



Quest



Eero Mattila
Principal Systems Consultant

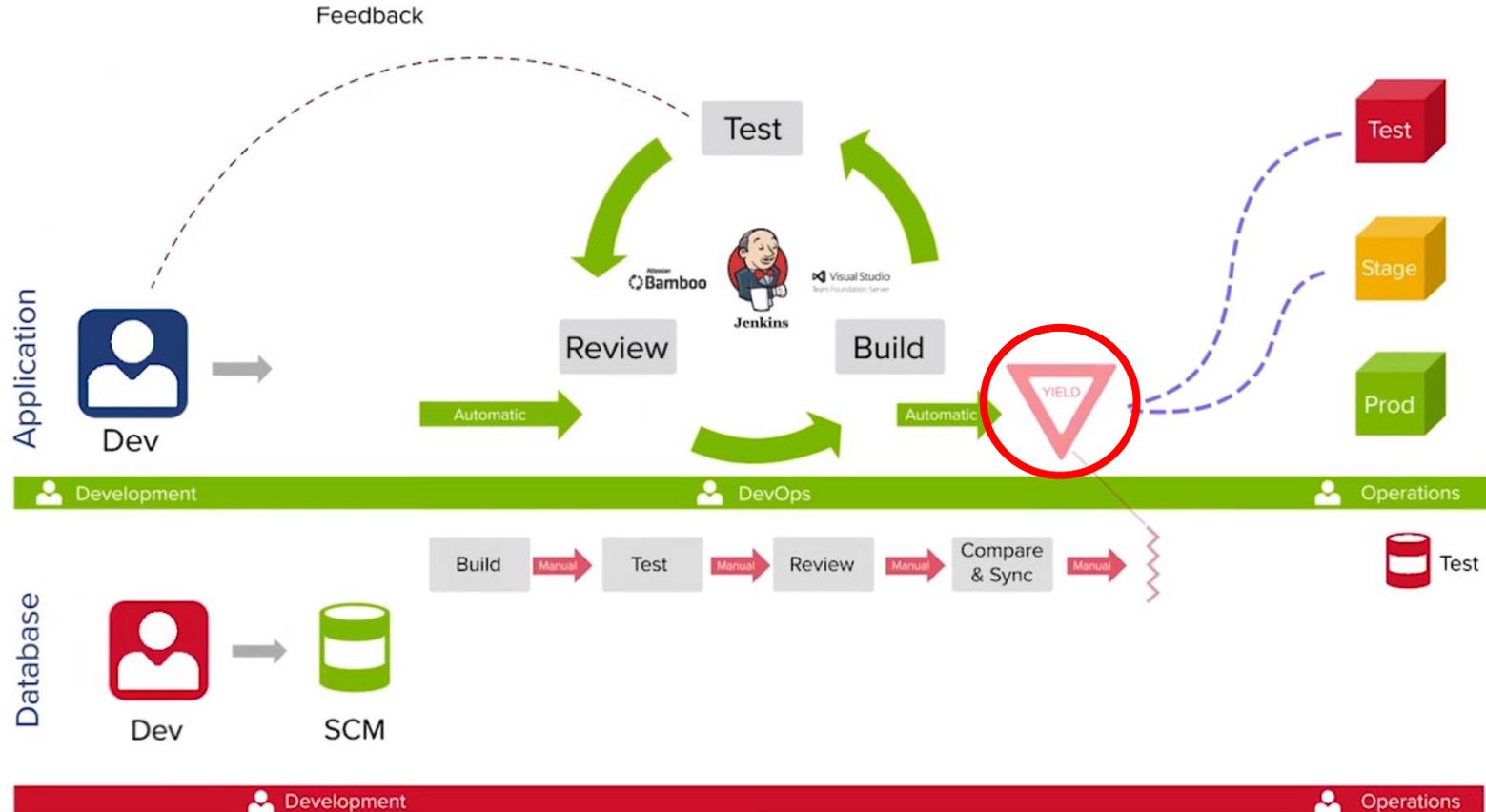
Continuous Database Operations For Your DevOps Infrastructure

Are You Trying To Future Proof Your Databases?

- Are you...
 - under pressure from your business to build, test and release software changes **faster**?
 - unclear about how to bring **database** processes into your **DevOps** infrastructure?
 - concerned about the **risks** of making **compromises** in your desire to shorten release cycles?
 - unsure how to effectively **monitor** the performance impacts of your database changes?
 - required to **replicate** fast moving changes to your hybrid database environments?



Consequences of not changing



Consequences of not changing



Application releases are often delayed when database changes are required.

Insufficient testing and monitoring of DB changes may result in poor application performance or downtime.

Your competition will be able to innovate faster than you.

But What If You Could...



Automate unit testing and code reviews and integrate them with your build process



Test, monitor and identify performance issues throughout your DevOps pipeline?



Automatically replicate production changes in near real-time to other database environments inc those in the cloud?

You can – with Toad!

- Easily integrate DB application development with Version Control Systems
- Create repeatable Unit Tests for PL/SQL to eliminate Bugs production.
- Continually control the quality and maintainability of PL/SQL code minimize unplanned code changes.
- Automate SQL Tuning for better performance of SQL statements





