SKA LMC Workshop

Assembly Integration and Verification

Craig Haskins | Software Engineer, ASKAP

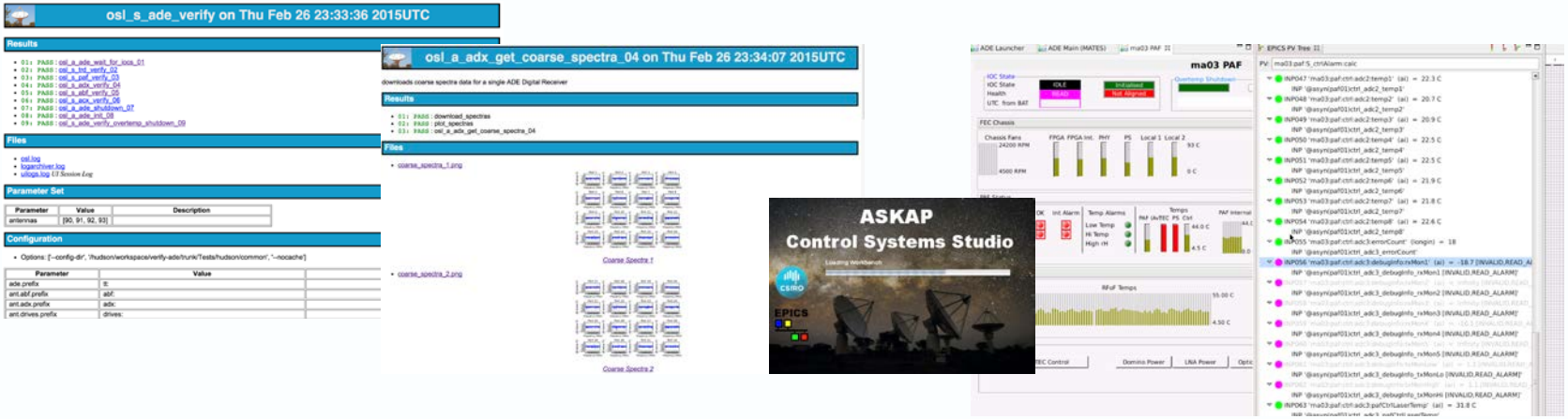
25th March 2015

CSIRO ASTRONOMY AND SPACE SCIENCE
www.csiro.au

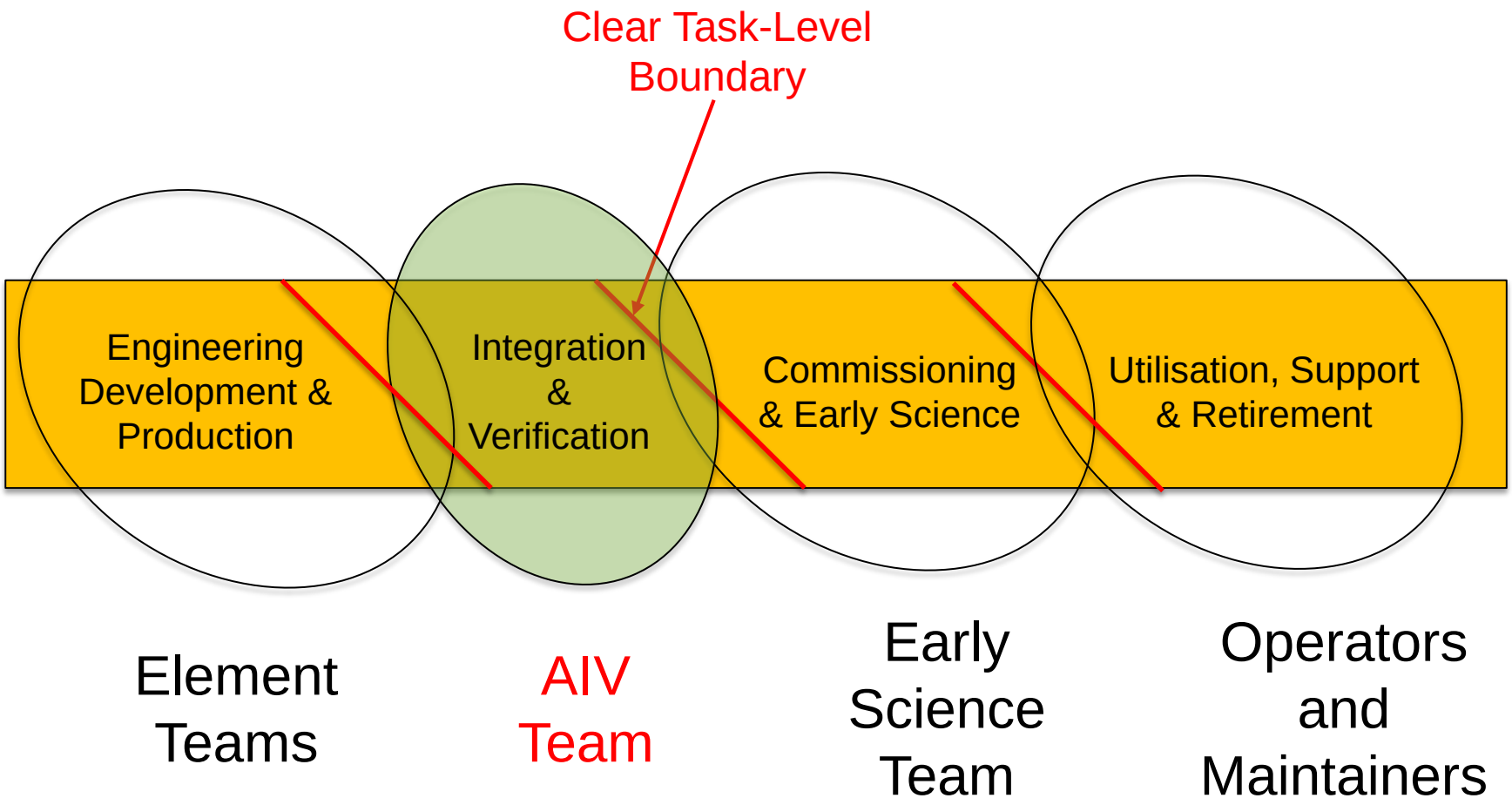
ASSEMBLY, INTEGRATION & VERIFICATION

Speaker Introduction

- Developed EPICS IOCs for ASKAP
