Jan Steinman

Barbara Yates

“Special” team members

LET’S FACE IT—WE’RE ALL “DIFFICULT” at some time or other. This can range from mild irritability over a bad hair day, to active sabotage between competing groups. These problems are best dealt with before they develop into a pattern of behavior, but the pace of Smalltalk development often results in people settling into behavioral patterns before anyone notices.

We divide special team members into three categories:

1. “Pluses” offer net productivity but can be much more productive if their unique strengths can be exploited while reducing the impact of their weaknesses.
2. “Zeros” are a wash and can be tolerated while your organization finds a place where they can become pluses.
3. “Minuses” detract from the productivity of others and can seriously impact a project if not dealt with in some way.

Keep in mind that we are writing about established behavior *patterns* here. Obviously, new assignments, emotional problems, family crises, etc., make the best of us “zeros” or even “minuses” from time to time, and a compassionate organization will help, or at least tolerate, these non-chronic productivity losses.

Here are some of the more prevalent behavior patterns we’ve found in Smalltalk projects and suggestions for dealing with them.

“THE LONER”

This person is an enigma to management. the Loner is often a Meyers—Briggs[1] “INTP” type, who may be perceived as “not a team player,” and might even be fired if he wasn’t so damned creative. Like Bramson’s[2] “Analyst,” the Loner will often miss deadlines, not because he isn’t working, but simply because he is *still* working!

The big danger on a Smalltalk project is that the Loner may disappear after having been given an assignment and come up for air several months later with a beautifully crafted, complete solution to the wrong problem. Because Smalltalk is so productive, it may be tempting to redefine the project around the Loner’s wonderful solution of the wrong problem, because he may well be far ahead of the rest of the team, who’ve been busy collaborating all these months!

At worst, a single Loner is a “zero,” but two or more on a team may quickly destroy a project if not guided by a skilled architect. Once you’ve discovered a Loner on your team, there are several techniques you can use to harvest his creativity without yielding control of the project:

- Schedule regular peer review, especially at the design level, before the Loner is able to write reams of code.
- If the Loner is also a Know-It-All (discussed later), call these peer reviews “educational reviews” to avoid wounding his fragile ego.
- Assign the Loner a “shadow,” “buddy,” or “stunt-double” —someone who keeps up-to-date on what the Loner is doing, in case it is necessary to fill in in an emergency and provide the communication that the Loner is unable to provide.
- Limit Loners to well-specified, well-defined tasks. This is a last resort, because junior people will not break their Loner habit and senior people will get bored and possibly become Slackers (discussed later).

“THE LOANER”

While discussing the Loner, we realized we have, on several occasions, experienced its pun! For various reasons, the project is running late and senior management decides they had better round up nine people so they can ship this baby in a month.

Almost 20 years ago, Frederick Brooks, Jr.[3] noticed that adding resources to a late project makes it even later. Loaners often consume more time than they add, and are, therefore, “minuses,” because they need to be integrated with the team’s procedures and conventions, and also absorb all the history that has gone into getting to this point of crisis. This is not so much a reflection on the

Jan Steinman and Barbara Yates are cofounders of Bytesmiths, a technical services company that has been helping companies adopt Smalltalk since 1987. Between them, they have over 20 years Smalltalk experience. They can be reached at Barbara.Bytesmiths@acm.org or Jan.Bytesmiths@acm.org, or via http://www.bytesmiths.com.

person as it is on the process that put them in this unfortunate situation.

This problem is amplified by Smalltalk because Smalltalk is not a language, it is an environment. Not only must this person learn your project before they can be useful, they often must learn exotic (to them) concepts, such as Dictionaries—concepts for which your team has developed a shared vision.

Even more insidious is the possibility that you might not be getting quality material to begin with. Think of it—if asked to loan one of your team members, would you give your best person, or perhaps someone whom you haven't quite been able to find a place for yet?

If your manager insists on doing you this "favor," keep the following in mind:

- Make sure they are doing you a favor—don't accept someone else's problem when you're already in schedule trouble!
- Keep loan lengths on the long side to better amortize the "interest" cost of borrowing a person—don't take a Loaner for less than six months, unless the person has enough history with your project to hit the ground running.
- Your Loaner must either be Smalltalk-knowledgeable, or must be able to contribute without ever touching Smalltalk. Growing Smalltalk talent is too much of an investment to return when the loan is due!
- Make sure Loaners document their work so others can pick up where they leave off.
- Assign them a "stunt double," who will work with them on a day-to-day basis. (Be aware that too much "time suckage" from the double may turn a Loaner into a "minus.")

> *If asked to loan one of your team members, would you give your best person, or perhaps someone whom you haven't quite been able to find a place for yet?*

"THE COWBOY"

The Cowboy typically learned Smalltalk in relative isolation and is used to being "king of the image". Cowboys delight in tricky code, sometimes doing it for sheer intellectual pleasure without the slightest rationale.

