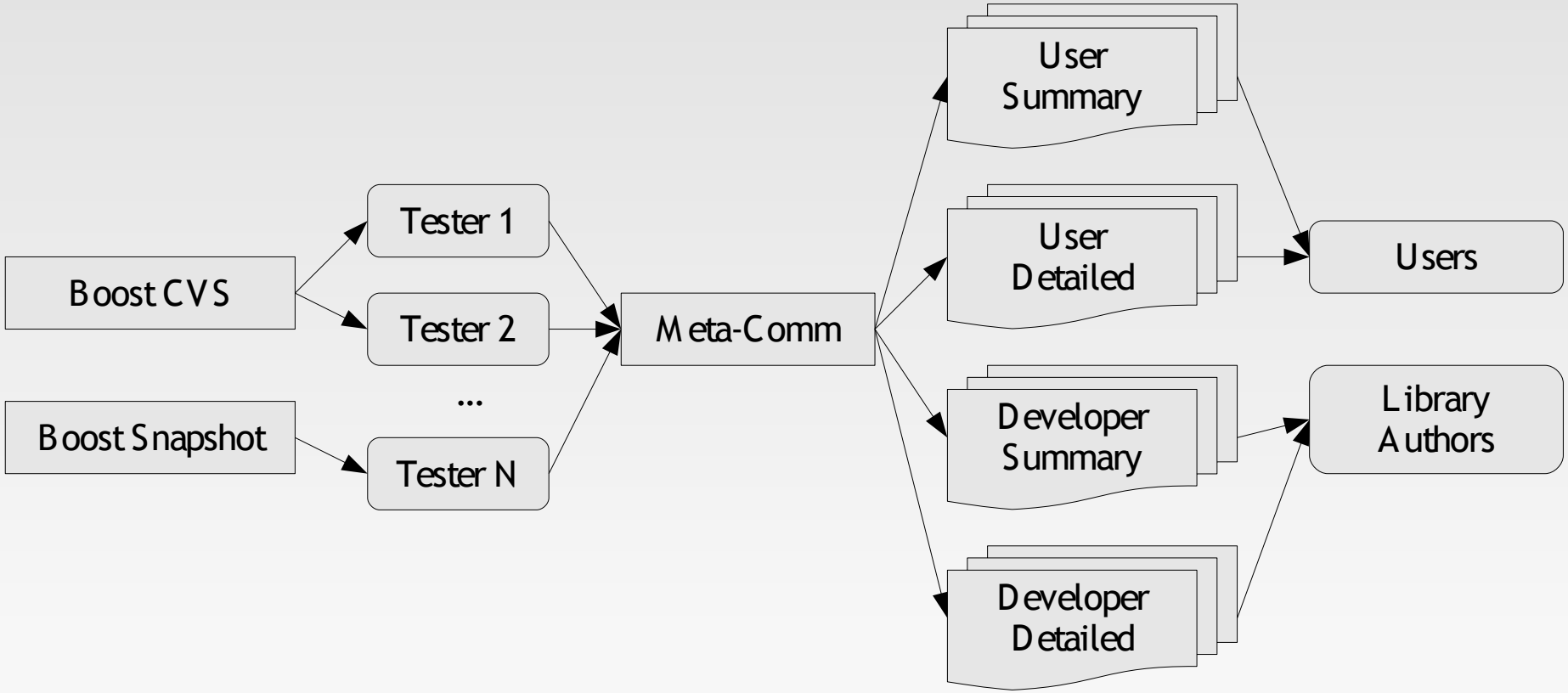
Testing Boost:
a Planning and Design Sprint

- Overview of current system, and it's problems.
- Boost requirements for a test system.
- Existing technology review.
- Design/plan solutions.