- Developed testing framework for IOCs in Python
- Developed Engineering GUIs (Control Systems Studio)
- Involved with system commissioning
- Software Improvements for ASKAP Design Enhancements (ADE)
- Software Representative for AIV



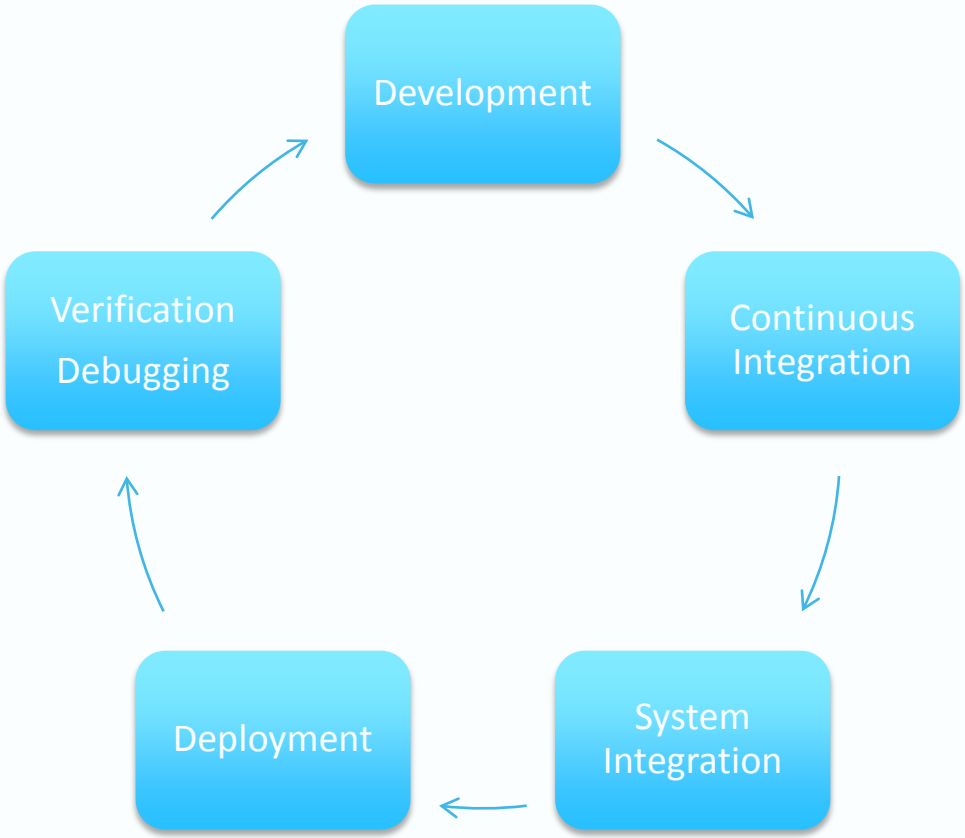
Boundaries of Responsibility during Construction Phase



