28th Annual INCOSE international symposium
Washington, DC, USA
July 7 - 12, 2018

A Framework for Testability Analysis from a System Architecture Perspective

Mahmoud Efatmaneshnik,
Mike Ryan & Shraga Shoval

www.incose.org/symp2018

Capability Systems Centre

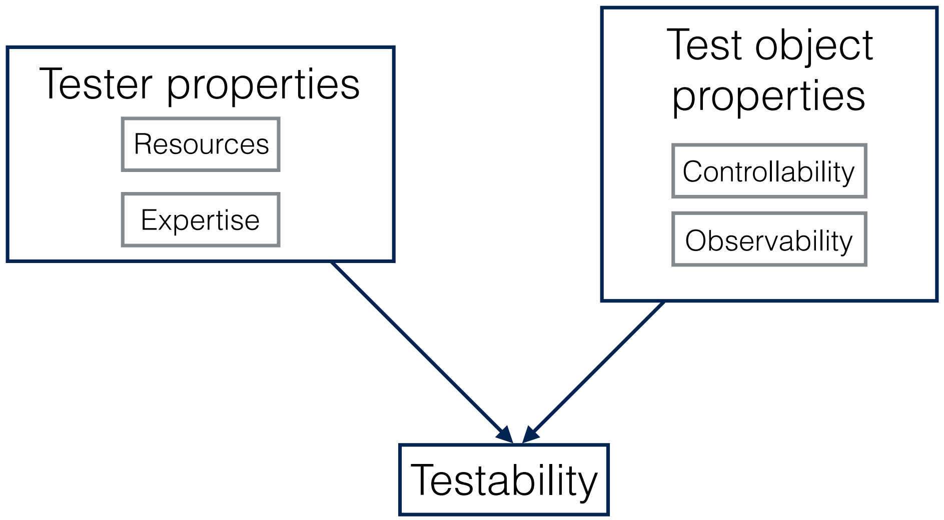
Aim

- We study the effect of test architecture on the efficiency and effectiveness of repetitive testing.
- We present a simple model of integration as a Markovian process and a number of simulations are used to estimate the expected number of the repetitive tests as well as the expected quality after test.
- We discuss the effects of test architecture on these two testability outcomes for several architectural settings.

Testability

- The degree to which a component or a system can be tested in isolation from other components or systems.
- Effort required for testing a system.
- The degree of effectiveness and efficiency with which test criteria can be established for a system.
- Testability can be a property of a requirement, a system, or any element of the system (assembly, subsystem, or component).

