Coding Therapy for Developers
a.k.a. How does this code make you feel?

Steven Feuerstein
PL/SQL Evangelist, Quest Software
www.quest.com  steven.feuerstein@quest.com
I am not making light of therapy

- Therapy can be a very important and meaningful process for healing pain and distress.
- I am not mocking *real* therapy nor do I suggest that my session today can substitute for a session with a real therapist.
- But perhaps I can help you with a few code-related issues....

Now, and later at PL/SQL Obsession: www.ToadWorld.com/SF
Coding Therapy

• Don't be afraid to ask for help.
  – With a touch of dream therapy

• How does this code make you feel?

• Shock therapy
  – SQL is *not* your best friend.

• Battling the inner code sociopath
  – The eternal optimism of the code-full mind

• Couples therapy
  – Developer and Database Administrator
  – Developer and Manager
Why might therapy help developers?

• Sure, some software is generated.
• But almost all software continues to be written directly by human beings.
• Software construction is a wholly intellectual endeavor.
  – That is, to write software we rely almost exclusively on our brains.
• Consequently, human physiology and psychology play an enormous role in how software is written - and mis-written.
Don't be afraid to ask for help.

"Predators look for signs of illness or weakness when choosing their prey, so a prey animal needs to appear healthy, or it will be a sure target. By the time they are showing signs of disease, in many instances, the birds have become too weak to be able to disguise it."

- From peteducation.com

- Our evolved survival instinct urges us to hide weakness.
- On top of that, we software developers are supposed to be really smart.
  - A big bunch of know-it-alls!
- Unfortunately, ignorance leads directly to bugs and sub-optimal code.
So what should you do when you're really stuck?

- **Try dream therapy.**
  - Let your subconscious do the work.
- **Ask for help, of course!**
  - It's not really that terrible a thing to do.
  - No one will think less of you because of it.
Dream therapy for developers

• The worst way to deal with a problem is to worry at it (for hours or days).
  – Sometimes conscious attention only makes it harder to find the solution.

• Maybe you should sleep on it, instead.
  – There's a good chance that when you wake up the answer will be waiting in your conscious "in box."

• If you can't go to bed and you're stuck on a problem, then...
  – Take a walk, run around the block, do some stretches or sit-ups, anything but stare at your screen.
Tips to make it easier to plead ignorance

• Most people *like* to be asked to help.
  – And simply asking (or *contemplating asking*) will often steer you to an answer. It doesn't matter *who* you ask!

• **Follow the Thirty Minute Rule.**
  – If you can't fix a bug in 30 minutes, ask for help.
  – You are usually too far inside the problem to step back and take a fresh look.

• **Senior developers and managers must take the lead!**
  – Ask more junior members for help. Show that you are fallible, that you can learn from anyone and everyone.
How does this code make you feel?

```sql
EXCEPTION
  WHEN OTHERS
  THEN
    IF SQLCODE = -24381
      THEN
        DBMS_OUTPUT.put_line ( 'Total number of errors = ' || SQL%BULK_EXCEPTIONS.COUNT);
      END IF;
    RAISE;
END;
```

• Puzzled? Deficient? Intimidated?

"Ha, ha. I know what -24381 is and you don't. You're not very good at this, are you?"
How does this code make you feel? Part II

• I need to iterate through a collection, but it might be empty and I want to avoid the VALUE _ERROR exception.

```sql
FOR rowind IN NVL (birthdays.FIRST, 0) .. NVL (birthdays.LAST, -1)
  LOOP
    DBMS_OUTPUT.put_line (birthdays (rowind));
  END LOOP;
END IF;
```

• You have to *interpret* that code. How about this one?

```sql
IF birthdays.COUNT > 0
  THEN
    FOR rowind IN birthdays.FIRST .. birthdays.LAST
      LOOP
        DBMS_OUTPUT.put_line (birthdays (rowind));
      END LOOP;
  END IF;
```

*for_loop_with_nulls.sql*
So how *should* code make you feel?

- A great program is one that *welcomes* the visitor, makes you feel at home.
  - And makes you feel like you can change the code without risking ripple effects of bugs that are hard to track down without regression tests.
- Aim for maximum readability and *not* high code density.
  - "Look how much I can do in ONE LINE!" is not a boast one should make.
- Bottom line: don't use code to show everyone how clever you are.
Shock Therapy

- Sometimes you need to be given a shock to be able to break out of your ruts, and gain a new perspective on your life.
- Same with software.
- So here's your shock therapy of the day:

```
SQL
```

It's so easy to write SQL in PL/SQL! No JDBC or ODBC nonsense needed here!

This INSERT will never raise a DUP_VAL error. No one else will ever need to write a SQL statement like this.

Copyright © 2008 Steven Feuerstein
What? No SQL for PL/SQL developers?

• That's right. I suggest that *application-level programs* should not contain any SQL statements. Why?

• Because every SQL statement you write is a *hard-coding* – and it is a more dangerous and problematic version of hard-coding than hard-coded *literals*.

• You shouldn't repeat "170546" and you shouldn't repeat "SELECT * FROM...."

• Yes, I know: shocking!
What's a PL/SQL developer to do?

- Create a data access layer. AKA:
  - Table API
  - Transaction API
  - Data encapsulation

- Instead of writing SQL statements, call functions and procedures that do the work for you....
  - Hiding your implementation (it will change)
  - Strengthening your error handling (you only write it – or generate it – once)
  - Increasing your productivity.

