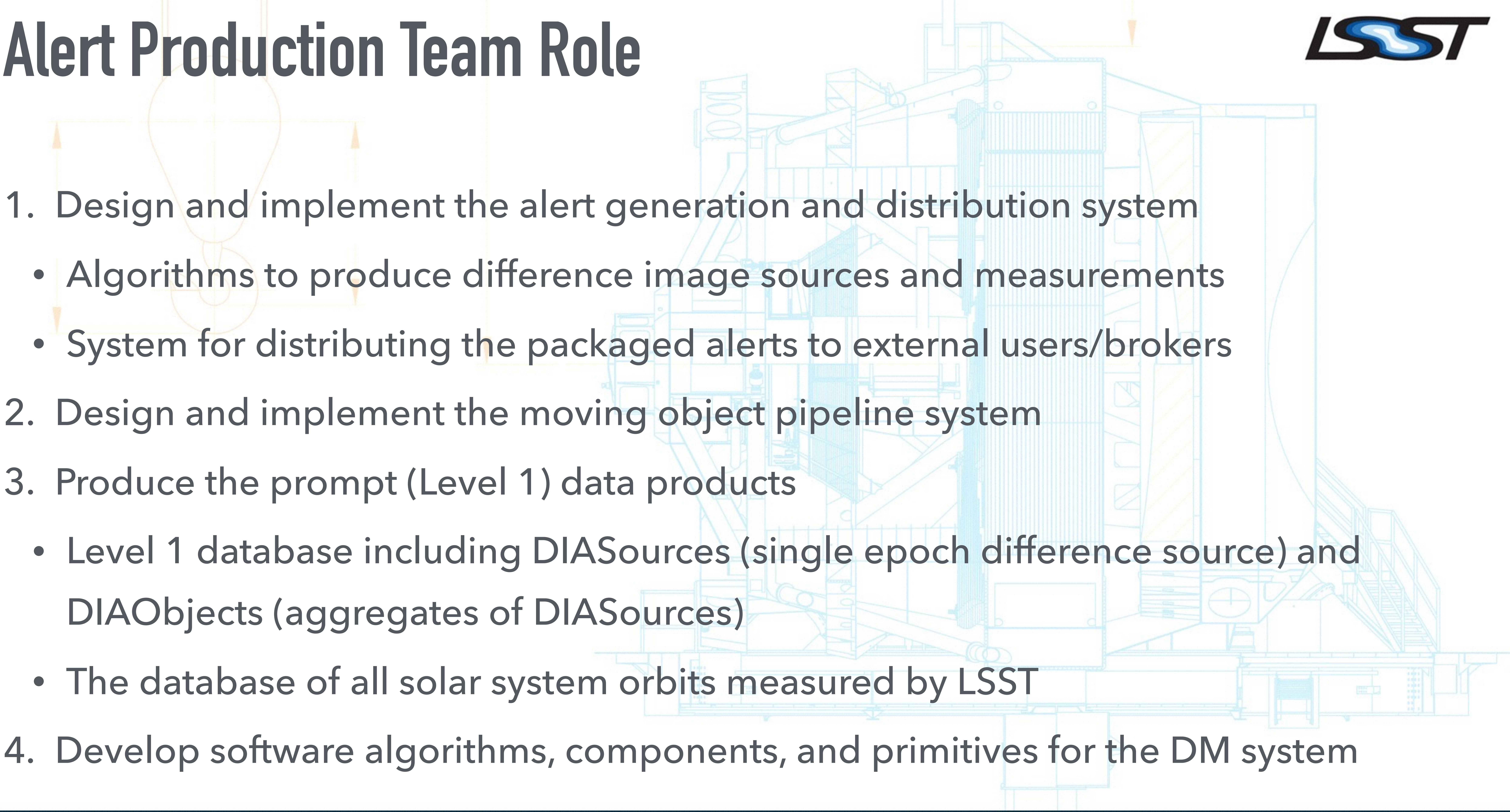
Alert Production
Team Introduction & Development Process

