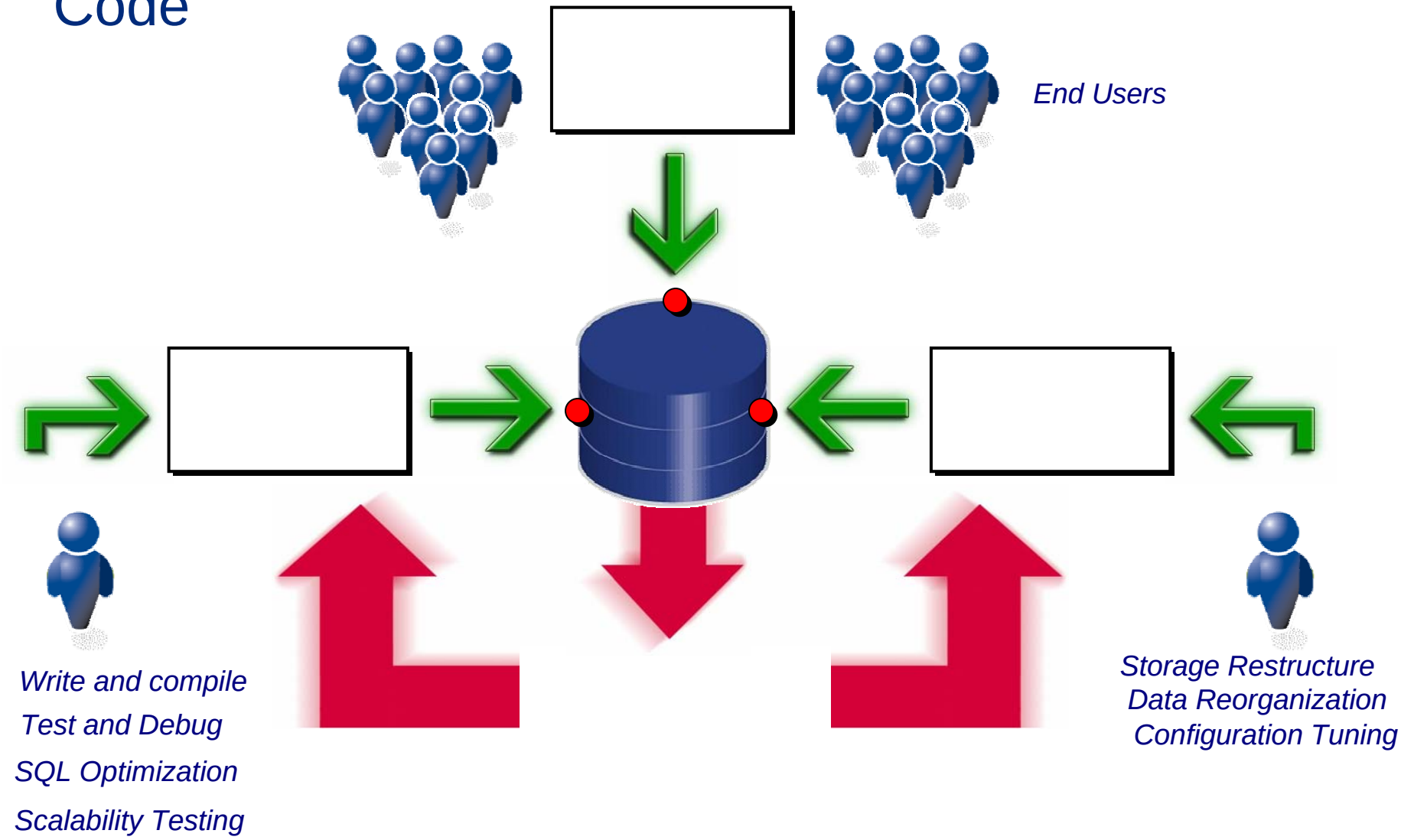
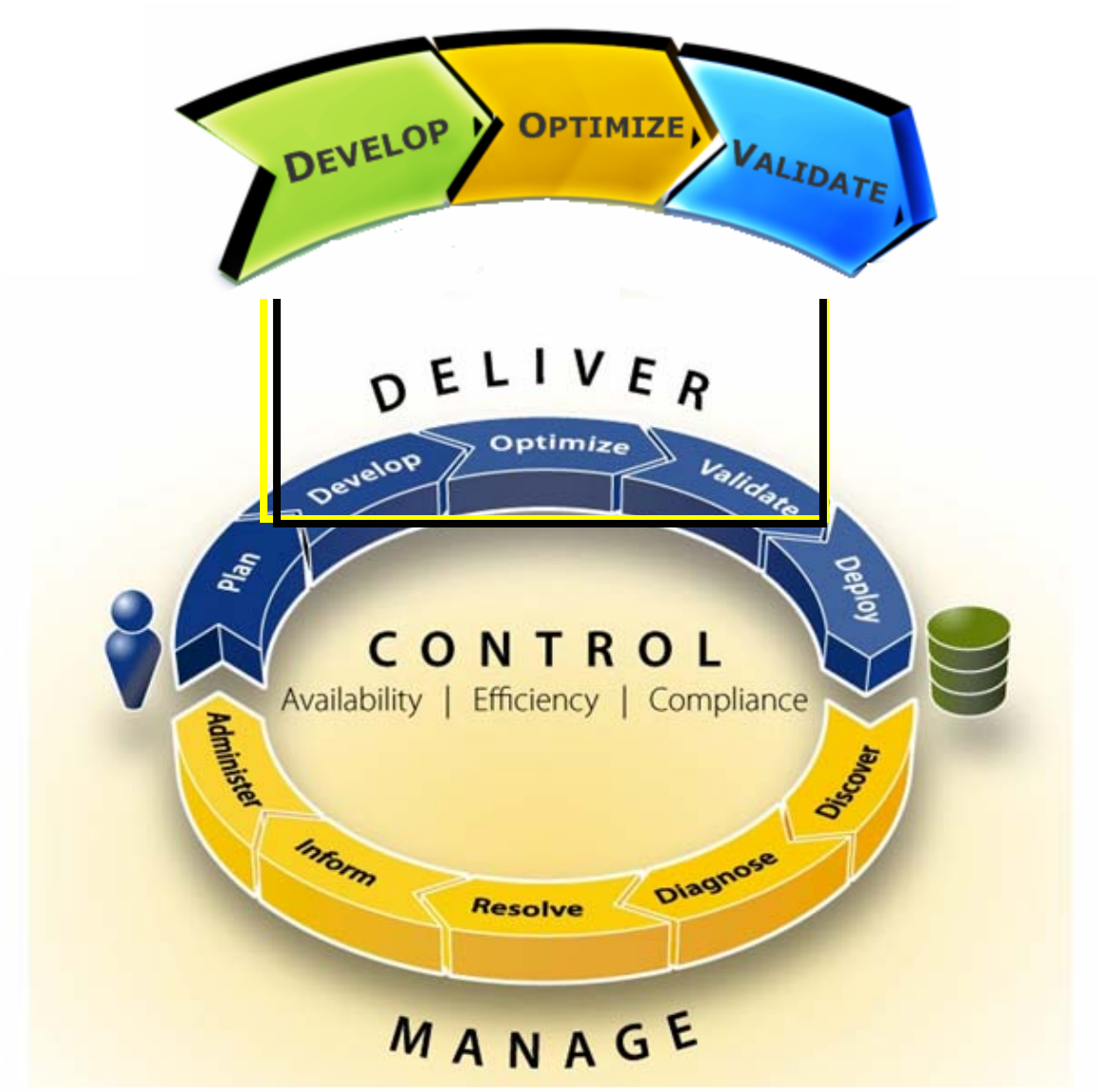
Database Development Best Practices