The Basics

1) Track changes with Team Coding & VCS integration

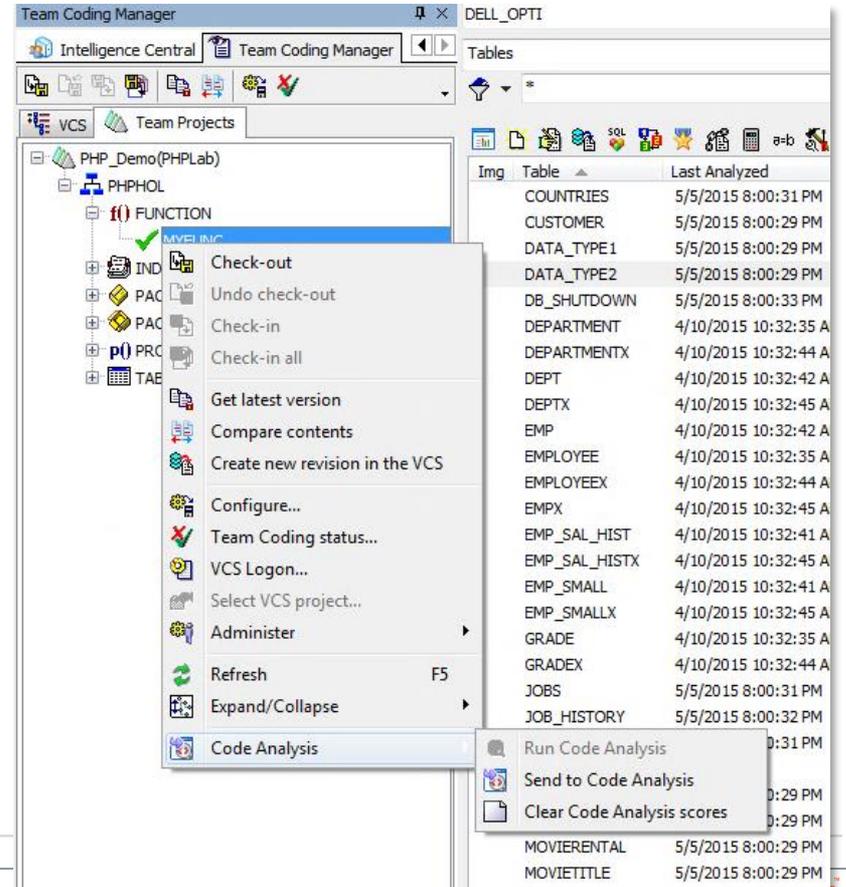


Why is this important?

- Ensures correct code versions are used during build.
- Ensures ongoing PL/SQL code/object integrity.
- Provides true collaborative development.

Benefits:

- **Intuitive** – enables you to work directly on DB objects without worrying about checking-out/-in files.
- **Improves standards**– helps promote a team-based environment for ensuring code changes/updates are consistent. Eliminates backdoor changes.
- **VCS enabled** – works with version control systems like Git, Microsoft TFS, CVS, ClearCase, Perforce, and SVN.
- **Productivity** – seamlessly check-in and check- out objects directly from within Toad.
- **Policies** – optionally force PL/SQL regression test executions and minimum code review standards prior to code check-in.



2) Automate PL/SQL unit testing with Toad Code Tester

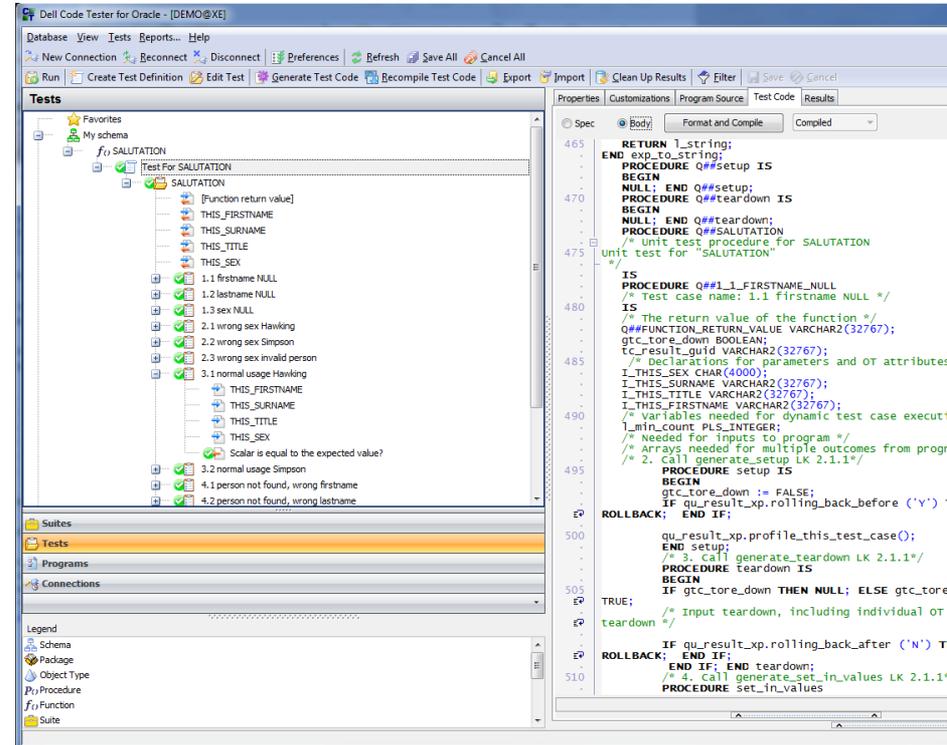


Why is this important?

- Ensures tests cover all operational use cases.
- Reduces the risk of bugs in production.
- Becomes part of Continuous Integration process.

Benefits:

- **Intuitive and fast** – enables you to think about code behavior as you define your tests – test code is generated automatically. Simply run your code and create your tests.
- **Persistent** – all unit tests are preserved for the life of the code, facilitating full regression test/code coverage reporting during ongoing code changes.
- **VCS enabled** – optionally require full regression test cycle during code check-in.
- **CI integration** – unit test executions become callable from a CI build process.



