The NASA Software Research Infusion Initiative:

Successful Technology Transfer for Software Assurance

Michael G. Hinchey
Software Engineering Laboratory
NASA Goddard Space Flight Center
Greenbelt, MD 20771, USA
1-301-286-9057
Michael.G.Hinchey@nasa.gov

Thomas Pressburger,
Lawrence Markosian
NASA Ames Research Center
Moffett Field, CA 94303, USA
1-650-604-4878
Tom.Pressburger@nasa.gov
zaven@email.arc.nasa.gov

Martin S. Feather
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
1-818-354-1194
Martin.S.Feather@jpl.nasa.gov

ABSTRACT

New processes, methods and tools are constantly appearing in the field of software engineering. Many of these augur great potential in improving software development processes, resulting in higher quality software with greater levels of assurance. However, there are a number of obstacles that impede their infusion into software development practices. These are the recurring obstacles common to many forms of research. Practitioners cannot readily identify the emerging techniques that may most benefit them, and cannot afford to risk time and effort in evaluating and experimenting with them while there is still uncertainty about whether they will have payoff in this particular context. Similarly, researchers cannot readily identify those practitioners whose problems would be amenable to their techniques and lack the feedback from practical applications necessary to help them to evolve their techniques to make them more likely to be successful. This paper describes an ongoing effort conducted by a software engineering research infusion team, and the NASA Research Infusion Initiative, established by NASA's Software Engineering Initiative, to overcome these obstacles.

Categories and Subject Descriptors

D.2.9 [**Management**]: Software Quality Assurance

General Terms

Management, Measurement, Documentation, Economics, Experimentation, Human Factors.

Keywords

Research Infusion, Technology Transfer

1. INTRODUCTION

Technology infusion—the maturation and transfer of research results into practical use—has long been a desirable, yet challenging, goal [1]. NASA, like many organizations, can benefit from successful technology infusion, but as with other organizations, technology infusion is often difficult. Shapiro [2] outlines some of the obstacles to technology infusion within the NASA context, and proposes some remedies, using microelectronics technologies as examples.

Software engineering is a technology area that is subject to these infusion obstacles. Zelkowitz [3] observed this a decade ago in the context of the NASA Software Engineering Laboratory.

Recognition of the growing prominence of software within the development and operation of NASA spacecraft has led to the establishment of the NASA Software Working Group, the purpose of which is:

> "...to develop and oversee the formulation and implementation of an Agency wide plan to work toward continuous, sustained software engineering process and produce improvements in NASA; and to ensure appropriate visibility of software issues within the Agency" [4].

One of the strategies of this group is to "Improve NASA's software engineering practices through research". A software research infusion team, involving representatives from various NASA centers, is charged with facilitating this. We will describe this team's approach and some experiences and lessons learned.

2. OBSTACLES

There are many obstacles to software engineering technology infusion within NASA, just as there are with many other organizations.

These include a significant gap in the perception of adequate maturity when viewed from a researcher's and practitioner's viewpoint; inadequacy of the NASA Technology Readiness Level (TRL) scale for quantifying the size of this gap; the risk-averse nature of most NASA software developers; and the differing motivation structures for researchers and developers. Rarely are there return-on-investment (ROI) models, competitive analyses, or other sufficient evidence to

demonstrate a research product's value in specific development environments.

There are many software engineering research products and it is difficult for practitioners to identify, evaluate, and track those that may be appropriate for them. The practitioner community is also somewhat fragmented, with many contractors—who develop the majority of NASA-funded software—unaware of previous NASA-funded software engineering research.

Finally, software development for NASA missions takes place in the larger context of project management of the entire mission, wherein there is reluctance to commit scarce resources to try out technologies that haven't been thoroughly proven, and even more reluctance to placing them on any critical path.

3. OVERCOMING OBSTACLES

Our team's approach to overcoming these obstacles includes:

1. *Information Gathering*: We identify and assess software engineering research that is of relevance to NASA's software development activities. Included in this is research performed both within and outside of NASA, commercially available products and tools, and tools and approaches developed by research groups (both academic and industrial). Some of this research has previously been funded by NASA, but much of it has not.
2. *Information Dissemination*: We identify the channels to reach the NASA software practitioners who might benefit from the research techniques. We use these channels to publicize the research techniques among NASA and its contractors' software development teams.
3. *Brokering Collaborations*: We identify and encourage promising collaborations between researchers and NASA software engineering practitioners. This is helped by the availability of funds specifically devoted to support such collaborations. Our infusion team helps recommend the allocation of this funding to worthy collaborations.