The Impact of Poor Quality and Performing Code



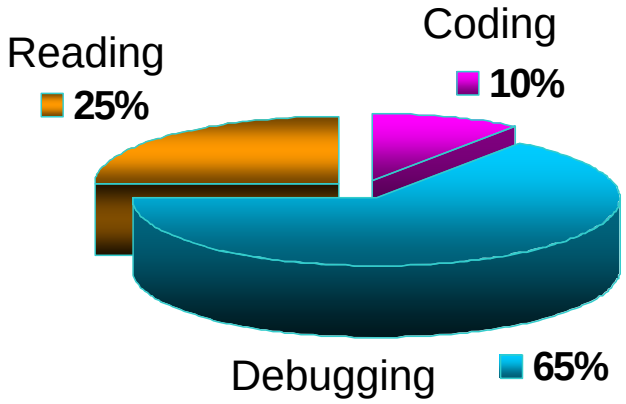
Development Best Practices



Deliver Quality Code

Step 1: Develop

- Define tests to measure success objectively
- Write code, focusing on single program unit at a time
- Test unit of code
- Debug code
- Apply standard formatting



Developer Resource Distribution

Source: GARTNER GROUP

•There can be as many as 20 to 30 bugs per 1,000 lines of software code. —Sustainable Computing Consortium

