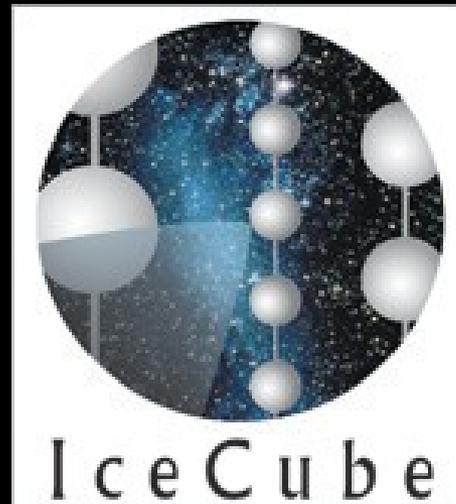


# IceCube DAQ: Lessons Learned during Distributed, Collaborative, Multi-Disciplined Software Development



Keith Beattie  
Lawrence Berkeley National Laboratory

# DAQ Development

A first-hand narrative

- Background
- Experiences
- Observations
- Conclusions



# The IceCube Detector

*Instrumenting 1km<sup>3</sup> of Antarctic Ice to detect extraterrestrial neutrinos*

## IceTop

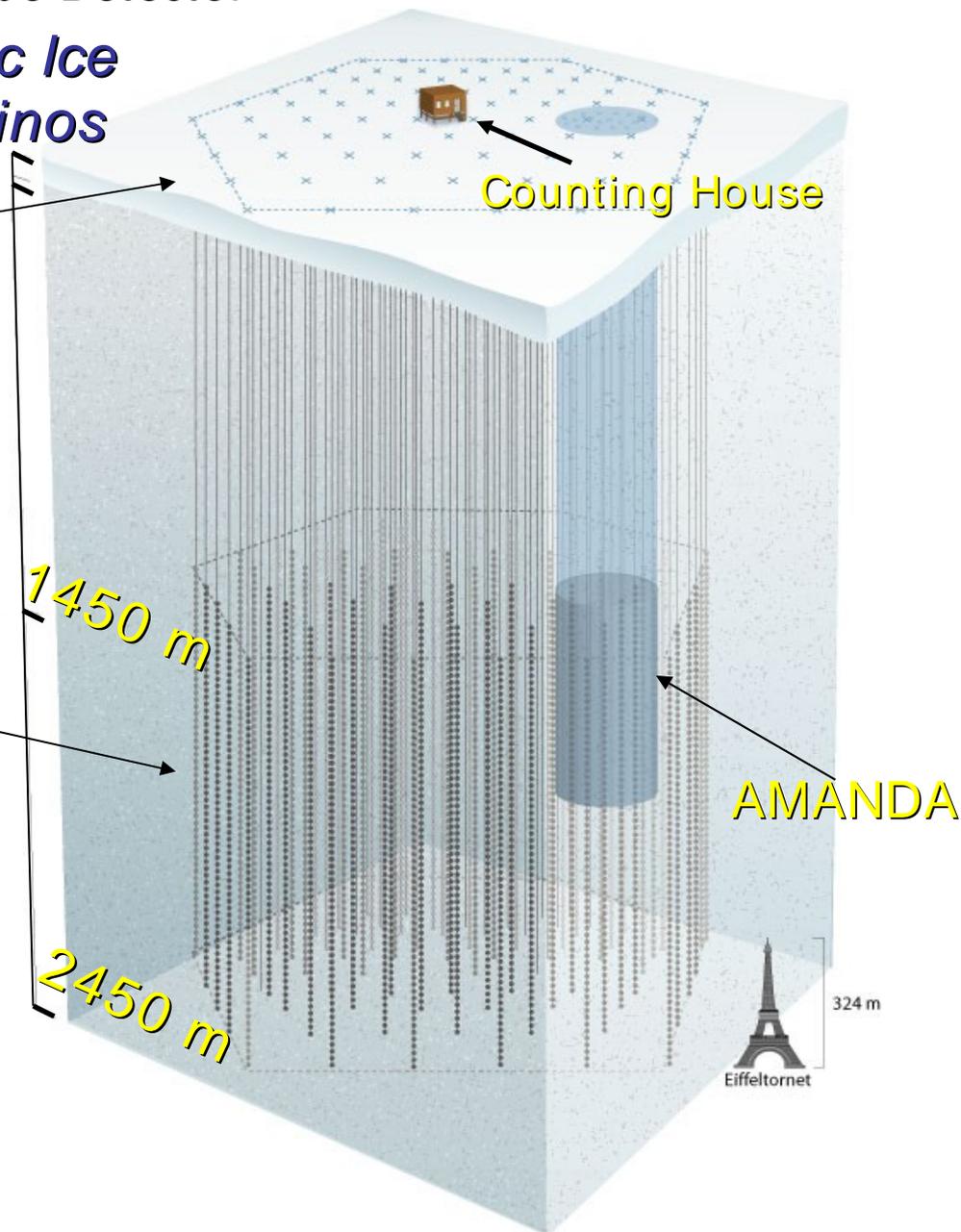
Surface air shower array

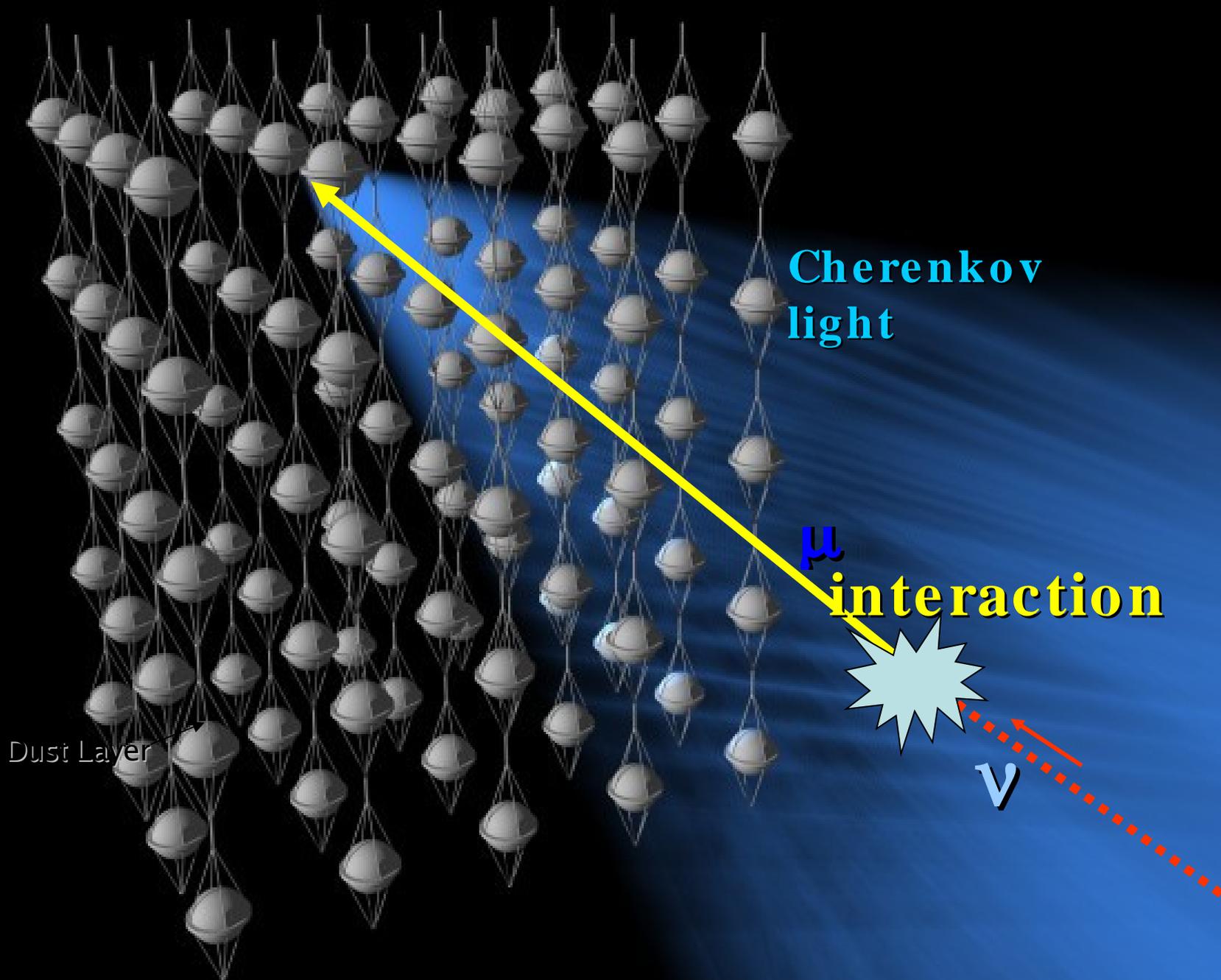
## InIce

70+ strings, each with 60 digital optical modules (DOM)

