Building an Effective CBSE Handbook

Position Paper for ICSE99 Workshop

"Developing a Handbook for Component-Based Software Engineering"

ICSE99, Los Angeles, May 17th-18th, 1999

Martin L. Griss

Hewlett-Packard Laboratories, 1U16

1501 Page Mill Road, Palo Alto, CA 94304-1126

Tel: (650)857-8715, fax: (650)813-3668

Email: griss@hpl.hp.com

URL: http://www.hpl.hp.com/reuse

Abstract

The community has made tremendous progress in understanding the critical technology, methods, processes, and organizational factors that lead to effective architected, component reuse. There is of course much more we can do, but still, far too few software developing organizations see reuse or CBSE as the solution to the problems they have, nor do they practice component reuse systematically. Far too few schools teach much about reuse and components. A CBSE handbook, summarizing industry best practice and state-of-the-art, could be an extremely useful vehicle for moving us forward.

However, experience with such handbooks in the past suggests that we should think very carefully about who the audience(s) for the handbook are, how this handbook should be used, and how it should consequently be structured. I propose several additions and modifications to the strawman handbook outline, to clarify these issues, as well as incremental adoption, organizational and process maturity, and CBSE methods.

Keywords: software engineering, components, education, technology transfer, best-practice, architecture.

1 Background

For the last 16 years, I have worked on reuse methods, modeling, component software, software education and software engineering process improvement at HP[griss95]. I have also gained a lot of experience with software engineering, VB and ActiveX components and reuse courses as an adjunct professor at the University of Utah[kessler97]. I have recently worked on the joint ACM/IEEE software engineering education taskforce and helped develop accreditation guidelines[griss98], and have reviewed the outline and content of a "software engineering body of knowledge".

Over these years, I have had the opportunity to lead and/or work on three handbook projects, a book on architected reuse and component-based software reuse, and several related projects. These include:

1. A "systematic reuse handbook" workgroup at WISR-4 (Workshop on Institutionalizing Software Reuse) 1991, in which we created an outline and discussed a process for building the handbook, and transferring the knowledge. See wisr91-hand.txt for a summary of this workshop. Several alternate handbook structures were discussed, such as one based on reuse process framework, one based on independent modules, one based on incremental adoption, and one with different sections for different audiences.
2. A reuse handbook for use at HP, started in 1990 and evolved during 1991-1992, incorporating the WISR handbook outline, 1. above. In this project we envisioned a "loose leaf" customizable handbook, in the spirit of some time managers or process handbooks: the idea was that a reuse manager could customize a process handbook for his group, selecting only material germane to the current process, reuse, and architecture maturity of his organization. This handbook was used extensively inside HP in the period 1992 to 1995 to establish reuse pilots as part of a corporate reuse program. A training class to introduce the handbook was prototyped. The experience gained in this work, influence the content and structure of the book I subsequently wrote.
3. An architecture handbook workshop at OOPSLA 1994(?), led by Bruce Anderson. This included a structure for architectural styles, mechanisms, standard architectures and the beginning of design patterns.
4. A book on "Software Reuse: Architecture, Process and Organization for Business Success" with Ivar Jacobson, and Patrik Jonsson[jacobson97].. This book provides a holistic view of model-based component software development (using UML), covering a variety of organizational, management, technical, architectural, process and business issues, in a consistent framework.
5. A web-based reuse process guide, produced as part of a 1996 consulting engagement by the HP PSO (Professional Services Organization) OO and subsequently refined for HP use [griss1997].

My related work on the UML standardization effort, on reuse and process maturity models, on technology transfer for an architected, component based product line at HP, on model-driven management agents, and on software engineering standards, provided the many opportunities for thinking and writing about architecture, components, models, patterns, and knowledge transfer. I have produced many training slides, run workshops, produced many small briefing documents and templates.

2 Position

I learned a number of things from the various handbook experiences above, which could be valuable in the proposed CBSE handbook effort, and the workshop itself.