Assembly, Integration & Verification

- AIV is responsible for verification of L1 requirements at site
- AIV supports the creation of an Integration Test Facility (ITF) where LMCs and hardware can be integrated, tested and debugged prior to installation on site.
- AIV does have an interest in suggesting the use of common software, frameworks methodologies and processes for testing and verification, even if the LMC specific test plans are not written and executed by AIV
- Common software reduces downstream integration risk

Development & Integration Cycle



Development Stage Guidelines

- Use a common LMC framework
- Use common software libraries (3rdParty, SKA libraries)
- Test driven development
- Hardware emulation (external emulators, HAL emulation)
- Integrate early and often
- Design in low level interface to LMCs
- Engineering level GUIs
  - Feature enable / disable
  - Debug features
  - Logging configuration

Example : ASKAP Development

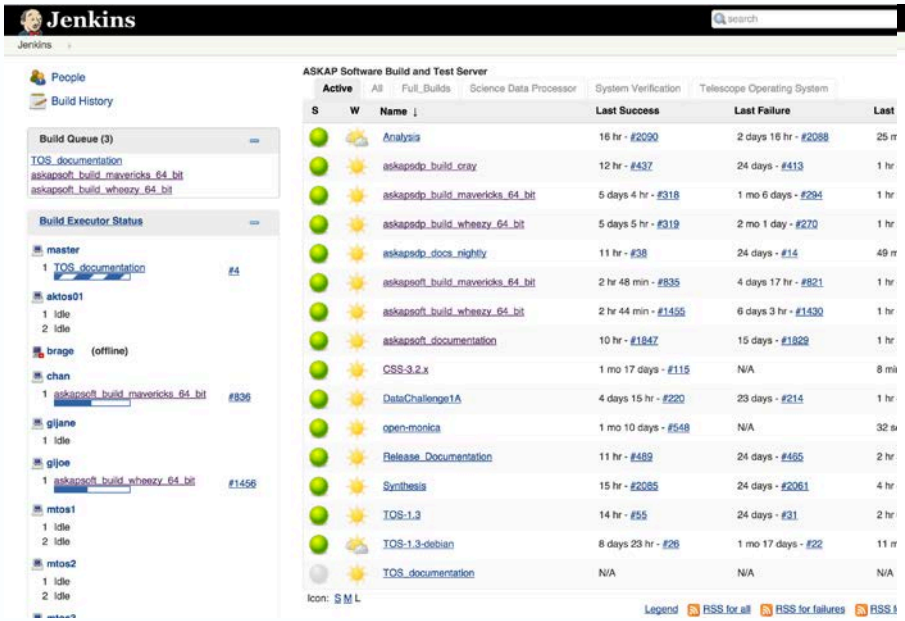
- EPICS Framework + select support libraries
- CPP Unit, JUnit, Nose Test (Python)
- Hardware Abstraction Layer (HAL)
- Target & emulated HAL
- Device emulators
- Common software libraries, common EPICS databases.
- Log4epixx (log4cxx with EPICS Wrapper, GUI)