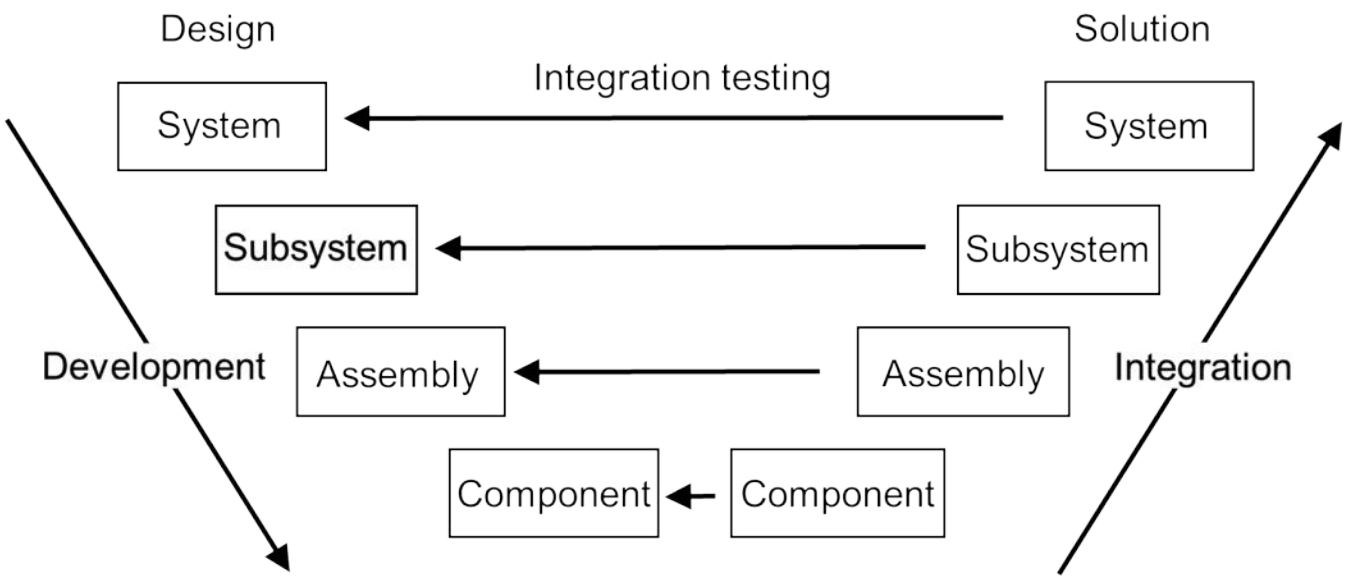
Design for Testability



- Testability is most commonly viewed as design for controllability and observability.
- However, tester properties (such as resources applied and the quality of the test) have a lot to do with testability.

Testing and the V-model

- A simple version of the SE V-model shows testing as components that are delivered, tested, integrated, tested, integrated, tested until the system is complete (and tested).

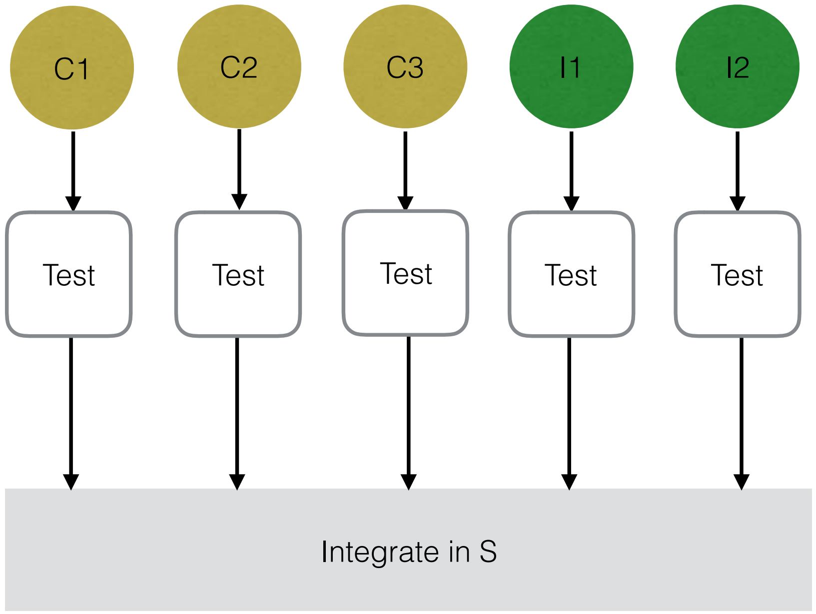


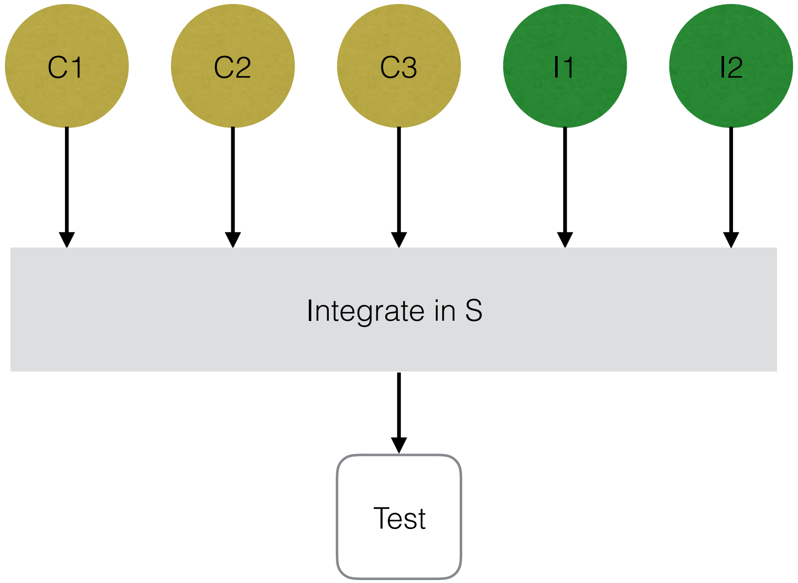
Testing and the V-model

- While being useful for visualisation of the SE process, using the V-Model as a test architecture may be costly and inefficient.
- For example, a unit test will require a custom test-bed (or stub-driver set in software case), and each unit may require a different test harness. For large and complex systems with many units and components, this could prove to be costly and problematic.
- It might be better to test a unit when connected to an actual assembly/subsystem or the system so that the subsystem/system itself is used as the test-bed.

