JOIN OVER 100,000 PROS! Join Log In | Member Benefits

STICKYMINDS

Agile Testing DevOps Security AI

Agile SCM 2005 - Reflecting back on the year in books

ARTICLE

By Steve Berczuk , Brad Appleton Robert Cowham | June 16, 2006

SUMMARY

We thought 2005 was a pretty gosh darn great year for Agile Software Development and Software Configuration Management alike. We wanted to share what we feel are some of the timeless classics that we have most looked to throughout the year, as well as the new books in the last year that we have been most impressed with.

Hello! My name is Brad Appleton, and I'm a book-a-holic! Hear my serenity prayer:

Lord, please grant me ... the serenity to accept that I can't read *everything*, the time to read and understand everything that I can, the wisdom to know the difference *[so I won't have to leave my estate to Amazon.com]*, and a sufficiently well-read network of friends *[to tell me all about the books they've read]*.

We thought 2005 was a pretty gosh darn great year for Agile and Software CM alike. We wanted to share what we feel are some of the timeless classics that we have most looked to throughout the year, as well as the new books in the last year that we have been most impressed with.

Let's start off with "the classics"!

We're not pretending these are the "only" classics, or even the "best" of the classics. These are just the ones from our personal list that we have found ourselves referring to most during 2005. A few of them might be surprising. In a later section we'll tackle some books that aren't really classics just yet, but we hope they will be someday soon.

Software Fundamentals: Collected Papers of David Parnas

Parnas' papers are most definitely fundamental classics in the field software

We use cookies to ensure you get the best experience on our website.

Accept

Read More

contraction apply every bit as much to configuration management as they do to software design.

I'm proud to have played a part in the creation of this book. I had the initial idea for it, mostly because I wanted it to exist and when I found out it didn't, I felt very strongly that it ought to. When asked if I wanted to be the one to collect the papers together into a book, I said (in my best *Wayne and Garth* voice) "I'm not worthy", but I knew David Weiss (through Jim Coplien) and was able to get him interested enough in the project to turn it into reality!

Are Your Lights On? and **Becoming a Technical Leader**, by Weinberg and Gause

Gerald Weinberg has authored and co-authored **so many classic books**, it's extremely difficult to choose just one or two. Steve likes these two because they are so timelessly helpful in problem-solving, and in working with others to lead problem-solving efforts.

Are Your Lights On?: How to Figure Out What the Problem Really Is is a bit of an easy-to-read "primer" on **General Systems Thinking**. One problem people encounter when they use SCM tools and techniques is that they do "too much," and their system ends up hindering rather than helping them. This book is not about SCM but rather how to figure out what is really wrong in a system. It is an entertaining read that teaches an important and often undervalued skill. This books is about figuring out what the problem really is before going off and "solving" it. A must for agile teams.

Becoming a Technical Leader: An Organic Problem-Solving Approach is a must-read not just for all would-be technical leaders, but for anyone trying to champion change in their workplace processes (be it CM-related or Agile-related). It's a great book at helping us understand ourselves and how we can help others understand and express their voice so that we can all put our heads together, collaboratively. What else is there to say - the title says it all!

Code Complete, by Steve McConnell

Steve McConnell's 2nd edition **Code Complete** just came out this year and has been comprehensively updated and expanded. This book is still a must read for any practicing software professional and covers far more than just development. It doesn't claim to be the first and last word on the subject, but it's definitely a solid "first word" with comprehensive resources and reading lists on where to find the rest.

DaMarco & Lister's **PeopleWare**

DeMarco and Lister's **Peopleware** is another well-known classic that was perhaps one of the first well known books to popularize the importance of the people-factor in software development, which Agile methods brought back into the limelight when so many who were focusing on the Software CMM were appearing to omit that important part of the equation (even though the CMM fully acknowledged it, and even devised a People CMM to underscore it). Some of the seminal research on the importance of flow and context-switch interruptions and workspace layout and office design is in this book.

DeMarco and Lister's most recent work together is on the critical subjecct of software

We use cookies to ensure you get the best experience on our website.

past burnout, busywork, and the myth of total efficiency are likely to become classics in the not-so-distant future.