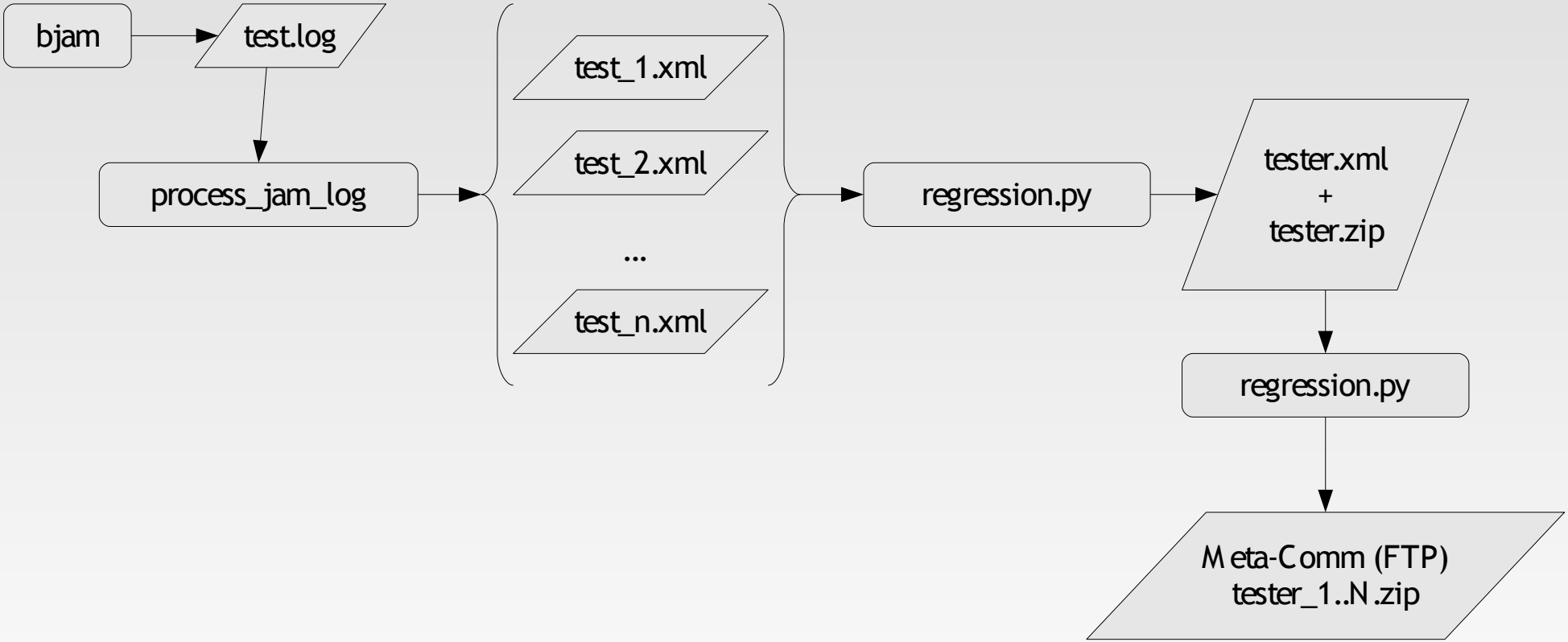
Current System: Overview

- From CVS to users and authors getting results:



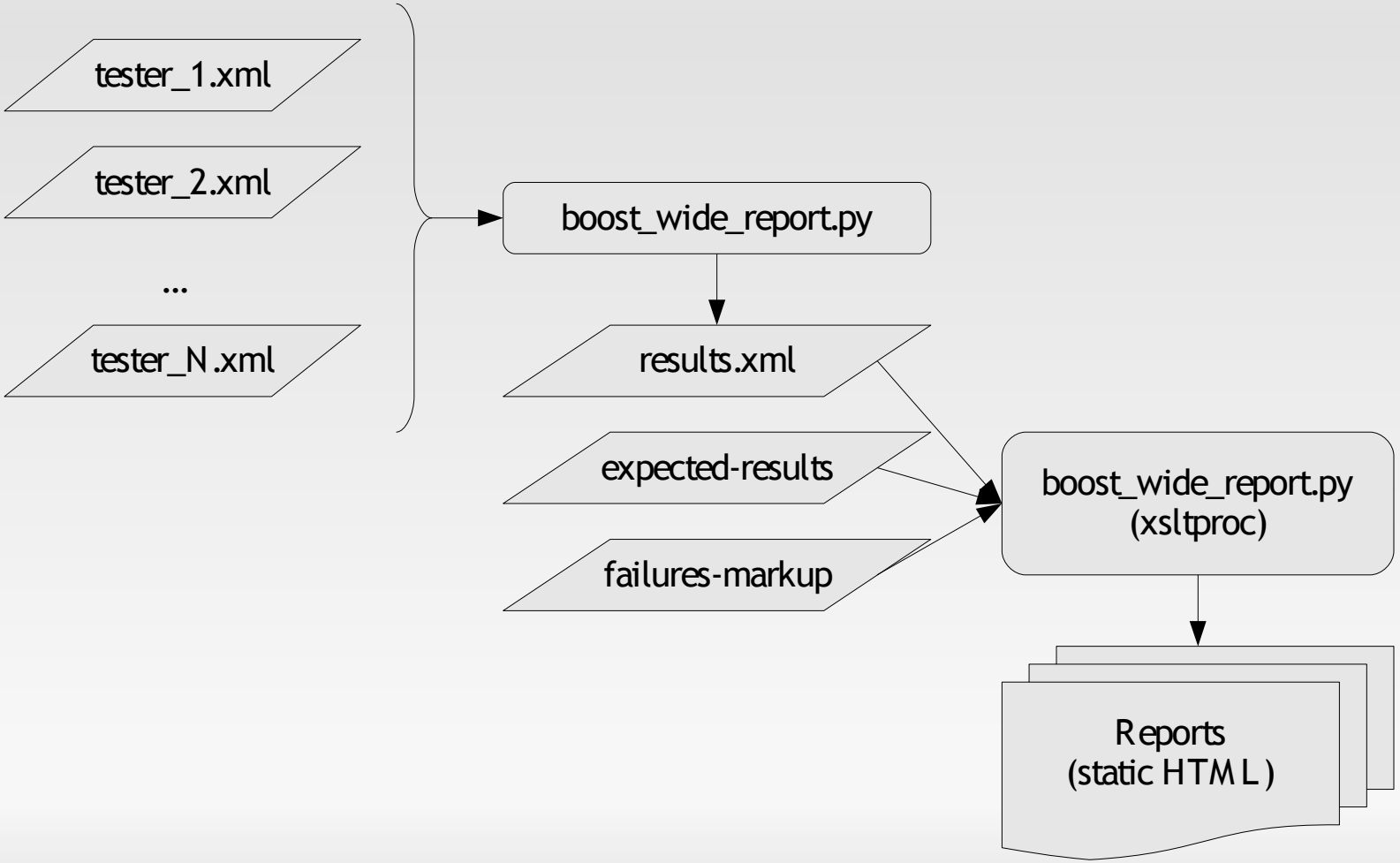
Current System: Overview

- Testers execute tests and generate a log:



Current System: Overview

- Server processes tester logs to results:



Current System: Problems

- Large results delay
- Not scalable
  - Hard for testers to control resource use
  - Results processed in one batch
- No tracking of state
  - Minimal test environment information
  - No historical results

Requirements

- Developers
- Testers
- Users
- System

Existing Technology

- Buildbot
- QMTest
- Dart
- CMake
- Boost.Test

Solutions

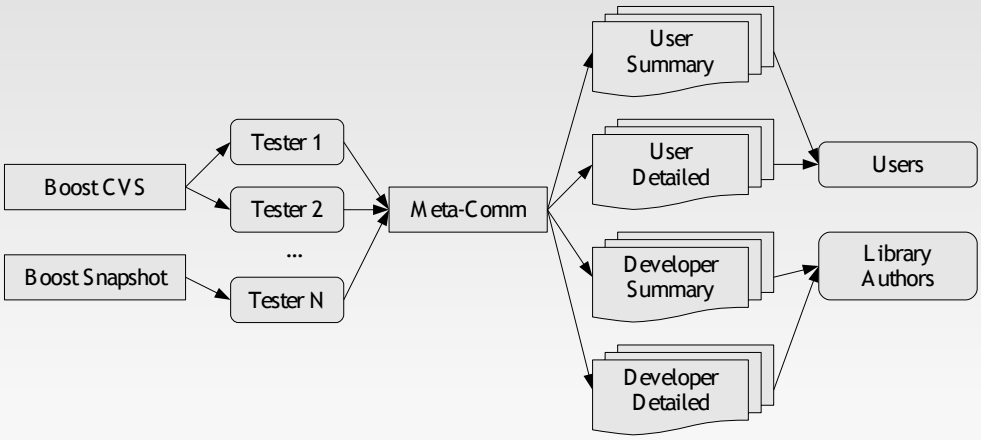
- Anything is possible... Nothing is perfect.