Testing Architecture

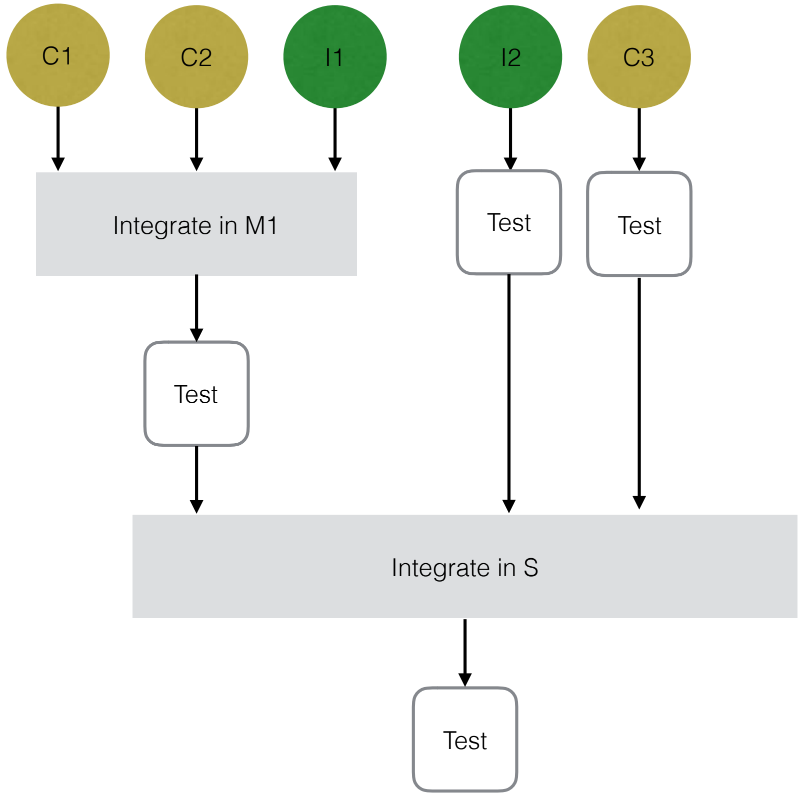
- So, we have a number of options for testing. Simplistically:





Testing Architecture

- Or, we could test as modules in many combinations.



Testing Architecture

- The advantages of module testing are:
  - *Efficiency:* testing can be focused on smaller or larger system elements.
  - *Effectiveness:* more effective in pinpointing and correcting faults, since, when a fault is found, it is known to exist in a particular module.
  - *Parallelism:* module testing introduces the opportunity to test multiple modules simultaneously, which can in turn reduce the required test time.
  - *Flexibility:* can be done at different times and locations and by different teams with heterogeneous expertise.
  - *Complexity Management:* increases traceability of faults, and allows for simpler test-bed setups.

Measures of Testability

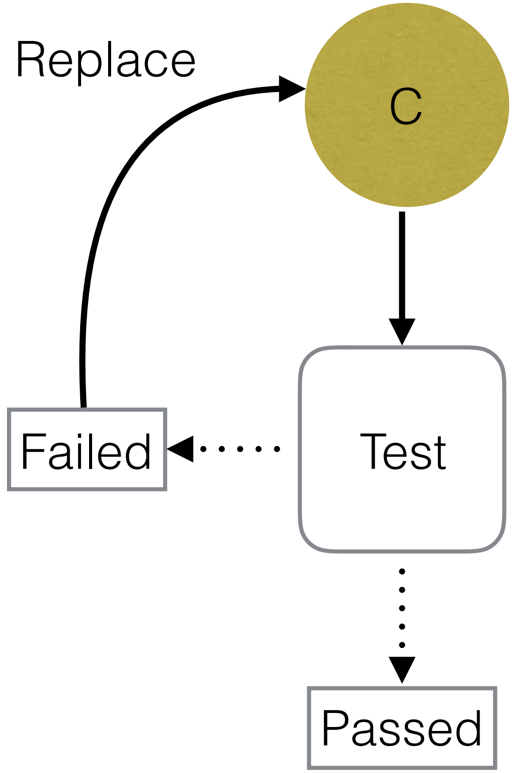
- Test Quality (TQ)
  - The average capacity of a test to identify a defect.
    $\mathrm{TQ} = Pr(Fault = Detected \mid Unit = Faulty)$
- Test Cost (TC)
  - Determined by the size of the tested system, TQ and the architecture of the testing. A low-cost test implies a high probability of test accuracy that facilitates quick identification of defects.
- System Quality after Test (SQaT)
  - Depends on TQ and System Quality before the Test

