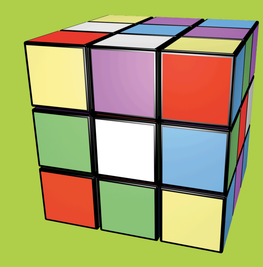
KATHOLIEKE UNIVERSITEIT
LEUVEN

Rik Huygen, et al.



Rubik's Cube® used by permission of Seven Towns Ltd. www.rubiks.com

Providing legacy access to astronomical data analysis software

Problem Statement

Astronomers need access to data reduction tools long after project resources for active developer support is available.

PREPARE DURING DEVELOPMENT PHASE

- ✦ Language choice / environment choice
  - ✦ Popular environments (IDL, Python) likely to be maintained on the infrastructure level
  - ✦ Popular environments close to user expertise, less support needed
  - ✦ Stable framework language which is hardware independent (Java)
  - ✦ Trade-off with provision of test harness, strong typing, etc.. that help quality
  - ✦ Backward compatibility track record
- ✦ Avoid complexity, keep simple things simple
- ✦ Standard product definition

REPAIR AFTER DEVELOPMENT PHASE

Provide a legacy archive with high quality end products

FREEZE

- ✦ Bundle in a robust way
- ✦ Include all libraries
- ✦ Include auxiliary and calibration data
- ✦ Test suite
- ✦ Weed out complexity
- ✦ Virtualization

FOSS

- ✦ Migrate the core legacy logic (pipelines, ...) to a popular environment
- ✦ Make it attractive to FOSS community
- ✦ Extendable and scalable
- ✦ Small packages, wide use
- ✦ Large user base