3) Automate code reviews with Toad Code Analysis

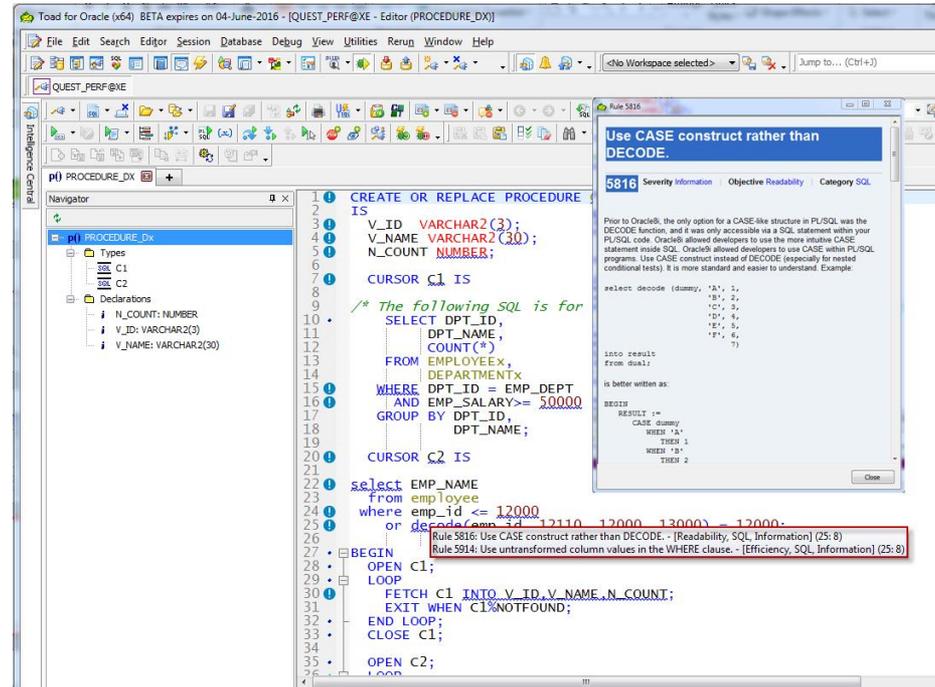


Why is this important?

- Ensures consistent code quality standards are applied across an entire project.
- Becomes part of Continuous Integration process.

Benefits:

- **Intuitive and fast** – continuous parsing of code enables you to instantly see where code quality violations occur as you code.
- **Customizable** – different sets of rules can be defined per project – rules can be customized.
- **VCS enabled** – optionally require minimum code quality level during code check-in.
- **CI integration** – code reviews become callable from a CI build process.



4) Automate SQL tuning with Toad Auto Optimize SQL



Why is this important?

- Provides a really simple way to tune poor SQL and PL/SQL before production.
- Cost of fixing in production can be exponentially higher and very disruptive for business.

Benefits:

- **Intuitive and fast** – based on patented re-write technology to directly fix the problem – not just offer advice.
- **No hidden costs** – does not use the Oracle EE Management Packs.
- **Evidence based** – provides different alternatives based on pre-defined performance goals.
- **Advanced optimization** – more advanced SQL optimization capabilities such as indexing available depending on developer's experience and permissions.

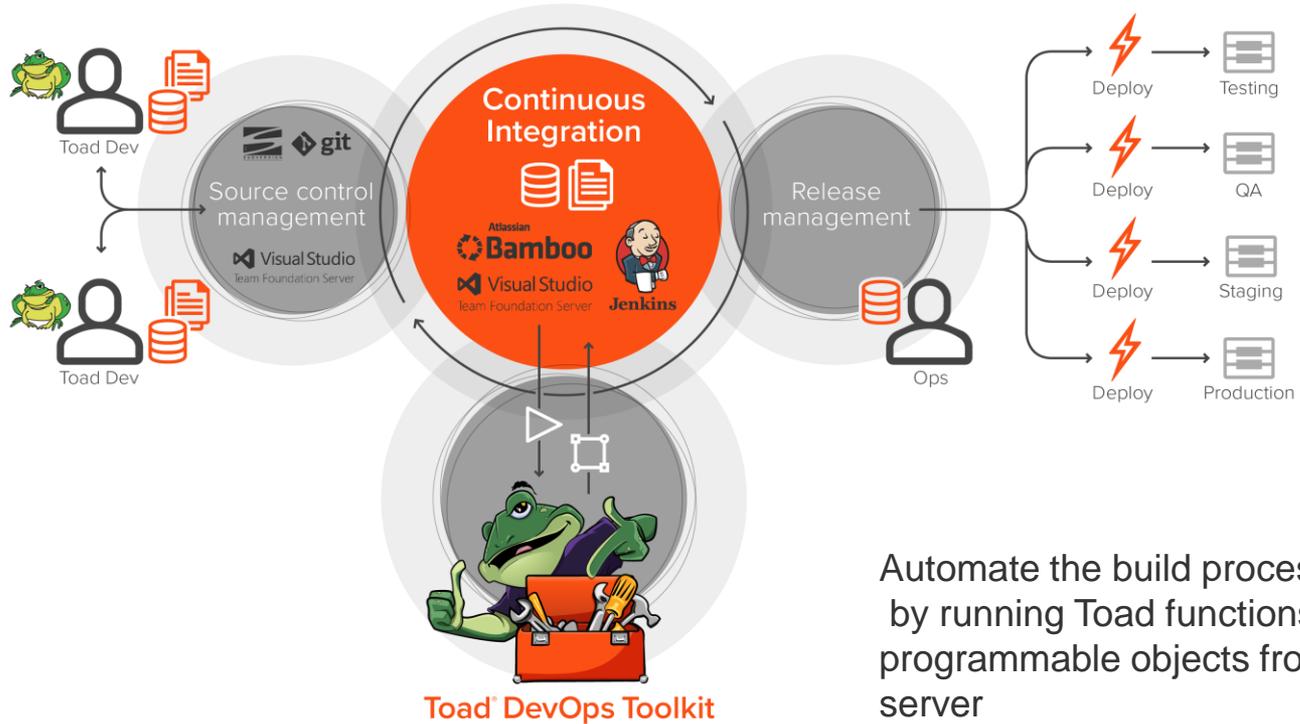
The screenshot displays the Toad for Oracle interface during an 'Auto Optimize SQL' session. The main window shows a list of alternatives with their respective elapsed times. The 'Original SQL' is highlighted, and its execution plan is visible in the bottom right pane. The execution plan for the original SQL shows a 'SELECT STATEMENT ALL_ROWS' with a cost of 2,487. The plan includes a 'SORT AGGREGATE' step, followed by 'NESTED LOOPS', a 'VIEW VIEW QUEST_PERF.index\$_joins_001', a 'HASH JOIN', and an 'INDEX RANGE SCAN INDEX QUEST_PERF.DEPART'. The execution plan for 'Alternative 22' shows a 'SELECT STATEMENT ALL_ROWS' with a cost of 747, which is significantly lower than the original SQL. The plan for Alternative 22 includes a 'SORT AGGREGATE' step, followed by a 'HASH JOIN', and an 'INDEX RANGE SCAN INDEX QUEST_PERF.DEPART'. The interface also shows a progress bar indicating that 99% of 104 tests were completed, with 369 rewrites checked.