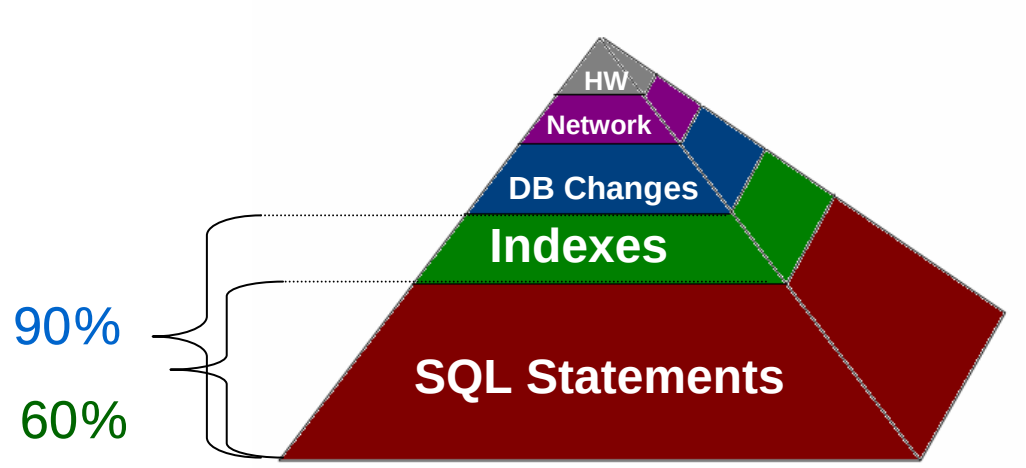
•32% of organizations say that they release software with too many defects.—Cutter Consortium

•Developers spend about 80% of development costs on identifying and correcting defects.—*The National Institute of Standards and Technology*

Deliver Optimal Code

Step 2: Optimize

- Review coding best practices
- Proactively identify problematic SQL directly from the source code
- Automatically rewrite SQL code in every possible alternative



