Challenges and solutions for the SKA TM Architectural Team (TMAT)

Di Carlo, Matteo, Bridger, Alan, Le Roux, Gerhard, Chaudhuri, Subhajyoti

Matteo Di Carlo, Alan Bridger, Gerhard Le Roux, Subhajyoti R. Chaudhuri, "Challenges and solutions for the SKA TM Architectural Team (TMAT)," Proc. SPIE 10707, Software and Cyberinfrastructure for Astronomy V, 107071O (6 July 2018); doi: 10.1117/12.2309342

Event: SPIE Astronomical Telescopes + Instrumentation, 2018, Austin, Texas, United States

Challenges and Solutions for the SKA TM Architectural Team

Matteo Di Carlo[a], Alan Bridger[b], Gerhard Le Roux[c], Subhrojyoti R. Chaudhuri[d]
[a]INAF Osservatorio Astronomico d'Abruzzo, Teramo, Italy; [b] UK Astronomy Technology Ctr. (United Kingdom); [c] SKA South Africa (South Africa); [d] Tata Consultancy Services, Ltd. (India);

ABSTRACT

The SKA (Square Kilometre Array) Telescope Manager (TM) [1] is the core package of the SKA Telescope: it is aimed at scheduling observations, controlling their execution, monitoring the telescope health status, diagnosing and fixing its faults and so on. Following the adoption of the Views and Beyond (V&B) approach [2] of the Software Engineering Institute (SEI [3]), it was discussed and agreed upon to take the opportunity to setup a TM Architecture Team (TMAT) composed by a combination of team members (who drove much of the work towards the main deliverables for the final review) and others (who can help shape the architectural design). The TMAT has to make sure that the main deliverables are well aligned with the overall TM architecture (including ensuring that the SEI approach is followed to the level agreed upon), and that there are no gaps and that cross-cutting issues are taken care of properly. This paper wants to analyze the challenges that the team has to face together with the solutions proposed to ensure that the quality of the deliverables are reached.

Keywords: SKA, TM, TMAT, Challenge, Solution, SEI, V&B

1. INTRODUCTION

1.1 SKA Project

The international Square Kilometre Array (SKA) project to build two radio interferometers is approaching the end of its design phase, and gearing up for the beginning of formal construction. A key part of this distributed Observatory is the overall software control system: the Telescope Manager (TM).

The two telescopes, a Low frequency dipole array to be located in Western Australia (SKA-Low) and a Mid-frequency dish array to be located in South Africa (SKA-Mid) will be operated as a single Observatory, with its global headquarters (GHQ) based in the United Kingdom at Jodrell Bank. When complete it will be the most powerful radio observatory in the world. The TM software must combine the observatory operations based at the GHQ with the monitor and control operations of each telescope, covering the range of domains from proposal submission to the coordination and monitoring of the subsystems that make up each telescope. It must also monitor itself and provide a reliable operating platform.

1.2 Telescope Manager

SKA TM consortium (the association of institutes and companies for the development of the TM software package) is composed by people and professionals located in different locations and comprises the following parties:

Table 1. TM Consortium member

Country	Consortium member
India	National Centre for Radio Astrophysics (NCRA) Tata Research Development and Design Centre (TRDDC)
South Africa	SKA South Africa (SKA SA)

* matteo.dicarlo@inaf.it; phone +39 0861 439711; fax +39 0861 439740; www.oa-abruzzo.inaf.it

Software and Cyberinfrastructure for Astronomy V, edited by Juan C. Guzman, Jorge Ibsen, Proc. of SPIE Vol. 10707, 107071O · © 2018 SPIE · CCC code: 0277-786X/18/$18 · doi: 10.1117/12.2309342

Country	Consortium member
United Kingdom	Astronomy Technology Centre STFC (ATC-STFC)
Italy	Istituto Nazionale di Astrofisica (INAF)
Portugal	Grupo de Radioastronomia do Instituto de Telecomunicações (GRIT)
Australia	CSIRO, Astronomy and Space Science (CSIRO)
Canada	NRC Herzberg Institute of Astrophysics (NRC-HIA)

At the beginning of the project, many tasks were found to ensure the correct design of the TM package out of which the most important were:

1. Telescope Management (TELMGT), that is the engineering management services for each of the three SKA instruments;
2. Observation Management (OBSMGT), that covers all activities for observing with the telescope that are associated with an astronomically motivated and described observation;
3. Local M&C (LMC), to ensure the continuous and proper operation and
4. Local Infrastructure (LINFRA), that is all the needed servers where TM will run.

