ATAM, the identified risks (9 of which are system level risks) have been recorded into a total of 25 OAR tickets. These will follow the same workflow of all the other OAR tickets, relying on input from the SDP Consortium before they can be mitigated or transferred to the system level.

All the actions related to the SDP baseline documents, emerging from those observations and from the CDR discussions are going to be completed, including re-issuing of the documents, by the 15th of March. This will leave 2 weeks, before the CDR closeout date, for the SKA Office, to complete the SDP Baseline and store it in the Configuration Management repository.

5 Responses To Specific Panel Charges

The Panel was asked to respond to 15 specific charges and they are detailed below:

- Is the submission complete, including all the documents called out in the Statement of Work and successive amendments done via ECP?
    - Submission is complete as agreed with the consortium. Not all changes to the specific documents in the original statement of work were explicitly amended by ECP
- Have the all artefacts associated with the architecture been delivered and are under suitable configuration control?
    - Yes
- Is the SDP architectural design, including interface descriptions, sufficiently mature to satisfy the allocated L1 Requirements without gaps?
    - The are some noted deficiencies in the ICDs, but otherwise the architectural design is sufficiently mature to satisfy the allocated L1 requirements. There are some gaps in the L1 requirements that have been noted by the consortium as L2 requirements that are not traceable to L1s
- Has the SDP architecture and development plan being analyzed to demonstrate its testability, integrability, among its components and with other software developments within SKA (e.g. TM, INFRA, …) at intermediate steps in Construction?
    - Yes
- Has the SDP baseline architecture been established and documented to enable software detailed design and coding to commence?
    - Yes. The architecture is not suitable for a defined deliverable contract, but detailed design and coding can commence with realistic risk management within the SAFe framework.
- Is SDP maintainable, according to its plans, along its whole lifetime?
    - Yes
- Are the SDP artefacts delivered for the CDR suitable to be developed, in Construction, according to the Scaled Agile Framework (SAFe)?
    - Yes
- Are the remaining risks in construction understood and identified, associated to proper cost contingencies, such they can be properly endorsed by SKAO in Construction?
    - There are significant uncertainties in costs and so the contingencies associated with the level of risk are optimistic. Cost uncertainties come from the cost and efficiency of hardware components at the time of procurement, but also on the computational cost of SDP workflows. The latter has been modelled parametrically, but significant uncertainty remains on the input parameters. The understanding of both aspects need to mature further to reduce the cost uncertainties.
- Is the construction schedule executable and in line with the overall SKA Project Construction Plan?
    - Yes, subject to addressing the staff ramp-up and continuity issues cited by the panel.
- Is the SDP software development in Construction executable according to the SDP Construction Plan and in line with the existing budget?
    - Yes
- Is the SDP associated hardware detailed design timely procurable within the Construction budget?
    - The hardware is presented as a conceptual architecture, not a detailed design and this is appropriate for this stage of the project life-cycle. Presenting a detailed design of the hardware architecture at this point would be unrealistic, both in cost and deployment terms, and so would be of little value. Consequently, there are assumptions about Moore’s Law which carry significant risk, and there are

unknowns relating to the suitability of future architectures to meet the projects needs. A detailed plan needs to be put in place to manage these risks. The plan should build on the strategies outlined in Appendix A of the Cost Basis of Estimate.

- Are all Critical Safety Items (if any) identified?
  - No critical safety items were identified.
- Have key SDP characteristics, having the most impact on system performance, assembly, cost, reliability, sustainment or safety, been understood and identified?
  - Yes
- Has the SDP cost model been updated and allocated to the SDP components and the SAFe methodology to show correspondence and feasibility?
  - Yes
- Have all Observation Action Register items for both Pre-CDR and CDR been closed out or if not closed out, are there credible short-term plans to close them out?
  - Yes. The remaining Observation Action Register items should be closed out by the time the consortium dissolves.

6 Recommendations

6.1 SDP - SRC Interfaces

SDP Is required to deliver Observatory Data Products to the SRCs. Further, in the Construction Plan p. 8 it states “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.”