Testing Architecture

- The way in which system elements are tested as they are integrated (that is, the testing topology of components, modules, subsystems and systems).
- Affects both SQaT and TC.
- We define a Test Setting (TS) as a quadruple $T_S = \{N_S, Q_S, A_S, T_A\}$
- N_S is the number of system components, with Unit Qualities (UQs) identified in $Q_S = \{Q_i\}$, $Q_i = 1 - Pr(Unit_i = Faulty)$,
- A_S is the modular test architecture for system *S*, that needs *m* tests with TQs in a test architecture $T_A = \{\tau_1 \ldots \tau_m\}$,

Test Cost Contributing Issues

- Number of required testbeds (stubs etc)
- Required test quality
- Number of test repetitions either planned or unplanned (due to failure of tested unit)

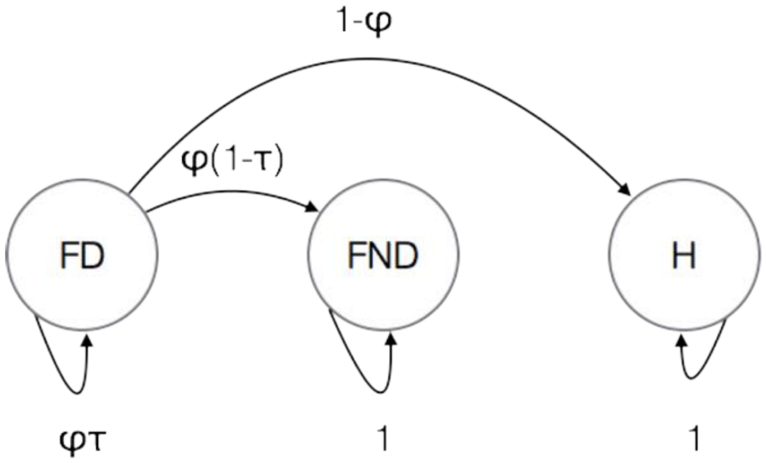


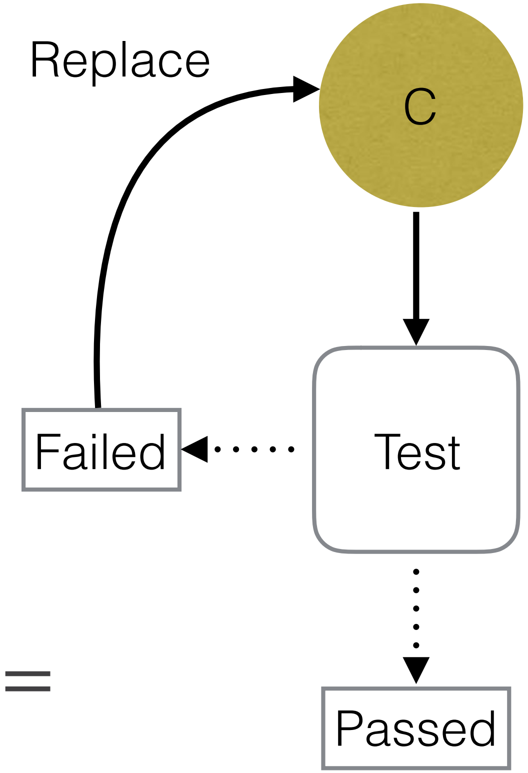
Problem Description

- To determine SQaT and Expected Number of Tests (ENT) given TQ and UQ.

A Markov Chain Solution

- Absorbing Markov chain state space for one unit testing





- FD = Fault Detected, FND = Fault Not Detected
  H = Healthy, $\varphi = Pr(Unit = Faulty), \tau = Pr(Fault = Detected \mid Unit = Faulty)$

Markov Chain solutions

- $[FND_\infty \quad H_\infty] = (1-T)^{-1}A = \begin{bmatrix}\frac{\varphi(1-\tau)}{1-\varphi\tau} & \frac{1-\varphi}{1-\varphi\tau}\end{bmatrix}$
  FND_∞ and H_∞ are the *FND* and *H* states probabilities after absorption.