Another important task were the system engineering (SE) work from the requirement analysis until the product breakdown structure definition. The term SE refers also to the team of system engineers working on the SE processes.

The above distinction brought the leading party (NCRA-India) to constitute 5 main team, which corresponds to the initial tasks definition.

More information about the Telescope Manage architecture is available in [1].

1.3 TMAT

The SKA TM Consortium has at various previous occasions tried to put an Architecture Team together but, with the adoption of the V&B approach, the need has become urgent. The team is composed by four people chosen within the teams composing TM (see 2.7 for more information about the TM teams).

The main focus of the TMAT is the TM Software Architecture while the TM Project Managers continue to coordinate the overall project activities and full set of TM CDR deliverables. Another crucial aspect is the responsibility to drive and coordinate the V&B SEI process within TM to deliver a coherent TM Software Architecture by CDR.

Another important aspect is the fact that four people cannot do the job of the entire TM consortium so the TMAT relies heavily on TM team members to whom they delegate certain activities, rather than doing all the work themselves. The TMAT has architectural decision making authority where their primary role is to define the TM Software Architecture. Secondary roles include ensuring that the main deliverables and supporting documents are well aligned as an overall TM architecture (including ensuring that the V&B approach is followed to the level agreed upon), that architectural decisions are clearly communicated, that the rationale for architectural decisions are captured sufficiently in the architecture documentation, that there are no gaps and that cross-cutting issues are addressed properly.

Together with the TMAT, a TM Review Team has been created with the role to assist the TMAT by providing guidance (where needed) through early inputs on topics during the TMAT meetings, and also to give inputs on the planned activities of the TMAT. Once the activities have been completed, the TM Review Team will review outputs and provide feedback.

1.4 Definitions

Table 2. Glossary

Term	Definitions
CDR	The term CDR (Critical Design Review) refers to a review in the environment of a system engineering process that need to demonstrate whether a design can proceed to the fabrication, assembly, integration, and test.
Product	According to the definition made by Duncan Haughey **Error! Reference source not found.Errore. L'origine riferimento non è stata trovata.**, a product can be defined as what the project deliver. For the purpose of this document, a product is defined as the result or deliverable coming out from the development effort on the SKA Telescope Manager. The product is what the artefact will look like when the customer for which the product is intended can accept and make use of the item.
Product breakdown structure	An exhaustive, hierarchical tree structure of components that make up a project deliverable, arranged in whole-part relationship [5].
System engineering process	It is a resolution process recursive and iterative [5]. It is mainly composed by: 1. Requirement analysis 2. Functional analysis 3. Technical design
V&B	Views and beyond (V&B) [2] refers to a proven approach to documenting software architecture developed by the SEI (Software Engineering Institute [3])
Architecturally Significant Requirement (ASR)	A requirement that affects the architecture of a software system.
SKAO	The Office for the SKA Organisation (SKAO) is responsible for coordinating the global activities of the SKA project. This includes engineering, science, site evaluation, operations and public outreach.
Mission thread	A mission thread is a sequence of end-to-end activities and events that takes place to accomplish the execution of a system capability.
Quality attribute	It is a requirements such as those for performance, security, modifiability, reliability, and usability that have a significant influence on the software architecture of a system [6]

Term	Definitions
Quality attribute workshop	The Quality Attribute Workshop (QAW) is a facilitated method that engages system stakeholders early in the life cycle to discover the driving quality attributes of a software-intensive system.
View (or view packet)	It is a description of a coherent set of elements and relations among them of a particular structure belonging to a software architecture that enable the reader to reason about the quality of the system [6]
LMC	The Local Monitoring and Control is a component present in every sub-system of the SKA project in order to provide TM with a consistent interface regarding message protocol, behaviour, naming conventions, level of abstraction and responsibility.
LMC Harmonization	Series of workshop with the aim of improving the commonality across the SKA Telescopes Monitoring and Control Systems architecture at levels beyond the LMC. The aim is to develop a community of practice, create common control framework, and establish a collaborative developer community.
ANT Team	Point of contact between the SKAO architect(s) and the various consortia involved in the SKA project within the LMC Harmonization process.
TM Board	The consortium board is a set of people representative of each consortium member country (or organisation) with decisional power.

