Monitoring & Control CoDR:
Risks
Tim Stevenson, SPDO

Scope of Talk

- Risk Management in SKA
- System Level Risks applied to M&C
- Review items received
- M&C Risks
  - Management & Organisation of Development
  - Engineering
  - Architecture
  - Design

Risk Management in SKA

- Working towards an active, useful and comprehensive RM regime
- Concept Phase risks:
  - Generic management risks
  - Vague technical risks
- Risks currently being collected in non-uniform ways across SKA
  - Harmonisation & calibration still to be done
- Risk Management vocabulary & principles remain a doctrinal problem in SKA

Risks from System dCoDR

- Section 4 Allocated/Paraphrased to M&C
  - Allocated due to software predominance in M&C
  - Those of particular concern ('Risk Exposure High')
    - Due to novelty of industrial approach in SKA
      - Vs R&D in Institutes
    - Due to R&D orientation of key players
    - Due to geographical and cultural dispersion

Exploring the Universe with the world's largest radio telescope

M&C M&O Risks

No.	Risks	Short Description	Proposed Plans to Manage Risks & Issues	Exposure: SKA1 *SKA2*	Retire by:
5.1.1.	Distributed Development: Task Distribution	Task uncertainty: WPC and SPO teams don't possess the knowledge and capabilities needed Task understanding: WPC and SPO teams don't understand the specification of the task Task coupling: the task is not logically divided into self contained subtasks	Communication and management of dependencies between project components Monitoring and managing progress towards delivering planned documentation Learn from Precursor and Pathfinder experiences	Very High *High*	SRR
5.1.2	Distributed Development: Knowledge Management	Knowledge creation, capture and integration is not managed in a sustainable manner Not enough documentation, and not of sufficient depth or quality	Implementation and mandated use of a widely accessible and searchable document management system Phased delivery of documentation Learn from Precursor and Pathfinder experiences	High *High*	SRR
5.1.3	Distributed Development: Geographical Distribution	Spatial distribution: many diverse WPC and the SPO sites may be involved Temporal distribution: many time zones may be involved Goal distribution: diverse objectives exist across WPC and SPO teams	Implement: Formal system requirements framework Common document templates and standards Clear definition and communication of expectations for delivered documents Selected review of documents prior to delivery to receiving teams Learn from Precursor and Pathfinder experiences Ensure interface simulators are developed by each team as a deliverable to other teams as far as possible.	Very High *Very High*	SRR
5.1.4	Distributed Development: Collaboration Infrastructure	Collaboration across WPC and SPO sites is difficult Coordination mechanisms across WPC sites not appropriate or sufficient Process alignment across WPC and SPO sites not appropriate or sufficient	Adopt effective, efficient and commonly available tools to support electronic collaboration Apply appropriate, internationally recognised standards for project management, including process descriptions and change management Learn from Precursor and Pathfinder experiences	Medium *Medium*	SRR
5.1.5	Aggressive schedule	The schedule for the SKA is ambitious	Plan to deliver software incrementally Allocate resources to problem areas before they escalate Where possible, decouple development dependencies Plan for resource contingency Learn from Precursor and Pathfinder experiences	Very High *Very High*	PRR
5.1.6	Estimation errors	Cost and effort for M&C development underestimated, due to missed tasks or optimistic task resource estimates	Multiple independent estimates thorough task identification comparison with precursor projects contingency for unexpected additional work	Very High *Very High*	PDR

COAR

No.	Page	Section	Reviewer's Comment	Type	Response	Description/Action	Done?
4	12	5.1.3	Mitigation: Ensure interface simulators are developed by each team as a deliverable to other teams as far as possible.	Minor	**Accepted**	Agreed - implemented	Yes