3.1 Information Gathering

Our information gathering efforts aim to identify software engineering research being undertaken that is relevant to NASA's software development activities. Since our effort was chartered in 2002, we have considered research performed by NASA researchers, by researchers (in academia and industry) funded by NASA, research from outside NASA, and commercial products.

Our team consists of members of the software engineering research community from several of the NASA centers, and JPL. Their experience and activity within the software engineering community gives the team a broad awareness of ongoing developments in that arena.

We have narrowed our focus to software engineering research results that:

(1) Have particular relevance to software assurance (partly influenced by the fact that funding comes from the Software Assurance Research Program).

(2) Can be incorporated into existing software development practices with a minimum of disruption.

(3) Are mid- to high-TRL (Technology Readiness Level) research, demonstrating success on a real project, and ready for use more or less "as-is". It is important to note that our activity is truly dealing with *infusion*, and that funds are not made available (from this initiative) for further development or "polishing" of the research product.

(4) Are either NASA-funded, or are related technologies, or have been suggested by NASA software developers. Future infusion is likely to concentrate more on technologies that have been funded by NASA, in order both to validate this research and to ensure transfer of the technology into NASA.

3.2 Information Dissemination

Information on the research techniques that we have identified is posted at the research infusion web site [5]. These are publicly accessible web pages, and so may be located by practitioners within NASA and its contractors by search, or by following links to these pages from various other NASA web pages (for example, the NASA Software Working Group's pages).

Our team members have contacts with NASA software practitioners at their respective centers and with contractors as well. Presumably other NASA software engineering researchers have similar contacts with software practitioners, and might be expected to pursue these to and make connections between practitioners and other research of which they are aware. Our infusion team, through its involvement in gathering information on suitable techniques, has at its fingertips deeper and broader knowledge of those techniques, and so is better able to recognize potential connections.

In addition, specific site visits have been conducted to NASA Centers and contractors to enable a familiarization with some of the issues facing software developers.

3.3 Brokering Collaborations

The research infusion team conducts an annual call for research collaboration proposals. Proposals for such funding must be submitted by a software practitioner (not the technology provider), and must be for application of the technique to actual project use (not for further research or enhancement of the technology).

Unlike other research programs, the Research Infusion Initiative optimizes the likelihood of a successful collaboration by communicating with each proposal team (wherever possible) prior to the proposal due date to ensure, initially, that there is a good match of technique and requirements, that the proposed collaboration is well-designed, and finally that the nominal outcome of the project will be a success by our standards.

Funding provided for each collaboration is in the $20,000–$50,000 range over a 6-month period. Funds are intended to be used for risk-reduction and mitigation in introducing the

technology—for example, for training, customer support, limited licenses where required, and collaboration management, data collection and analysis. Despite the low level of funding in comparison to typical NASA project budgets, we have seen an increase in the number of proposals over the three-year history of the initiative.

The Research Infusion team established the following evaluation criteria for submitted proposals. Software developers complete a proposal template which includes sections crafted to gather information on each of these criteria:

a) Feasibility: Is the proposed collaboration feasible? Are the skills of the participants relevant? Is the funding adequate? Is the management plan sound?

b) Impact on NASA: What will be the impact on NASA? Is the technique being applied to an important project?

c) Likelihood that, if successful, the technique will be adopted as part of the development team's practice: What is the likelihood that the technique, if successful in the proposed collaboration, will be adopted as part of the development team's practice? Will that usage become commonplace, or only in particular circumstances (e.g., when problems arise).

d) Adequate feedback provided to researchers: Is adequate feedback provided to the researchers during the collaboration? For example, are uncovered bugs, metrics, data, and a final report being provided to the research team.

e) Good use of NASA funds: Is the proposed collaboration a good use of NASA funds? The proposal's budget section addresses this question directly by stating how the funds will be used. We also ask that the proposer indicate what the impact will be on the development project if the proposal is *not* implemented.

4. COLLABORATIONS

Our effort was chartered in 2002. We held NASA-wide videoconferences in August of 2003, May of 2004 and March 2005. At each of these we featured seven or more promising assurance techniques (in the second and third events, repeating some of the ones from previous years as well as new ones), and announced a "call for collaboration proposals".

Following the selection process this lead to funding for a selection of Research Infusion collaborations. Ten such collaborations were initiated during 2004 and 2005. The technologies included a technique for conducting more efficient formal inspections; software defect classification for process improvement; requirements analyzers; code analyzers; and tools and a method for design rationale capture. The complete list is shown in Table 1.

The target application projects included spacecraft flight software, a ground antenna controller, International Space Station payloads, Space Shuttle and Space Shuttle Main Engine software, and a mission design activity. Table 1 also illustrates the outcomes of each of these collaborations.

An additional six collaborations have been approved for 2006, as shown in Table 2. Of these, one will feature the use of a commercially available tool that has been used in a prior infusion.