2. CHALLENGES AND SOLUTIONS

At the kick-off meeting of the TMAT team, the TM management setup the process to follow in order to let the team rapidly get started and ensure that the role of the TMAT as technical coordination both for teams and outcomes resulting as TM software architecture. Together with this, things like work ethic, goals, channels of communication, quorum and decision making process were discussed and agreed.

Since the beginning, a list of tasks (or issues), called challenges in the present paper, was given to the team such as:

1. Identification and prioritization of the architecture significant requirements together with the identification and prioritization of the main TM requirements considered not clear or "unexpected" such that these can be registered and resolved through cooperation with the SKAO.
2. Consolidation of the main overarching quality attributes for the TM software architecture from the outputs of the mission threads and quality attribute workshops and view packet development.
3. Manage and actively resolve the TM internal technical/architectural issues. All TM teams can register issues for clarification and decision making with TMAT.
4. Ensure that the TM architecture and supporting documents (like mission thread analysis, quality attribute trees, view packets and so on), are coherent and consistent.

While point 1 and 2 are clear and well understood for a software architect (because it means basically to identify the requirements that mostly affect the architecture and the qualities needed for the system), it is not the same for point 3 and 4. In fact, point 3 (TM internal technical/architectural issues) means to identify, prioritise and work through the TM issues uncovered by the mission thread workshops and view development work. Point 4 is the most challenging item in the list because, for this, the TMAT should give guidelines and inputs to the teams working on the various aspects to direct the outcomes before too much detail is generated in the separate teams. This, together with point 3, basically means to drive the decision making process.

2.1 General challenges

When developing a software application, an architect has to face many problem in order to achieve the qualities needed by the stakeholders involved in the project. This is a challenge per se, because there is the need to a vision of the design patterns to include in the architecture developed. As TMAT together with the basic challenge of the software architect, others are:

- awareness of every documents wrote since the beginning of the project from SKAO, TM and other element teams (like the ANT team of the LMC harmonization),
- awareness of every quality needed by the TM,
- awareness of the main processes used (i.e. system engineering process, View and Beyond documentation approach),
- awareness of the project management models (i.e. cost model),
- be able to harmonize all the documents to be delivered for the CDR.

2.2 Software vs System Engineer

All those initial tasks come from the responsibility to drive and coordinate the V&B SEI process within TM and specifically in term of documents to release by CDR. From the various discussion made, it emerged a different vision between the processes of software engineer and system engineer, particularly the way various terms are used. One of the most controversial discussion was related to the products and the product breakdown structure (PBS) of the entire TM.

The analysis made by the TMAT showed that the initial decomposition of TM (see 1.2) was incorrect because the main products (of the PBS) were, in reality, software modules.

The concept of PBS and product were deeply analyzed and two definitions were taken into account:

- "PBS is an exhaustive, hierarchical tree structure of components that make up a project deliverable, arranged in whole-part relationship. A PBS can help clarify what is to be delivered by the project and can contribute to building a work breakdown structure" (see [4]);
- "PBS is a hierarchical structure of the complete set of physical systems and subsystems including operational system, training system, development support, production support, etc. which identifies the configuration items"(ANSI/EIA-632 Standards).

Even if, in the SKA project, the formal definition of the end product is done by means of requirements leading to the rule of thumb: "if it has a requirement it is a product", the first definition described above was taken as primary input and a clear separation between modules and product was made. In particular, three main products have been found:

- TM Mid: every TM subsystems needed at the SKA Mid (intended as mid frequency) hosted in South Africa;
- TM Low: every TM subsystems needed at the SKA Low (intended as low frequency) hosted in Australia;
- TM Observatory: every TM subsystems needed at the SKA GHQ (Global Head Quarter) hosted in UK.

Together with those products, three main software modules were setup: OSO (Observatory Science Operations), TMC (Telescope Monitor and Control) and SER (TM Services). The first two modules derives from an early division of activities into online and offline where online means activities happening at telescope "live" and offline means activities happening in advance, anytime and anywhere.

The third module, SER, came out after a discussion and an analysis of the commonality between OSO and TMC that is, the products that should be in every location of the SKA project. In specific, the analysis covered the architecturally significant requirements, the rationale for having it and the products included in it, taking into account the common framework selected for the entire project.

2.3 Bureaucracy

The system engineering helps in design and manage a complex system over its lifetime; it is mainly a waterfall approach and, because of that, not every changes are welcome. On the other hand, software system always changes (otherwise it becomes progressively less satisfactory, see [7]) and the need for a process that helps it is necessary.