And, on p. 9 “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.”

The lack of a formal SDP – SRC ICD prevents a comprehensive elaboration of the DELIV component of SDP, as well as features in or characteristics of SDP that would enable effective use the SDP instances in SRCs.

Recommendation: Develop a “reference” SDP – SRC ICD in Bridging (even in advance of complete clarity within SKA on the funding, number, location, and capabilities of the SRCs) as a design and planning vehicle, and maintain it until it can be baselined.

6.2 Staffing Ramp Up

The Construction Plan Figure 3 Staffing profile assumes ~42 people on staff at C0. It is not clear this is feasible or realistic since it depends on the level of early staffing that will be authorized and how many people will transition from the Consortium/Design phase to Construction, providing needed continuity.

Recommendation: In Bridging, incorporate a "ramp up" period in the Construction schedule to account for staff increasing over the initial period of the Construction phase. Identify a core of staff that will transition from the Consortium into Construction.

6.3 End-to-End Tests

The Construction Plan contains a well-defined set of Learning, Program Increment, and Fixed Date milestones. The plan takes into account logical dependencies within the SDP architecture, and maps the SDP development into the overall SKA program schedule in order to ensure SDP capabilities needed at program milestones are provided. These milestones are sufficient to manage the SDP development, integration, and commissioning.

While the development of the algorithms is planned and will proceed in parallel with the development of the non-domain services of SDP, there are integration points at which it will be informative to conduct an end-to-end processing of a representative volume of pre-cursor or simulated data, with the algorithms developed to date, in the actual SDP software environment.

A particular value of this type of testing is to enable assessment of the scientific quality of the algorithm output, and the computational performance of the algorithms as executed in the actual SDP system. This enables tradeoffs to be made between those two dimensions and can uncover modifications to the architecture that optimize across these dimensions.

While much of this will occur when the AA3 antennas are available and beyond, with actual SKA data, it is valuable to do this type of testing with pre-cursor or simulated data prior to that point.

Recommendation: Before SDP closeout, the SDP team should incorporate at least one or two "end-to-end" Test Milestones into the construction plan to test algorithm scalability and quality, and to identify desirable/necessary algorithm-driven modifications or enhancements to SDP non-domain services.

6.4 Architecture

The SKA SDP team has developed a conceptual architecture, which while complex in comparison to existing radio astronomy processing and archiving systems, is appropriate to the scope and scale of SKA SDP. The architecture contains features to address the functional, performance, and growth requirements, is well documented, and has been comprehensively exercised analytically, and by prototyping in key areas.

The scaling and performance aspects of the problem space <u>may</u> require the full architecture as currently designed, however simplification may be possible without adversely affecting the delivered performance. We endorse the planned development of an initial simple system (the minimum viable product, MVP) and the realization of the full architecture incrementally through the SAFe process adding complexity only when it delivers needed value.

We believe that further work at the abstract/conceptual level is not likely to significantly advance SDP readiness for construction, except in isolated "gap" areas (mostly in external interfaces e.g. the lack of flowed down System requirements for Security, the lack of a formal ICD with SRCs, lack unified specifications for SKA Common, including Observatory Support Tools). Only by down-selecting technologies, implementing SDP, and testing it, will the architecture be truly validated.

Recommendation: Begin concrete construction activities in Bridging.

6.5 Algorithm Development

The refinement of domain algorithms for a specific telescope or observatory is a process requiring very specialized skills. Currently no resources have been allocated for advancing the technical readiness level of algorithms required by the observatory, nor for the tailoring of these algorithms for application for the SKA telescopes.

While the larger research community will continue to develop and evolve algorithms, they may not address the specific challenges of the SKA.

Recommendation: Add staff to the project plan dedicated to the advancement and tailoring of algorithms for the SKA, before SDP closeout.

6.6 Commissioning and Early Operations

The SDP design provides a highly efficient and automated system suitable for routine operations. Automated data transfer between the Buffer and LTS components as well as the automated mapping of data to processing elements are examples of this type of optimization. While this automation is appropriate and necessary for steady state operations, this automated behavior may be counter-productive during commissioning and early science activities.