Root of Database Performance Impact

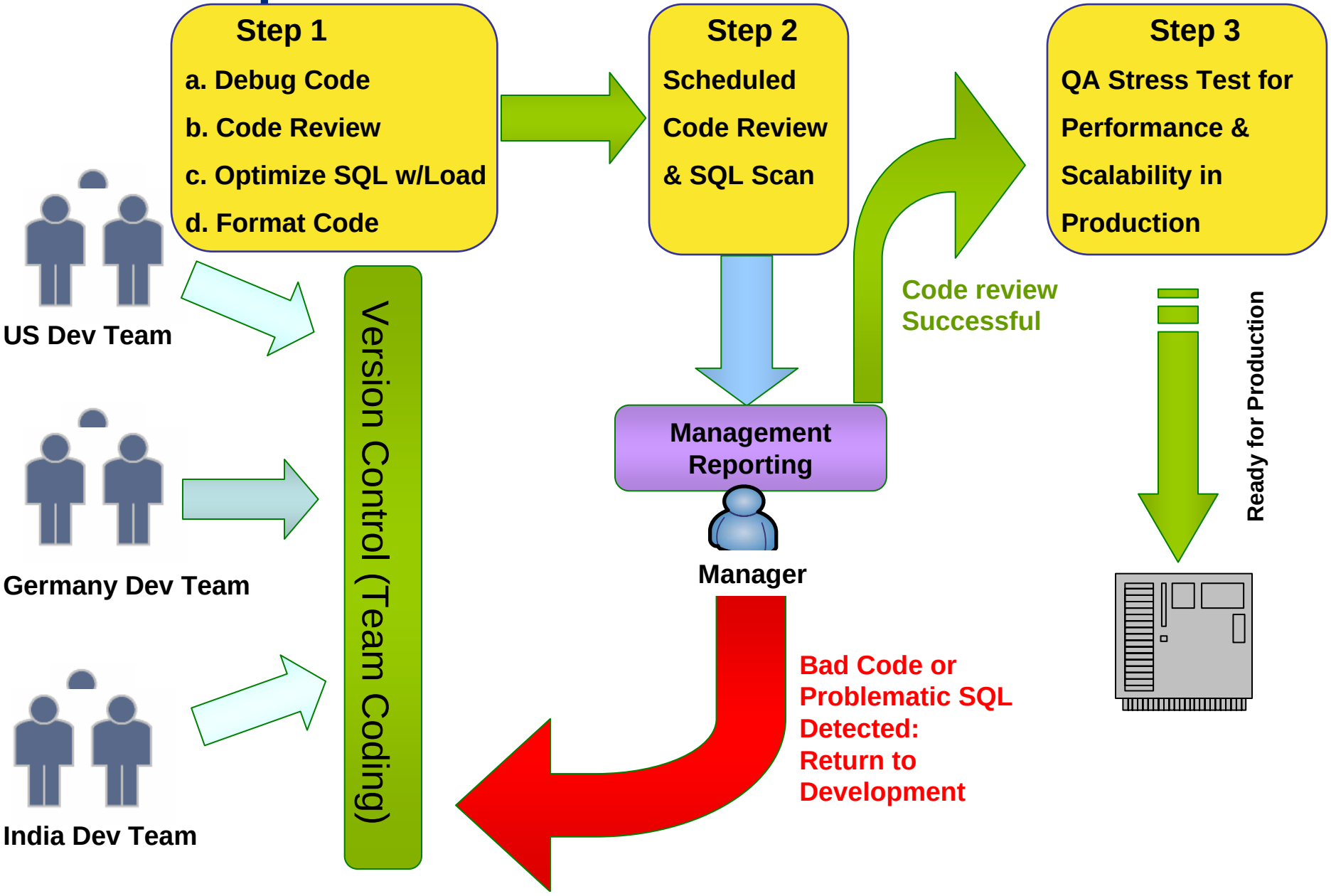
Source: Forrester Research

Deliver Optimal Code

Step 3: Validate

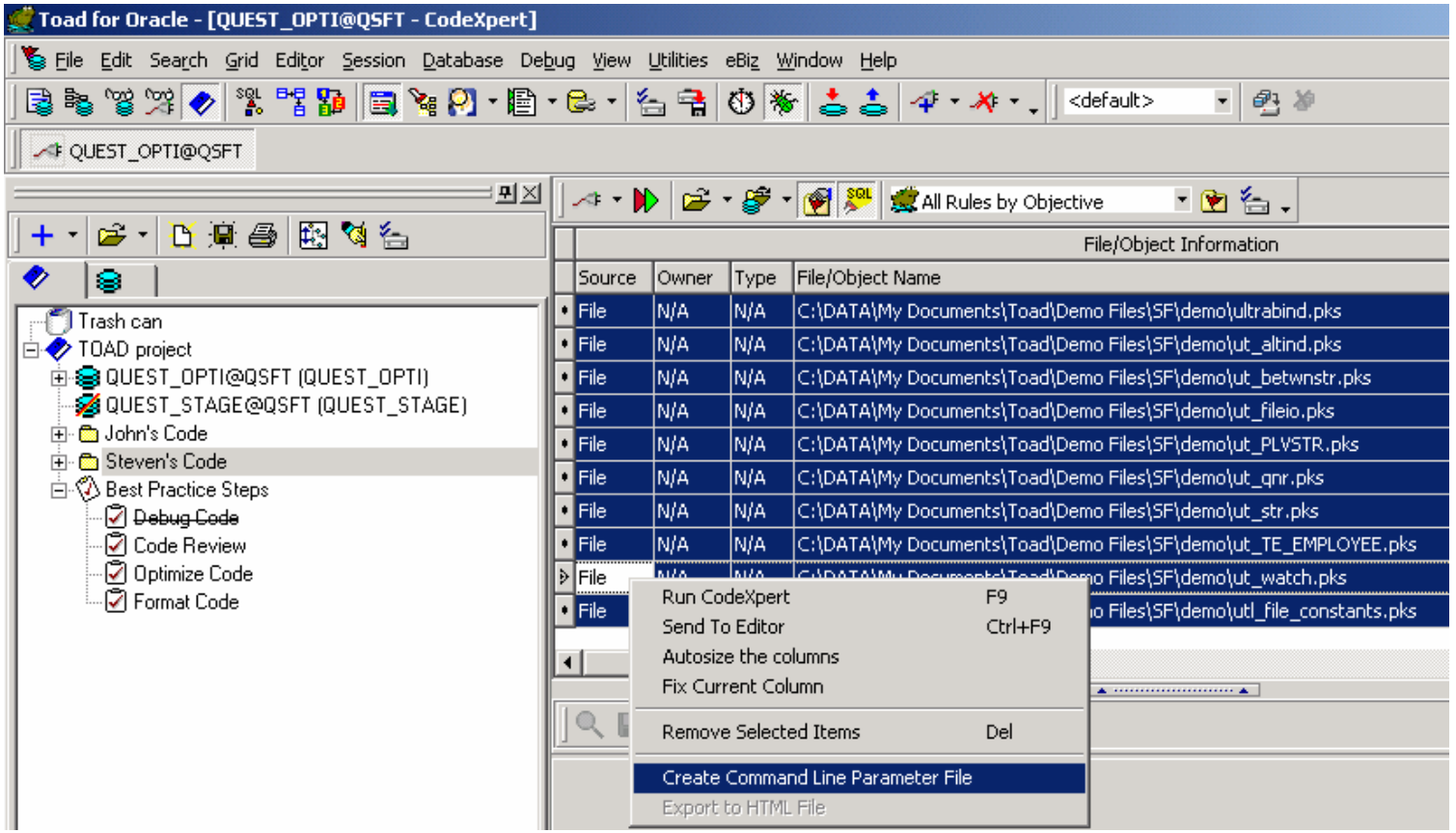
- Benchmark SQL alternatives for performance and scalability
- Ensure code will perform for production requirements before deployment
- Create management report that shows best practices have been adhered to

