



benchmarkQA

REUSE
RECYCLE
RENEW



your Automated Testing

Larry Green

'GREEN' AUTOMATION

WHEN YOU THINK OF AUTOMATION..?

- ◎ Utility
 - ◎ Applicability, Value, Cost?
- ◎ Places and People
 - ◎ GUI, Non-GUI?
 - ◎ Functional test, performance test, unit test, static analysis ?
 - ◎ Just for <name the group/person>?
- ◎ When
 - ◎ Next project? Yesterday?



(OVERLY) GREAT EXPECTATIONS

- ⊙ Silver bullet
- ⊙ Trivial implementation and maintenance
- ⊙ Minimal skills to utilize
- ⊙ Short development, late in design cycle
- ⊙ It's not code
- ⊙ Finds all bugs
- ⊙ Replaces all manual tests
- ⊙ Any tool will do



- ⊙ Results:
 - 'Late' automation
 - Underutilized, unmaintained and abandoned automation
 - Shelf ware for pricey tools

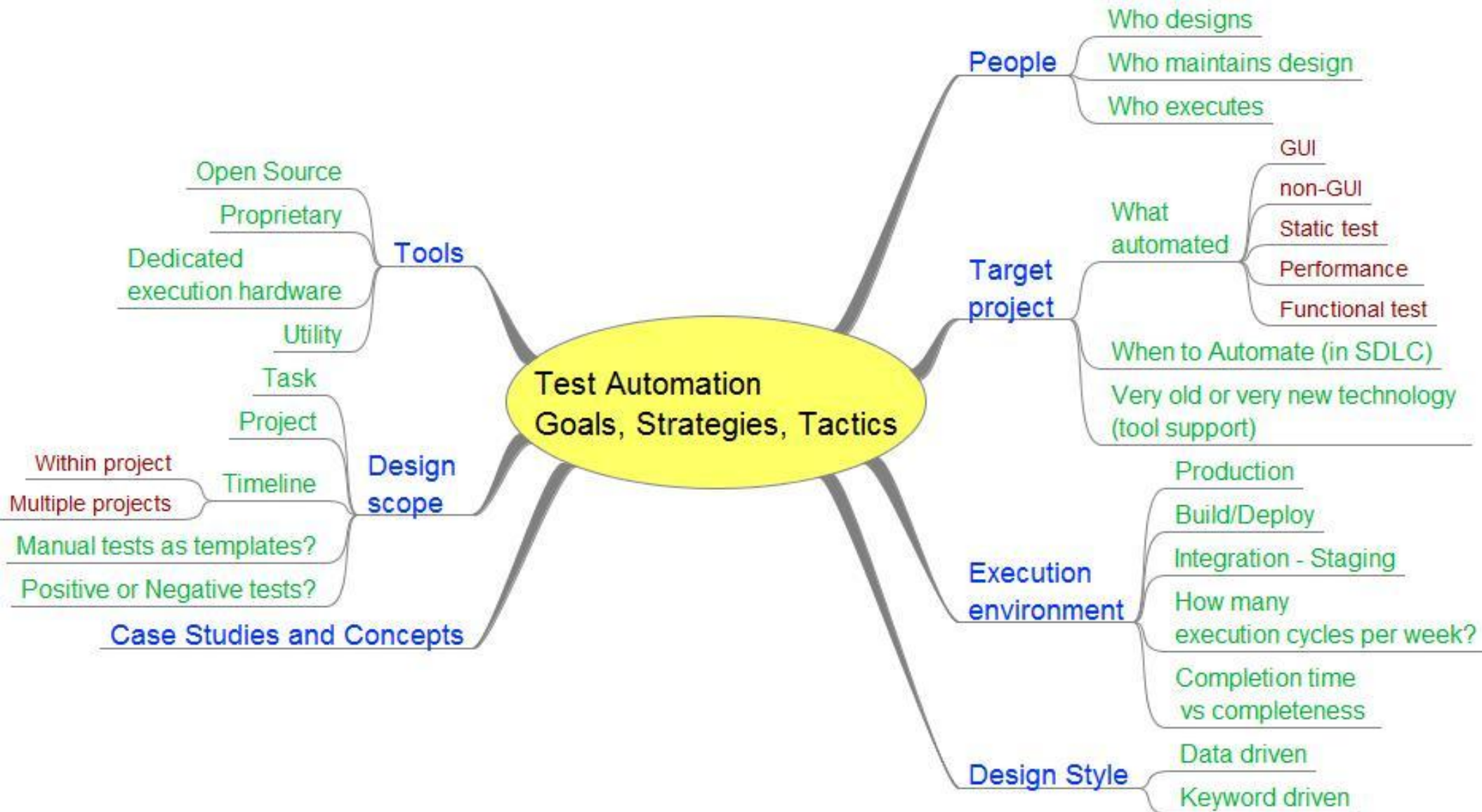


PREFERRED EXPECTATIONS

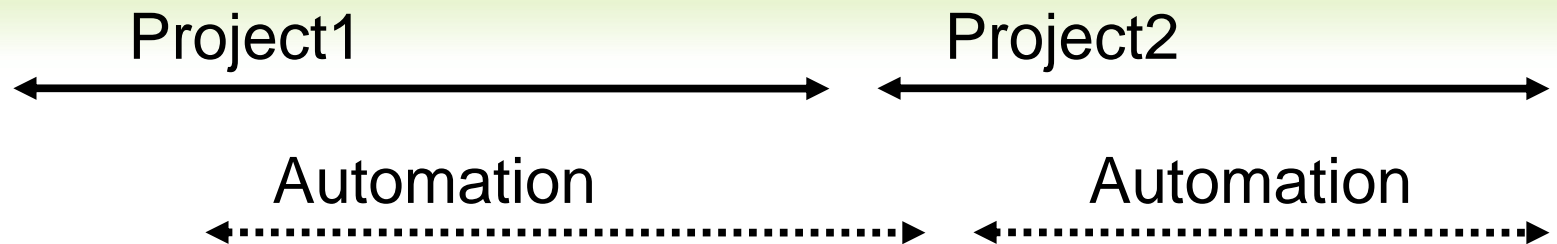
- ◎ Target for reuse, payback
- ◎ It's Code
 - ◎ Part of project, managed source tree, Pass/ Fail evaluates target to known reference.
- ◎ Common usage, functional regression test.
- ◎ Measure what automation finds and provides.
 - ◎ # of defects, broken builds etc.
 - ◎ Developer/Tester utilization improvements



AUTOMATION MIND MAP



AUTOMATION PROJECT SCOPE AND TIMELINES