Recommendation: Ensure that the system supports non-standard use cases through suppression of undesired automated behaviors, in Construction.

6.7 Reliability, Availability, Maintainability

Given the conceptual nature of SDP at this point, full FMECA Analysis is not practical or desirable, but additional failure modes analysis in selected areas, such as the Buffer, is desirable prior to System CDR. The actual System CDR deliverable in this area needs better definition.

Recommendation: Define the RAM deliverable in Bridging and schedule analyses necessary to produce it.

6.8 SKA Common

The SDP Consortium classified some software in their design that was shared with other parts of the system as “SKA Common”. This term has been used by others to represent different software and is already a named part of the PBS. No detailed design exists for some SDP related aspects of SKA Common, including Observatory Support Tools. So, while it is a clearly defined Construction PBS element, there is no team with assigned responsibility for this design. As a result, there may be budget gaps for deliverables that are assumed to be at system level but are not in any of the consortia silos.

Recommendation: The SKA must develop the design of all software that is shared across the system during Bridging. This includes the software that the SDP has called "SKA Common" as well as that defined by other consortia (particularly TM).

6.9 Security

SDP is a complex software and hardware system, but not unlike similar systems for which known, best practice security processes, models, and technologies exist. The team has demonstrated via discussion and documentation (Security View document) that they are aware of the security elements. The SDP architecture has no clear flaw in terms of security.

However, the System level Security program and requirements have not yet been fully elaborated, and therefore not flowed down to SDP. The team appears able and committed to implementing the system level security requirements when those are specified (see risks).

Recommendation: The SKAO should define the System level Security program and requirements, based on existing formal standards and processes (e.g. NIST, TrustedCI), and flow those requirements down to SDP in Bridging.

6.10 Adoption of the SAFe process

The project wide adoption of the Scaled Agile Framework (SAFe) addresses many of the issues of the more traditional waterfall methods. This process provides mechanisms to balance architectural and scientific priorities, and is designed to deliver maximum value to the stakeholders. Acceptance and buy-in across the project seems strong.

The SKA Project is pioneering the use of SAFe in a large scientific project, and there are associated risks. While the methodology guarantees the delivery of maximum value it does not ensure a viable product.

Recommendation: The project should be careful to routinely evaluate the efficiency of SAFe in addressing issues of varying types, and not assume it is a "magic bullet" for all issues.

7 Progress since Pre-CDR

Due to the nature of the review process the areas covered in the Pre-CDR and the CDR differed somewhat, and there are still some outstanding issues from the Pre-CDR. The Pre-CDR panel made 18 explicit recommendations, which are given briefly below, and the common members of both panels made the following observations about them.

1. Before CDR, the Consortia, working with the SKA Office, produces an clear process or plan to manage the hardware technology risk.
    - This topic is discussed in appendix to the Basis of Estimate document.
2. Where the design calls for use of off-the-shelf software solutions, the SDP CDR should present named examples of software that has been analysed, or preferably prototyped for use.
    - Prototypes based on actual off-the-shelf software solutions have been demonstrated as discussed in Section 4.4
3. The SKA project should explore the development of a common, cross-element compute platform.

- This is an action on the SKA project and will be taken up as part of the overall harmonisation work during bridging.

