# Software Quality at The GLANCE Project

Rodrigo **Torres**, Fernando **Ferreira**,
José Manoel **de Seixas**, Ana Clara **Cruz**,
Carlos **Brito Filho**, Carolina **Rodrigues**,
Gabriel **Aleksandravicius**, Gabriel **Souza e Silva**,
Gabriela **Pinhão**, Gustavo **Machado**,
Jomar **Pereira**, Leandro **Alves**,
Mario **Simão**, Pedro **Afonso**

Laboratório de Processamento de Sinais, COPPE / Poli / UFRJ

ATLAS EXPERIMENT

lps
COPPE/Poli/UFRJ
Laboratório de Processamento de Sinais
Inteligência Computacional. Inovação
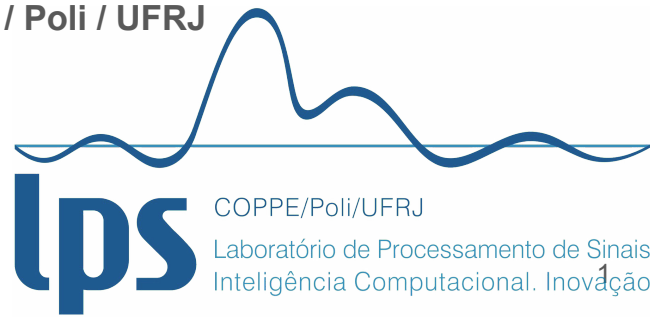
ATLAS and Its Dynamic Environment

- 182 institutions in 41 countries
- Students, physicists, engineers and technicians:
    - Geographically dispersed
    - Different time zones
    - Different experiences
    - High turnover

The challenge is:

- Information centralization
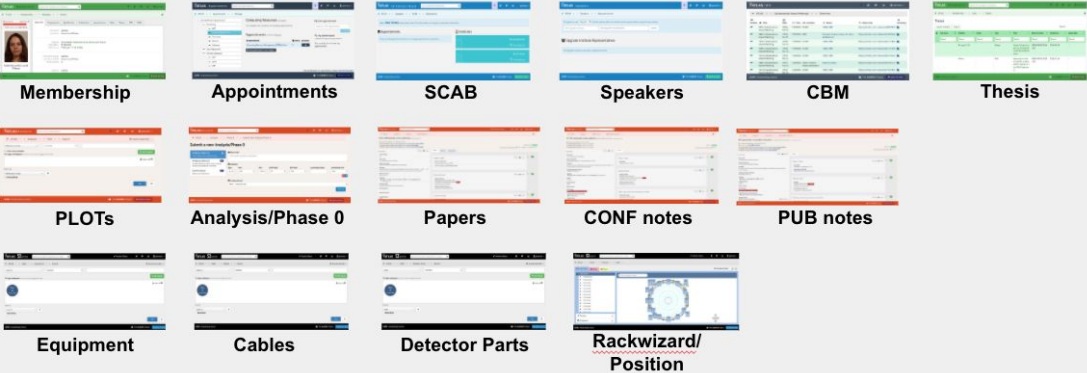- Communication between groups
- Continuity of workflows

What is Glance?

- Technology agnostic framework to integrate data sources
- Search queries can be posed to this data integrated environment, reducing the latency to get the information needed
- Web user interfaces are easy to be developed on top of Glance, allowing the automation of many ATLAS-related processes
- Migration from "super search" to a set of REST API services further improving its usability

The Glance Project

	ATLAS	LHCb	ALICE
Systems	15	4	3
Users	6k	1.5k	1.3k
Requests per week	10	3	5

ATLAS:
Membership | Appointments | SCAB | Speakers | CBM | Thesis
PLOTs | Analysis/Phase 0 | Papers | CONF notes | PUB notes
Equipment | Cables | Detector Parts | Rackwizard/Position

ALICE:
SAMS | Membership

LHCB:
Cables | Equipment | Membership

The Glance Project Team & Resources

- 12 current developers (~30 developers over the project history)
  - ATLAS: 7
  - LHCb: 2
  - ALICE: 3
- Geographically scattered
- Mainly undergraduate (part-time) students from POLI / UFRJ
  - Scholarships
  - Internships
  - In loco internships
- 2 startups whose founders are COPPE alumni with ATLAS background
- Budget: 100,000 SFR/years (CERN + UFRJ)
- Contributions from other institutes (mainly UDINE / Italy and LIPE / Portugal)

Analysis

- Supports ATLAS documents' lifecycle
- Automates email notifications
- Keeps track of documents and reviews stored in CERN Document Server

Membership

- System to mainly manage ATLAS's:
  - Personnel
  - Institutes
  - Authorship
- All information from other systems in one place:
  - Contract dates
  - Appointments
  - Analysis papers
  - Given and future talks
  - Operational Task Planning data
  - Thesis
  - Volunteer for talks
- Institutes and Funding Agencies
- Plot generator

Glance Main Users

- ATLAS deputy spokesperson

- Physics coordinators

- Publication Committee chairs

- ATLAS Secretariat

- Authorship Committee

- Physics Office

- Resource Coordinator

- Technical Coordination

- TDAQ Upgrade Project Leader

Software Quality in GLANCE

- Centralized requests

- Requirements documentation

- Virtual communication

- Agile method

- Automation

Goals

- Ensure your long-life software
- Support business processes grow
- Quick start new developers
- Overcome remote team limitations

How to make it all work?

