

EVOLVE™ VERIFICATION MANAGEMENT

Engineering teams regularly go on 4-hour coffee breaks when their code base is broken. Engineers face a bewildering repository of poorly described tests when debugging or changing code. Leveraging legacy tests into new systems and tracking progress is difficult. Managers have trouble estimating when the chip will be done and have poor confidence in their estimates. EVOLVE™ solves these problems. Only EVOLVE™ guarantees individually quality-controlled releases. Tests and regressions are centrally controlled, well described, and uniformly invoked. EVOLVE™ presents transparent statistics and reporting to management. The result: Coffee consumed directly fuels a working piece of silicon.

VERSION CONTROL MADE EASY – NO MORE BROKEN HEAD-OF-TREE

Why do engineering teams regularly go on four hour coffee breaks when their code base is broken? When using the release methodology known as “Head of Tree”, the latest commit is always presumed to be good. This policy only works when engineers are meticulous with their commits and verify their work. As teams grow in size, the probability of a mistake increases until there are multiple daily “coffee breaks”.

Head of Tree has several other problems. Merging a given user’s workspace with the head of tree can be difficult. Two engineers trying to release a common feature (with test and design) cannot commit their individual changes. Engineers cannot snapshot their internal development efforts. In one customer's case, this required a metal spin due to incorrectly self managed files.

Modern software methods (Agile) address these problems. Unfortunately, automated Java or Android build systems designed for software will not help your chip design effort.

EVOLVE™ solves these problems by layering a modern release server on top of the existing revision control system. The entire system is tuned specifically to chip design, blending the best practices from software and hardware efforts.

Eliminate the fear of commits; commit early and commit often: The release server will manage the rest. Only when a release is requested will the code go to a clean test-space and a quality control regression run against it. Only good, working releases are available to the team.

The result? No more coffee breaks due to broken code bases. No more copying files across workspaces. No more lost days of work when an engineer deletes the wrong file. No more

wondering what files go with what changeset. Releases have an annotated, commented searchable history. No more wondering who broke the repository. Every release is attributed to a user, and verified.

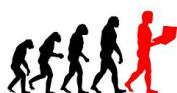
REGRESSION AND TEST TRACKING – NO MORE TOILING AT DEBUG

As chip complexity grows, so does test and regression complexity. No single engineer understands every test, and there is no central location that describes what a test does or how it works. This lack of structure overburdens weekly test efforts. Real test failures cannot be discerned from false failures due to IT infrastructure issues. Failures repeat themselves. The result: 1-2 engineers each week try to triage, debug, and file functional bugs on the code base. It starts all over again the next week; a Sisyphean task.

Schedules become difficult to maintain. The lack of a “clean” regression crushes engineering morale. The fear of breaking something worse can be crippling to a team, even though that team has validation tests available. This lack of understanding delays the validation of critical features.

EVOLVE™ overcomes these problems by ensuring that:

- 1) The release server divides failures due to individual code submissions into manageable chunks. These failures are resolved by the engineer that introduced them.
- 2) Tests and regressions are stored in a searchable central database. A uniform, central repository with this metadata simplifies running tests – just launch the named test! Even with a uniform interface, EVO provides multiple levels of flexibility on the test. And the wealth of meta-data makes it easy for anyone to drill down and debug a test.
- 3) Tests and regressions have a full repeatable history.



4) Tests and regressions can be automatically rerun with debug information turned on. This eliminates the need to have an engineer make a first pass analysis.

5) Tests and regressions can be automatically “binned” or triaged according to a failure signature, allowing the debug team to focus on the largest (or oldest) failures first.

6) I.T. infrastructure failures can be automatically filtered and resubmitted. During this process EVOLVE™ gathers statistics to submit to I.T. so that the underlying hardware problem can be resolved.

TRANSPARENT VIEWS OF PROJECT STATUS: REPORTING EVOLVED

An Engineer needs to answer several debug related questions. *Why did this test suddenly start failing? When did this test last succeed? Why did this release fail? When was this file last touched and thus cause this failure?*

An Engineering Manager needs to answer project related questions. *How complete is the test effort? How many bugs were introduced/solved this week? What is the closure rate over time? How much longer will the chip design process take based on that closure rate? What is the confidence in the numbers generated by the engineering team? How close to “done” is this design?*

Most chip design efforts today fail to ask (let alone answer) these questions because the tools are not in place collect the necessary data. There is no history of tests. There is no automatically generated relationship between tests and bugs. Project status is often derived from qualitative emails sent by members of the engineering team.

EVOLVE™ addresses these issues by retaining a full history of test and regression statistics. Built-in coverage, test planning, and releases are linked to these statistics. Combine this with search and reporting mechanisms, and engineers can now rapidly answer their debug related questions.

EVOLVE™ includes an Engineering Manager’s dashboard to improve both time to market and product quality. By knowing a project’s quantitative past, present, and projected future, EVOLVE™ can assist decision makers. Reports may be generated at a high level or in great low level detail.

Still need more? Custom reports may utilize the simple hooks that EVOLVE™ provides.

TRANSITION THE TEAM FROM LEGACY TO STATE OF THE ART

Many teams are struggling to transition to coverage-based verification techniques. Directed tests still play a large role in legacy test strategies. They still have value in simple, focused debug for designers. As functional and random-based techniques are increasingly used, the problem becomes one of both integration and data management. How do teams integrate existing test suites and complement them with random efforts? How do teams manage the results from these billions of random vectors to give meaningful data?

EVOLVE™ solves this complex problem with simple, intuitive coverage and test planning interfaces. The test planning tool allows engineers and managers to express major areas to be tested. These efforts may be through directed, random, or coverage based efforts, or a blend of all three. Combined with releases, EVOLVE™ provides a solid, enforceable, reportable methodology. No other toolset has this capability.

Since EVOLVE™ maintains a full history across every user, it is simple to report on coverage numbers in the past week, month, or lifespan of the project. Groups can subdivide tasks and EVOLVE™ brings the results together.

Confidence in schedules and quality improves with EVOLVE™. *There is 97% coverage “today” on the feature freeze day, but is that consistent over the last 5 weeks? How many exceptions are there? How much has the design changed in the last 5 weeks? Is it smarter to delay 2 more weeks for stability?* EVOLVE™ provides quantitative answers to these questions and the ability to drill down into the details.

For all project phases, EVOLVE™ ensures maximum confidence and coverage for any level of design complexity. EVOLVE™ minimizes time-to-market and release risk with the ability to serve the largest engineering efforts. EVOLVE™ provides regression, test tracking, and transparent views of project status. Superior designs are driven by the adoption of the best practices and the best tools available. Now is the time to EVOLVE™.

Contact Missing Link Tools at sales@missinglinktools.com or call 720-246-1139 for more information.