4. The SDP work with the SKA System Scientist to select a limited number of algorithms for further analysis and/or prototyping.
    - It was agreed to prototype a Direction Dependent calibration routine and this led to the development of the new MPC generalised calibration algorithm. A report on this work was made available during the CDR meeting represented satisfactory progress, but there is more work to do.
5. SDP enhances the Parametric Model to allow an estimate of the arithmetic efficiency of the different stages of each algorithm be included.
    - This activity is planned for the bridging phase
6. To complement the previous recommendation, the SDP should continue the ongoing work to estimate the arithmetic efficiency of major parts of the key algorithms.
    - This activity will also continue in the bridging phase
7. The SDP works with a resolution team from the office to develop a canonical model for SDP utilisation in the early years of operation.
    - This work is still outstanding and will be part of the ongoing refinement of the construction plan, and will depend heavily on the deployment baseline.
8. The SDP and TM consortia should release the next version of the SDP to TM ICD (version 4) as soon as possible, and before the dissolution of the TM consortium.
    - An updated version of the SDP to TM ICD has been presented at the review
    - The document will continue to be maintained during the bridging phase
9. The SKA Office should convene a Resolution Team to harmonise the concept of resources and capabilities across the system.
    - It is ongoing as one of the bridging tasks addressed by SAFe Agile teams
10. The SDP should work with the office and people who are currently part of the TM consortia to further update the SDP to TM ICD
    - Same as #8
11. The SDP Consortium should work with the System Scientist to develop a set of requirements for software required for commissioning and array calibration, particularly in the early phases of deployment when only the commissioning support system is available.
    - This activity is planned for the bridging phase
12. The SKA Office translate the ATAM risks observations into OAR Observations.
    - Done
13. The consortia work with the office to agree resolutions for all outstanding observations (which will be Minor, Typo and ATAM observations).
    - Done
14. Any resolution that requires work by the SDP consortium should be resolved before SDP CDR.
    - Done
15. The SDP consortium completes the list of views that they are planning to incorporate into the final documentation.
    - Done and reviewed in this CDR
16. The SDP should pay particular attention to updating the SDP Architectural Overview so it is understandable from outside the SKA family.
    - Done and reviewed in this CDR
17. The CDR submission should include a clear rationale for architectural decisions.
    - Done and reviewed in this CDR

18. The SKA Office should work with the SDP consortium to generate a first version of the SDP-SRC interface.
    - This activity is planned for the bridging phase

8 Appendices

8.1 Review Process Improvements

The SDP CDR review has been carried on in a shared effort involving a large group of people from all over the globe, agreeing on a single process, and supported by IT tools and meeting logistics.
The process adopted is the result of previous efforts carried on during TM CDR and SDP pre-CDR reviews, and as such it incorporates many suggestions and lessons learned. While some process optimization has been adopted during the review, a process retrospective analysis has been executed at the end of the meeting, using a simple Glad-Sad-Mad reporting technique. The retrospective was realized in the form of a Trello board[1] where each CDR meeting attendee could add card to the one of the Glad, Sad, or Mad columns, describing aspects of the CDR that have been successful, that might be improved or that really prevented from an effective participation. Below it is reported a summary of the main findings, divided by argument.

8.1.1 Jira OAR

The Observation Action Register (OAR) has been implemented using the Jira system provided by the SKA Office[2]. This system resulted extremely effective, enabling the organization of observations in discussion topics and keeping stats about the overall progress of the review. Some improvements in the workflow have been suggested:

- For OAR review, Nick had to act as moderator AND highly interested party AND scribe, which meant at certain times we went down rabbit holes.
- The workflow for JIRA tickets has been expanded as a result of the CDR meeting, in order to enable traceability of actions resulting from OAR tickets.

8.1.2 CDR Process

The process examined SDP under two different points of view with two different means: SDP architecture has been examined using the ATAM, other programmatic aspects of the documentation submitted have been analyzed using a more classical review process based on reviewers observations. It has been noted that, for external reviewers, the ATAM approach based on scenario analysis results easier to follow thus allowing them to contribute more effectively.

8.1.3 ATAM review

The ATAM technique resulted effective in highlighting the positive aspects of SDP architecture as well as in identifying the architectural gaps and the major risks and trade-offs presented by the proposed software architecture. The blame-less nature of the ATAM process itself contributed to promote a collaborative environment during the meeting, resulting in a better overall outcome. One of the criticalities that have been noted in following this process is that:

[1] https://trello.com/b/TOfTjRRQ
[2] https://jira.skatelescope.org/projects/SDPCDR

- ATAM has been only a qualitative exploration of the architecture. I wish to have more quantitative analysis to better understand the trade-offs identified.