Software CM Classics

We wish there were more to say here. The "classic" books on Software CM are the ones by **Whitgift**, **Babich**, Berlack, and **Bersoff**. Despite their technology-independent content being "timeless", the books are nonetheless out-of-print or hard to find, or very much "dated" by the amount of references to old or outdated SCM technology.

What should the "new" SCM classics be? Great question! There were a dearth of good books on the subject for awhile, and then sometime around 2000 or 2001, the subject started becoming popular again and several good titles came out. Unfortunately, some of those too were either tool-specific and/or now outdated.

If I had my pick, my first choice would be Susan Dart's **Configuration Management: The Missing Link in Web Engineering**. I'm aware of Alexis Leon's comprehensive SCM Guidebook (now in it's second edition) but Susan's work is among my favorite in the field and this book really combines together most of her best writings on the topic.

My second pick would probably be Mario Moreira's **Software Configuration Management Implementation Roadmap** followed closely by Anne Hass' **Configuration Management: Principles and Practice**

Michael Bay's **Software Release Methodology** warrants an honorable mention here too.

Robert is allowed to mention **Software Configuration Management Patterns: Effective Teamwork, Practical Integration** by Steve Berczuk with Brad Appleton! This book would not have been written without encouragement from John Vlissides, one of the authors of **Design Patterns**, the seminal work on the subject of patterns if not on all of O-O software design, and one of the best selling computer-science books of all time. Unfortunately John **died recently** after a long bout with cancer -- there is a **Wiki page to memorialize John** showing some of his contributions and the lasting impressions he created.

SCM Books in 2005

The jury is still out on what will be the "new" classics of SCM though, particularly since 2005 was a banner year for Software CM books, including all of the following:

- the 2nd edition of Brian White's outstanding **Software Configuration Management Strategies and Rational ClearCase**, updated by David Bellagio and Tom Milligan
- the 2nd edition of Alexis Leon's **Software Configuration Management Handbook**
- Pulsipher and Buckley's **The Art of Clearcase Deployment**
- **Practical Perforce** by Laura Wingerd
- **Pragmatic Version Control w/Subversion**, by **Mike Mason**
- **Subversion Version Control**, by Bill Nagel

We use cookies to ensure you get the best experience on our website.

and/or Version Control, particularly the titles from **Laura Wingerd**, **Brian White**, and Vincent Maraia (though I have several quibbles with numerous statements in "The Build Master").

Agile Management and IT Career Development

There have been a number of books in the last year or two that seem less about CM or agile *development* and more about CM toolsmithing and administration, or agile *management*. While they might not be what you would expect to see on a list of "Agile CM" readings, I think they are nevertheless worth your attention!

Agile Estimating and Planning and **Managing Agile Projects**

Mike Cohn's brand new book **Agile Estimating and Planning** looks pretty incredible. Whether or not your project is Agile, this seems to be the nuts and bolts of real-world estimating and planning. This is not what formal project management textbooks say to do - it is what you end up having to do anyway in the real-world, at least if you want to succeed.

In the same vein, Sanjiv Augustine's **Managing Agile Projects** gets at the heart of the social and political dynamics of real-world project management, and managing interactions and relationships with your stakeholders and project team-members (rather than burying your head in the GUI of the project management tool).

Behind Closed Doors and **Ship It!**

Esther Derby and **Johanna Rothman** have collaborated together on an indispensable "tour de force" in **Behind Closed Doors: Secrets of Great Management**, from **The Pragmatic bookshelf**. Every budding and first-time software manager should grab this and read it.

Also from **The Pragmatic bookshelf** is Jared Richardson and Will Gwaltney's **Ship It! A Practical Guide to Successful Software Projects**, which is exactly what the title suggests it is.

Both these books are just published in 2005 and both look to be winners by giving insider knowledge of what happens in the real-world and what's necessary, successful, and ethical, in order to deal with it.

Scripting, Automation, and Staying Employed

Just about anyone involved in Software CM has had to do some scripting here and there, and Perl is the language predominantly used to script and automate many such tasks. No Perl-scripting SCM automaton should be without Peter Scott's 2004 book **Perl Medic: Transforming Legacy Code**.

For those first starting out as toolsmiths and automators, Mike Clark's 2004 Jolt award winning **Pragmatic Project Automation** is another must-have.