SIMON KRUGHOFF • KRUGHOFF@UW.EDU

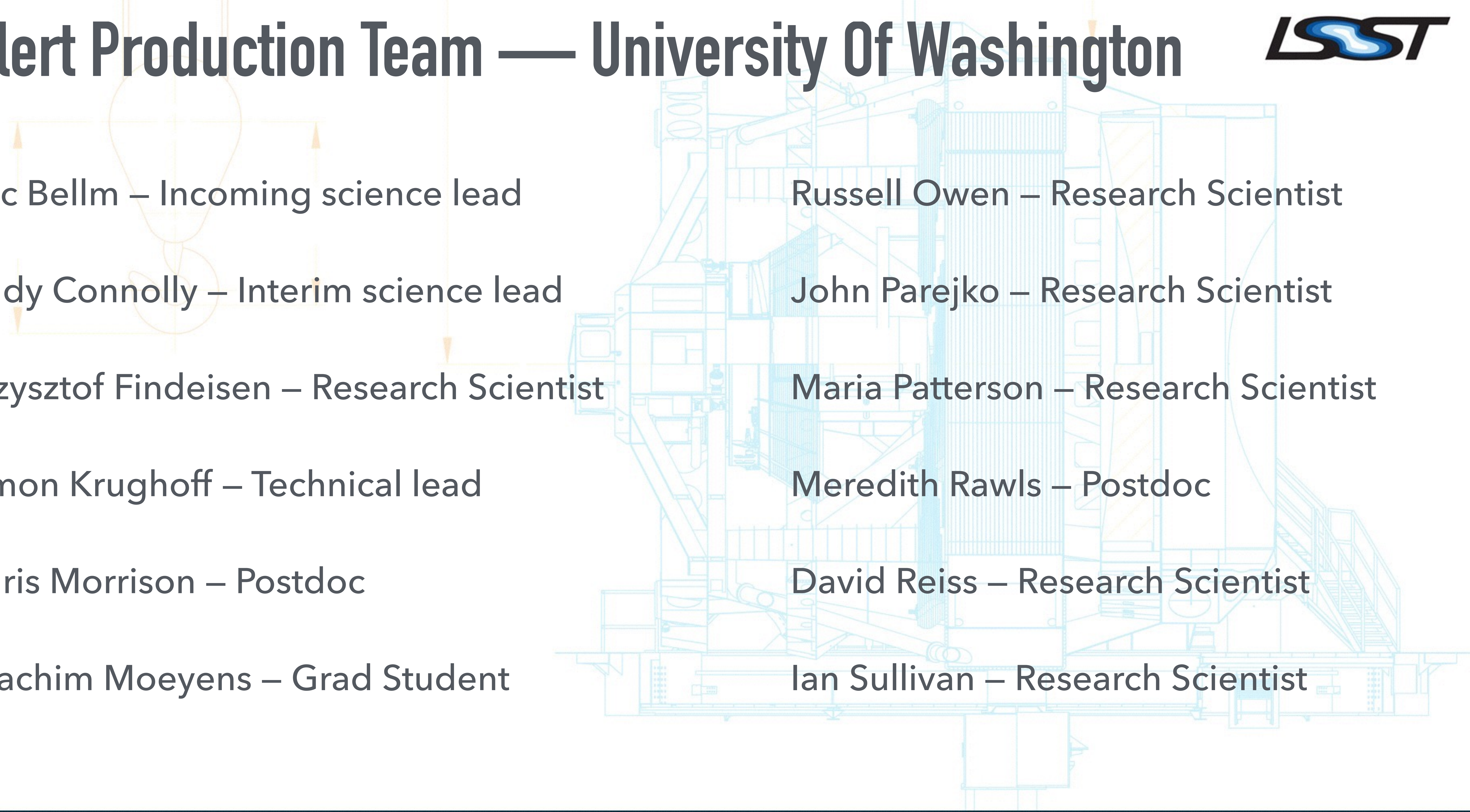
Large Synoptic Survey Telescope

Alert Production Team Role



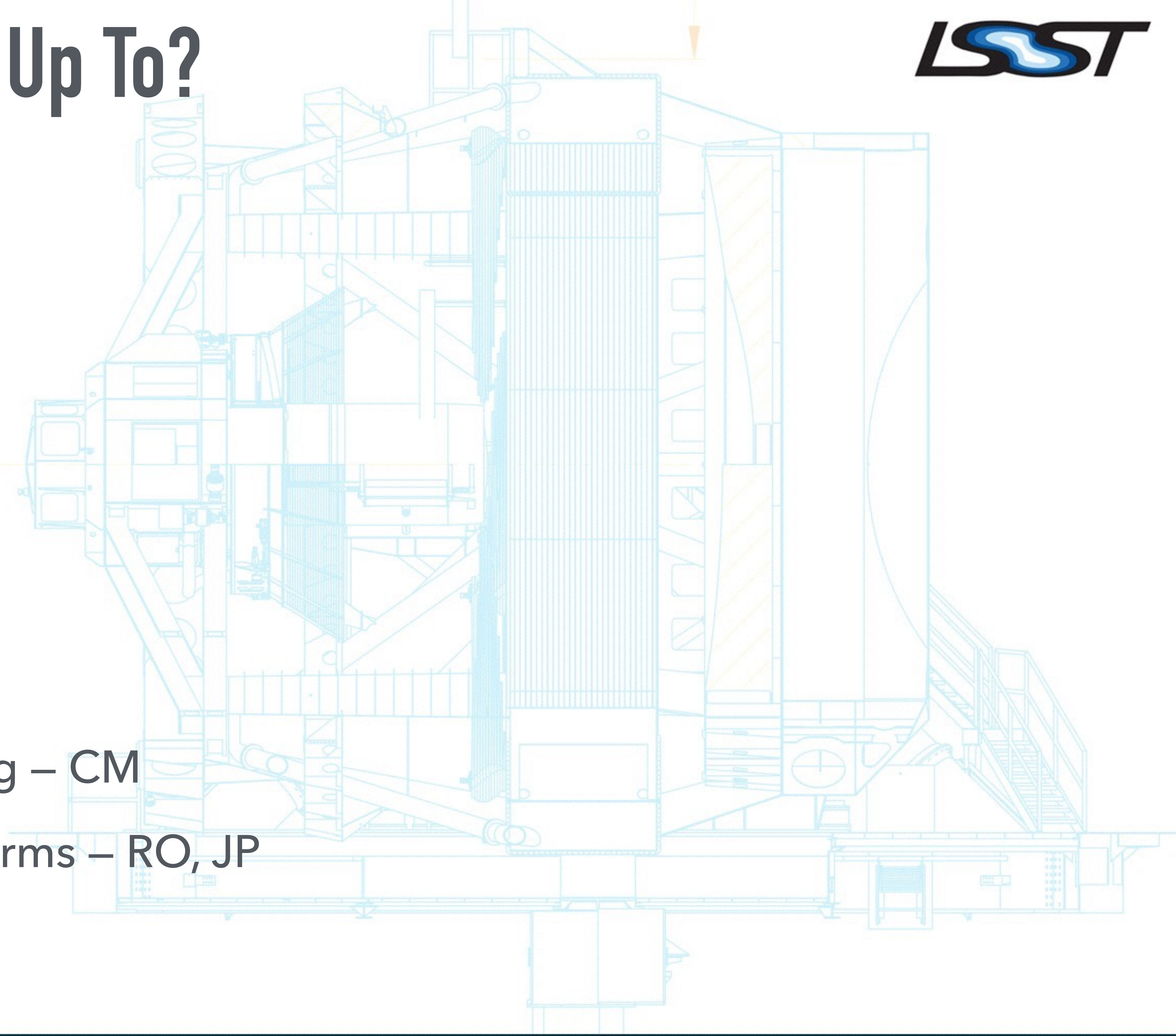
1. Design and implement the alert generation and distribution system
   - Algorithms to produce difference image sources and measurements
   - System for distributing the packaged alerts to external users/brokers
2. Design and implement the moving object pipeline system
3. Produce the prompt (Level 1) data products
   - Level 1 database including DIASources (single epoch difference source) and DIAObjects (aggregates of DIASources)
   - The database of all solar system orbits measured by LSST
4. Develop software algorithms, components, and primitives for the DM system

Alert Production Team — University Of Washington



- Eric Bellm – Incoming science lead
- Andy Connolly – Interim science lead
- Krzysztof Findeisen – Research Scientist
- Simon Krughoff – Technical lead
- Chris Morrison – Postdoc
- Joachim Moeyens – Grad Student
- Russell Owen – Research Scientist
- John Parejko – Research Scientist
- Maria Patterson – Research Scientist
- Meredith Rawls – Postdoc
- David Reiss – Research Scientist
- Ian Sullivan – Research Scientist

What Is Alert Production Up To?

