

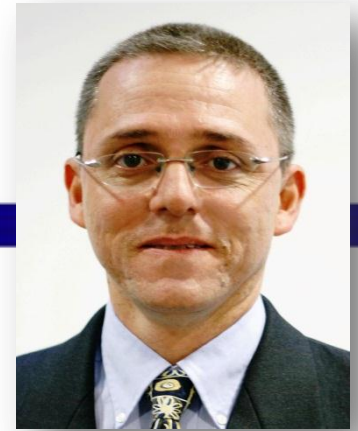
Optimizing Your Regression Tests

The Best are **Getting Better**

Alon Linetzki

2

- Quality, Testing and product assurance coach and trainer – in the last 28 years, with 10 years in development
- Vast experience in SW and embedded systems
- Trained > 1000 professionals and coached in over 20 companies
- Keen on people, how we think, how well we work, how we can do better...
- Established the Israeli branch of ISTQB® - ITCB (2004), serve as VP and marketing director, lead the ISTQB® Partner Program w/w
- Established the SIGiST Israel – Israeli Testing Forum (2000)
- Train and coach internationally since 1995



BestTesting
The best are getting better

We shall cover...

3

- **Definition and motivation**
 - ▣ What is Regression Testing
 - ▣ The motivation for performing regression testing
- **Major Factors of Regression**
 - ▣ Present major factors influencing regression testing
 - ▣ How much effort is invested in regression tests?
- **Diversified approaches & Selection Criteria**
 - ▣ Present different approaches for regression testing, and their compelling event
 - ▣ Creating the selection criteria
- **Discuss regression test strategy and optimization**
- **Summary**

What is Regression Testing?

4

- “...testing that is performed after making a **functional improvement or repair to the program**. Its purpose is to determine if the change has regressed other aspects of the program” (Myers, 1979)
- “...any repetition of tests (usually after software or data change) intended to show that the **software's behavior is unchanged** except insofar as required by change to the software or data” (B. Beizer, 1990)
- “... selective retesting of a system or a component to verify that **modifications have not caused unintended effects** and that the **system still complies with its specified requirements**” (IEEE 610)
- “The intent of regression testing is to ensure that *a change, such as a bug-fix, did not introduce new faults*” (Wikipedia)

What is Regression Testing?

5

- Regression Impact might occur when:
 - ▣ Bug Fixes are introduced
 - ▣ Enhancements are introduced

- When do we find regression bugs?
 - ▣ Engineering & development missed an impact caused by the changes that were introduced

Motivation

6



- ▣ Each new release of the system could cause existing features to Fail,
- ▣ Clients are depending on the fact that the current installed product or system will work properly and safely,
- ▣ A Failure in the new system, would lower the confidence of clients and would harm sales and revenue,



- ▣ Until the end of a release, or big cycle, regression tests are done to make sure that no defects are introduced after bug-fixes and code enhancements. Such defects can degrade the system quality.

5 Regression Major Factors

7

- Scope of the enhancements within a release
- Resources and effort needed to develop & run tests
- Estimated Coverage of current regression pool
- Test Case management approach/tool used
- Regression approach used:
Manual/automated



How much effort is invested?

8

- Some industry companies indicate an average of **60-80%** of the test execution is focused on regression tests,
- The absolute average number from **total project budget** is estimated to be **8-10%**

9

Optimizing your Regression tests

Process Assumptions



Assumption - 1

10

- ❑ Most of the Regression test cases are focused on testing the software in the way it was designed to work
- ❑ Other types of tests, should have been tested on previous releases

A good ST execution phase, according to some industry statistics, may reach **only 25%** code coverage

Assumption - 2

11

- Test cases of each release, are designed to cover most (if not all) of the product/system functionality that is at risk
- Non-functional testing is conducted as well to the maximize risk coverage
- Hence; Regression pool of test cases has the potential of covering all that critical parts of the product

Assumption - 3

12

- Regression pool of test cases has minimal scenarios redundancy
- It is assumed being kept across past versions that went into production

Average redundancy of test cases within the regression tests might be **15-25%** and more...