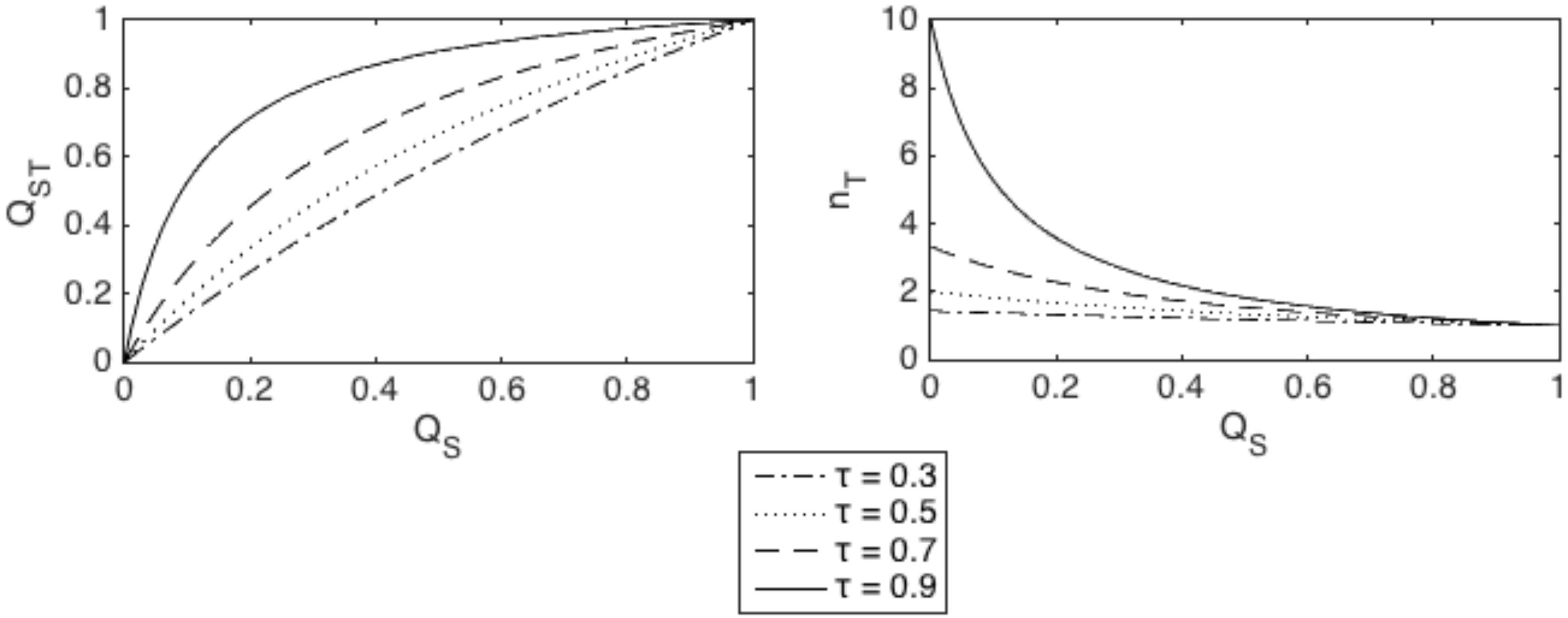
$$SQaT = Q_{UT} = \frac{1-\varphi}{1-\varphi\tau} = \frac{Q_U}{Q_U\tau-\tau+1}$$

- Latent defect probability: $P_{LD} = 1 - Q_{UT} = \frac{\varphi(1-\tau)}{1-\varphi\tau}$
- Expected number of steps (tests) to absorption:

$$\mathrm{ENT} = n_T = \frac{1}{1-\varphi\tau} = \frac{1}{Q_S\tau-\tau+1}$$

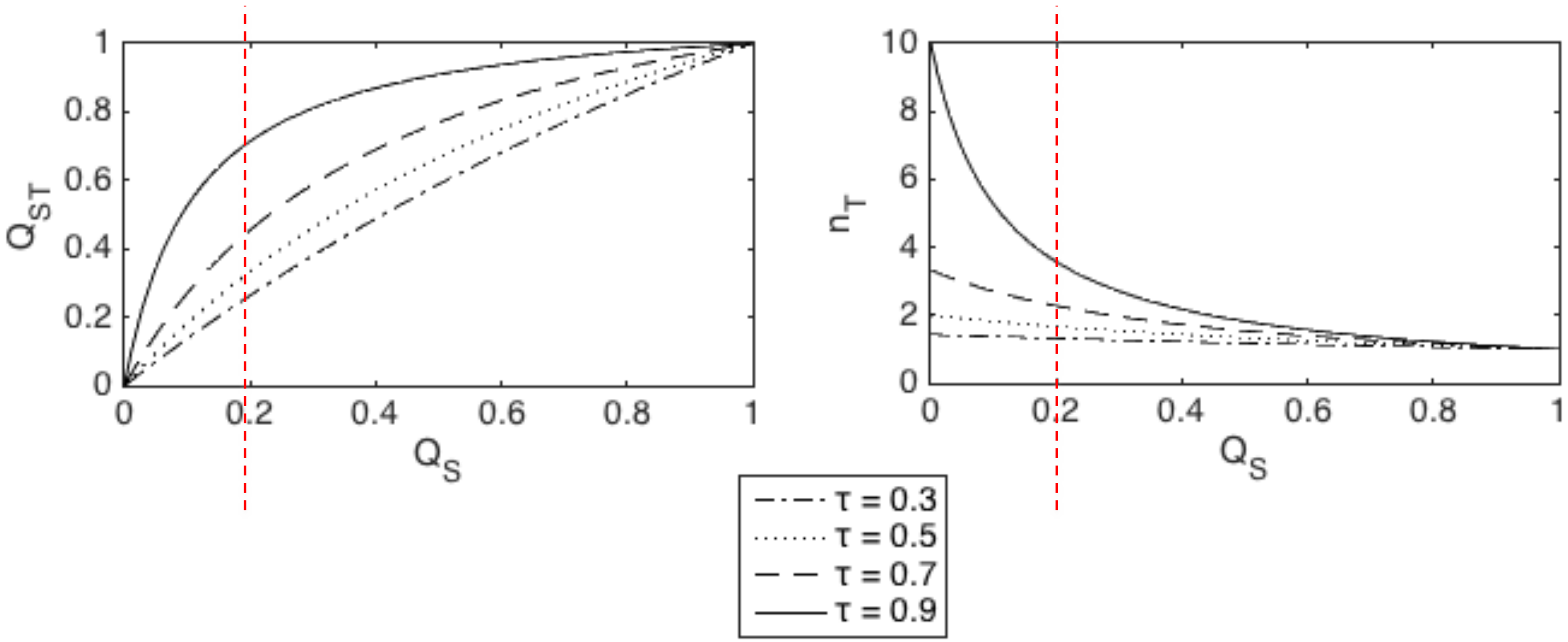
Unit Testing

- There is a tradeoff between acquiring quality components and setting up quality tests.