Name	Elapsed Time
Alternative 22	0.23 secs
Alternative 19	0.24 secs
Alternative 23	0.24 secs
Alternative 25	0.26 secs
Alternative 24	0.28 secs
Alternative 4	0.31 secs
Original SQL	0.33 secs
Alternative 1	0.33 secs
Alternative 5	0.33 secs
Alternative 3	0.34 secs
Alternative 21	0.34 secs
Alternative 18	0.36 secs
Alternative 6	0.37 secs
Alternative 20	0.44 secs
Alternative 27	0.45 secs
Alternative 2	0.46 secs



Make Database Development a part of Continuous Integration

Continuous Integration with Toad® DevOps Toolkit

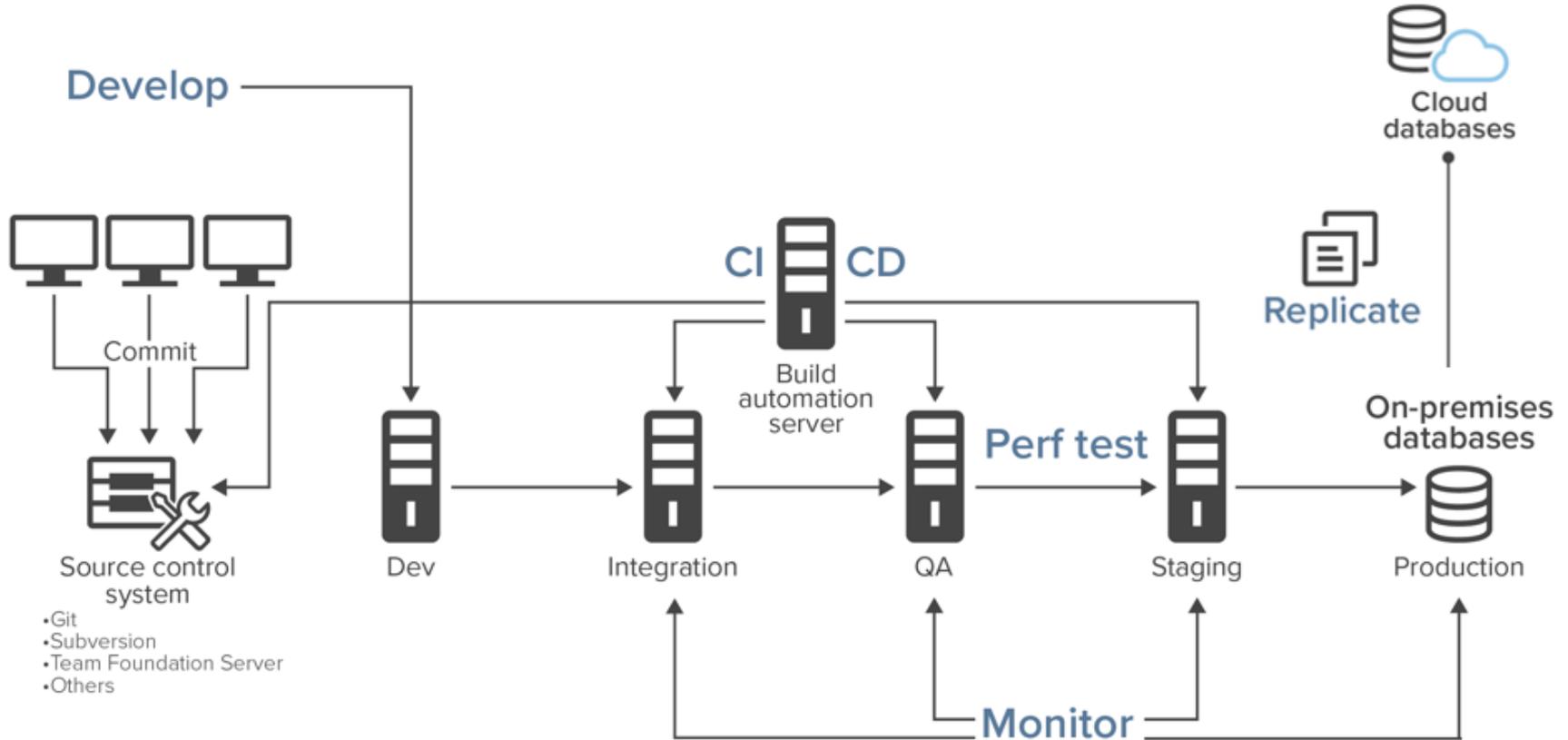


Toad DevOps Toolkit – feature scope

- PL/SQL Unit Tests
- Code Reviews
- Database compare
- Schema compare
- Data compare
- Script execution