- Fast iterations

- Testing & Automation

- Concrete and visual artifacts such as mockups,

  diagrams, and description of scenarios

Anticipate validations as much as possible!

Cost of Change

Requirements | Analysis and Design | Coding | Testing in the Large | Production

http://www.agilemodeling.com/essays/costOfChange.htm
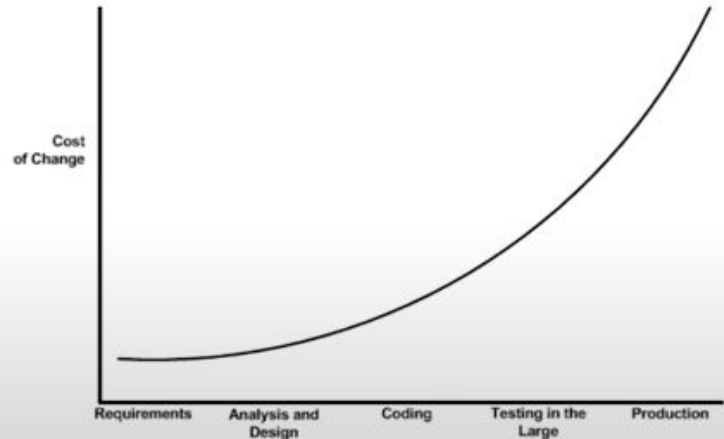
Tooling

Engage the whole team in the development process:
programmers, QA team, stakeholders, clients.

- Pair programming
- Continuous Integration
- Test Driven Development
- Design Guided By Examples
- Ubiquitous Language
- Communicate design and architecture decisions

Cost

Programming defect found via
Pair Programming

Programming defect found via
Continuous Integration

Design or programming defect found
via Test Driven Development (TDD)

Requirements or design defect found via
Active Stakeholder Participation

Requirements or design defect
found via Model Storming

Defect found via independent
parallel testing

Requirements defect
found via traditional
acceptance testing

Design defect
found via traditional
system testing

Programming defect
found via traditional
system testing

Defect found via a
review or inspection

Length of Feedback Cycle

Copyright 2006-2009 Scott W. Ambler

http://www.agilemodeling.com/essays
/costOfChange.htm

Testing

Release smaller working pieces of software to catch missing bugs with the help of users!

- Keep documentation to a minimum using tests as specifications
- Automate your test suites to have a short feedback loop and keep your codebase safe
- Hierarchical testing structure:
  - Unit
  - Functional
  - System
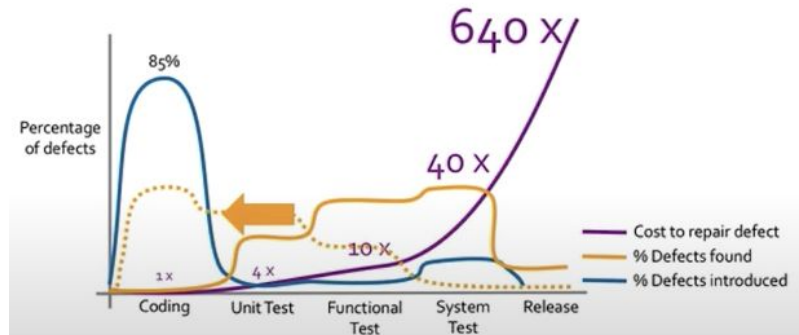
Software defect-removal efficiency
Capers Jones et all, 1996

Key Takeaways

- Make sure everyone is following up
  - Define communication channels and reinforce its proper use
  - Make use of visual representations of the domain being modelled (diagrams)
  - Create a single source of truth (Confluence)

- Unburden common bottlenecks
  - Code Review policies
  - Documenting architecture decisions (ADRs)
  - Creating discussions around every new implementation (RFCs)
- Keep technical debt visible and tackle it (JIRA)
  - Consider paying it when negotiating new implementations with stakeholders.
  - Include technical milestones in medium to large sized projects.

Foreseen Challenges and Next Steps

- Measure the development progress and efficiency success
    - Can the team move forward or keep re-implementing old requests?
    - How to measure developers' engagement to the process?
    - How much time a new requirement takes to be released?
- Assess software quality continuously by using machine learning techniques.