- ⊙ Difficult to synchronize into development
 - ⊙ Discover /Recover cycle post build
- ⊙ Regression (positive) tests faster, simpler than negative tests.
 - ⊙ Observability/Controllability/Cleanup/Restore



CASE STUDY: MANUAL BEATS AUTO

- ◎ Goal: Test 12 countries ~75 products/country, internationalization of enrollment/fulfillment/reporting web site.
- ◎ Option1; manual test (many weeks per execution cycle, tester burnout).
- ◎ Option2; replace manual tests with GUI automation (3-5 minutes per product x ~1000 products → ~1.5+ weeks).
- ◎ Option3; manual test SQL query and text compare to validate all 1000 products (~ 3 minutes) plus data driven automated tests for each county (~ 1 hour).



CASE STUDY: WIDGET AUTOMATION

- ◎ Multi-platform API Automation
- ◎ Master calling script, function library scripts for each component.
- ◎ Libraries reusable for web pages that incorporate components. 'Fine grain' extensible automation.

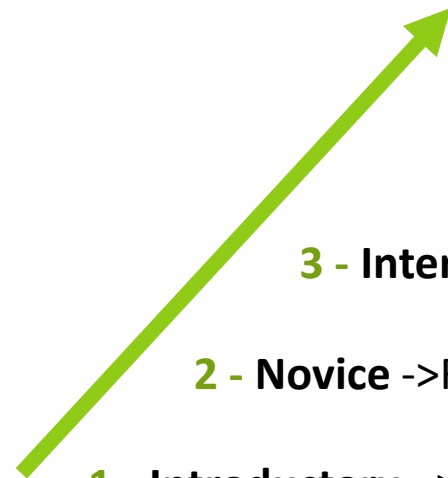


CONTINUOUS INTEGRATION MATURITY MODEL'S TEST LEVELS

CI Actions

BUILD → DEPLOY → TEST → REPORT

CI Test Maturity Levels



4 - Advanced ->Extensive Automated Function Tests,
Risk-Based Manual Tests, Security Scans