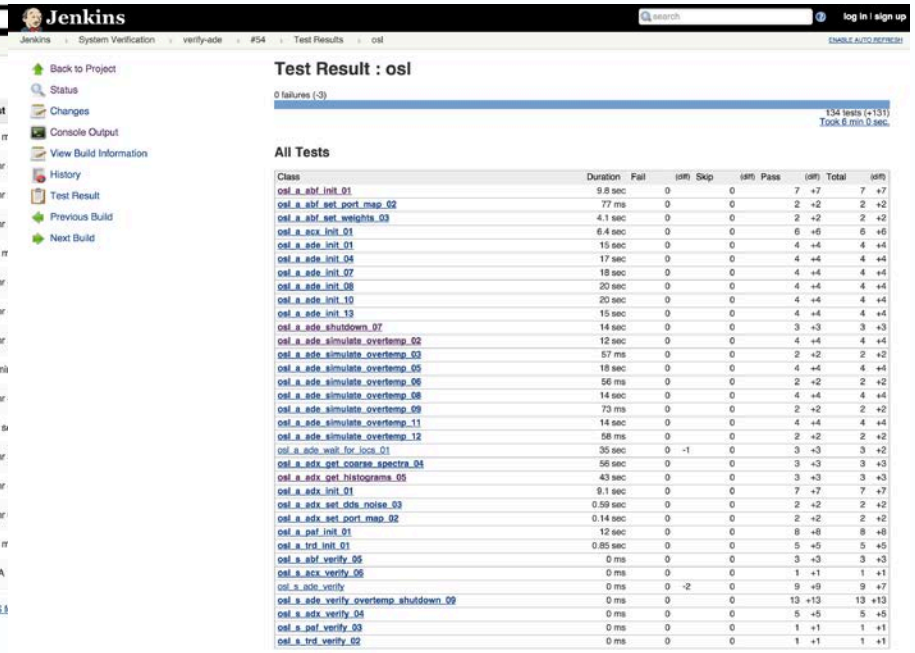
Continuous Integration

- Continuous Integration can assist downstream system integration by finding issues early on & reduces integration risk
- Provided test coverage is adequate
- Requires good unit & functional testing
- Requires good Regression testing
- When a bug is found, write a test against it
- Downside of Continuous Integration: Continuous Noise
- Multi-Stage Continuous Integration might be applicable

Example: ASKAP Continuous Integration

- Continuous Integration using Jenkins
- Build & testing triggered by revision control commits
- EPICS IOCs tested against emulated hardware





System Integration

- Bringing Element LMCs together
- Integration tests derived from commissioning tests
- Software only integration test using emulators
- Hardware Integration at Integration Test Facility

Example : ASKAP System Integration

- Hardware Abstraction Layer (HAL)
  - Target HAL and Emulated HAL For Most EPICS IOCs
  - Switchable at Runtime
- Protocol emulators for simple devices (e.g. weather station)
- OSL (Operator Script Library)
  - Python scripting for EPICS IOCs
  - Scripting Framework for creating Automated Tests
  - Generates web based reports & junit xml for CI tool
- OSL Used for
  - Automated system integration tests with Emulated HAL
  - Manual System commissioning test with Target HAL + Hardware

AIV Software Deployment

- Software packaging
- Dependency management
- Software configuration management
- Flexible deployment
- Automated testing of deployment