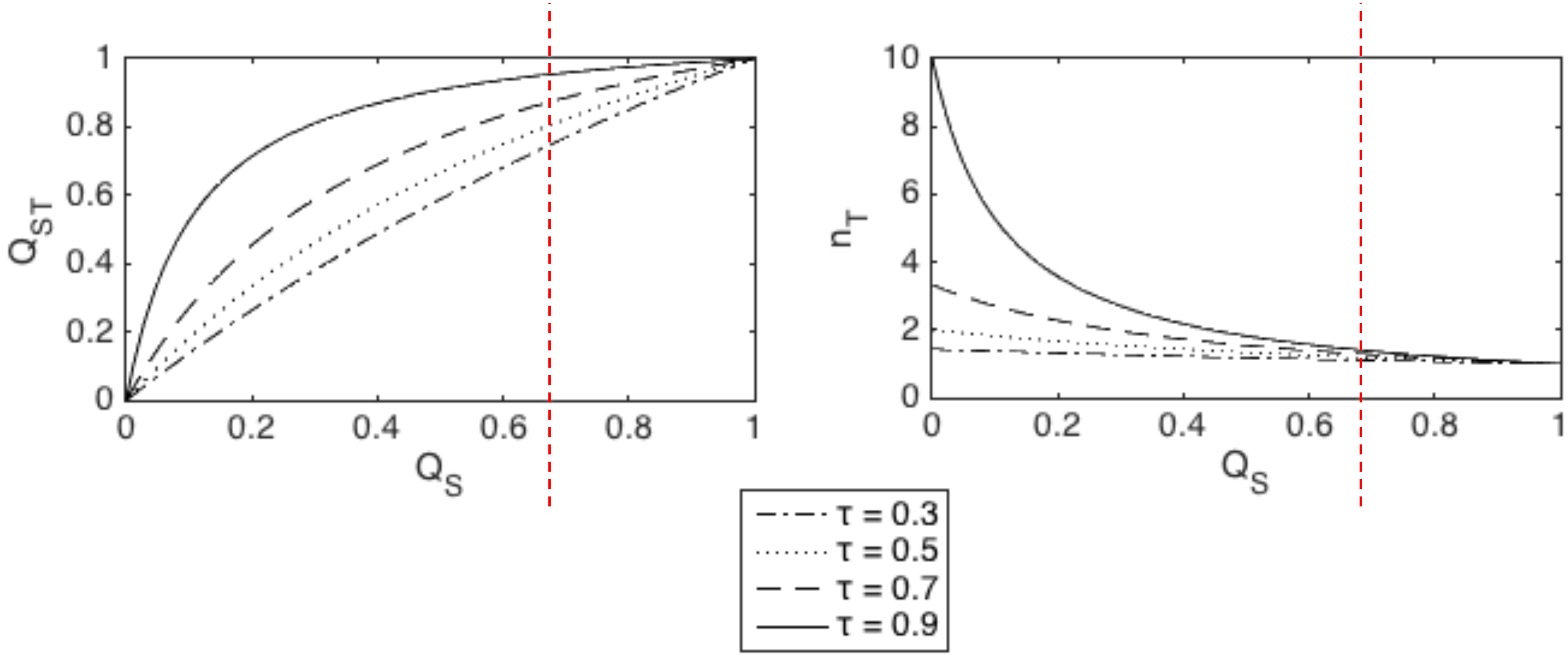
Unit Testing

- There is a tradeoff between acquiring quality components and setting up quality tests.



Unit Testing

- There is a tradeoff between acquiring quality components and setting up quality tests.



System Testing

- A system S is considered as collection of n elements:

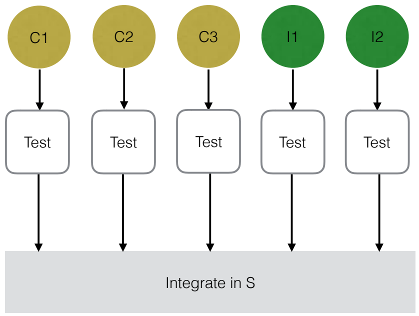
$$Q_S = \prod_{i=1}^{n} Q_i = \prod_{i=1}^{n} (1 - \varphi_i)$$

Single-level System Testing

- N-test

$$n_{T|n} = \sum_{i=1}^{n} \frac{1}{Q_i\tau - \tau + 1}$$

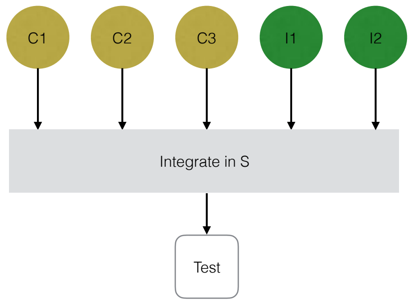
$$Q_{ST|n} = \left(\frac{Q_i}{Q_i\tau - \tau + 1}\right)^n$$