Development Best Practices



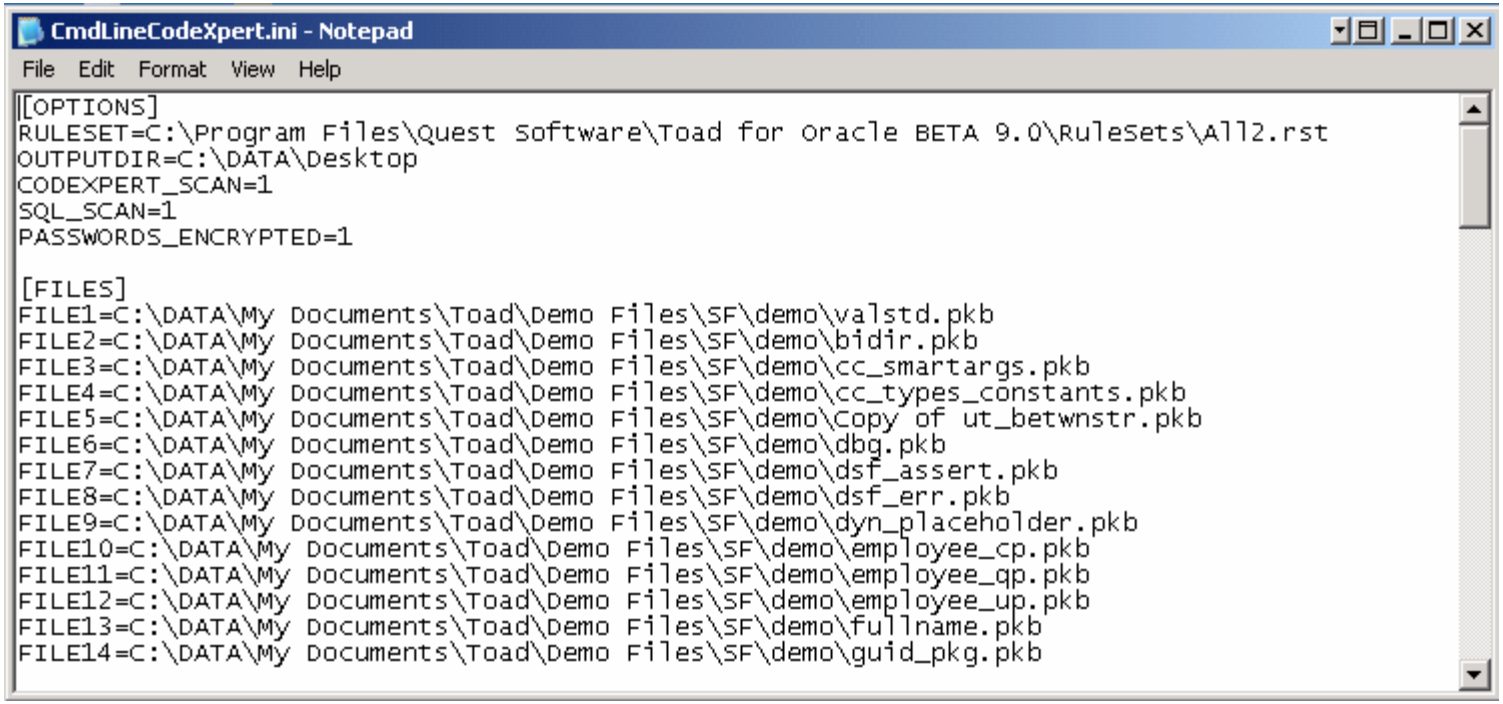
Automated Code Reviews

1. Add the code folder to Toad Project Manager
2. Right-click to send to Code Xpert console
3. Right-click to create Code Review command line file