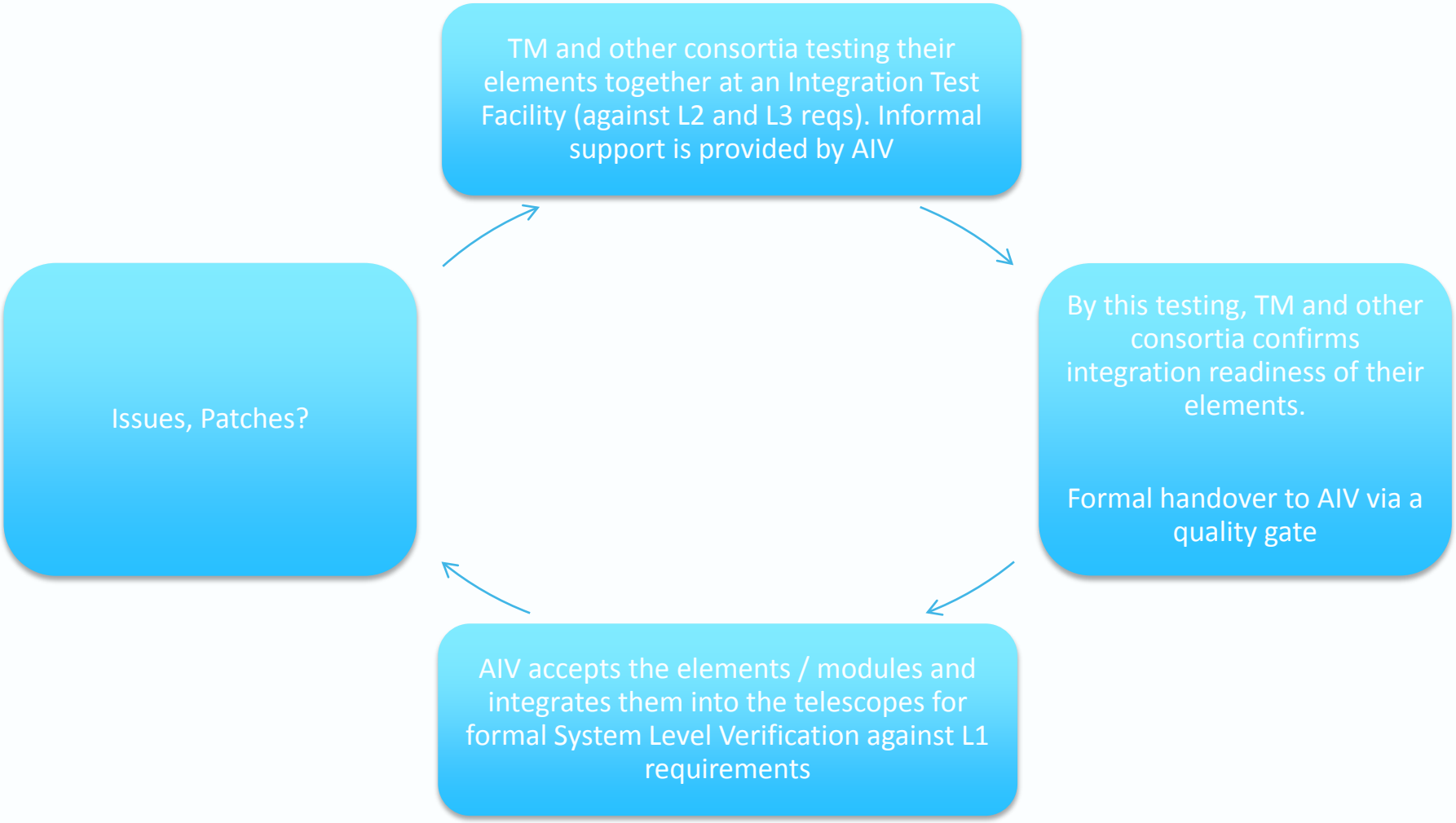
Example: ASKAP Software Deployment

- Using APT/Debian package management
- Single package per LMC
- Runtime configuration on installation via machine name
- Deploy to virtual machines
- Deploy to a test facility
- Deploy to site

Need to Define a Software Release Process

- What are the deliverables?
- What are the processes?
  - How to manage hand-over of SW to AIV
  - Version control / Numbering
  - Quality gates
  - Issue resolution
  - Regression testing
- Roles and Responsibilities

High Level Software Release Process



Thank you

CSIRO Astronomy and Space Science
Craig Haskins

Software Engineer

t +61 2 9372 4308
e Craig.Haskins@csiro.au
w http://www.atnf.csiro.au