The Cowboy's nemesis is ENVY/Developer, because he doesn't like people looking at his tricky code, he can't imagine others actually *working* on his tricky code, and absolutely hates the constraints imposed by a code management system—if changing the implementation of basicNew suits his purpose, he cannot tolerate the thought of getting the permission of the Library Supervisor!

Cowboys can be wonderful "pluses" if carefully managed; they can also be "minuses" if they consistently destabilize your environment or if their escapades consume an entire "stunt-double" resource. To deal with the Cowboy, try the following:

- Use and enforce your code management system's security features. This includes passwords for all accounts and no shared accounts, especially privileged accounts such as ENVY's Library Supervisor.
- Never, ever let the Cowboy use a privileged account to work on the base image!
- Find tasks for Cowboys that exploit their curious nature—some tasks demand tricky code!
- Establish a culture where the only tricky code tolerated is well documented, complete with the rationale for being tricky.
- The Cowboy is often a Loner, and some of those coping strategies, such as extensive peer-review and "stunt-double" coverage, work well for him also.

"THE SLACKER" OR "ROBINSON CRUSOE"

The Slacker often knows the best web sites and is fluent on the latest Usenet newsgroup gossip. He may often quickly collapse a window as you approach his desk and you may notice his long print jobs that are totally unrelated to work. When others are at his desk, they often seem to be doing the typing or mousing.

We sometimes call this pattern "Robinson Crusoe" because it seems that Slackers always expect to have their work done by Friday, even though they haven't started it by Thursday. (And if on Friday they are inconveniently stranded on some desert isle, Slackers are perfectly content to arrange for other team members to pick up the slack!)

The Slacker never meets a deadline and never works a full week, but neither does he ever report that he is behind schedule and, of course, there is always "The Good Excuse."

Slackers come in two varieties: Dumb and Lazy, and Bored. It is difficult to distinguish between them but the difference is vital:

- A Dumb and Lazy Slacker is in over his head but won't admit it and doesn't really care. He may become a minor "plus" if given a simpler task.
- A Bored Slacker is in well *under* his head and may become a major "plus" if properly inspired.

If you do not raise The Slacker to at least a "zero," your project will suffer much more than the mere loss of effective head count. Sometimes you can do this by the following:

- Give Slackers additional training or mentoring. Put a hard limit on this "time suckage," and let Slackers know it or their mentors will merely end up doing all the work.
- Give Slackers many intermediate deliverables, which may help determine whether they are Dumb and Lazy or merely Bored.
- Micromanage the Slackers with daily progress checks, but recognize that this activity alone may take enough of your time to keep them a "minus."

Often, despite your best efforts, a Dumb and Lazy Slacker cannot be raised to a “plus.” This cannot be tolerated and the person must go. If removing a Dumb and Lazy Slacker from your project is not possible, you need to minimize his impact on your team.

- Isolate the Slacker; forbid him to seek help, and forbid others to help him with his work.
- Perhaps you can turn your Slacker into someone else’s Loaner? (Nah, we wouldn’t suggest that!)

“THE KNOW-IT-ALL”

This person often actually knows a lot, but the Know-It-All’s insecurity causes them to “know” more than they actually do. (In the immortal words of Bo Didley, “It ain’t what you don’t know; it’s what you know that just ain’t so!”) We’ve found this often results from taking someone who has been the “big cheese” on a traditional project and immersing them in Smalltalk, which is strange, different, and frightening to someone who has become used to being an acknowledged expert.

This is the only pattern that Bramson also uses, and he divides them into two categories: the “Bulldozers” and “Balloons,” the primary difference being that “Bulldozers” know what they’re talking about whereas “Balloons” do not. Of the two, “Bulldozers” are merely obnoxious—although they may demoralize others with their strong assertions, they are still strong “pluses,” even in context of the entire team. We’re more concerned with “Balloons,” who can lead an entire project astray if they have the ear of someone important!

Don’t let the insecurity of the Know-It-Alls blind you to what they can be contributing. To deal with the Know-It-All try the following:

- Be quick to acknowledge and reward the greatness of Know-It-Alls when you know they are right—give them strokes freely when they deserve it and they will be less likely to seek strokes for false knowledge.
- Ease Know-It-Alls out of their comfort zone—carve off a bit of the project, such as designing C primitives or RDBMS access, which will allow them to use their expertise while slowly coming to grips with Smalltalk.
- A Know-It-All can be responsible about his or her lack of knowledge when not threatened and may do well if assigned a junior “buddy” to mentor. The mentoring can surreptitiously become two-way, especially if the junior person is farther along the Smalltalk learning curve, but monitor them carefully to make sure The Know-It-All is not filling an impressionable mind with puffery.

CONCLUSION

There are very few truly useless people in this world, but there are many people who are viewed in light of their weaknesses, rather than being put to work using their strengths.