Content is of course really important. We have lots of promising technology, methods, and guidelines that can be adapted from the reuse, architecture, OO and patterns community. But too little is taught systematically [bucci98], or practiced widely. As an example, we believe that we have a good understanding of how issues and choices in several areas could influence critical success or failure of architected, component reuse -- CBSE:

- Technology - OO; architecture; patterns; components; interfaces; generators; library systems and classification schemes; UML, ….
- Process - domain analysis, CFRP, DFR-DWR, incremental pilots, process, product and reuse metrics; process maturity models; economics, Catalysis, RSEB, …
- People: distinct create, reuse, manage, support organizations; architecture teams, explicit high-level management leadership; domain- and component- engineering skills; roles; …

As an expert in many of these areas, I can contribute to content in many sections of the strawman handbook, such as 1b -- "how does CBSE relate to similar fields ", 2a - "processes and methodologies", 2b - "organizational issues", and to many of the sub-sections of 3, such as "UML", "libraries", tool, etc.

A key learning from the HP handbook project, was the tension between size and usability of the handbook (partially addressed by a loose leaf approach). Another was the need to provide customized material to different groups. Another was the difficulty of keeping such a handbook up-to-date. And finally, the difficulty in getting groups to use such handbooks.

Thus an important concern for the handbook effort proposed is to discuss the goals and audiences(s)of the handbook and how we expect it to be used. This will certainly have an impact on the structure of the outline and mechanisms used for customization and adoption, if any.

Some other observations and suggested changes to the strawman outline:

- The preamble specifically mentions *engineering handbook.* Is the intent to exclude, or segregate management material, perhaps to later produce a management handbook for CBSE? Certainly material on adoption, organization, economics have a large management as well as process content. One idea would be to have a separate management section along with the engineering sections, or for each section to have the managerial implications as well as a technical implications.
- I would separate recommended processes and methods from the adoption section, 2. That is, I would collect managerial and adoption issues in one section, adding material on CBSE and process maturity models, the role of pilots etc.. I would then have a separate section, covering recommended CBSE processes and methods. For example, we should mention domain engineering (e.g., FODA, ODM, Synthesis), Catalysis, Rational Unified process, RSEB, etc., or least integrate these into a recommended CBSE processes for architecture, component design and component use. The CFRP (common framework for reuse process) could provide a useful source of process material. Alternatively, the methods could be considered part of technology or although that is not the current scope of technology in section 3 -- certainly methods and process are more than just notation and modeling languages.
- Section 4, could be more usefully structured to distinguish "best practice", "state-of-the-art", "advanced", "domain-specific" and "research" areas, much in the spirit of the Software Engineering Body of Knowledge [IEEE98].
- It is important to add a reference material, bibliography and some templates and tables.

Key outline elements of the HP reuse handbook (developed before the growing use of modeling, UML) are included in the appendix. Book also provides an interesting overall outline, separating architecture, process and management issues[jacobson97]. The knowledge captured in the CBSE handbook, as a component community best-practice consensus could be injected into the evolving software body of knowledge (SWEBOK)[IEEE98].

3 Comparison

There are now many books on reuse, components and architecture, and a growing number of reuse courses, both at schools and from vendors. However, apart from the DOD Reuse technology roadmap [DOD96], and the SPC adoption guidebooks, there does not seem to be a community consensus on CBSE best practice, or how best to drive adoption. No one seems to be addressing this in the context of the new licensing and accreditation context.

References

[bucci98] Bucci, Paolo, and Timothy J. Long and Bruce W. Weide, "Teaching Software Architecture Principles in CS1/CS2", Position Paper, Third International Software Architecture Workshop, Orlando, Nov 1-2, 1998, pp. 9-12. (See http://www.cis.ohio-state.edu/).

[DOD96] Reifer, Don, *"Reuse Technology Roadmap"*, Department of Defense, 1996.