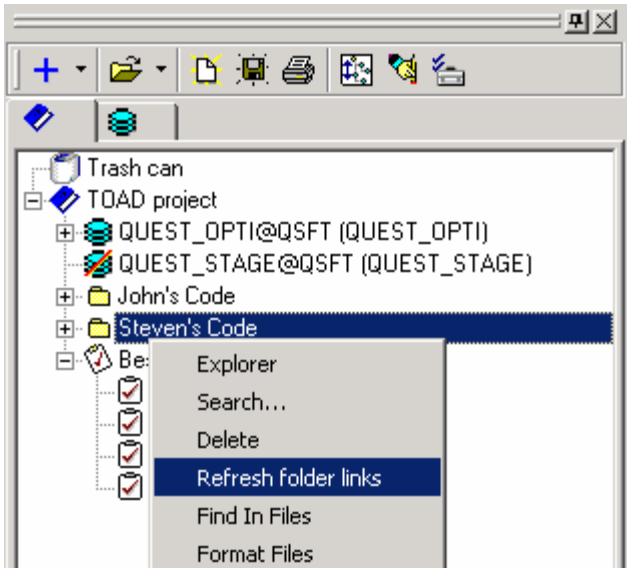
Schedule Code Reviews

- Use a simple command line to schedule code review
  - Toad.exe CE=CmdLineCodeXpert.ini
- Passwords are encrypted



Adding New Code for Review

- In Toad Project Manager R/C to refresh folder contents before creating command line file
- Or, just manually add new source to command line text file



Code Review Analysis

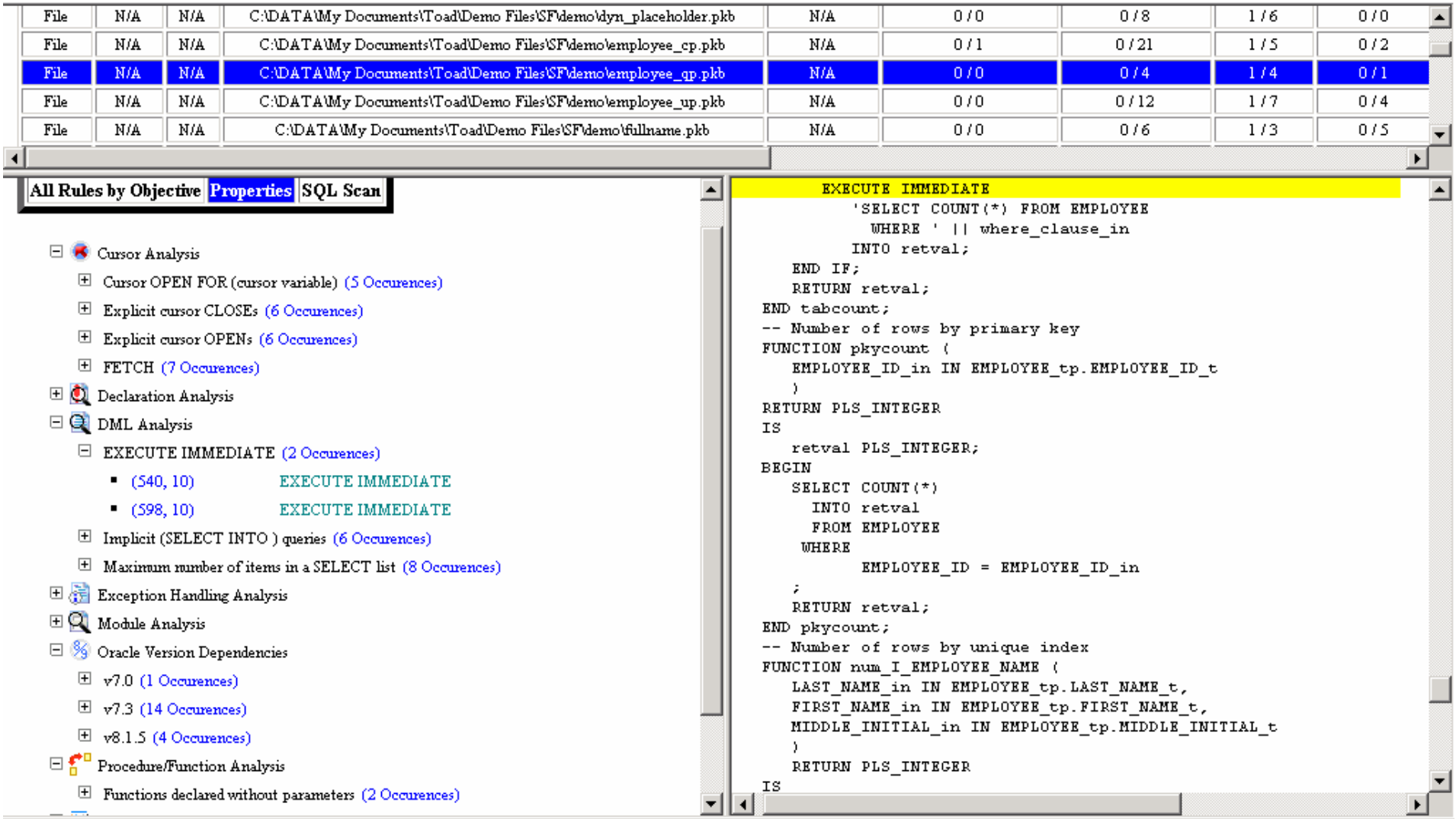
File/Object Information					Totals by Objective			
Source	Owner	Type	File/Object Name	Connection	Code Correctness	Maintainability	Efficiency	Readability
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\valstd.pkb	N/A	0 / 0	0 / 6	1 / 6	0 / 3
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\bidir.pkb	N/A	0 / 0	0 / 0	1 / 8	0 / 1
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\cc_types_constants.pkb	N/A	0 / 0	0 / 27	1 / 10	0 / 12
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\Copy of ut_betwnstr.pkb	N/A	0 / 0	0 / 4	1 / 4	0 / 2
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\dbg.pkb	N/A	0 / 0	0 / 7	1 / 41	0 / 28
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\dsf_assert.pkb	N/A	0 / 0	0 / 0	1 / 9	0 / 0
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\dsf_err.pkb	N/A	0 / 0	0 / 2	1 / 0	0 / 0
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\dyn_placeholder.pkb	N/A	0 / 0	0 / 8	1 / 6	0 / 0
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\employee_cp.pkb	N/A	0 / 1	0 / 21	1 / 5	0 / 2
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\employee_qp.pkb	N/A	0 / 0	0 / 4	1 / 4	0 / 1
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\employee_up.pkb	N/A	0 / 0	0 / 12	1 / 7	0 / 4