Testing Boost:
a Planning and Design Sprint

- Overview of current system, and it's problems.
- Boost requirements for a test system.
- Existing technology review.
- Design/plan solutions.

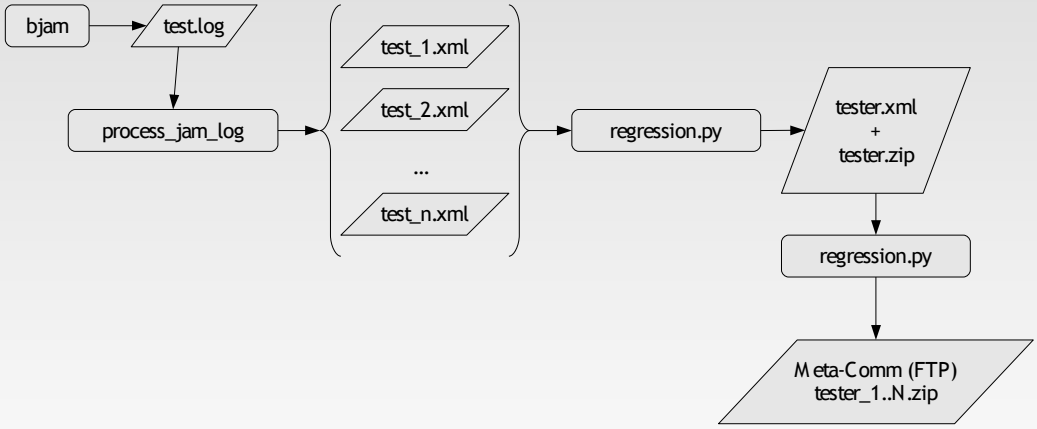
Current System: Overview

- From CVS to users and authors getting results:



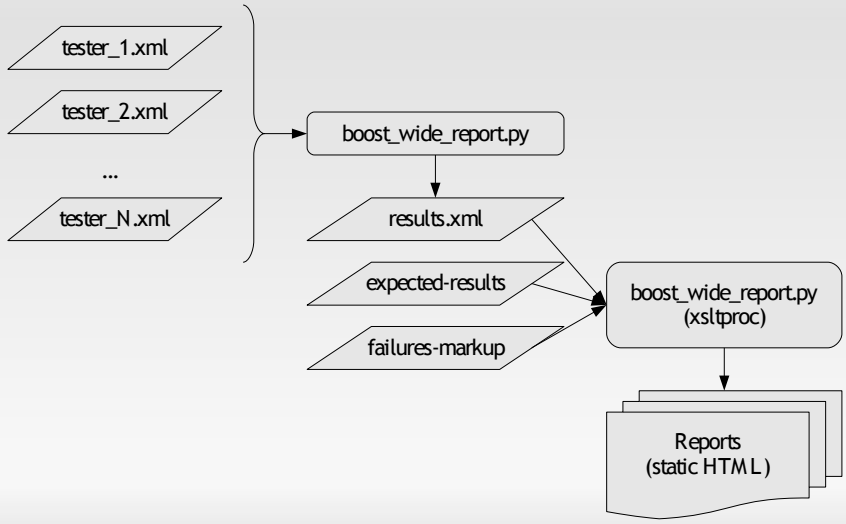
Current System: Overview

- Testers execute tests and generate a log:



Current System: Overview

- Server processes tester logs to results:



Current System: Problems

- Large results delay
- Not scalable
  - Hard for testers to control resource use
  - Results processed in one batch
- No tracking of state
  - Minimal test environment information
  - No historical results

Requirements

- Developers
- Testers
- Users
- System

Developers

Immediate results (10 minutes)
Experimental tests

Testers

Minimal setup (zero-install)
Specify tests to execute
Specify machine resources
Running tests when changes affect real code
No need for manual intervention for testing changes