Table 1. Existing Research Infusion Collaborations.

Technology	Technology Provider	Technology Description	Customer Sites and Target Applications	Outcome/Benefits
Perspective-based Inspections	Fraunhofer Maryland, SARP	Software Manual Inspection Technique	**GSFC** (Spacecraft FSW) **USA** (ISS power analyzer)	Defects found in legacy code and that escaped previous inspections. Adopted.
Software Cost Reduction (SCR)	Naval Research Laboratory	Requirements Analysis Tools	**ARC** (ISS Payload)	Personnel trained. Reqts validated.
SpecTRM	Safeware Engineering Corp. & MIT	Requirements Capture and Analysis	**JPL** (Capture of Mission Design Rationale)	Personnel trained. MIT student hired.
Orthogonal-Defect Classification	JPL, SARP	Process Improvement Methodology	**JPL** (Ground SW)	SQA and project personnel trained.
CodeSurfer/CodeSonar	Grammatech, Inc.	Reverse Engineering/defect detection	**JSC** (ISS, Shuttle), **IVVF** (Spacecraft FSW)	Found defects that escaped previous inspections.
C Global Surveyor (CGS)	ARC – Intelligent Systems Program	Software defect detection tool	**ARC** (ISS science payload) **MSFC** (ISS payload)	Found defects. Good feedback to provider.
Coverity SWAT/Prevent	Coverity, Inc.	Software defect detection tool	**MSFC** (ISS, Shuttle FSW)	Found defects that escaped testing. Will be adopted.

Table 2. Approved Future Research Infusion Collaborations.

Technology	Technology Provider	Technology Description	Customer Sites and Application
Design Advisor	Siemens Corporate Research	UML style checker	GSFC (Spacecraft Science Instrument Module)
Software Architecture Evaluation	Fraunhofer Maryland	Code/Architecture Consistency Analysis	JHU/APL (Ground SW)
Klocwork Inspect	Klocwork, Inc.	Software defect detection tool	JPL (Ground SW), GSFC (FSW)
CodeSurfer	Grammatech, Inc.	Reverse Engineering/defect detection	KSC (Shuttle Processing CY06) GRC (ASMS)
CASRE	JPL	Software reliability estimation	JPL (MONTE)
RTLinux	FSMLabs	Real time operating system	LaRC (RSC)

Prior collaborations have been evaluated according to the following criteria:

a) The previously stated success criteria of the collaboration projects funded under this proposal must be met. This includes a positive rating for each product on the collaboration's evaluation criteria metric(s).

b) The research product is adopted by the collaborating software development team for current (routine) use.

c) The research product is included in a list of recommended development practices at a NASA Center or by contractor.

d) The software development team using the product provides feedback, including performance data, to the research team to guide future development of the product.

e) Six months after the funded collaboration period, the research product is still being used by the development project or by a successor development project.

f) Independent of the success of the collaborations, "lessons learned" regarding the challenges and success factors for software development technology infusion within NASA are provided.

To date, six collaborations have fully completed and have submitted their final reports. All of them have achieved a "penetration factor" of 9 (as measured on the NASA Software Assurance Research Program's scale of 1 – 9), meaning that the results of applying the technology were actually used on the project. In the historical context, this level of penetration of new software engineering technologies is rare.

One collaboration resulted in success criterion (e) – technology is still in use 6 months after the end of the collaboration – and (c) – the technology is in the center's list of recommended development practices; two other collaborations are planning to adopt (and so would lead to (e)); and yet two more are investigating adoption in their context.

It is too early to tell if the collaborations which are still running or have completed less than six months ago, will achieve criterion (e).

5. LESSONS LEARNED

We believe that our initiative has been very successful, based on the above results.

Clearly we are getting results where technologies are being taken on board and will be used in future best practice within the relevant NASA organizations and by the relevant development teams. This is notwithstanding a very small budget and limited timeframe for the infusion (6 months).

Much of the success of the approach is based on prior planning. The approach (originally suggested by Pat Schuler) involves receiving competitive proposals (typically, and unfortunately, two to three times more proposals than we can fund) written by development teams rather than technology providers. The team communicates both with potential proposers and the relevant technology providers in advance of the proposal deadlines, to ensure a good match of technologies and to determine goals for successful infusion *a priori*.

In the past, we have identified the technologies we are seeking proposals to work with. This has been in part to focus the emphasis on software assurance (part of our remit), and also to ensure the feasibility and relevance of using the particular research results. We have however brokered specific collaborations where a development team was anxious to have an opportunity to try to apply a particular technology to its practices, and we anticipate doing more of this in the future.