And for those who are primarily doing IT and scripting and only dabbling into the SCM process area out of necessity (because no one else is there to do it), keep learning about

We use cookies to ensure you get the best experience on our website.

important as a wake-up call for not only keeping skills current, but keeping one's career "marketable" and making oneself indispensable.

Would-be and Lesser Known Classics

Agile Software Development: Principles, Patterns and Practices by Robert C. Martin

This book won a much deserved 2002 Jolt Award. For me, few other books successfully combine and convey the essence of software design, design patterns, refactoring and agility in a single tome. "Uncle Bob" successfully inter-relates principles of object-oriented design to design patterns and refactoring and agile methods in what is likely to become a classic. I find myself referring to this book at least weekly if not more often. The principles of object-oriented design relate not only to agility and software architecture, but also to good **SCM solution architecture** and sound **principles of SCM**.

My "runner's up" and "honorable mentions" for this one are:

-
    - **The Pragmatic Programmer**, by Dave Thomas and Andy Hunt
    - Martin Fowler's **Refactoring**
    - Josh Kerievsky's **Refactoring to Patterns** (2005 Jolt award winner)
    - Michael Feather's *Working Effectively with Legacy Code*
    - **Software Factories** (2005 Jolt award winner)

This book isn't truly a classic (yet). I feel that it should be, and hope that it will be some day soon. I think it is a profoundly important book for all involved in software development/engineering to understand. If more businesses understood Armour's perspective about why "Software is not a product", but rather a medium for the capture of knowledge (and hence software development is a knowledge creation activity rather than a production manufacturing activity), they would find so much more in alignment between software products and services and Peter Drucker's expansions on Deming's theory of profound knowledge that it might well revolutionize the way they choose to develop software.

Contradictory as it might seem, Armour's writing and "Laws of Software Process" are extremely well aligned with Agile methods and values. Anyone who considers themselves an Agile advocate would do well to read and understand this book.

My "Runner's up" for this one would have to be:

These may not be as well known as Kent Beck's seminal book on **Extreme Programming Explained**, but the above two books, along with Armour's are quite possibly the best out there for understanding much of the underpinnings of Agile methods. (Of course I'm leaving out Nonaka and Takeuchi's **The Knowledge Creating Company** and the existing literature on complex adaptive systems, but I can't fit everything in.)

We use cookies to ensure you get the best experience on our website.

Agile Management for Software Engineering by David Anderson (2005 Business Blogging Award winner)

The Ascent of Man, by **Jacob Bronowski**

This is a book about neither CM nor Agility. It's not even about software development. It is a book about both science and humanity. And it's probably the book that had the most profound impact upon me of all the books I read during my high school years. I find myself regularly referring to it more and more as Agile development method keeps injecting the human focus on people and interactions into the art and science of software development.

Bronowski's book has a chapter called "Knowledge or Certainty" that is about dealing with the inherent uncertainty in some things, and also in attempts at absolute knowledge/power attempted without grounding in real feedback, ending up doing things that dehumanize ourselves and others.

Also, in the first chapter is a quote that:

Man ascends through the discovery of the fullness of his own gifts. What he creates along the way are monuments to the stages of his understanding of nature and of self."

I think the same is true of how we develop software and the iterative discovery and learning along the way about the system's real requirements, and the nature of the software development process itself and what software architecture can and cannot do to sustain both over time.

***Robert reviews Laura Wingerd's* Practical Perforce**

Having just been published by O'Reilly in November 2005, Laura is the VP of Product Technology at Perforce and co-author with Chris Seiwald of one of the most quoted papers in SCM. She brings an engaging writing style and tone to the explanation of a variety of subjects ranging from the arcana of merging through to the care and feeding of codeline management.

The ideas presented are of course mainly directed at Perforce usage, but some of the principles apply to tools using similar "path space" branching mechanisms (what Perforce calls "inter-file branching") such as Subversion and Microsoft Team Foundation.