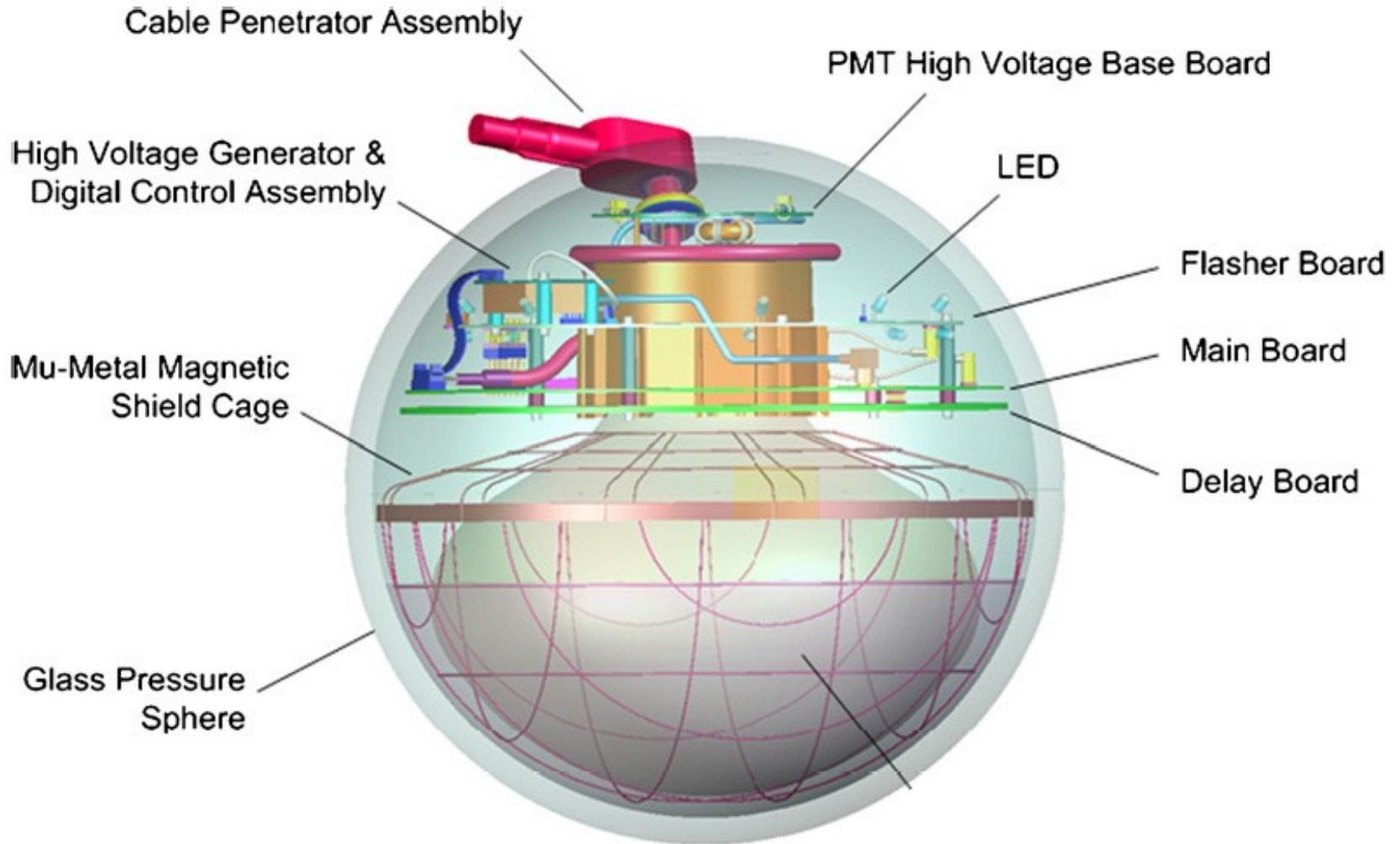
17 m between modules

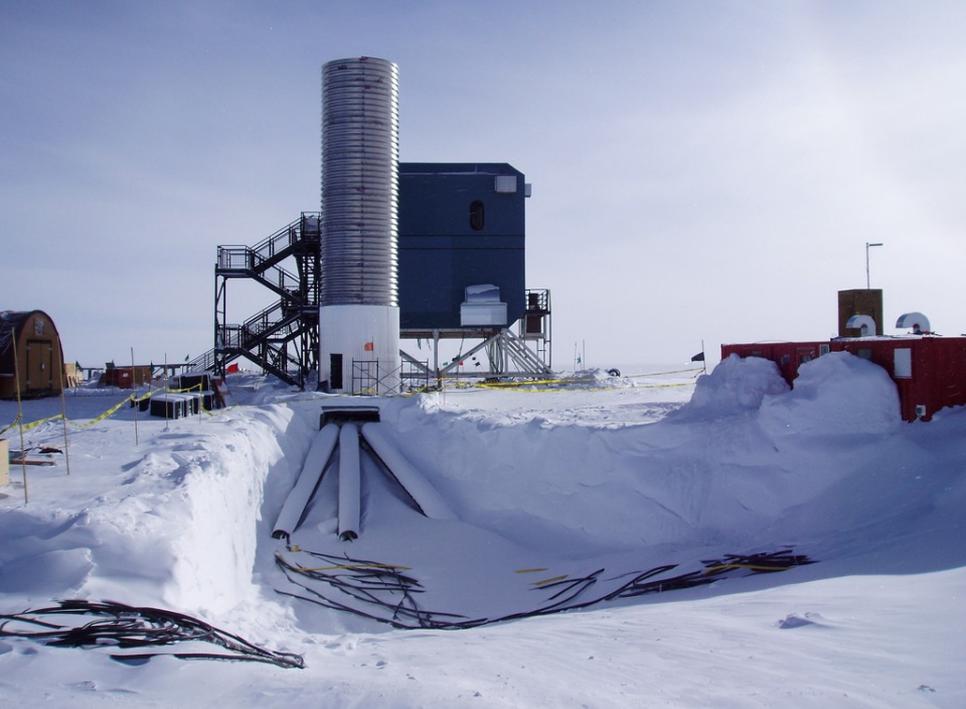
125 m string separation



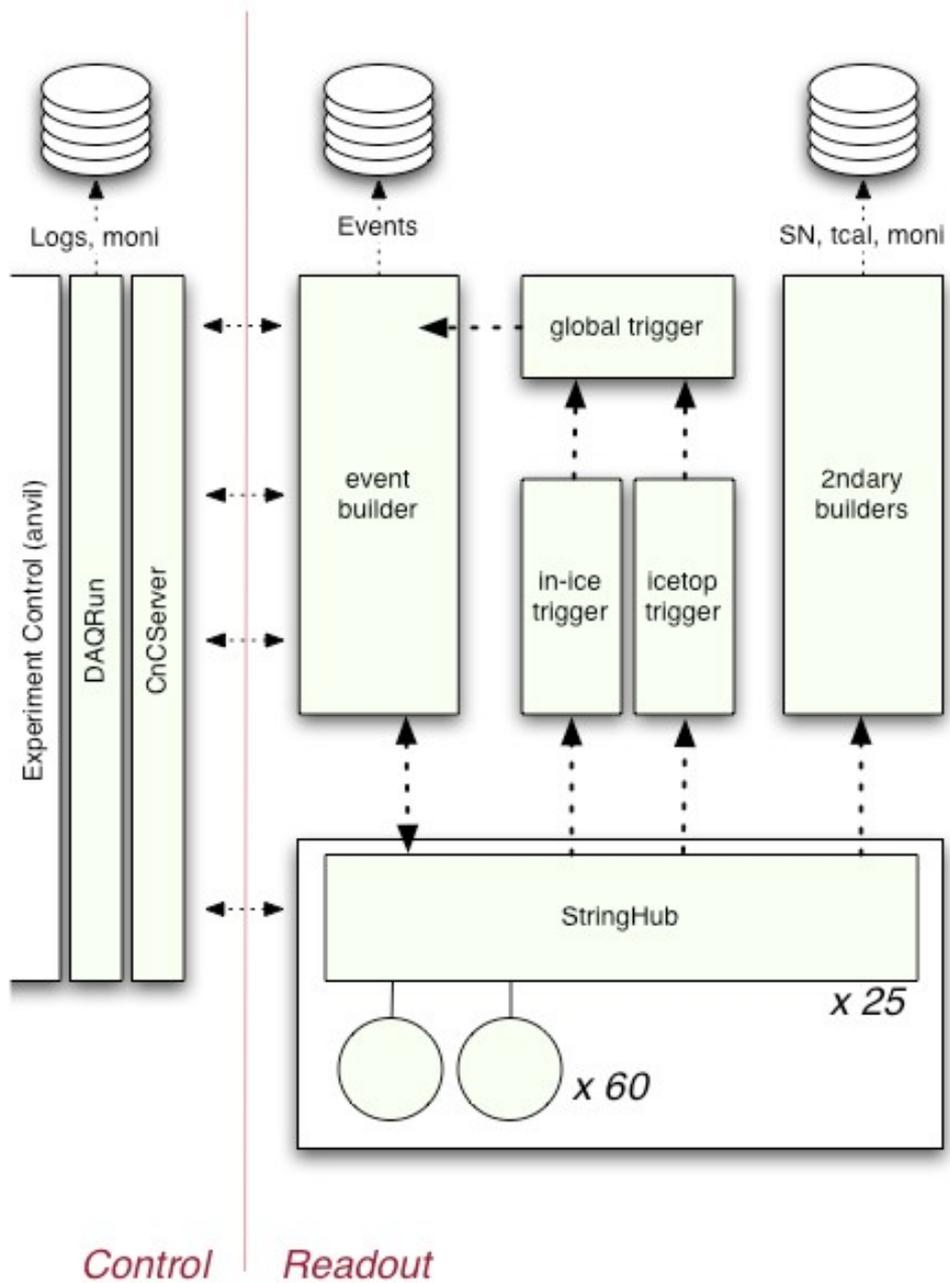


# IceCube's "Eyes": Digital Optical Module (DOM)





# pDAQ System



# Project Management

- Distributed, Multi- Institutional Collaboration
- Remote groups increases complexity
- Multiple Institutions allows for separation of project and supervisory management
- Separation of authority and responsibility
- Bottom- up vs. top- down strategy required at times