Users

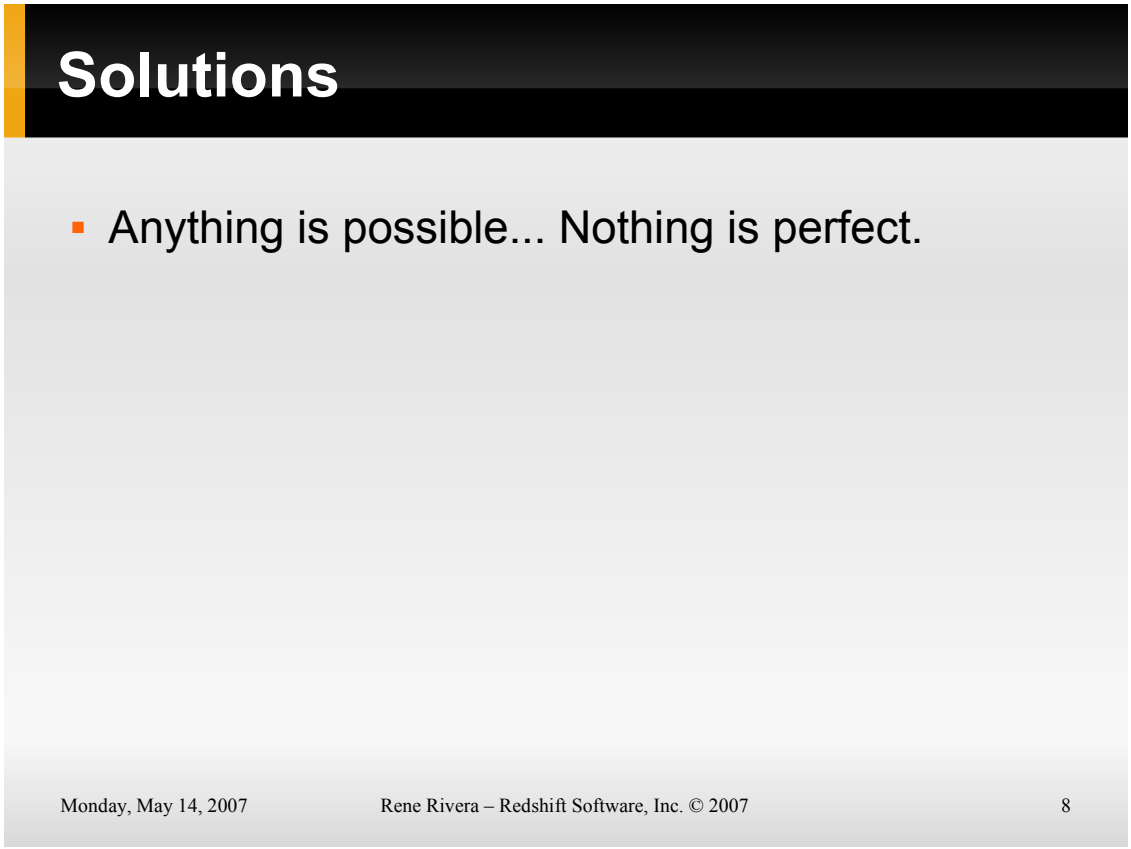
System

Piecemeal result posting
Test groups, to define tests to run
Reliability
    No false results
    Monitoring of system
    Self-testing
    Must work 95% of the time

Existing Technology

- Buildbot
- QMTest
- Dart
- CMake
- Boost.Test

- **Buildbot** <http://buildbot.net/>
  - Manages testing resources.
- **QMTest** <http://www.codesourcery.com/qmtest/>
  - Defines and runs tests.
- **Dart 2** <http://www.na-mic.org/Wiki/index.php/Dart2Summary>
  - Web based test results reporting.
- **CMake** <http://www.cmake.org/>
  - Build system frontend. Generates projects/makefiles for existing build tools.
- **Boost.Test** <http://boost.org/libs/test/doc/index.html>
  - Implementation and execution of C++ tests.



Solutions

Dart Test reporting system

- Test the feasibility of using Dart to replace the XSLT reports.

Plan: Near Term

Replace reporting system

- Doug & Troy; will make a test case of using CMake+CTest+Dart for at least one of the Boost libraries.
- Rene; will make changes to regression.py and collect_and_upload_logs.py to additionally submit results to Dart. (COMPLETED, Dart server at <http://beta.boost.org:8081>)
- Noel & Rene; will make changes to Boost.Jam and Boost.Build to submit results directly to Dart.
- Everyone; evaluate results as available in Dart.

Plan: Long Term