One of the processes put in place, in the field of system engineering, is the change management process. It means that, when there is the need for a change, an algorithm has to be followed which comprises:

1. Identification of the change,
2. Analysis (feasibility) and evaluation (costs and benefit) of the change,
3. Plan for the change,
4. Implementation of the change and
5. Final review.

Without going into many detail of this process (that can change depending on the specific needs), this imposes a sort of bureaucratic problem if the process is strictly followed. In specific, common problems could be:

1. the involvement of too many people so that following the process creates a bottleneck,
2. the lack of interest in the process itself that is when there is a clear process but still it is slow,
3. The scalability, when too many steps are in place or there are wrong enabling system (see [8]).

One of the most important example of change management process is the software development in the maintenance context. When a software user see a bug, he/she reports it and start a process of analysis, design, implementation, etc. until the bug is solved (or the new functionality implemented).

In SKA project, the process is enabled for documents when they are under control of the configuration management system. From that time, every change of it has to follow the process and every implementation needs authorization. Unfortunately, this brought the problems above described and, in many case, the only solution was to avoid it, that is not putting the item under the control of the configuration management system. A lighter process (or a different/smarter tool) would have helped to improve the involvement of more people resulting in a fast process with benefit for the output material.

2.4 Reasoning about the design

Even though engineering is fundamentally about the application of science, the application itself is a social activity and are very much determined by social rather than physical structures.

Argumentation and debating about ideas, the key activity performed by architects, is a field receiving considerable attention by researchers lately. Good decisions can only come forth when competition between ideas exist ([9]), which in turn often leads to conflict. However, Morton Deutsch in his seminal paper on task oriented groups ([10]) concluded that most conflicts are not about one person preventing another from achieving his personal goal but rather about both persons arguing about how to achieve the same goal. Thus the best behaviour in resolving these types of collaborative conflicts popularly named the “win-win” strategy, is markedly different than the more traditional adversarial approach would suggest.

The architecture team worked best when they were conscious of their role as collaborators instead of adversaries, yet the complexity of human relationships, of which culture (see 2.5) played a primary role, quickly revealed that the task of collaboration, of resolving conflicts in a win-win manner, is a much more sophisticated process than what appears on the surface.

In thinking about conflicts in terms of goals, Hoker and Wilmot ([11]) structures conflict into something that involves multiple shared goals, each taking place on a different layer of the relationship:

1. The topic goal: the obvious issue on the surface of the conflict - “e.g. is this a good idea for us?”

2. The relational goal: the effect the issue have on interdependent relationship goals between members - ‘e.g. will this harm our friendship?”
3. The identity goal: the effect the issue have on the perception of self - “e.g. will this change how he/she sees me?”
4. The process goal: the going about in resolving this process “e.g. should I allow him to state his side of the story?”

Thus the conflict as it appears to the outsider are often difficult to understand or justify given the hidden layer of many of the interpersonal goals being jeopardised by the conflict.

For TM, it was found that structures that harmonised the various levels of goal conflicts improved conflict management and communication in general. In other words, processes, roles and relationships between members were defined so as to maximise their coincidental positive outcomes when resolving the topical goal. For example, the TM architecture team were officially authorised (and recognised) to address architecturally cross cutting issues, ensuring people had a strong correlation between their identity goals (being architects instead of systems engineers of software developers) and topical goals (addressing the architectural issues of TM). This prevented people from feeling conflicted in resolving a topical issue.

Moreover it was often found that conflict on one level of goal seeking affected how conflicts on other levels were resolved. If two people often struggled with relationship issues, it “spilled over” into relatively arbitrary topical issues. Conversely two people struggling with topical issues inevitably started having conflict with relationship and sometimes even identity conflicts. The upshot of this is of course that positive outcomes (or the absence of conflicts) reinforced rather than damage goal seeking on another layer. This ties in with recent studies ([12]), suggesting - somewhat counter intuitively - that in order to address issues of excessive diversity in teams, resulting in “fault lines” and creating in-groups and out-groups, they should focus instead on the shared tasks at hand, rather than the relationships or social aspects of the team. Over time, the successful outcomes of tasks shared by the group, reinforces new social relationships and disable fault lines and in-group/out-group situations to form (e.g. the software engineers versus systems engineers).

The TM architecture team was formed as members from divergent functional backgrounds and cultural backgrounds; yet the shared tasks and issues that was resolved together over time prevented people from forming clusters around shared culture, training or background, resulting in a new culture, a TM culture being formed instead.