All Rules by Objective | Properties | SQL Scan

- Code Correctness (0 Rules / 0 Occurences)
- Efficiency (2 Rules / 4 Occurences)
- Maintainability (2 Rules / 4 Occurences)
  - CURSOR - 2610 (1 Occurences) — Limit use of weak REF CURSOR types.
    - (248, 7) TYPE weak_rc IS REF CURSOR;
  - LITERAL - 4602 (3 Occurences) — Avoid use of literals in non-declarative parts
- Program Structure (3 Rules / 4 Occurences)
- Readability (1 Rules / 1 Occurences)

```
   TYPE weak_rc IS REF CURSOR;
   allrows_cur weak_rc;
   l_rows PLS_INTEGER;
   retval EMPLOYEE_tp.EMPLOYEE_tc;
BEGIN
   OPEN allrows_cur FOR
      'SELECT
            EMPLOYEE_ID,
            LAST_NAME,
            FIRST_NAME,
            MIDDLE_INITIAL,
            JOB_ID,
            MANAGER_ID,
            HIRE_DATE,
            SALARY,
            COMMISSION,
            DEPARTMENT_ID,
            EMPNO,
            ENAME,
```

Code Review Properties



File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\dyn_placeholder.pkb	N/A	0 / 0	0 / 8	1 / 6	0 / 0
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\employee_cp.pkb	N/A	0 / 1	0 / 21	1 / 5	0 / 2
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\employee_qp.pkb	N/A	0 / 0	0 / 4	1 / 4	0 / 1
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\employee_up.pkb	N/A	0 / 0	0 / 12	1 / 7	0 / 4
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\fullname.pkb	N/A	0 / 0	0 / 6	1 / 3	0 / 5