# Communications & Tools

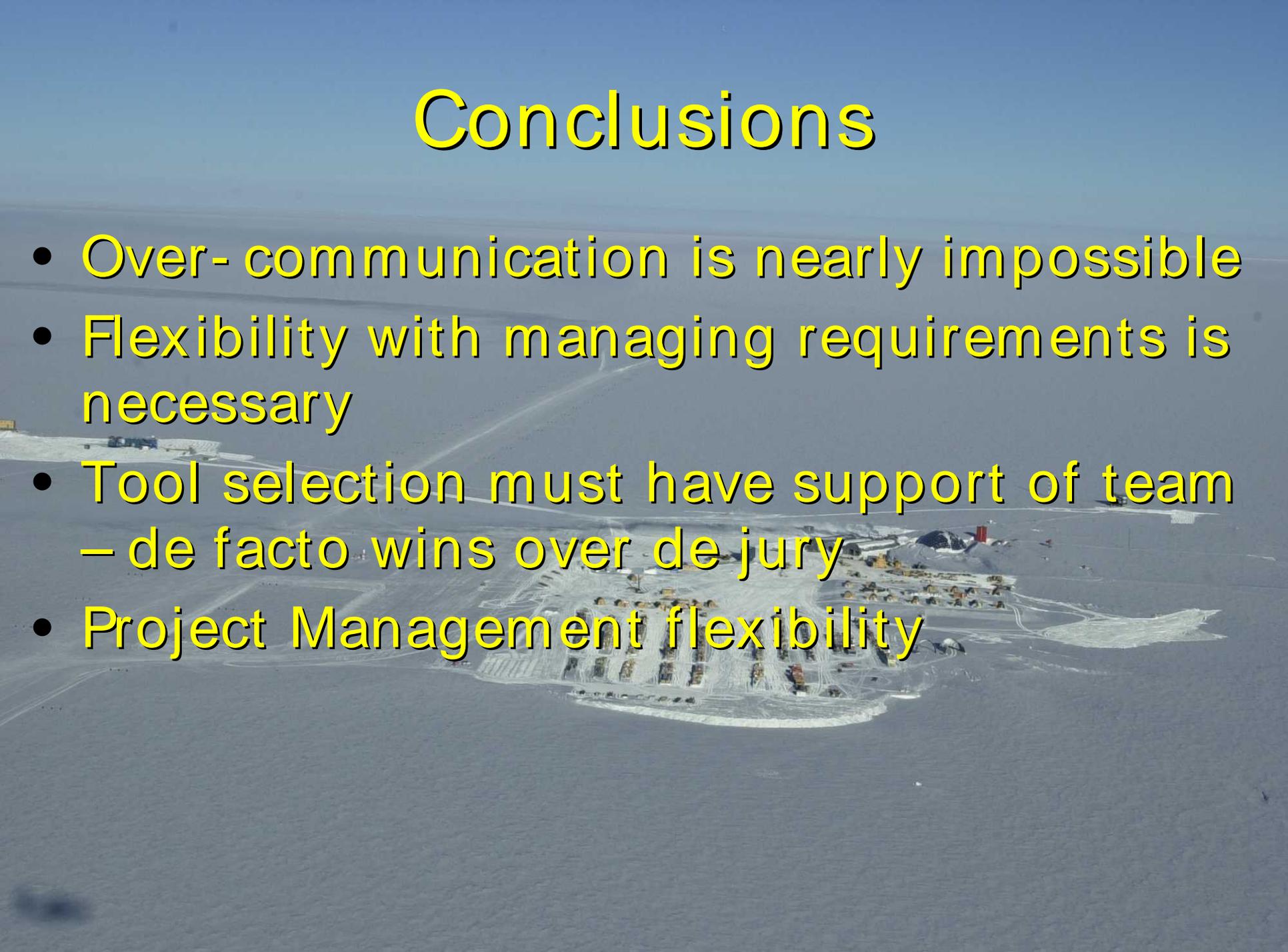
- Over-communication to overcome 'loss'
- Collaborative Tools
  - Phone, conference calls, email, lists, IM, Chat rooms, VoIP
- Development Tools
  - Version Control, Issue Tracking, Build, Test, Dependency, Release systems.
- Nothing beats face-time
  - Build Rapport & team cohesion

# Development Methodologies & Requirement Specs

- Extreme / Agile
  - Assumes requirements change
  - Short dev cycles, then re-eval requirements
- High level- design in place
  - Low level details came with working code
- Flexibility required
  - Multi- Institution
  - Multi- Discipline

# Conclusions

- Over-communication is nearly impossible
- Flexibility with managing requirements is necessary
- Tool selection must have support of team – de facto wins over de jure
- Project Management flexibility





The End