2.5 Cultural differences

Culture is the set of values and experiences of an individual that contributes to forming his or her personality and, overall, the way he/she reasons. Cultural experience is not just geographical - there is also the difference between academic and commercial cultures. It is important for the team to be aware of these differences, and accommodate them. Since culture is not easily measurable like cost rates or effort estimations, the tendency is to underestimate its impacts on collaborations. It is also important to be aware that these impacts can be both positive and negative.

Within the TM consortium there are at least two different major cultures, and several micro-cultures. The differences between those cultures can be understood according to many fields but even if one understands the culture, there is still the practical challenge for an international team to build the necessary synergy. From this point of view, there are many tips to be aware of, some of them described in [13].

Although English is the standard language for communication within SKA, its usage obviously is not uniform across teams. For people from non-english speaking backgrounds, the usage of the language appeared to happen in two steps, formulating the expressions first using native language in the minds and then translating the same in English while communicating. For native english speakers the temptation to use idiom and slang can be high, especially in the extremes of both stressful and relaxed surroundings. This did give rise to occasional miscommunications, confusions and misinterpretations. Given that the SKA is a globally distributed project with much of the interactions happening over emails and phone calls, the possibility for miscommunication and misinterpretation was quite high as a result. While it is possible to imagine formalizing the usage of English in such projects to mitigate such problems, however defining such formal and constrained representations of a spoken language is not easy and it may take years of work [14]. More practically it is incumbent on both sides to show patience and awareness of the potential issues.

In meetings it is important to have a structured approach to debate, agreed at the beginning. Usually this will mean meeting sessions are chaired and that the Chair must control the discussions, usually by mechanisms such as allowing the current speaker to finish and requiring signals (raised hands) to allow others to make their interjections. This is especially important earlier in the collaboration when the individuals know each other less well, but in fact remains important throughout in order to get the best from the meetings. Compromise, and being prepared to compromise, is also important as is realising that cultural and individual approaches to meetings can be different, for instance in the apparent level of attention.

Communication is different from language and it is intended as the ability to share a concept, an idea. It goes without saying that any problem in communication starts when somebody says something and the other party does not understand it. This is particularly dangerous when one of the parties is working in a foreign language - sometimes, the English words are fully understood, but the understanding is something different from the original concept. Even the simple word "yes" can mean agreement, a direct answer, or "carry on, I an understanding", or sometimes just an an encouragement to continue speaking even if the understanding is not there. In general solving the communication problem is not easy and it becomes important to not only be careful in constructing arguments but also to ensure that a clear conclusion is written, in reasonably simple english, and agreed following the debate.

Cultural differences in a team are important to consider. They can cause problems, but they can also bring positive contributions and a considerable amount of enrichment to the team and individuals. The Office for the SKA Organisation has published a "Code of Ethics" [15] to help guide consortia in this and similar key areas.

2.6 Boundary between OSO and TMC

One of the first challenges tackled by the TMAT concerned the boundary between what became the TMC and OSO software modules. In essence this distilled into which module took responsibility for the software that executes the Scheduling Blocks and associated scripts that will result in telescope configuration, control and data-taking. This particular boundary was at serious risk of falling foul of Conway's Law, with responsibilities allocated according to a somewhat artificial Work Breakdown Structure imposed upon the consortium, which was then further allocated geographically. Many discussions had been held within the consortium, and with external reviewers, without reaching a clear conclusion. The formation of the TMAT provided a focal point and small team that could approach the issue in a more structured way. A supported argument for a preferred solution was put forward, with alternatives proposed and discussed. The following is an extract from the minutes of a meeting that discussed this particular issue:

Argument:

- The boundary should be based on keeping the details of planning and scheduling block representation internal to a single product. This suggests that the OSO should include all of planning, SB design, Scheduling and SB execution and OET.
- TMC is an instrument management system that executes commands from OET.
- This also separates knowledge of domain-specific observing strategies from more general instrument control and monitoring.
- A consequence is that the "new" interface will be more or less the present OET/OST – Central Node interface, which is already quite mature. This boundary avoids creating a new ICD that splits what is currently considered a single system.
- Verification of TM-Low and TM-Mid will need the creation of a (command-line) interface that supports scripting to invoke the Central Node commands. This should not be a problem. Such an interface will also be required for commissioning.
- Verification of OSO will need an instrument simulator, which is already planned, though its scope needs elaboration.