8.1.4 IT support tools

Many tools have been used during the review and during the meeting in order to have a better collaboration, involving people on site and remote participants:

- **Confluence** and **Jira** have been very effective. Organizing Jira tickets into Confluence pages for further discussion produced good results.
- **Slack** was used for real-time messaging during the meeting. It helped remote participants to raise questions, links to relevant documents were posted on the chat. Sometimes a realtime chat helped in overcoming issues with the AV system.
- **Audio / Video** equipment of the council chamber has been used to a great extent in many possible configurations in order to have split projections of meeting notes and presentations at the same time. In more than one occasion during the meeting, the audio connection with remote participants was lost, not allowing remote intervention. Audio equipment failures in the council chamber often delayed meeting start.
- **Internet access** for externals was not managed and it was provided to participants only after the first meeting day. Being wireless access non reliable, it required IT to set up dongles for every attendee.
- **Vidyo** was used by remote participants to join the meeting. It is noted by some participants how this system is not supported on Linux.
- **Google drive** was used during the preparation of the ATAM review in order to collaboratively edit the necessary artefacts such as a preliminary list of scenarios. ATAM scribe notes have been edited using google docs during the meeting and shared in real-time. This enabled remote people to open the notes file in view mode to follow the meeting progress.
- **Trello** was used to realize the process retrospective. This was considered very effective.

8.1.5 CDR Meeting logistics

Meeting facilities and logistic support have been evaluated overall positively by on site participants. The adoption of the council chamber resulted in a less crowded space for the meeting, thus allowing a more relaxed atmosphere. Food and coffee facilities have been appreciated. Some consideration and suggestions for future improvements have been collected:

- Room sometimes a bit cold.
- Too many plastic water bottles. Could use water pitchers and glasses.
- Straddle one weekend, so external/overseas participants only lose one and not two weekends.
- Scheduling the review period over Christmas was not ideal.

8.2 SKA SDP Business Drivers

The following list of high-level principles are drivers and influencers for the architecture of the Science Data Processor (SDP). Some flow down from a selection of SKA Board Principles that are applicable to the SKA Observatory as a whole, while others derive from constraints and expectations on the system. In all cases, they have been interpreted in the context for the SDP design.

1. The SKA Observatory will be operated as a **single observatory**. Telescope specific functions will be deployed at the relevant sites, while common functions will be deployed at the GHQ. The SDP must thus support a distributed workflow encompassing one integrated observatory, two telescopes, operating on three sites.
2. The primary success metric of the SDP within the SKA Observatory will be to effectively support and enable scientists to pursue **world-leading scientific programmes**, to organise and conduct improvements and upgrades of the SKA telescopes in order to provide and maintain facilities that are at the forefront of science and technology.
3. The scope of the SDP will be to support the SKA Observatory to **commission and upgrade** the SKA Telescopes, and to **produce scientifically viable data products** from the telescopes.
4. The expected lifetime of the SKA Observatory is **50 years**. The SDP architecture will be designed to fulfill this expectation, allowing for the system to **grow and evolve** with minimum impact on the existing programme. Science data products must be **archived** for 50 years.
5. The SDP shall allow the SKA Observatory to operate at a **high observing efficiency**, taking advantage of its flexible design and managing the **high data rates and volumes** from the Telescopes.
6. The SDP design shall allow the SKA Observatory to operate for 24 hours every day in order to **maximise its scientific productivity and impact**. It will be responsive to a flexible observing programme to allow it to continue to be scientifically productive in the face of adverse circumstances (e.g. faults, etc.).
7. The SDP will enable the SKA Observatory to interface with an evolving, globally distributed network of **SKA Regional Centres**, delivering scientifically viable data products that enable the community to undertake world-leading scientific research programmes.
8. A goal of the SDP design will be to minimise the overall total **cost of ownership** (construction plus operations).
9. The SDP design will be constructed within a **construction cost cap** as agreed with the SKA Observatory.

8.3 ATAM Scenarios

The following scenarios formed the basis of the ATAM review of the architecture. A detailed analysis of each scenario can be found in the proceedings of the CDR meeting. First column list the ID associated with each one.