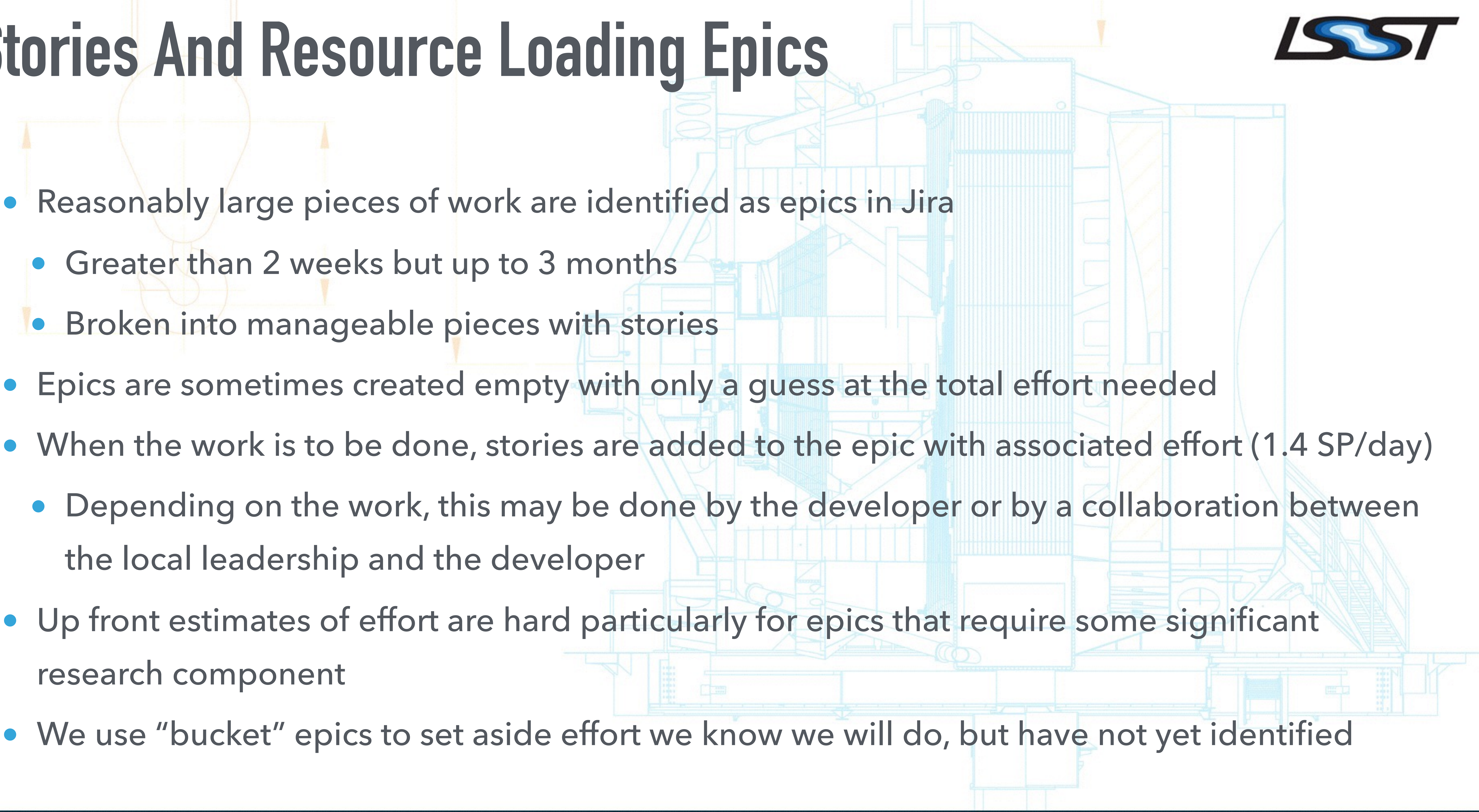


- pybind11 port – RO, KF
- jointcal – JP
- Optimal image difference – DR
- DCR corrected templates – IS
- Alert distribution system – MP
- AP CI system – MR
- Single frame reference matching – CM
- Composable coordinate transforms – RO, JP
- MOPS – JM