Alternatives:

- Include the Scheduling (OST) in OSO, but make OET a part of TM Low and TM Mid. This would keep TANGO-related software together. A consequence would be that SB design and SB execution are de-coupled, requiring detailed knowledge of the SB data model - and dependency on the ODA - to be added the TM-Mid and TM-Low systems. Changes to the SB design and observing strategies would affect both Products and the ICD.

- Keep OSO as all the functionality deployed at GHQ, and TM-Mid and TM-Low as all the functionality deployed at the telescopes. This will mean considerable functional coupling between the Products in terms of both observation design and planning functionality. It would also imply that planning and design cannot be seen as functions of TM-Mid and TM-Low

As a result of this discussion, a formal decision was made and recorded by the TMAT, with diagrams created to illustrate it.

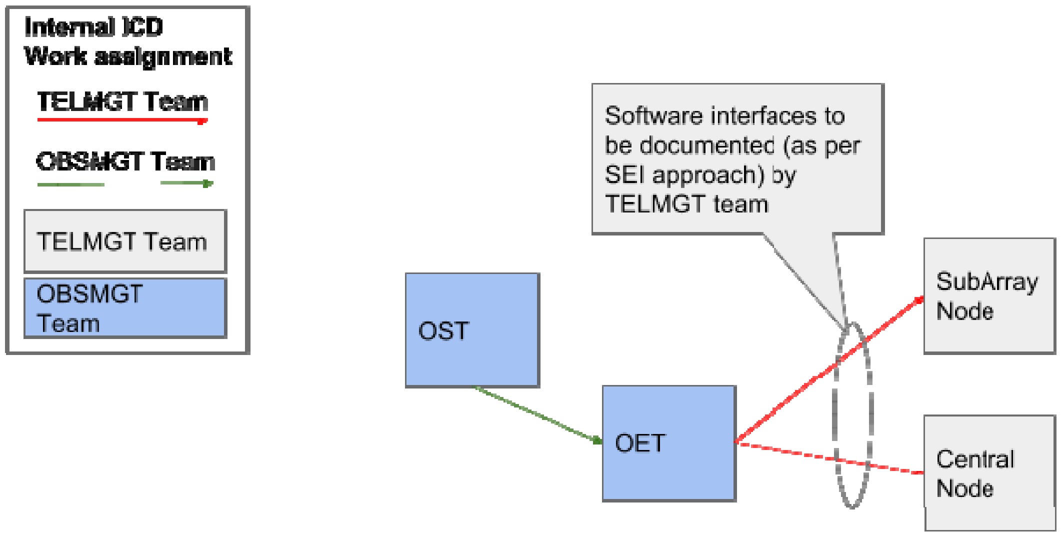
Internal ICD Work assignment
TELMGT Team
OBSMGT Team
TELMGT Team
OBSMGT Team
Software interfaces to be documented (as per SEI approach) by TELMGT team
OST
OET
SubArray Node
Central Node

Figure 1. Boundary between OSO and TMC

2.7 ARID-Like Review

One of the main task of the TMAT is ensuring that the TM architecture and supporting documents are coherent and consistent. In order to perform this activity with success, the main need is to read all the design documents and make a cross check between all of them. But the adoption of the Views and Beyond (V&B) approach [2] invalidated all the design documents that were written before and considering also the deadline of the CDR review, the only possibility was to make a set of reviews as soon as the views (parts) of the documents were ready.

The decision to go with an approach of the SEI [3], suggested to select one of their indicated review methods such as the ADR (Active Design Review [16]), the ATAM (Architecture Tradeoff Analysis Method [17]) and ARID (Active Reviews for Intermediate Designs [18]).

Since the reviews involved the ongoing work, the only possibility was the ARID method. It comprises three main teams (the review team, the lead designer and the reviewers), two separated phases (consisting of nine steps in total) and it is usually done in 2-3 days in a face to face meeting. Considering the deadline of the project, the TMAT decided to tailor the ARID process into something more suitable for this situation, omitting some steps (for instance, the reviewers for each document were already identified), providing the materials for reviewers as early as possible (this time varied from the day before the meeting to one week before depending on the work needed to complete the view), and then a shorter review meeting (typically one or two hours), held via teleconferencing facilities, to exercise and discuss the scenarios. These "ARID-like" reviews were organized on a regular, weekly, basis to address each key area of the overall TM Architecture. The initial 3 teams were collapsed in mainly two, the reviewer and the designer team. The reviewers had to test the architecture according to a particular real scenario so that they could check whether the architecture works for the highlighted problem. The core of the approach was maintained so that each reviewer should not question the rationale nor suggest alternatives rather test the design, find issues and only after that propose one or more alternatives.