ID	Stimulus	Environment	Response

9	TM issues request to observe an imaging SB including reference pointing and SDP responds that the resources are available. Once the visibilities are processed by SDP into Observatory Data Products, they are archived into Long Term Storage and delivered to an SKA Regional Centre.	Normal Operations	The observing commences and the data are ingested, flagged, averaged, direction independent calibration ensues followed by a direction dependent-calibration imaging pipeline. The provenance is recorded and updated for all steps of the processing and the quality assessment is calculated and recorded throughout. Finally the calibrated Observatory Data Products are staged for delivery to SRCs, entered into the Data Product Catalogue, and archived into Long Term Storage.
4	A project requires the processing of visibilities using the SDP instance on the SRC platform, using new algorithms developed by the project team. This may include the processing of new and old visibilities into new products, and perhaps from different telescopes (e.g. single dish).	Commissioning and Operations	The SDP at the SRCs support generation of different instances allowing new algorithms and workflows to be combined with existing capabilities. The operations capabilities and staff must be able to support the entire workflow in collaboration with the SRC, while still permitting other large-scale observing programs.
6	The parameters for an imaging workflow need to be changed after the Scheduling Block has been submitted for execution. There are 3 possible variants to this scenario. The change to the Processing Block is required: (i) while the SBI is waiting in the queue to be executed (this is in TM's domain); (ii) after the SBI is executed/observed and the visibilities are calibrated and waiting in the buffer for processing; (iii) while the data are being processed, requiring that the processing be aborted and started again with the new PB.	Commissioning and/or Normal Operations	In each case, changes to the PB should be possible via an interface with TM. Note that only (ii) and (iii) are relevant for SDP – (i) is here for completeness only. For (ii), the scheduled processing of the PB by the SDP is deferred until the PB is updated at some point in the future. For (iii), the processing is aborted on command from the operator. The PB is updated at some point in the future and rescheduled by the SDP for processing. In each case, the SDP resource model is updated accordingly allowing TM to change the scheduled observing if necessary.

23	A new data processing algorithm is proposed The new algorithm is not scaling because of data access patterns.	Normal Operations	The observatory staff overcomes the limitation getting the scalability back, implementing the right data access pattern
27	Regional Data Centre connection is degraded for a long period of time.	Normal Operations	Telescope continues to observe with data going to LTS, minimizing impact on current schedule. Data catch-up while still operating
29	Unauthorized access is executed on the SDP	Operations	Access is detected and root cause analysis information is available to understand and close problem within 1 week.
1	Budget changes require that the SDP software system be developed for 25% less than orignially budgeted.	Early Construction	Where in the documentation do you discuss possible descope options that would permit this type of management.
12	During construction, a correlator upgrade results in an increase in the data rate within the known data rates, on the CSP-SDP link and the result is >50% packet loss on the link when ingested	Construction	Packet loss records are available and identify this as a problem in the ingest node. Update to the receive driver fixes loss problem within 2 week development time. In the meantime system is run at low data rates.
5	A transient event is confirmed following the receipt of data from the single pulse search in the PSS, falling significantly above the detection thresholds for an "interesting" event! The SDP schedule is significantly disrupted	Normal Operations, buffer fill pattern is currently non-optimal.	An alert is issued that triggers a follow up SB to be scheduled for execution (the SB has override status). An IVOA event is also broadcast that is picked up by the other SKA telescope and, potentially, other observatories. SDP is re-scheduled according to new priorities. The buffer is allocated accordingly.
15	Config database goes down.	Normal Operations	It is brought back up without major impact to operations.
16	A component is first to be implemented by someone locally on its laptop to first see how the component works on a small subset of the data. Once that works on a small subset of data, it needs to be implemented on SDP.	Commissioning	Developer has the tools, documents, interfaces etc to develop parts locally, and tools to test things locally. The architecture allows scaling up of the component without major reimplementation.