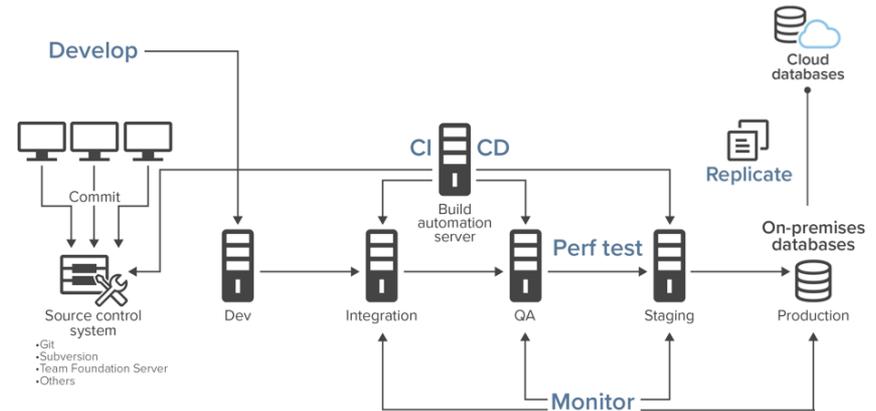
Typical IT Infrastructure



Typical Database Management DevOps Use Case

- The production DBA discovers a consistent month-end performance problem on various critical databases.
- Operations needs to quickly identify the root cause and, if necessary, have development implement any necessary changes.
- Development needs to include any SQL and code changes into their next sprint and also ensure data privacy is not compromised.

- Any SQL and procedural code changes need to be performance tested before releasing into production to ensure scalability.
- On-prem changes need to be replicated quickly to cloud databases.



Performance Monitoring

- A baseline deviation alert informs the DBA of a performance problem with an on-premises production Oracle 12c EE database supporting an Oracle eBiz application used by finance.
- Long-term history shows it's a recurring problem at month-end.
- The problem is a poorly executing SQL statement, but it's part of a procedural code program, so a request is made to development to fix and stage the schema changes for delivery.

The screenshot shows the Oracle Enterprise Manager Performance Monitoring console. At the top, there are summary cards for various database types: All Instances (22), SQL Server (6), SQL Server BI (5), Oracle (5), Sybase (1), DB2 (2), MySQL (1), and PostgreSQL (1). Below this is a table of all instances. The instance 'alvscdw24-ORAPROD12CDB' is highlighted with a red box, indicating a performance issue. The table columns include Sev, Name, Instance, Version, Up Since, and Workload.

Sev	Name	Instance	Version	Up Since	Workload
●	ALVSCDW05-SQL2008	SQL Server	12.0.5000.0	06/04/17 18:44	3.0
●	alvscdw18-ORAPROD	ORACLE Instance	11.2.0.1.0	—	3.0
●	ALVSCDW76-SQL2016	SQL Server	13.0.1601.5	06/04/17 18:46	2.0
●	Sybase_MDA@alvscdw94.qscprod.domain.corp	Sybase	ASE 15.5 EBF 18157 SMP E5D#2	06/04/17 18:43	2.0
●	ALVSCDW07-SQL2008	SQL Server	12.0.5000.0	09/16/17 19:15	0.0
●	STCHYPSQLW1201	SQL Server	13.0.1601.5	02/06/17 11:10	0.0
●	ALVSCDW08-SQLPROD1	SQL Server	11.0.2100.60	09/16/17 03:21	0.0
●	alvscdw93-DB2	DB2	10.5.500.107	06/04/17 18:47	0.0
●	PostgreSQLAgent@alvscpgsqlv01.qscprod.domain.corp	PostgreSQL	9.5.3	06/04/17 18:48	0.0
●	MySQLAgent@stchymysql01	MySQL	5.7.15	09/08/16 15:38	0.0
●	alvscmgdbw01.qscprod.domain.corp	Microsoft SQL Server	3.4.2	06/04/17 18:52	0.0
●	RAC-rac11vm-RAC11GA	ORACLE RAC	11.1.0.6.0	07/19/17 08:10	0.0
●	alvscdw25-ot12102	ORACLE Instance	12.1.0.2.0	06/04/17 18:47	0.0
●	alvscdw24-ORAPROD12CDB	ORACLE Instance	12.1.0.2.0	06/04/17 18:47	0.0
●	alvscdw24-ORAPROD12C	ORACLE Instance	12.1.0.2.0	06/04/17 18:47	0.0
●	ALVSCDW09-SQLPROD2	SQL Server	11.0.2100.60	09/16/17 03:18	0.0
●	ALVSCDW07-SQL2008	SQL Server	12.0.5000.0	Instance Down	—
●	ALVSCDW07-SQL2008-SSISDB	Analysis Services	12.0.5000.0	09/16/17 19:15	—
●	ALVSCDW08-SQLPROD1-ReportServer\$SQLPROD1	Reporting Services	11.0.2100.60	—	—
●	ALVSCDW05-SQL2008-ReportServer\$SQL2008	Reporting Services	12.0.5000.0	06/27/17 10:19	—
●	ALVSCDW07-SQL2008-ReportServer\$SQL2008	Reporting Services	12.0.5000.0	08/18/17 16:28	—

Identification of Performance Problem Using History

Performance Tree | Tops: 25

- Instance View
- SQL Statements
 - SELECT COUNT(*) FROM EMPLOYEEX WHERE EMP_DEPT IN (SELECT NVL(DPT_... WHERE DPT_NAME NOT LIKE %C% GROUP BY NVL(DPT_ID, DPT_ID))
 - SELECT COUNT(*) FROM EMPLOYEEX WHERE EMP_DEPT IN (SELECT NVL(DPT_... WHERE DPT_NAME NOT LIKE %C% GROUP BY NVL(DPT_ID, DPT_ID))
 - PL/SQL Blocks
 - QSR.exe
- DB Users
- Machines
- Actions
- Modules
- Client Info
- Command Types
- Services
- Consumer Groups
- Sessions
- Client PIDs
- Locked Objects
- Files
- Objects I/O
 - QUEST_PERF.EMPLOYEEX

History | Change Tracking

Resource Consumption | Baseline | Breakdown

Resource Breakdown

Resource	Percentage
CPU Usage	43.2%
Lock Wait	26.7%
I/O Wait	12.7%
Other Wait	7.2%