Cycle Planning

- Cycles are 6 months, but we have only been loading 3 months worth into Primavera at a time
- Assess carryover
  - Epics sometimes slip so some time in the next cycle needs to be devoted to closing them out
  - Sometimes changing priorities dictate that we need to just leave some epics until later and carry the schedule variance in PMCS until then
- Identify priorities by talking with other teams, the science leadership and the DMLT as a whole
- Choose candidate epics to schedule given team preference, epic story point estimates, and priorities
- Fill a spreadsheet with epics, their start and end dates, and the team member assigned (epics are first put in Jira).
  - This spreadsheet is used by Kevin to create a resource loaded plan in PMCS

Stories And Resource Loading Epics



- Reasonably large pieces of work are identified as epics in Jira
  - Greater than 2 weeks but up to 3 months
  - Broken into manageable pieces with stories
- Epics are sometimes created empty with only a guess at the total effort needed
- When the work is to be done, stories are added to the epic with associated effort (1.4 SP/day)
  - Depending on the work, this may be done by the developer or by a collaboration between the local leadership and the developer
- Up front estimates of effort are hard particularly for epics that require some significant research component
- We use “bucket” epics to set aside effort we know we will do, but have not yet identified

Sprinting

- Sprints are aligned with the DRP team sprints on calendar month boundaries
- Sprints are planned largely by the developers
  - Some stories usually fall over from the previous sprint
  - Choose stories from active epics to fill out the time
    - Active epics are any epic scheduled for the current cycle that are not marked "Done"
  - I then review the sprint, and meet with each of the team members to get sign-off from them that they think what they've signed up for is reasonable (we frequently eliminate some stories at this point)
  - We use the Jira Agile backlogs to plan our sprints by having the developers drag the stories from the backlog into the upcoming sprint

Standups

- We have two of these a week
  - Scheduled in the half hour before my T/CAM coord meetings so time boxed to 30 min
- Each person has a chance to mention
  - Progress
  - Plans
  - Announcements
  - Request for information
- I confirm each persons blocked status and if blocked ask what would unblock them.
- These frequently spawn side conversation and questions to bring up in the T/CAM meetings

Completing Work

- A story is complete when the assignee sets the status of the story to "Done"
  - In practice this almost always requires a review of some sort: i.e. code review for coding stories or a review by interested parties for documentation/research stories
- "Bucket" epics are closed when we have completed enough stories in them to fill the allotted allocation
- Regular epics are complete once all the stories in them have been marked "Done"
  - Again, in practice this usually involves me discussing with stakeholders to make sure that what they expected the product to be is what was done.
- There is currently no formal forum for us to get direct feedback from higher level science leadership on deliverables

Commentary

Large Synoptic Survey Telescope

What Works

- Standups
  - Gives me a chance to catch up with the team and identify blockers
- Jira for tracking work
  - The concept of epics and stories is a very useful and effective way to break big problems into something that can be effectively executed
- Jira sprint planning
  - We are not great at estimating effort yet, but the Jira tools give are good for getting feedback
- Sprint work is largely self assigned
  - Typically, I have to take work out, not add it in
  - Lets the team take ownership

What Has Been Harder

- Defining "Done" has been a little tricky
  - Having a CI system with metrics will help this
  - Having well defined product owners would also help with this (see next bullet)
- Sprint demos haven't happened
  - Clarifying who the product owner is would help with this – it can't be the T/CAM unless we install another scrum master
- We don't sprint together
  - I'm not sure this is actually a totally negative thing and I don't know if it's something we can fix unless we completely change how we have planned to do the work
- Design work requires a lot of overhead
  - Being allowed to ask forgiveness rather than permission would help out a lot with this, but I am unsure how possible that is with a system this distributed