As Smalltalk amplifies this problem, an out-of-place

continued on page 28

INFO@SIGS

SIGS Publications, Inc., 71 West 23rd Street, 3rd Floor, New York, NY 10010; 212.242.7447; Fax: 212.242.7574

ARTICLE SUBMISSION

To submit articles for publication, please contact:
John Pugh & Paul White, Editors, 885 Meadowlands Dr. #509,
Ottawa, Ontario, K2C 3N2 Canada;
email: streport@objectpeople.on.ca

PRODUCT REVIEWS AND ANNOUNCEMENTS

To submit product reviews or product announcements, please contact the Editors at the address above.

CUSTOMER SERVICE

For customer service in the US, please contact PO Box 5050, Brentwood, TN 37024-5050; 800.361.1279; Fax: 615.370.4845; in the UK, please contact Subscriptions Department, Tower Publishing Services, Tower House, Sovereign Park, Market Harborough, Leicestershire, LE16 9EF, UK; +44.(0)1858.435302; Fax: +44.(0)1858.434958

SIGS BOOKS

For information on any SIGS book, contact: Don Jackson, Director of Books, SIGS Books, Inc., 71 West 23rd Street, New York, NY 10010; 212.242.7447; Fax: 212.242.7574; email: donald_jackson@sigs.com

SIGS CONFERENCES

For information on all SIGS Conferences, please contact: SIGS Conferences, 71 West 23rd Street, 3rd Floor, New York, NY 10010; 212.242.7515; Fax: 212.242.7578; email: info@sigs.com

BACK ISSUES

To order back issues, please contact: Back Issue Order Department, SIGS Publications, 71 West 23rd Street, 3rd Floor, New York, NY 10010; 212.242.7447; Fax: 212.242.7574

REPRINTS

For information on ordering reprints, please contact: Reprint Management Services, 505 East Airport Road, Box 5363, Lancaster, PA 17601; 717.560.2001; Fax: 717.560.2063

ADVERTISING

For ad information for any SIGS publication, please contact:

East Coast/Europe:	Gary Portie
Central US:	Elisa Marcus
Recruitment:	Michael Peck
Exhibit Sales, West Coast:	Kristin Viksnins
Exhibit Sales, East Coast:	Sarah Olszewski

212.242.7447; Fax: 212.242.7574
email: sales@sigs.com

West Coast:	Diane Fuller

408.255.2991; Fax: 408.255.2992
email: dhfsigs@hooked.net

SIGS HOME PAGE

Access the SIGS Home Page at: http://www.sigs.com.

THE BEST OF COMP.LANG.SMALLTALK

continued from page 14

enough decisions to be able to work, but few enough that our code doesn't become brittle. That's one of the things that makes software difficult.

Passing off decisions to another object is often referred to as using policy or strategy objects. This is discussed in DESIGN PATTERNS[1] as the Strategy pattern.

Other related ideas are "Open Implementations," which can allow important decisions to be postponed so far that even the end user of the module can control them. I can't do justice to this topic here, but there's a web page available at http://www.xerox.com/PARC/spl/eca/oi.html

Because web pages change so rapidly, I'll also mention that I found it using the search terms *open implementation* and *Gregor Kiczales* (the project leader).

POSTSCRIPT

Although there is a significant element of humor in these principles, I do take them quite seriously and urge you to do the same. They illustrate some very important aspects of OO design and coding. I've even come up with enough of them to fill another column, so the next issue will continue this theme.

Reference

1. Gamma, E. et al. DESIGN PATTERNS: ELEMENTS OF REUSABLE OBJECT-ORIENTED SOFTWARE, Addison-Wesley, Reading, MA, 1994.

MANAGING OBJECTS

continued from page 17

person can cause damage more quickly on a Smalltalk project than they can on a traditional project, and corporate cultural checks that normally help such people, such as peer reviews, management one-on-one meetings, and performance reviews, are tuned to the slower beat of the traditional project.

Beginning a Smalltalk project offers the opportunity for a "behavioral context switch," in which old patterns can be broken. By catching behavioral difficulties early, you can keep them from becoming established patterns. Once behavioral patterns are established, their impact on productivity must be carefully monitored and humanely dealt with.

References

1. Kroeger, O. and J.M. Thuesen. TYPE TALK AT WORK, HOW THE 16 PERSONALITY TYPES DETERMINE YOUR SUCCESS ON THE JOB, Tilden Press, New York, 1992. [*This book concentrates on applying Jungian personality type theory in the workplace, and is much more approachable than defining works on the topic.*]
2. Bramson, R.M. COPING WITH DIFFICULT PEOPLE, Anchor Press/Doubleday, Garden City, NY, 1981.
3. Brooks, Jr., FP. THE MYTHICAL MAN-MONTH (20th anniversary ed.), Addison-Wesley, Reading, MA, 1995. [*A wonderful classic.*