Our technology selection criteria have remained largely unaltered through several years of scrutiny and application. However, several modifications are recommended for the future based on experiences over these years.

A greater emphasis should be placed on the criterion "How easily can the research product(s) be integrated into a software development project?" While this is stated as a constraint on the technology, it is a relation between the technology and the development environment, and it requires more careful

evaluation by the collaboration team prior to proposal submission. For example, several collaborations have had unexpected difficulty due to incompatibilities in the compiler (or other development tool) used on the project and the requirements of the technology. This can be a more serious issue at NASA than elsewhere because of the very conservative nature of NASA software development, supporting long-obsolete development platforms, in contrast to the most current environments that are typically supported by new software engineering technologies.

Also, the evaluation criteria for collaboration proposals need to take into account contractual risks (this has not been made explicit to the collaboration teams to date). The question can be interpreted in "cost/benefit" terms—will so much time be spent on handling contractual issues that the collaboration is put at risk. Again, this is a particularly significant issue for NASA (and other governmental agency) projects where there can be a high administrative overhead (including long delays as well as personnel effort) in getting necessary approvals. These obstacles have the potential for derailing projects with low funding and short duration.

In particular, with such low levels of funding, it is difficult to keep the interest levels of collaborators high should there be any significant delays. This is true both for the technology provider and the development team. For this reason, the period for submission and selection of proposals needs to be kept relatively short, and work on putting contracts in place must begin quickly after the proposals are selected. For this reason, our call for infusion proposals in 2006 will be later than in prior years.

A significant issue is that, if there are delays in start dates, the needed personnel may be lost due to prior commitments elsewhere.

Another risk that should be recognized and mitigated results from the classification of the collaboration's target software. Software that is classified as export-controlled may limit collaboration participation by technology developers. Unfortunately, the most safety- and mission-critical code is often classified as ITAR (International Traffic in Arms Restricted) at NASA.

6. DISCUSSION AND CONCLUSIONS

The overall impact and benefits of research infusion to space systems are several: previously-inaccessible software assurance technologies have been successfully infused; some have been adopted for inclusion in an organization's development practice; several have continued to be used for some time following the end of the collaboration; the software development team has provided feedback to the technology developers; and lessons learned have been identified regarding the challenges and success factors for software development within NASA [6].

The NASA Research Infusion Initiative has demonstrated an inexpensive and effective process for brokering matches between software engineering researchers and NASA software development practitioners that can be incorporated into NASA's overall strategies for infusion of software engineering research products, and specifically for research products that can improve software safety and mission assurance.

As our procedures are codified and the research infusion team has gained experience, our approach is likely to scale to a greater range of software engineering technologies (not just those addressing software assurance and related issues) and to larger numbers of collaborations.

Expansion of scope to more "revolutionary" technologies—technologies requiring a more significant change to an existing software development process model, or to the required infrastructure—is likely to require adaptations in the Research Infusion business model.

7. ACKNOWLEDGMENTS

The research infusion team was lead by Tom Pressburger from 2002 to early 2006. Mike Hinchey now leads the team.

Team members included the authors, along with Ben Di Vito (NASA Langley), Luis Trevino (NASA Marshall), and Tim Menzies (West Virginia University); Wes Deadrick (NASA IV&V Facility) has been an advisor and reviewer of collaborations.

We are indebted to John Kelly, NASA Office of the Chief Engineer, who leads the NASA Software Working Group and provides support for the initiative. Martha Wetherholt in the NASA Office of Software and Mission Assurance also provides financial and other support for the initiative, administered in part by the NASA IV&V Facility under the direction of Marcus Fisher.

We would like to acknowledge the many researchers who have lent their support and the many software developers who have submitted collaboration proposals, and worked with us to make this initiative such a success.

This paper is substantially based on [6].

8. REFERENCES

[1] Rogers, E. *Diffusion of Innovation,* The Free Press, New York, 1983.

[2] Shapiro, A.A., Technology Infusion for Space-Flight Program, *Proc. 2004 IEEE Aerospace Conference*, Vol. 1, pp 662-667, 6-13 March 2004.

[3] Zelkowitz, M.V., Software engineering technology infusion within NASA, *IEEE Trans. On Engineering Management*, 43(3): 250-261, August 1996.

[4] NASA Software Working Group website, available at: http://software.nasa.gov/about/

[5] NASA Software Engineering Research Infusion website, available at: http://ti.arc.nasa.gov/researchinfusion

[6] Pressburger, T., Di Vito, B., Feather, M.S., Hinchey, M.G., Markosian, L. and Trevino, L. Infusing Software Assurance Research Techniques into Use. *Proc. 2006 IEEE Aerospace Conference,* Big Sky, Montana, 4-11 March 2006.