Active Time

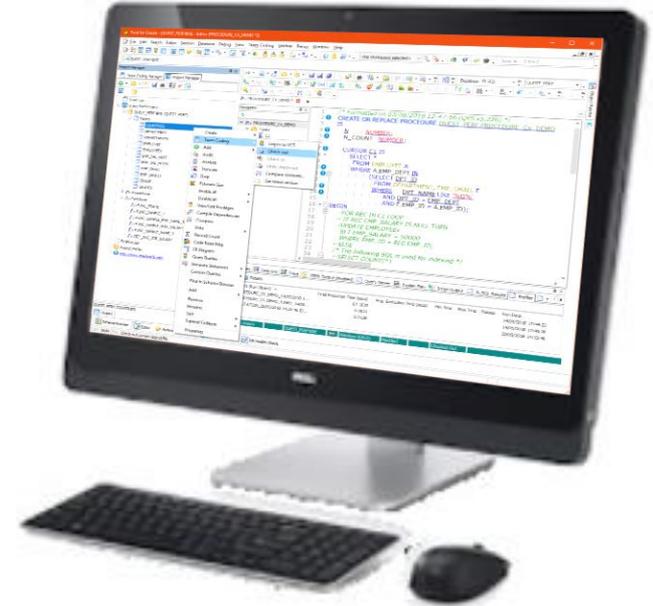
Top SQL Statements

Select Metric | View SQL Text | Analyze Plan | Tune SQL | Compare | Drill to SQL Statement | Export to CSV

SQL ID	Active Time	Active Time Percent	Average SQL
05vccz1dk098v	3,313.04	94.82	
fd6ksrzbu0hk	162.93	4.66	
9a9jmbx5a5r6	5.46	0.16	
9nv9k94w0ph1	2.16	0.06	
308kz67zjg725	1.62	0.05	
2snxp0r81cus6	1.33	0.04	
64caw4s3b2w7s	1.31	0.04	
6sq8qq41vjytm	1.25	0.04	
dxgny5t5fy3tx	1.20	0.03	

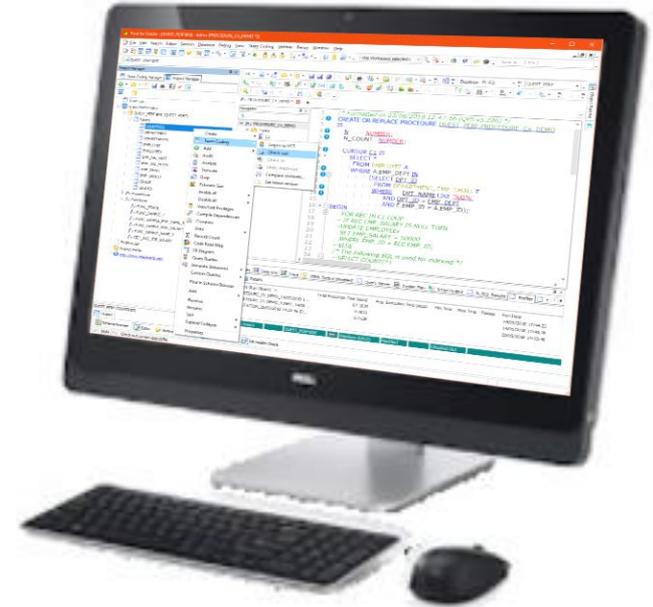
Database Development - 1

- A developer checks-out the detected procedural code program from source control and isolates the affected SQL statement via code profiling.
- The developer optimizes the execution of the SQL statement using automatically generated re-writes, finding a better alternative within seconds.
- Using the new, optimised SQL statement, the procedural code changes are regression tested using pre-saved unit tests and reviewed to ensure adherence to coding standards.



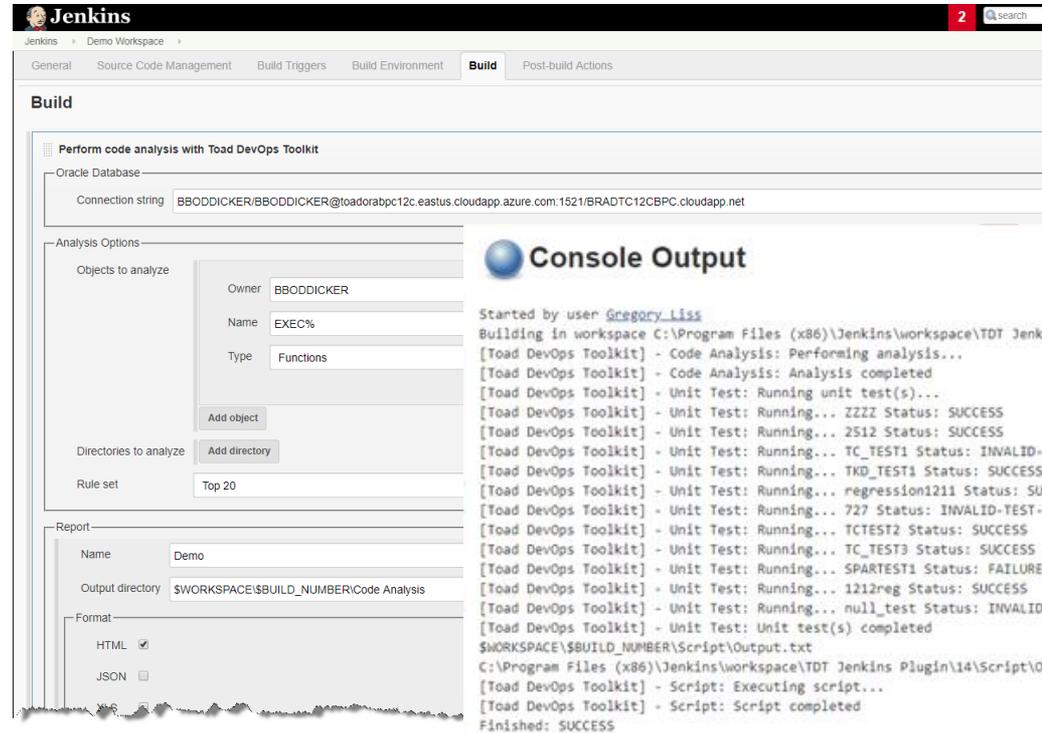
Database Development - 2

- Using pre-defined rules, the developer or DevOps engineer scans the database schema in order to identify personal or sensitive data and ensures it's properly protected before deployment into production.
- The developer will also be notified if a procedural code change exposes personal or sensitive data.
- The developer checks-in the procedural code program and associated unit tests to source control to ensure they're included in the next automated build cycle.