These reviews were crucial for the coherence and consistency of all the architecture documents but also for understanding the real behaviour of the design. Besides, in case of any inconsistencies, the analysis of the scenarios already pointed to some solutions and, in some cases, also to future developments.

Table 3. ARID-like steps.

ARID	ARID-like	Rationale
1. Identify Reviewers	**Prepare the view**	The specific view to be reviewed is the only thing needed. Scenarios could be taken from use cases (or requirements). The design presentation is not needed because all reviewers were already involved in TM and much of the preparation activities were already done.
2. Prepare Design Presentation		
3. Prepare *seed* scenarios		
4. Prepare for review meeting		
5. Present ARID method	**Present the view (if needed)**	The TMAT encourages the reviewers to consider using the "scenario" method: think of realistic scenarios and try to exercise the design.
6. Present Design		
7. Brainstorm and prioritise scenarios	**Perform review (2 hours)**	
8. Perform Review		
9. Present Conclusions		

2.8 Other taken solutions

For every challenge, in order to find the right solution is important to follow an approach, a method. One of the first decision made concerned the decisional process. It can be summarized with the following workflow diagram:

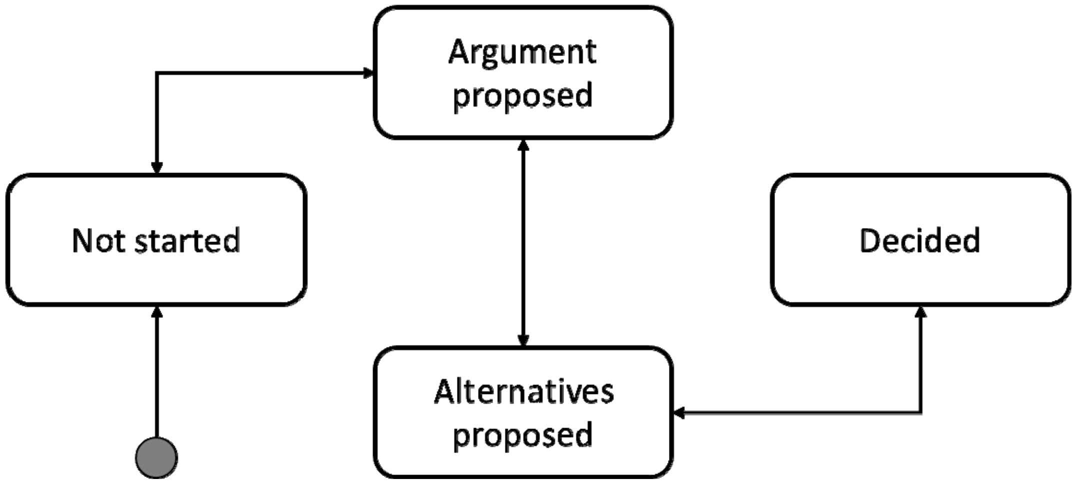

Figure 2. Workflow for decision

What is important to notice is the fact that only when two alternatives were given, then a decision can be taken. In order to take decisions, a number of actions has to be performed mainly for analysis for example with a classical "Generate and Test" approach (see [6]).

Among the various decisions taken, some of them are to be highlighted for their important. The following subsections highlight those decisions.

Architecturally Significant Requirements (ASRs)

According to the SEI approach on software architecture, one of the most important task to perform is the identification of the Architecturally Significant Requirements (ASRs) and quality attributes. For the ASRs, the decision made was in favor of assigning the task to every single team with the TMAT as supervisor of the process; for the identification of the quality attributes, the quality attribute workshop was promoted [19].

Terminology clarifications

The adoption of the Views and beyond approach of the SEI, brought a reworking of both the software modules and PBS as explained in "Software vs System Engineer" section. Table 1 summarizes the decision taken on the terminology in term of teams, software grouping and PBS.