[griss95] Griss, Martin and Wosser, Marty, "Making reuse work at Hewlett-Packard." *IEEE Software*, January 1995, V12, #1, pp. 105-107. .

[griss97] Griss, Martin and Balasubramanian, Ramesh, "Reuse Process Guide", *HP PSO/HPL working document*, Nov 1997.

[griss98] Griss, Martin, "Letter from the Executive Committee," *Software Engineering Notes*, Vol. 23, No. 5, Sept. 1998, pp. 1-2.

[IEEE98] Bourque, Pierre at. al., "Guide to the Software Engineering Body of Knowledge (SWEBOK)", Strawman Version, IEEE Computer Society, September 1998. (See http://www.ieee.org/).

[jacobson97] Jacobson, Ivar, and Martin Griss and Patrik Jonsson , *"Software Reuse: Architecture, Process and Organization for Business Reuse,"* Addison-Wesley-Longman, 1997.

[kessler97] Kessler, Robert R.,"CS451-CS453 - Software Engineering Laboratory", Computer Science Department, University of Utah, Salt Lake City, UT, 1997. See http://www.cs.utah.edu/~cs451.

[tracz98] Tracz, Will, and Mary Shaw, and Martin Griss, and Don Gotterbarn, "Panel: Views on the State of Texas Licensure of Software Engineers", *Proceedings of FSE-6: ACM SIGSOFT 6th International Symposium on the Foundations of Software Engineering*, Nov 3-5, Orlando, ACM SIGSOFT, 1998, pp. 203-208.

Biography

Martin L. Griss (griss@hpl.hp.com) HP Laboratories, *http://www.hpl.hp.com/reuse.*

Martin is a Principal Laboratory Scientist in the Software Technology Laboratory at Hewlett-Packard Laboratories, Palo Alto, California. He has over 30 years of experience in software development, education and research. For the last 16 years at HP, he has researched software engineering processes and systems, systematic software reuse, object-oriented development and software process improvement at HP. He created and led the first HP corporate reuse program and participated in the development and execution of the HP corporate software initiative. He led HP efforts to standardize UML for the OMG, and is a member of the OMG UML revision taskforce. He was previously director of the Software Technology Laboratory at HP Laboratories, and an Associate Professor of Computer Science at the University of Utah.

Martin is co-author of the influential book "Software Reuse: Architecture, Process and Organization for Business Success" (with Ivar Jacobson and Patrik Jonsson), which holistically addresses technology, people and process issues in a UML framework. He writes numerous articles on software engineering and a reuse column for the "Object Magazine/Component Strategies". He lectures widely on systematic reuse and software process improvement. He is a member of the ACM SIGSOFT Executive Committee, and of the joint IEEE/ACM committee on software engineering education and accreditation. He is an adjunct professor at the University of Utah, co-developing component-oriented software engineering courses, and advising on software engineering curriculum design.

Appendix: Extract from HP reuse handbook, circa 1994.

- **Introduction**
    - *Audience and purpose*
    - *Philosophy and assumptions: definitions, guidelines vs processes, …*
    - *How to use this handbook*
- **Preparing for a reuse program**
    - *Starting a reuse project*
    - *Work product inventory*
    - *Factors influencing effective reuse*
    - *Maturity models*
    - *Effective adoption*
- **The reuse process**
    - *Managing the reuse process; planning, organization , economics, metrics, …*
    - *Producing reusable work products: architecture, domain engineering, coding guidelines,…*
    - *Using reusable work products: finding, adapting requirements to match assets, adapting assets to meet requirements*
    - *Supporting reusable work products:: testing, certification, documentation, library issues*
- **Reference documents**
    - *Working references*
    - *Glossary*
    - *Available resources*
    - *Online templates*
        - *Assessment*
        - *Metrics*
        - *Inventory*
        - *Costing*
        - *Architecture discovery*
        - *Best practice capture*
        - *Domain dictionary*
- **Indices**

- *Topics*
- *Concepts*