Software Engineering Research/Developer Collaborations (CI04)

Final Report

(External release)

March 11, 2005

Table of Contents

1 Overview

In 2004, six collaborations deployed software engineering technologies on NASA projects. The main purposes were to benefit the projects, infuse the technologies if beneficial into NASA, and give feedback to the technology providers to improve the technologies. Each collaboration project produced a final report. This report: 1) summarizes each project, drawing from the final reports; 2) indicates paths to further infusion of the technologies into NASA practice; and 3) summarizes some technology transfer lessons learned.

Below we restate our success criteria from our proposal:

"We would like one of the main success criteria to be that the research products used in the collaborations are adopted for future software development by the teams (or organization). However, this is unrealistic for mid TRL-level research products that may lack productization, and it may be unrealistic for high TRL or even for commercial products (for example, the license fee may be too high for a single team to bear). Thus we have identified several other success criteria.

1. The success criteria of the collaboration projects funded under this proposal are met. This includes a positive rating for each product on the evaluation criteria metric.
2. The research product is adopted by the collaborating software development team for current use.
3. The research product is included in a list of recommended development practices at a NASA Center or by contractor.

4. The software development team using the product provides feedback, including performance data, to the research team to guide future development of the product.
5. Six months after the funded collaboration period, the research product is still being used by the development project or by a successor development project.
6. The researchers and consumers recommend to the CTO methods of making future versions of the research products available within NASA (for example, by Open Sourcing or by licensing the technology commercially or to organizations such as the Southwest Research Institute).
7. Independent of the success of the collaborations, "lessons learned" regarding the challenges and success factors for software development technology infusion within NASA."

A modification of 3 is "The research product is recommended for a branch, division, or directorate at a center". That is the statement for which column 3 applies in the table below.

Also relevant to judging the impact of the collaborations is the penetration factor (PF) used for SARP quarterly reviews:

PF 8: Data passed back to project;
PF 9: Results actually used by the project.

Project	PF	1	2	3	4	5	6	7	Impacts
ARC CGS on ISS payload software	9				Y			Y	Found 2 errors to be fixed. Useful feedback to the CGS developers.
GSFC PBI in Flight Software Branch	9	Y	A	Y	Y	A		Y	PBI led to changes in a project's development plan. Expect roll out o PBI in FSB standards.
JPL ODC on ground software	8 now, will soon be 9	Y	Y	Y	Y	A		Y	Training occurred in several JPL organizations. ODC led to several recommendations that will be used project maintenance phase. Collaboration is continuing to infuse ODC on another project.
JSC CodeSurfer for Inspections of ISS software	9	Y	Y		Y			Y	Found 12 additional (minor) defects Tool is continuing to be used and promulgated.
MSFC SWAT & CGS on Flight Software	9	½	Conditional on cost	Conditional on cost	Y			Y	Useful feedback to the CGS developers. SWAT found 9 defects worth fixing in the software, some o which had escaped formal testing.
USA PBI on ISS software	9	Y	Y	Y	Y			Y	Found 6 major defects, several of which had escaped previous inspections, and/or occurred in reused code. Will continue to be used and is recommended as ar

									optional process.

Y = Yes **A** = Anticipated in 2005 – 2006 timeframe

2 Summary of technology provider/software development project collaborations

This section describes briefly each collaboration: its objectives, what transpired, its impact on the project, and the success criteria that were met.

2.1 GSFC: "GSFC FSB Application of Perspective-Based Inspections"

The goal of this collaboration was to produce Flight Software Branch (FSB) process standards for software inspections which could be used across three new missions within the FSB. The standard was developed by Dr. Forrest Shull (Fraunhofer Center for Experimental Software Engineering, Maryland) using the Perspective-Based Inspection approach, (PBI research has been funded by SARP), then tested on a pilot Branch project. Because the short time scale of the collaboration ruled out a quantitative evaluation, it would be decided whether the standard was suitable for roll-out to other Branch projects based on a qualitative measure: whether the standard received high ratings from Branch personnel as to usability and overall satisfaction. The project used for piloting the Perspective-Based Inspection approach was a multi-mission framework designed for reuse. This was a good choice because key representatives from the three new missions would be involved in the inspections.

The perspective-based approach was applied to produce inspection procedures tailored for the specific quality needs of the branch. The technical information to do so was largely drawn through a series of interviews with Branch personnel. The framework team used the procedures to review requirements. The inspections were useful for indicating that a restructuring of the requirements document was needed, which led to changes in the development project plan.

