A Field Study of Collaborative Software Development Teams (Initial Results)

Cleidson de Souza[1,2]

John Penix[2]

Maarten Sierhuis[2]

[1]UC, Irvine Graduate Student and
NASA/Ames Summer Intern

[2]NASA/Ames Research Center

Overview

- The Setting: CTAS
- Methodology
- Initial Findings
- Future Work
- Conclusions

The Setting:
Center TRACON Automation System (CTAS).

- A suite of automation tools developed at NASA/Ames designed to help air traffic controllers to manage air traffic flow at large airports.
- In 1991 it was chosen by the FAA as the future automation system for the terminal area.
- Since then it has been used in 6 different airports.

The Setting:
Center TRACON Automation System (CTAS).

- CTAS is composed of 10 different tools.
- Source code:
  - C and C++. GUI's are being ported to Java.
  - 1,000,000 LOC .
- Development Team:
  - Number of developers: 31 .
  - Two groups: V &V and Developers.
  - Work in processes, instead of tools

Methodology

- **Field Study**
  - Five weeks in the field until now, four more weeks to go.
- **Data Collection**
  - Participant Observation
    - "Shadowing" developers with different roles.
  - Interview Techniques
    - 4 interviews until now ranging from 45 to 120 minutes.
- **Data Collected**
  - Several artifacts collected
  - What developers do, how, when, where they do, and most importantly WHY they do it.

Initial Results

- Most important tools:
  - configuration management; and
  - bug tracking system.
- These tools provide shared repositories for source code and change requests.
- The CM and the bug tracking tool provide automation of some tasks like:
  - Version control, identification of releases, report generation, and so on.

Initial Results

- Developers adopt conventions to use these tools so that they users might cooperate effectively.
- Examples:
  - Naming conventions for creating branchs and views to work with the CM tool;
  - Priorities and severities of the bugs in the bug tracking tool.

Initial Results

- However, the conventions adopted by the developers are not automated.
- Examples:
  - Previous naming convention;
  - E-mail sent by developers right before the check-in.

Initial Results

- **Important communication using e-mail:**
  - **Is it the most effective tool to provide notifications?**
  - **On the other hand, e-mail is also used as a learning tool by new developers, so that they can be aware who is responsible for what process. This information is later used when one has to fix a bug in that process.**

Short Summary of Results

- Coordination using CM and bug tracking
- Use of Conventions
- Communication using E-mail
  - Problematic in some cases; but
  - Provides awareness of others work.
- Intense Parallel Development

Future Work

- Data Collection for 3 more weeks.
- Analysis of the data
  - Grounded Theory
  - Brahms multi-agent model
- Ultimate goal:
  - Identify requirements for technology support for this group.
  - If necessary, develop this technology.

Conclusions

- CTAS:
  - Successful project developed at NASA/Ames.
- Methods
- Initial results
  - Important tools used by the developers; and
  - Problems with these tools
- Future Work