Automating Build and Release of Database Changes

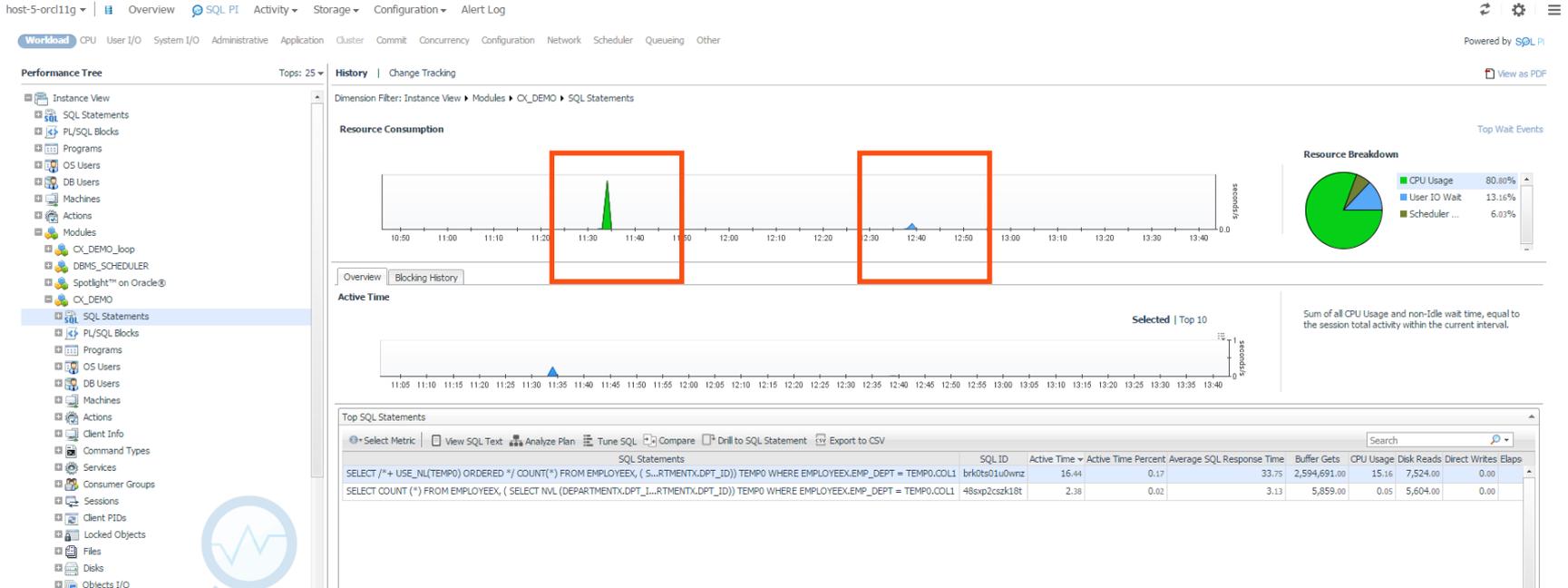
- Using Jenkins or other build server, procedural code unit tests and code quality reviews are automatically executed against code checked into source control.
- Pass/Fail notifications help to determine whether build is ready to be deployed.
- Deployment scripts based on source/target comparison are used to automatically promote the DDL changes into target DB.



The screenshot shows the Jenkins interface for a build job named "Perform code analysis with Toad DevOps Toolkit". The job is currently in the "Build" phase. The configuration includes an Oracle Database connection string, analysis options (Owner: BBODDICKER, Name: EXEC%, Type: Functions), and a report format (HTML checked, JSON unchecked). The console output shows the build process, including code analysis and unit tests, which completed successfully.

```
Started by user Gregory Liss
Building in workspace C:\Program Files (x86)\Jenkins\workspace\TDT Jenkins
[Toad DevOps Toolkit] - Code Analysis: Performing analysis...
[Toad DevOps Toolkit] - Code Analysis: Analysis completed
[Toad DevOps Toolkit] - Unit Test: Running unit test(s)...
[Toad DevOps Toolkit] - Unit Test: Running... ZZZZ Status: SUCCESS
[Toad DevOps Toolkit] - Unit Test: Running... 2512 Status: SUCCESS
[Toad DevOps Toolkit] - Unit Test: Running... TC_TEST1 Status: INVALID-TEST
[Toad DevOps Toolkit] - Unit Test: Running... TKD_TEST1 Status: SUCCESS
[Toad DevOps Toolkit] - Unit Test: Running... regression1211 Status: SU
[Toad DevOps Toolkit] - Unit Test: Running... 727 Status: INVALID-TEST
[Toad DevOps Toolkit] - Unit Test: Running... TC_TEST2 Status: SUCCESS
[Toad DevOps Toolkit] - Unit Test: Running... TC_TEST3 Status: SUCCESS
[Toad DevOps Toolkit] - Unit Test: Running... SPARTEST1 Status: FAILURE
[Toad DevOps Toolkit] - Unit Test: Running... 1212reg Status: SUCCESS
[Toad DevOps Toolkit] - Unit Test: Running... null_test Status: INVALID
[Toad DevOps Toolkit] - Unit Test: Unit test(s) completed
$WORKSPACE\BUILD_NUMBER\Script\Output.txt
C:\Program Files (x86)\Jenkins\workspace\TDT Jenkins Plugin\14\Script\O
[Toad DevOps Toolkit] - Script: Executing script...
[Toad DevOps Toolkit] - Script: Script completed
Finished: SUCCESS
```