Engineering Risks

No.	Risks	Short Description	Proposed Plans to Manage Risks & Issues	Exposure: SKA1 *SKA2*	Retire by:
5.2.1 (5.1.7)	Scope of work greater than expected	Infeasible or unrealistic requirements Underestimation of the costs of complexity Required functionalities are overlooked Insufficient interface definition of – and integration with – other software implementations such as science processing Unnecessary development of software is initiated when existing codes could be re-used or configured Excessive interface complexities/diversity	Establish and maintain mechanisms to capture and assess early signs of negative scope risk Define standard interfaces for all hardware and local M&C and do it early To constrain software development: Use existing codes wherever possible, especially COTS Top-down budgeted 'cost as a design constraint' approach to development, i.e.: First order parametric estimating models for estimating the level of software development achievable within specified cost budgets Capture and use software productivity metrics from radio astronomy incremental software developments, other science domains and industry to inform the parametric estimating models Expert judgement as checks on parametric models	Very High *Very High*	PDR
5.2.2 (5.1.8)	Scope creep	Failure to baseline and traceably manage requirements held in a central repository	Set up Change Control Board and related governance structures to tightly manage all requests for change Use agreed common processes and tools to share requirements-related information Ensure requirements for component work deliverables are well documented and understood by all teams Learn from Precursor and Pathfinder experiences	Very High *Very High*	PDR
5.2.3 (5.1.9)	Misinterpretation and erroneous analysis of requirements	The flow down of requirements is open to misinterpretation particularly when this is solely via document handover	Use agreed common processes and tools to share information related to requirements Close collaboration between the parties involved in generating requirements including regular reviews of requirements Learn from Precursor and Pathfinder experiences	High *High*	PDR
5.2.4 (5.1.10)	Over-reliance on software development processes appropriate only for small scale projects	Lifecycle development models that are appropriate for small scale early stage development and preliminary implementations are very unlikely to work for development of large scale robust systems.	Peer with organisations – both industrial and research (e.g. CERN) – that have experience in management of development of large scale software intensive systems. Adopt appropriate "industrial" software development processes. Link into ICALEPCS body of knowledge to learn from experiences and activities on other big physics instruments (International Conference on Accelerator and Large Experimental Physics Control Systems http://icalepcs2011.esrf.eu/)	High *Medium*	SRR

COARs

No.	Page	Section	Reviewer's Comment	Type	Response	Description/Action	Done?
1	10	4.1	Description: Another bullet: * Excessive interface complexities/diversity Mitigation: * Define standard M&C interface for all hardware and local M&C early	Minor	Accepted	4.1 is a Risk paraphrased from System level; the corresponding Element level Risk is 5.1.7 which has been amended accordingly.	Yes
1	14	5.1.7	The statement "use existing codes wherever possible, especially COTS" is not entirely accurate. Use of Open Source software is widely used in very successful large experimental physics projects and should also be considered.		Not accepted	We believe that the term 'existing codes' encompasses Open Source software.	Yes
1	15	5.1.9	Column "Risks Becomming Issues Results in:" might also contain "Software integration is difficult/impossible". The next column could than contain "continues/regular integration".	Minor	Discuss	We believe that this refers to 5.1.7 'Scope of work greater than expected" and that the text already reflects.	
3	11	4.4	Mitigation: Link into ICALEPCS body of knowledge to learn from experiences and activities on other big physics instruments (International Conference on Accelerator and Large Experimental Physics Control Systems http://icalepcs2011.esrf.eu/)	Minor	Accepted	4.4 is a Risk paraphrased from System level; the corresponding risk is 5.1.10. Suggested Mitigation added.	Yes
2	15	5.1.10	What is an appropriate "industrial" software development process? Should SKA follow a software development process of similar organisation like ALMA, CERN and ITER rather than commercial industries?		Not accepted	The term was intended to signify a process that conforms with industrial standards such as, but not limited to, IEEE/ANSI 983-1986 "Guide for Software Quality Assurance Planning," IEEE/ANSI 1008-1987 "Standard Software Unit Testing," and IEEE/ANSI 1012-1986 "Standard Software Verification and Validation Plans." The exact standards to be applied are TBD, so are not cited here.	Yes

Architecture Risks

No.	Risks	Short Description	Proposed Plans to Manage Risks & Issues	Exposure: SKA1 *SKA2*	Retire by:
5.3.1	Lack of attention to non-functional performance requirements	Performance requirements can sometimes be overlooked	Undertake: Measures of performance of existing and prototype systems Simulation Benchmarking	Medium *High*	SRR
5.3.2	Incomplete interface identification or definition	There may be some interfaces that are not well specified or overlooked until integration testing and commissioning is in progress.	Collaborative interfaces definition and management. Undertake ongoing reviews of design documents, and in particular Interface Control Documents Learn from Precursor and Pathfinder experiences Focus on Interface Control Documents early	Medium *High*	PDR
5.3.3	Localization of safety	Incorrect assumption that all safety concerns can be localized to regions, so that Central M&C is not safety-critical, and so that connectivity with Central M&C is not safety-critical	Safety threat modeling and analysis to confirm assumption.	*High* *High*	PDR
5.3.4	M&C as part of Scientific Computing	M&C viewed as purely software development as a component of Scientific Computing rather than a system design problem. Insufficient attention to systems issues.	M&C as independent work package with strong collaborative relationships to System as well as Science Computing, plus linkages to other domains. Software & Computing redefined as governance layer.	*High* *High*	SRR
5.3.5	Scale creep	M&C scale (number of Components, monitoring points) much higher than anticipated, keeps increasing as design proceeds	Scalable M&C architecture. Interactions with domains to establish workload. Investigate scalability issues for software platform.	*High* *High*	PDR
5.3.6	Command conflicts	Auxiliary points of control e.g. domain M&C. Possibility of control conflicts among roles e.g. operators, engineers, scientists	Architectural principles to ensure single point of command, possibly augmented with carefully designed delegation models.	Medium Medium	SRR
5.3.7	Late Safety requirements definition	Insufficient effort in defining the ruling safety requirements leads to commitment to an architecture in which compliance is unnecessarily costly	Prioritise safety requirements capture	Medium Medium	SRR