20	A new approach to container management is released by Google, vastly superior to Kubernetes, and promising significant improvements in SKA processing.	Commissioning	The existing technology choice (Kubernetes for example) can be backed out and replaced with no negative impact on SKA workflows, and is deployed without significant impact on operations in less than 3 months.
30	Data in the LTS is corrupted (bitrotting) or data is corrupted upstream before being written	Normal Operations	Data corruption is identified as soon as possible. Sufficient forensics are available to evaluate what can be corrected. Relevant checksums are in place at different stages.

8.4 ATAM Risk Analysis

The analysis of scenarios lead to capturing a number of risks, which have been further analyzed and grouped into major themes as described in section 4.2.2. Risk IDs are formed by an Scenario ID —from the table above— followed by the individual risk number within the scenario.

The headings of the table are mapped to those risk themes as follows:

Sys	System-SDP boundary
Doc	Documentation
Sec	Security and Data Integrity
Port	Portability
Test	Testing
Mon	Monitoring
Impl	Implementation
Cplx	Complexity
Rec	Recovery

		Risk themes								
ID	**RISK**	**Sys**	**Doc**	**Sec**	**Port**	**Test**	**Mon**	**Impl**	**Cplx**	**Rec**
R9.1	Recovering from slow delivery can have cascade effects									x
R9.2	Interface to TM for buffer control is incomplete or may need an additional interface for direct operations.	x	x				x			
R9.3	Data models are defined conceptually and not physically. How quality related information are stored alongside data products is undefined.							x		

R4.1	While attributes and characteristics of a re-deployable SDP are captured, the work required to deliver an installable SDP to the SRCs is not fully elaborated. Responsibilities for architecture definition and costing are not clearly defined and allocated.	x	x		x			x		
R6.1	Race condition in the SDP scheduling between cancellation and submission of a replacement PB, may require atomic "cancel and resubmit" operation or other solutions.									x
R23.1	There is no software performance and diagnosis profiling tool. Profiling responsibilities are not allocated.					x	x			
R23.2	The architecture documentation is not clear enough in describing logical and physical separation of data access patterns and data chunks storage. Concrete example would help clarifying.		x					x		
R23.3	Operationally, there is need to monitor hardware performances. We need to detect processing inefficiencies and drill down to causes. No solution is documented.		x				x			
R27.1	There is no dump/restore interface to an externally provided storage without burdening the network e.g. for disaster recovery									x
R29.1	An explicit security model is missing for integrating the external elements with SDP			x						
R29.2	A system level security policy TBD will impose new requirements on SDP which might pose architectural challenges, especially if introduced in later stages of development	x		x						
R29.3	A conceptual architecture can not be secured, security implementation needs to be addressed while developing the SDP			x				x		
R1.1	The descoping options are not documented. Descope analysis shall continue both at system and element level.		x					x	x	
R12.1	SPEAD tests are not documented officially.		x							

R12.2	NIC upgrades and supplier time scale might take longer than 2 weeks							x		x
R12.3	SDP simulator might not fully replicate the behavior of the CBF. CBF data output representative hardware might not be available for testing	x				x		x		
R12.4	At scale, testing environment will not be representative enough of the real system to replicate the error					x		x		
R15.1	Reloading Processing Blocks from TM upon restart is not documented in the ICD as a use case. This identifies a gap in the SW architecture.	x	x							x
R15.2	The RPC calls protocol is currently undefined. Care should be taken that it does not become stateful and break the architectural principles building the SDP		x					x		
R16.1	Portions of the SKA Common deployment are not within SDP scope and properly part of the System team. Some SDP specific aspects must inform this development.	x	x		x	x				
R20.1	Inter module coupling through implementation technologies might be difficult to avoid or understand		x					x	x	
R20.2	People with technical knowledge on pre-construction need to flow into construction, operations	x								
R30.1	There is no documented way of addressing bitrotting at the moment	x	x	x						
R30.2	Long term storage is more subject to undetected corruption unless periodically checked			x			x			
R30.3	There is no awareness of how sensitive we are to bit-flipping, we should try to quantify that	x		x						
	Number of risks per theme:	**9**	**11**	**6**	**2**	**4**	**4**	**10**	**2**	**5**