My main quibble with the book is the focus on the command line - understandable as it is much easier to present examples than via screen shots which are more difficult to maintain - which might put some people off, but this is *a "must have" for any site using Perforce.*

Robert reviews David Allen's Getting Things Done

This book (GTD) has created something of a cult following in technological circles. From its author's website:

Sophisticated without being confining, the subtle effectiveness of GTD lies in its radically

We use cookies to ensure you get the best experience on our website.

at any moment. GTD embodies an easy, step-by-step and highly efficient method for achieving this relaxed, productive state. [...]

Implementing GTD alleviates the feeling of overwhelm, instills confidence, and releases a flood of creative energy. It provides structure without constraint, managing details with maximum flexibility. The system rigorously adheres to the core principles of productivity, while allowing tremendous freedom in the "how." The only "right" way to do GTD is getting meaningful things done with truly the least amount of invested attention and energy.

I have had some good experience personally with using this method to get to grips with my inbox in particular - reducing hundreds of messages to only a few on a regular basis together with generating appropriate lists of tasks to complete!

The principles of tracking and managing items and co-ordinating them into "done" have direct relevance to SCM. The feeling of being more in control is also something that I feel with a well implemented SCM process - it reduces worry and frees energy for being productive with a relative lack of risk.

There are a flood of web sites and resources discussing GTD but the book is a relatively easy read and very quickly gives you the idea. One of the big attractions is how easy it is to implement using a wide variety of different technologies ranging from pencil and paper (**Hipster PDA**) to an **Outlook plugin**.

Happy Holidays and Hopeful New Years

That's our Agile SCM Librarian's retrospective of 2005. We hope you all have a **Very Happy Merry ChristmaHannaValiRamaKwanzaakah** (or non-denominational solstice celebration) in 2005 and look forward to what 2006 will bring to all of us in the coming year!

Topics: agile, configuration management,

We use cookies to ensure you get the best experience on our website.

ABOUT THE AUTHOR

Steve Berczuk

Steve Berczuk is a Principal Software Engineer with experience as a manager, Scrum Master and technical lead in Boston, MA. The author of *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*, he is a recognized expert in software configuration management and agile software development. Steve is passionate about helping teams work effectively to produce quality software. He has an M.S. in operations research from Stanford University and an S.B. in Electrical Engineering from MIT, and is a Certified ScrumMaster. Contact Steve at steve@berczuk.com or visit berczuk.com and follow his blog at blog.berczuk.com.

Brad Appleton

Brad Appleton is an Enterprise Agile+DevOps leader, coach & manager, and seasoned DevOps/ALM/CM solution architect at a large Fortune 100 company. Currently he helps organizations and teams scale, adopt and apply lean/agile development methods and DevOps/ALM/CM practices and tools. He is co-author of *Software Configuration Management Patterns*, the Streamed-Lines branching patterns, and a former section editor for *The C++ Report*. You can read Brad's blog at blog.bradapp.net.

Robert Cowham

Robert Cowham has long been interested in software configuration management while retaining the attitude of a generalist with experience and skills in many aspects of software development. A regular presenter at conferences, he authored the Agile SCM column within the CM Journal together with Brad Appleton and Steve Berczuk. His day job is as Services Director for Square Mile Systems whose main focus is on skills and techniques for infrastructure configuration management and DCIM (Data Center Infrastructure Management) - applying configuration management principles to hardware documentation and implementation as well as mapping ITIL services to the underlying layers.

UPCOMING WEBINAR

We use cookies to ensure you get the best experience on our website.

LETS HANG!

FEATURED RESOURCES

AGILE
Implementing Agile eGuide
White Paper | StickyMinds

AGILE
Leadership for Today's Agile Team eGuide
White Paper | StickyMinds

AGILE
Scrum Master Essentials eGuide
White Paper | StickyMinds

Log In or Register to post comments

USER COMMENTS

0 comments

We use cookies to ensure you get the best experience on our website.

YOU MAY ALSO LIKE

AGILE

From Silence to Self-Organization: What Social Interaction Games Achieve That AI Cannot

Dinesh Sharma | June 17, 2026

AGILE

Why Your Agile Team Is Failing Even When Everyone Knows Their Stuff

Ella Mitkin | March 2, 2026

AGILE

Solving the Problem vs Treating the Symptoms

Payson Hall | January 16, 2026

Join the crew for FREE! Access exclusive content, contribute, and more.

See All Member Benefits

We use cookies to ensure you get the best experience on our website.