3 - Intermediate ->Static Analysis , Automated Functional Tests

2 - Novice ->Regression Tests Run with Build

1 - Introductory ->Some Test Automation

http://www.anthillpro.com/html/resources/elements_of_ci_USLetter.pdf

CASE STUDY: AUTOMATED BUILD VERIFICATION TEST AND MANUAL TESTING WITH SAME SCRIPTS

- ◎ Solution: Build a suite of standalone automation libraries and calling program.
- ◎ Manual testing usage: Data driven, more exhaustive test data than build verification test, document test, more quickly analyze test results.
- ◎ Automation library script examples:
 - ◎ Discover and document test execution environment (read windows registry for build and setup info), write test run results into directory tree.
 - ◎ Read Windows event logs and filter them for 'error' events within the timeframe of the test execution.
 - ◎ Read task manager for selected process ID values (Changes during test indicate code crash and recovery).
 - ◎ Read application's log file for 'error' events within the timeframe of the test execution.



OPEN SOURCE AND PROPRIETARY TOOLS CONSIDERATIONS

- ◎ Known
 - ◎ 'Buzz' or Advertising
- ◎ Functional
 - ◎ Useful for your application
 - ◎ IDE/ Development support
 - ◎ Regular language
- ◎ Available
 - ◎ 'Popular' and 'Active' community or profitable company with support
- ◎ Affordable



CASE STUDY: OPEN SOURCE TRADEOFFS

Implement GUI automation using multiple open source tools with blended libraries; Pluses and Minuses.

1. + Low tool acquisition cost, + no per seat restrictions on who can use tool, + no maintenance contracts, + management support.
2. + Adjust tool to fit. - Pain as you go plan (tool instability).
3. - Tool set API inconsistencies, and tool function discovery difficulties. - Minimal IDE capability. - Silent false test pass results possible (tool's API lacked exception handling robustness).
4. - Product's GUI widget discovery difficulties. - Slow test execution.



TOOLS WISHLIST

- ◎ Robust development environment
 - ◎ Integration to version control tools, Consistent, well documented API , Robust IDE support (Text editor, debugger, API helper), Common language, Legible SQL execution, Readily discover and manipulate App's objects, Fast Execution
- ◎ Open, active development community
 - ◎ Code snippets published, solutions published, libraries published (all at zero or low acquisition cost).



CONCEPTS (IN NO PARTICULAR ORDER)

- ◎ Target maximum code execution and reuse: Build verification, Support/aid productivity for manual testing. Data driven scripts. Production support.
- ◎ Its a code development project. Check in scripts to CM tool. Perform code reviews. Practice DRY (don't repeat yourself) and make functions instead.
- ◎ Reduce test complexity before automation (Just because automation executes faster [than some manual test] does not guarantee value in adding more tests.) Number of tests executed is likely a meaningless metric unless tests structured by boundary values, equivalence portioning etc.
- ◎ Keyword driven; domain specific micro language test tactic, not a strategy.
- ◎ Design manual tests for automation capability.



MORE CONCEPTS

- ⊙ Rather than directly translate manual tests into automation. Refactor to better fit automation's strengths.
- ⊙ Automation strengths are internal repetitions (e.g. data driven code, functions, code blocks) and external repetition (broadly reusable) and site navigation.
- ⊙ Design automated tests to support manual test execution. Let tools discover and document test execution, and gather results.
- ⊙ Match tool to needs. Be wary of automation tool capabilities for bleeding edge or trailing edge projects.
- ⊙ Consider automation as a technique within continuous integration scope.
- ⊙ Google knows all (about your site), why not you? Consider GUI automation as a widget and navigation discovery tool.



QUESTIONS



BenchmarkQA

952.392.2400

benchmarkQA.com

QA IS ALL WE DO!

BenchmarkQA helps project teams deliver higher quality software through:

- Quality Assurance Consulting
- Contract Staffing
- Training



benchmarkQA