Assumption - 4

13

- Test cases are managed according to risks
 - ▣ ...and has a risk index attached to each test case, covering a certain area of function or non-functional capability of the system
- Hence, regression test cases, has a risk index attached as well

Risk Based Testing management, has a potential of reducing your test cases by up to **30%**, while improving your **DDP by 20%**

14

Optimizing your Regression tests

Regression Test Approaches



Regression test approaches

15

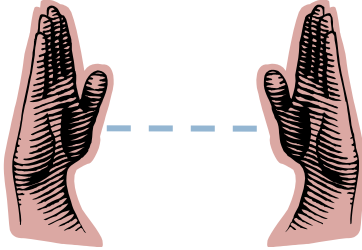
How much testing should be done:

- **Run all test cases from previous releases**
 - ▣ Applicable for some systems/products
 - ▣ Safety critical systems
- **Run most of the regression test cases pool**
 - ▣ High automation level/maturity
- **Run a sub-set from the regression test cases pool**
 - ▣ Very common approach
 - ▣ Derived from the risk based testing



Aspects to determine Regression

16



Size of the changes

- Number of modules
- Number of business processes



Past bug logs



Risk of the change i.e. in
system critical business
processes



Complexity of the change

Regression test approaches

17

Sub-conclusion:

There is no single, very good, regression test approach, but rather many fine Good Practices which are very much context dependant,

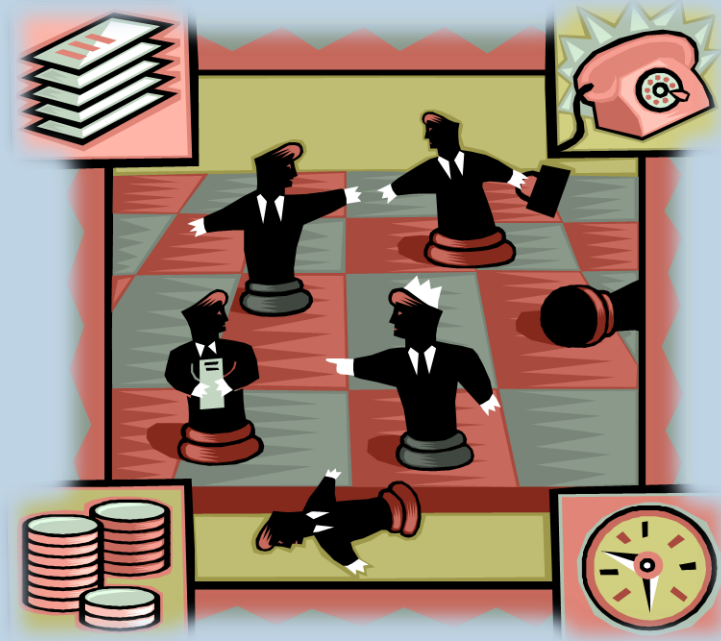
BUT...

what is common at the basis of their decision making process?

18

Optimizing your Regression tests

Regression Test Strategy



Regression test strategy



19

- Regression test strategy should consider:
 - ▣ Regression on any test level
 - Unit
 - Integration
 - System
 - Acceptance
 - ▣ Regression on any test type
 - Functional
 - Non-functional
 - ▣ Regression on any configuration/platform



Regression test strategy



20

- Regression strategy (selection criteria) should be based on 3 main knowledge items:

- ▣ Risk assessment results

Where are the impacted areas?

- ▣ Coverage information

Which areas of risk and functional coverage exist in my regression pool [relate to item 1]

- ▣ Past bug logs

Which areas had many errors in the past [and relate to item 2]

Regression test – Next Steps



- Identify risk assessment, coverage, and past bug logs - processes, artifacts, information gathered,
- Analyze with respect to the testing operations and product delivery, and define **selection criteria**,
- Pilot and evaluate
- Adjust, implement and measure...

Test Automation should have a separate analysis, and should be integrated into this action plan

Regression Point of View

22



- Things look different, depending on where you are...



Regression testing Optimization

Success factors – implementation process

- Include assumption's implementation as first step
- Define your 'regression selection criteria'
- Select the right TCs from the regression pool,
- Check your selection vs. risks and coverage,
- Review areas of regression,
- Check which can be run automatically, and which manually
- Prepare: impacted TCs, test suites & data, environments & configurations

Automation can get up to **60-80%** of the functionality,
and have a good ROI...



Summary

- Verify Assumptions are implemented – they are the real meat in this story...
- Measure your regression coverage –
 - ▣ [you may] Run Code coverage once for the whole Regression pool,
 - ▣ see how much you are covering (manual and automated)
 - ▣ do that for every new TCs/TA you write
- Invest in keeping Regression pool with minimal redundancy

A changing world...

25

“It is not the strongest of the species that survives, nor the most intelligent but the one that is **most responsive to change**”

Charles Darwin





THANK YOU!

Optimizing Your Regression Tests

The Best are **Getting Better**