Quest CodeGen Utility: www.ToadWorld.com  Downloads
More shocks to the system...

• You should consider every hard-coded VARCHAR2 declaration to be a bug.
  – Unless it's the "single point of definition."

 DECLARE
 l_full_name VARCHAR2(100) :=
   first_name_in || ' ' || last_name_in;
l_big_string VARCHAR2(32767);

• What should you do instead?
  – Use %TYPE, %ROWTYPE and SUBTYPEs.

 DECLARE
 l_full_name employees_rp.full_name_t :=
   employees_rp.full_name (first_name_in,
   last_name_in);
l_big_string plsql_limits.maxvarchar2_t;

Three bugs!
No, I am not accusing you of moral irresponsibility or lack of conscience.

A sociopathic coder is one who is totally consumed by the program at hand, by the deep need to solve the logical puzzle.

– Completely indifferent to or ignorant of the need for that program to live in a long-lasting community of code.

And these coders are generally the most dangerously optimistic of our kind.
The eternal optimism of the code-full mind

- Many programmers live in denial, stubbornly optimistic that their code will defy every known law and experience.
  - And that each of us know the best way to do everything.
- Throwaway, non-repeatable testing
  - "After I get it working today, why would it ever need to be tested again?"
- Minimal exception handling
  - "This INSERT statement will never raise an error."
- No comments
  - "My code is self-documenting."
Where does this optimism come from?

• Our brain has evolved to let us project into the future (plan) and remember the past.
  – But for survival, the present dominates and overwhelms our perceptions.

• This "hard-wired" emphasis on the present makes it very hard to see when, where and why we need to explain ourselves.

• As we are writing our code it always seems transparently obvious.
  – Even as we acknowledge and then repress that what we are doing is quite complicated.
Mental devices to avoid Now Coding

• The "Memento" Factor
  – As in the movie about a guy who forgets everything by the time he wakes up each day.
  – Pretend you will have total amnesia tomorrow about the code you wrote today. (Or tattoo yourself with reminders about your code.)
  – How will that change the code you write, right now?

• Think of the (your) next generation.
  – Your child may grow up to be a programmer.
  – That programmer may have to maintain your code.
  – What sort of code will you inflict on your progeny?
Be ready to learn from non-programmers

- Two very different electricians....

"Spaghetti" Wiring

Well-documented and modularized wiring
And from my woodworking teacher...

The fastest way to get the job done is to take your time.

• We all know this, right?
• When you hurry, you make mistakes.
• When you hurry, you take shortcuts that you know will cause pain and delay – later.
• So probably the most important piece of advice I can give me, I mean you is:

Fight against that urge to hurry.
TAKE YOUR TIME AND DO IT RIGHT.
Couples Therapy: Developer and DBA

• Can't we all just get along?
• Apparently not....
  – Developer: "Why can't I just add that column myself?"
  – DBA: "No, you tune the application's performance."
• Stay on good terms with your DBA.
  – Let's face it; a DBA has lots more responsibility and privileges (which translate to "power" in a database).
  – You can't win a battle with a DBA.
• Application development should be a collaborative process.
Developer and DBA: a close collaboration is best

- As you change the way you write your code, you should notify your DBA.
  - Start to work with collections? Notify!
  - Using FORALL and BULK COLLECT? Notify!
  - Discovered Oracle11g function result cache? Notify!
- Ask (negotiate) for the resources you need to analyze code impact yourself.
  - For example, with access to just a handful of V$ views, you can monitor PGA memory usage of your application.
Couples Therapy: Developer and Manager

• Are you satisfied with the way your manager communicates with you?
• Which of these does your manager ask:

  - How maintainable is your code?
  - When will you be done?
  - How thoroughly have you tested your code?
  - Did you follow the coding standards?
  - How can I help you get your job done?
  - Can you make it go faster?
Developer and Manager: pretense is critical

• Most managers need to know that you, the developer, know "who's boss."
  – Even though, really, the only point of a manager is to help you succeed (after all, who does the real work?).

• So avoid direct challenges to authority and instead...
  – Buy your manager a copy of Peopleware ("Thanks for being such a great manager!").
  – Ask frequently for help, advice and direction. ("How should I improve my testing?")
  – Ask for coding standards and mechanisms to verify that you are following them.
But seriously, Development Managers...

- Managers represent the interests of the employer.
- Software that is written today becomes a part of the intellectual assets of the company.
- So the manager should take primary responsibility for ensuring that code works today, but will also work for years to come.
- And that takes resources, tools, changes in culture.
What every development manager needs

- Coding standards and naming conventions
  - [http://examples.oreilly.com/orbestprac/](http://examples.oreilly.com/orbestprac/)

- Automated regression tests
  - Quest Code Tester, utPLSQL, PL/Unit, PLUTO, dbFit

- Automated code analysis and review
  - Toad CodeXpert, PL/SQL warnings framework, ClearSQL

- Automated SQL analysis and performance tuning.
  - It doesn't make sense for developers to have to learn the ins and outs of tuning SQL!
Coding Therapy: We're in this together.

• OK, so you're not the perfect coder. Neither am I.
  – We've all got issues.

• In fact, if you are anything like me, you are:
  – Lazy, undisciplined, short-term focused
  – Deeply suspicious that there might be more to life than sitting in front of a computer screen and keyboard.

• So deal with it!
  – Recognize that our brains and our bodies are involved and sometimes work (subconsciously) against our best coding intentions.
  – Work with your managers, developers and DBAs so we can all help each other stay off the coding therapist's couch.