Table 4. Terminology clarifications

Teams	OBSMGT, TELMGT, LMC, LINFRA (Should only be used for allocating work, should not appear in diagrams or architecture documents)
Software groupings	OSO, TMC and SERvices
PBS	TM OBServatory, TM MID and TM LOW

TM Context Document

As stated in the “General challenges” section, one of the most important obstacle within this project is the number of documents to read. To mitigate this problem and to set the context for all the stakeholders interested in understanding the architectural aspects that the Telescope Manager (TM) of the Square Kilometre Array (SKA) addresses, the TMAT decided to work on a document, called Context. The intention is to aid the readers of the TM architecture documentation pack in understanding the broad context of TM. It also aims to facilitate navigation through all the details covered across multiple architectural document sets delivered as part of the TM architecture pack, and to show the relationships between the three key system architectures that together form the Telescope Manager.

Requirements mapping and QAs

Every software is built to satisfy one or more requirements. According to [2] and [3], a view has to answer to at least one question that can be a requirement or a generic quality that the software should have. But how is it possible to map the requirements assigned to a package to an analysis made (i.e. a view)? The solution that the TMAT found was:

1. A full mapping of the Level 2 TM requirements available as a separate document and prepare by the SE team,
2. For each SAD, an overview of the Architecturally Significant Requirements (ASRs) has to be provided in the "beyond views" section of it, with reference to particular views,
3. Specific mentions of ASRs can be made where relevant in the views, with some author discretion as to whether or not it is useful to do so.

The same approach - overview in the context document, optional specific mentions in views - can be made for the identified Quality Attributes.

3. CONCLUSION

Every software application, even the easiest one, has an architecture. Regardless the techniques used to build the architecture and the design of it, taking decisions is a task for a software architect (or architecture team). The goodness of those decisions can be understood only after they have been taken and for TM, the fact that the Critical Design Review has been passed is an indication of the work done.

In specific, the goodness of the TMAT experience can be expressed by the work done for the time that it has been alive and in particular:

- More than 50 decisions has been taken;
- More than 50 actions has been performed in order to take the related decision;
- One context document has been created.

4. ACKNOWLEDGEMENT

This work has been made possible thanks to the financial support by the Italian Government (MEF - Ministero dell'Economia e delle Finanze, MIUR - Ministero dell'Istruzione, dell'Università e della Ricerca).

REFERENCES

[1] Bridger, A., et al., “SKA Telescope Manager - A Status Update", in [Software and Cyberinfrastructure for Astronomy V], Chiozzi, G. and Guzman, J. C., eds., Proc. SPIE 10707-02 (2018)

[2] David Garlan, Felix Bachmann, James Ivers, Views and Beyond, www.sei.cmu.edu/architecture/tools/document/viewsandbeyond.cfm

[3] Carnegie Mellon University, Software Engineering Institute (SEI), www.sei.cmu.edu

[4] Duncan Haughey, Project Management Tools, https://www.projectsmart.co.uk/project-management-tools.php
[5] International Council on Systems Engineering (INCOSE), “What is Systems Engineering?”, https://www.incose.org/AboutSE/WhatIsSE
[6] Len Bass, Paul Clements, and Rick Kazman, Software architecture in practice
[7] M M Lehman et al., Metrics and Laws of Software Evolution - The Nineties View, http://users.ece.utexas.edu/~perry/work/papers/feast1.pdf.
[8] ISO, ISO/IEC 12207, “Systems and software engineering - Software life cycle processes”, 2nd edition 1 February 2008 for the definition of Enabling system
[9] Mill, John Stuart, On Liberty and Other Essays, OUP Oxford, 1998
[10] Deutsch, Morton, “The Resolution of Conflict.” The American Behavioral Scientist 17 (2): 248–248, 1973.
[11] Hocker, Joyce L. and William W. Wilmot, Interpersonal Conflict. McGraw-Hill Education, 2017
[12] Gratton, Linda, Andreas Voigt, and Tamara Erickson, “Bridging Faultlines in Diverse Teams.” IEEE Engineering Management Review 39 (1): 80–90, 2011.
[13] Wolfgang Messner, Working with India
[14] Wikipedia, Controlled natural language, https://en.wikipedia.org/wiki/Controlled_natural_language
[15] SKA Organization, “Code of Ethics”, https://www.skatelescope.org/ska-organisation-code-of-conduct-for-meetings/
[16] David Parnas, David Weiss, Active Design Reviews: Principles and Practices
[17] ATAM, https://resources.sei.cmu.edu/asset_files/TechnicalReport/2000_005_001_13706.pdf
[18] ARID, https://resources.sei.cmu.edu/asset_files/FactSheet/2018_010_001_513474.pdf
[19] Software Engineering Institute, Quality Attribute Workshop, https://resources.sei.cmu.edu/asset_files/TechnicalReport/2003_005_001_14249.pdf