The standard was sent out to other Branch personnel for review. Branch personnel were very positive. However, important changes were identified because the perspective of Attitude Control System (ACS) developers had not been adequately represented, a result of the specific personnel interviewed.

The net result is that with some further work to incorporate the ACS perspective, and in synchrony with the roll out of independent Branch standards, the PBI approach will be implemented in the FSB. Also, the project intends to continue its collaboration with the technology provider (Dr. Forrest Shull) past the end of the grant, to allow a more rigorous quantitative evaluation.

Success criteria 1, 3, 4, and 7 were met, and 2 and 5 are anticipated.

3　Paths to further infusion of the technologies

3.1　Perspective-Based Inspections

Dr. Forrest Shull has developed a course syllabus for formal inspections. The syllabus includes tailoring for the attendees. The Knowledge and Training subgroup of the intercenter Software Working Group (SWG) is soliciting interest across NASA in the course, with the intent of funding it in FY05 if interest is sufficient.

At the moment, implementing Perspective-Based Inspections has a tailoring component. Dr. Shull expects that there is a limit to the number of perspectives that need to be produced for software. He provided a tailoring kit to USA.

4　Technology transfer lessons learned

1. Some developers are not proficient at research-oriented activities and need guidance and oversight. These teams are likely to benefit from more detailed *pro forma* documentation or templates (kick-off meeting agenda, project plan, reports). For specific categories of tools (such as source code analysis tools) we can provide very detailed templates. They also require frequent oversight (a) to be sure communication is occurring between developers and tech vendors and (b) to be sure that the schedule is being followed. Not all the projects require this level of support but it is likely to benefit Research Infusion by promoting uniform, higher-quality collaboration practice.

2. There are various answers to the question "What is the next step" – from research infusion to technology transfer. A general solution is unlikely. Some technologies are readily integrated and generalized into a parent organization's existing processes (for example, Perspective-based Inspections at GSFC) – they are modifications to existing processes. Various other technology-specific approaches may be appropriate; e.g., PBI may be supported by the Software Engineering Initiative's Training strategy.

3. Tighter qualification of technology / project combination may be needed. One of the source code analysis tools used at ARC and MSFC had previously been successfully applied to NASA software, but software that had different technical features. The tool did not transition well to software that did not have these features. Also, the appropriate lifecycle context and purpose for the tool (in this case) may not have been clear to the development teams.

4. Collaborations' project plans should explicitly include an iterative approach to technology application, scaling up with each iteration, as cited in the GSFC and JPL collaborations' final reports.

5. To succeed, training and continued support are needed. For example, the Coverity SWAT tool lacked training, and minimal support was provided. The technology vendor did not visit the development team to train and consult on the tool's application. In contrast, USA received onsite training on applying PBI technology to its own application. This reduced risk and cost as well, since part of the target application was used in the training session. "The most successful way to do tech transfer is to put a member of the [technology vendor team] on the development team" – Matt Barry, JPL, (paraphrased) communication to the authors.

6. Overall, Research Infusion's first set of completed collaborations supports the hypothesis that with selection of appropriate technologies, matching of technology with software development team, and guidance and oversight, infusion of new software engineering technologies can be performed successfully on a minimal budget.

5   Acronyms

ACS: Attitude Control System
ARC: NASA Ames Research Center
CGS: C Global Surveyor, a static analysis tool for C software developed at NASA Ames.
FSB: Flight Software Branch (GSFC)
FSW: Flight Software
GSFC: NASA Goddard Space Flight Center
ISS: International Space Station
ITAR: International Traffic in Arms Regulations
IVVF: NASA Independent Verification and Validation Facility (West Virginia)
JPL: NASA Jet Propulsion Laboratory
JSC: NASA Johnson Space Center
MER: Mars Exploration Rover
MSFC: NASA Marshall Space Flight Center
ODC: Orthogonal Defect Classification
PBI: Perspective-Based Inspections
PF: Penetration Factor
PFR: Problem Failure Report

PRS: Problem Reporting System
SQA: Software Quality Assurance
SQI: Software Quality Initiative (JPL)
SWAT: SoftWare Analysis Toolset
SWG: Software Working Group
TRL: Technology Readiness Level
USA: United Space Alliance