[1] The term Scientific Computing is this document refers to data processing directly related to the (real time) production of science output, excluding the low level execution of telescope control and the provision of low level engineering data, some of which may be at the behest of such data processing. The aforegoing is not (yet) official SKA terminology as the architecture is TBD.

COARs

No.	Page	Section	Reviewer's Comment	Type	Response	Description/Action	Done?
2	10	4.3	Mitigation: Focus on Interface Control Documents early	Minor	Accepted	4.3 is a Risk paraphrased from System level; the corresponding Risk is 5.3.2. The proposed Mitigation has been added.	Yes
3	16	5.3.3	I disagree that M&C as subset of Software and Computing is a risk. If you read all the provided documentation for the CoDR, including this document, M&C is 99% software development cost. If M&C gets separated from "software and computing" then who is responsible for the overall SKA software architecture? Who will define the data and data communication framework between observation preparation, execution, processing and archiving? Why is "Software and Computing" called like that and not "Data Processing"? Based on ALMA and other large experimental physics (ITER), the overall software (including M&C) should be part of one governance layer.		Discuss	The Risk wording has been changed so as not to rely upon a narrow definition of the term 'Software & Computing'. To do so is misleading for those who are not familar with the scope of this Domain in SKA.	Yes

Exploring the Universe with the world's largest radio telescope

Design Risks

No.	Risks	Short Description	Proposed Plans to Manage Risks & Issues	Exposure: SKA1 *SKA2*	Retire by:
5.4.1	Human factors are overlooked in developing interfaces	Overlooking human-machine interface design goals such as: Effectiveness: helping users achieve their intentions Efficiency: reducing the time taken and the incidence of user errors Satisfaction: offering an experience to users conducive to productivity Underestimating the effort required for UI development. Due to the scale of SKA it will be necessary to develop novel techniques for visualisation and operation of such a large instrument.	Include human factor considerations in the high level architecture design activities for interfaces for users and operators Learn from the experiences gained from Precursor and Pathfinder projects Assign a sub-team within M&C with the specific responsibility to develop SKA scalable UIs	Medium *Medium*	PDR
5.4.2	Diverse M&C development	Local M&C is developed independently by dozens of Component providers, leading to potential quality, integration and maintainability issues	Standardized Component Interface, guidance to Local M&C developers, System M&C prototype or dedicated facility as test/validation fixture certification of Local M&C prior to integration	High *High*	CDR
5.4.3	Excessive technology heterogeneity	Independent choices of sensors, actuators, fieldbuses by each Component provider.	Identify standard / preferred technology choices, with slightly weaker rules for off-the-shelf Components. Trade off imposition of standards against whole lifecycle costs	High *High*	CDR
5.4.4	Technology obsolescence	M&C technology choices and capabilities get outdated over long lead time to first science and long lifetime.	Modularized architecture realized with replaceable off-the-shelf components adhering to 'future proofed' standards. Anticipation of and sensitivity to technology trends, particularly in terms of advanced capabilities.	Medium *Medium*	CDR
5.4.5	Metadata Plan	Definition of and approach to metadata not complete or clear	Define metadata requirements early and conservatively	*High* *High*	CDR

COARs

No.	Page	Section	Reviewer's Comment	Type	Response	Description/Action	Done?
6	18	5.4.1	Involve expertise in HCI at universities and academic institutions like CNAM (Pierre-Henri Cubaud) and INRIA (Emmanual Pietriga) in HCI design and workshops. They are currently working on ALMA interfaces and we will involve them on MeerKAT interfaces too.	Minor	**Accepted**	Point accepted. Background is that HMI is the subject of discussions with a strategic partner, which should they fail, an alternative will be pursued.	Yes
7	18	5.4.1	Additional risk: Underestimating the effort required for UI development. Due to the scale of SKA it will be necessary to develop novel techniques for visualisation and operation of such a large instrument. Mitigation: Assign a sub-team within M&C with the specific responsibility to develop SKA scalable UIs		**Accepted**	See also [6] above. Wording implemented as proposed.	Yes

2	18	5.4.5	What I miss is the risk that the metadata plan is not complete or clear which might result in incompatible components.	Minor	Accepted	This has been added as 5.4.5	Yes

Exploring the Universe with the world's largest radio telescope