All Rules by Objective | Properties | SQL Scan

- Cursor Analysis
  - Cursor OPEN FOR (cursor variable) (5 Occurences)
  - Explicit cursor CLOSEs (6 Occurences)
  - Explicit cursor OPENs (6 Occurences)
  - FETCH (7 Occurences)
- Declaration Analysis
- DML Analysis
  - EXECUTE IMMEDIATE (2 Occurences)
    - (540, 10) EXECUTE IMMEDIATE
    - (598, 10) EXECUTE IMMEDIATE
  - Implicit (SELECT INTO) queries (6 Occurences)
  - Maximum number of items in a SELECT list (8 Occurences)
- Exception Handling Analysis
- Module Analysis
- Oracle Version Dependencies
  - v7.0 (1 Occurences)
  - v7.3 (14 Occurences)
  - v8.1.5 (4 Occurences)
- Procedure/Function Analysis
  - Functions declared without parameters (2 Occurences)

```
      EXECUTE IMMEDIATE
         'SELECT COUNT(*) FROM EMPLOYEE
           WHERE ' || where_clause_in
         INTO retval;
   END IF;
   RETURN retval;
END tabcount;
-- Number of rows by primary key
FUNCTION pkycount (
   EMPLOYEE_ID_in IN EMPLOYEE_tp.EMPLOYEE_ID_t
   )
RETURN PLS_INTEGER
IS
   retval PLS_INTEGER;
BEGIN
   SELECT COUNT(*)
     INTO retval
     FROM EMPLOYEE
    WHERE
          EMPLOYEE_ID = EMPLOYEE_ID_in
   ;
   RETURN retval;
END pkycount;
-- Number of rows by unique index
FUNCTION num_I_EMPLOYEE_NAME (
   LAST_NAME_in IN EMPLOYEE_tp.LAST_NAME_t,
   FIRST_NAME_in IN EMPLOYEE_tp.FIRST_NAME_t,
   MIDDLE_INITIAL_in IN EMPLOYEE_tp.MIDDLE_INITIAL_t
   )
   RETURN PLS_INTEGER
IS
```

Code Review – Problematic SQL

TOAD CodeXpert

File/Object Information					Totals by Objective			
Source	Owner	Type	File/Object Name	Connection	Code Correctness	Maintainability	Efficiency	Readability
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\valstd.pkb	N/A	0 / 0	0 / 6	1 / 6	0 / 3
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\bidir.pkb	N/A	0 / 0	0 / 0	1 / 8	0 / 1
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\cc_types_constants.pkb	N/A	0 / 0	0 / 27	1 / 10	0 / 12
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\Copy of ut_betwnstr.pkb	N/A	0 / 0	0 / 4	1 / 4	0 / 2
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\dbg.pkb	N/A	0 / 0	0 / 7	1 / 41	0 / 28
File	N/A	N/A	C:\DATA\My Documents\Toad\Demo Files\SF\demo\dsf_assert.pkb	N/A	0 / 0	0 / 0	1 / 9	0 / 0

All Rules by Objective | Properties | SQL Scan

- Invalid SQL (0 Occurrences)
- Problematic SQL (19 Occurrences)
  - (22, 10) SELECT
  - (58, 10) SELECT 1
  - (81, 10) SELECT
    - Full table scan with table size larger than the Problematic SQL Full Table Scan Thresho
  - (109, 10) SELECT
  - (221, 10) SELECT
  - (254, 11) 'SELECT
  - (287, 10) SELECT
  - (404, 10) SELECT

```
SELECT
       EMPLOYEE_ID,
       LAST_NAME,
       FIRST_NAME,
       MIDDLE_INITIAL,
       JOB_ID,
       MANAGER_ID,
       HIRE_DATE,
       SALARY,
       COMMISSION,
       DEPARTMENT_ID,
       EMPNO,
       ENAME,
       CREATED_BY,
       CREATED_ON,
       CHANGED_BY,
       CHANGED_ON
  FROM EMPLOYEE
 WHERE
```

Why Best Practices and Toad

- Toad has a huge user community and it is likely development teams are already familiar with Toad
- Toad has/ the needed functionality to significantly enhance the development process (code review and tuning)
- The Best Practices process is a simple extension of Toad's use
- By spreading the responsibility for quality code you help to remove the bottle necks
  - Manual code review team (they can't catch all the bad code)
  - QA responsible for quality and performance (To much code to tune)

Toad for Oracle v 9.0 is a Big Release!

- Toad for Oracle 9.0 has many enhanced capabilities, including:
  - **Reporting** – provides IT management with detailed insight into development practices through customizable reports that document team code quality.
  - **Code Xpert and Code Quality Repository** – enables database best practices through automated code reviews to ensure the highest code quality.
  - **Integrated Editors** – simplifies coding and increases productivity through a new single editor for application and database code with the ability to work fully offline.
- This is a HUGE deal because
  - New features simplify user workflow and enable Best Practice workflow

Database Development Best Practices

QUEST SOFTWARE®