Review of Database Changes in Production



Achieving Continuous Database Operations with Quest's Database Solutions

Foglight



Foglight[®]

Toad for
Oracle

Toad[®] Developer
Edition
for Oracle



*With Sensitive Data
Protection*

Toad DevOps
Toolkit

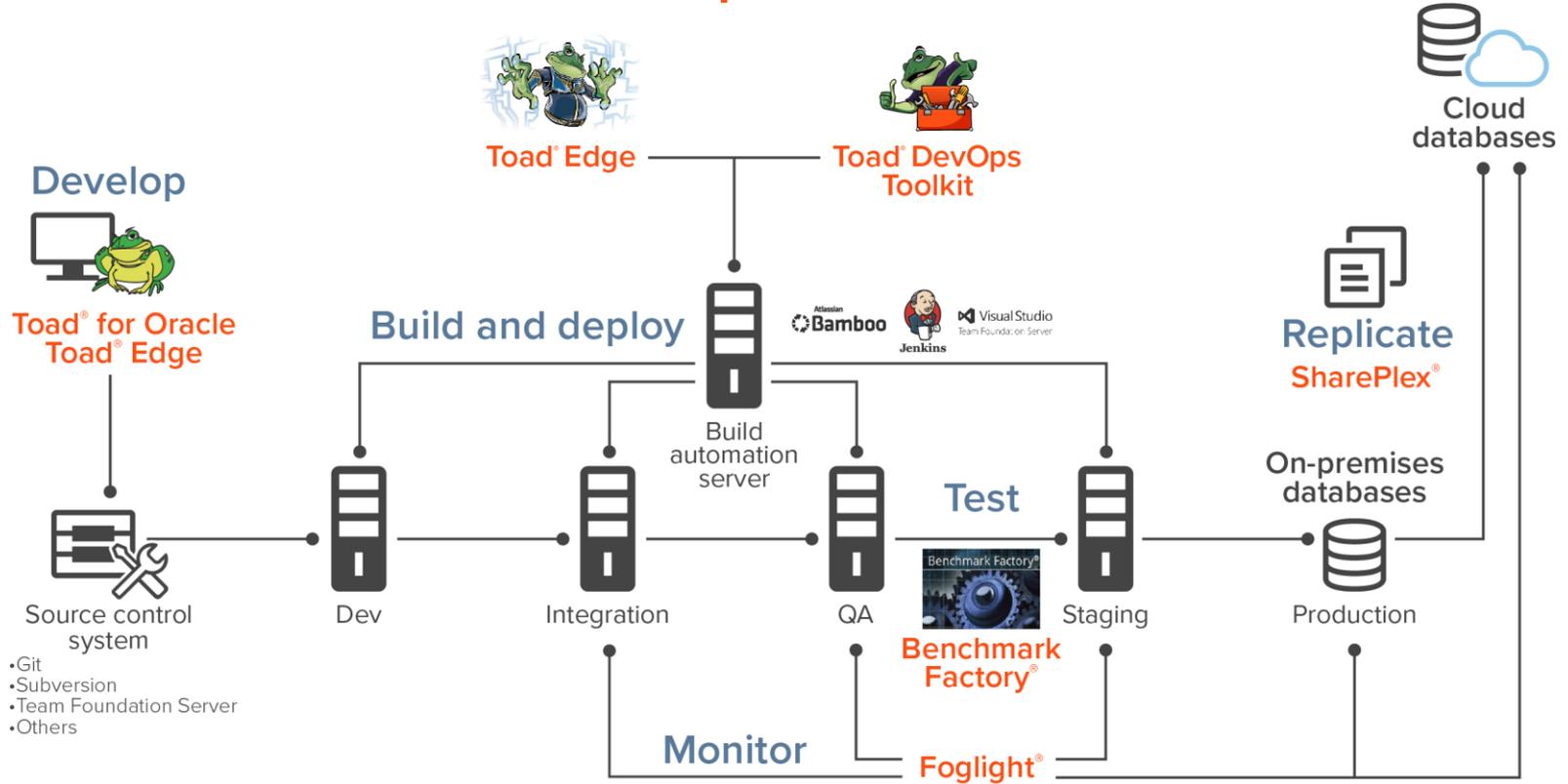


SharePlex



SharePlex[®]

Quest's Database DevOps Solution



Resources

- <https://www.quest.com/casestudy/major-financial-firm-advances-devops-with-trusted-database-tools8133654/>
- Toad for Oracle trial download: <https://www.quest.com/products/toad-for-oracle/software-downloads.aspx>
- Toad DevOps Toolkit trial download: <https://www.quest.com/register/113064/>
- Quest Database DevOps Solutions: <https://www.quest.com/solutions/devops/>

Major financial firm advances DevOps with trusted database tools

A Fortune 500 company used Toad® and Toad DevOps Toolkit to integrate database development into their new DevOps initiative.



"It's a challenge ... because we have all these databases and the data is so pristine to us. It's our Holy Grail."

Senior manager, database management, major financial company

Quest

CUSTOMER PROFILE

Company	Major Financial Company
Industry	Investment and Insurance
Country	United States
Employees	15,000

BUSINESS NEED

Senior executives committed to Agile development and DevOps to spur innovation. But database professionals needed proof that integrating the company's vital databases into the workflow would not lead to data loss.

SOLUTION

Using Toad® and Toad DevOps Toolkit, the company successfully completed a year-long proof of concept integrating their database development into the DevOps workflow. Database developers rely on Toad for easy unit testing and code analysis to meet quality control standards. The company will now initiate a pilot project that uses Toad and Toad DevOps Toolkit to automate code quality and testing.

BENEFITS

- Better quality code, based on code analysis evaluations
- Faster testing, using a repository of reusable code test cases
- Anticipated improvements to software release velocity in the DevOps pilot, based on Toad DevOps Toolkit automation

SOLUTIONS AT A GLANCE

- Database management



**MANAGE
MONITOR
MOVE**

Thanks for listening!

eero.mattila@quest.com