- 1-test

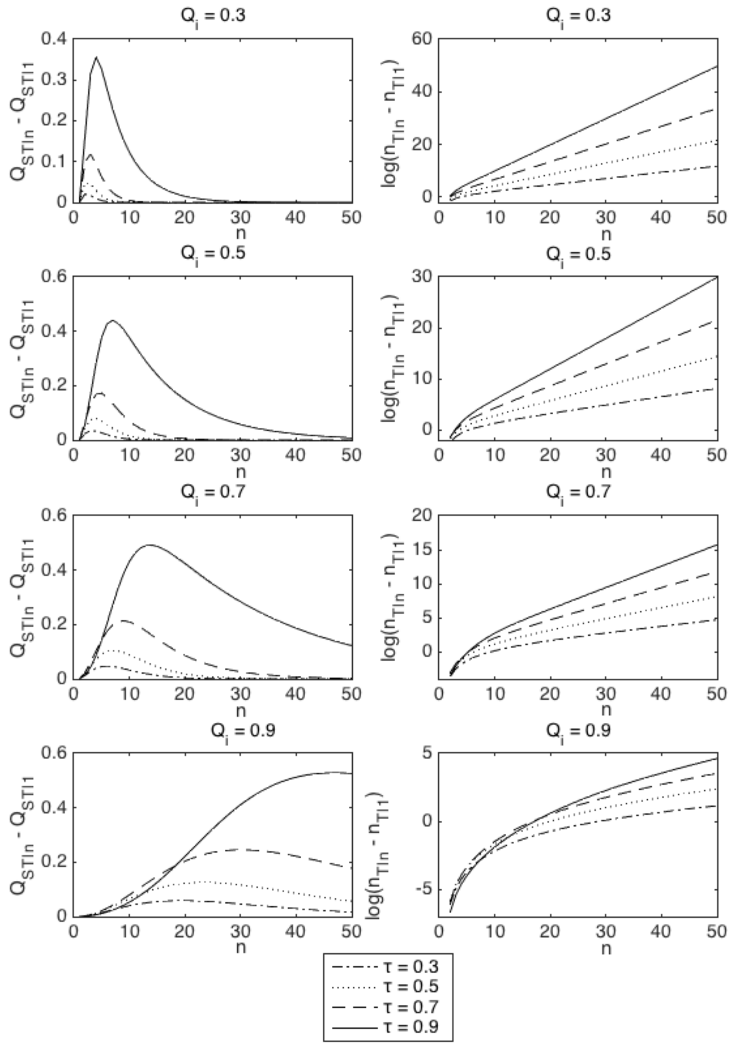
$$Q_{ST|1} = \frac{Q_i^n}{Q_i^n\tau - \tau + 1}$$

$$n_{T|1} = \frac{1}{Q_i^n\tau - \tau + 1}$$



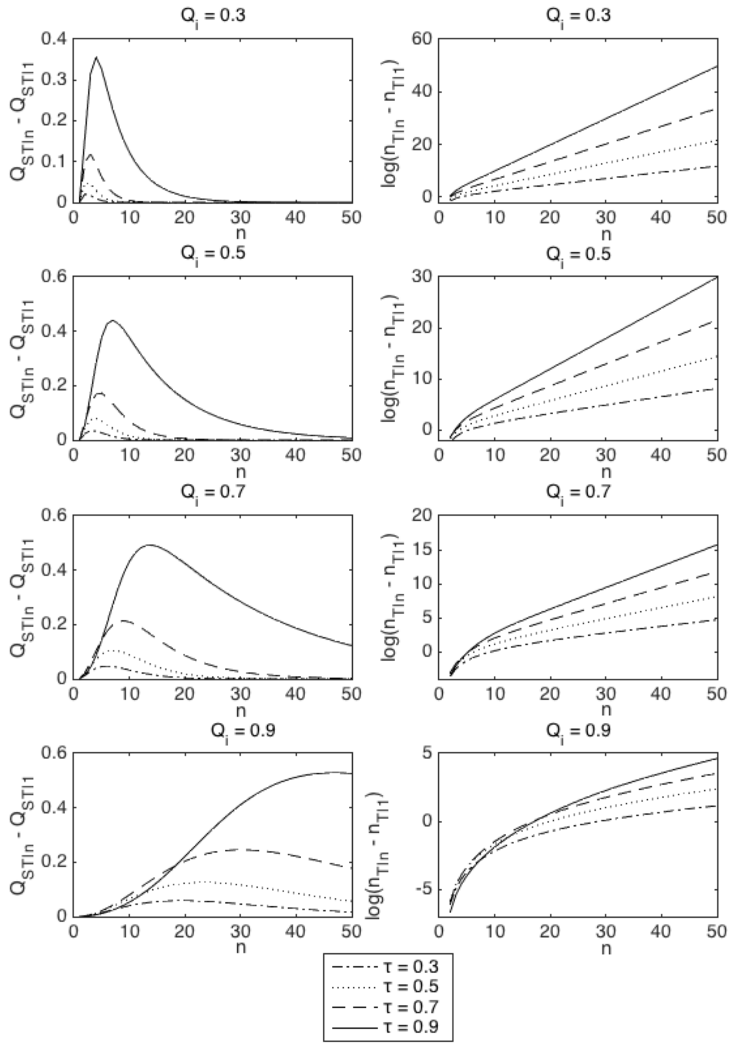
Heuristics for Single-level Test - 1

- For low quality tests, regardless of the number and quality of components before test, perform *1*-test.
- This is very good for test time/cost, while leading to not much difference in quality.



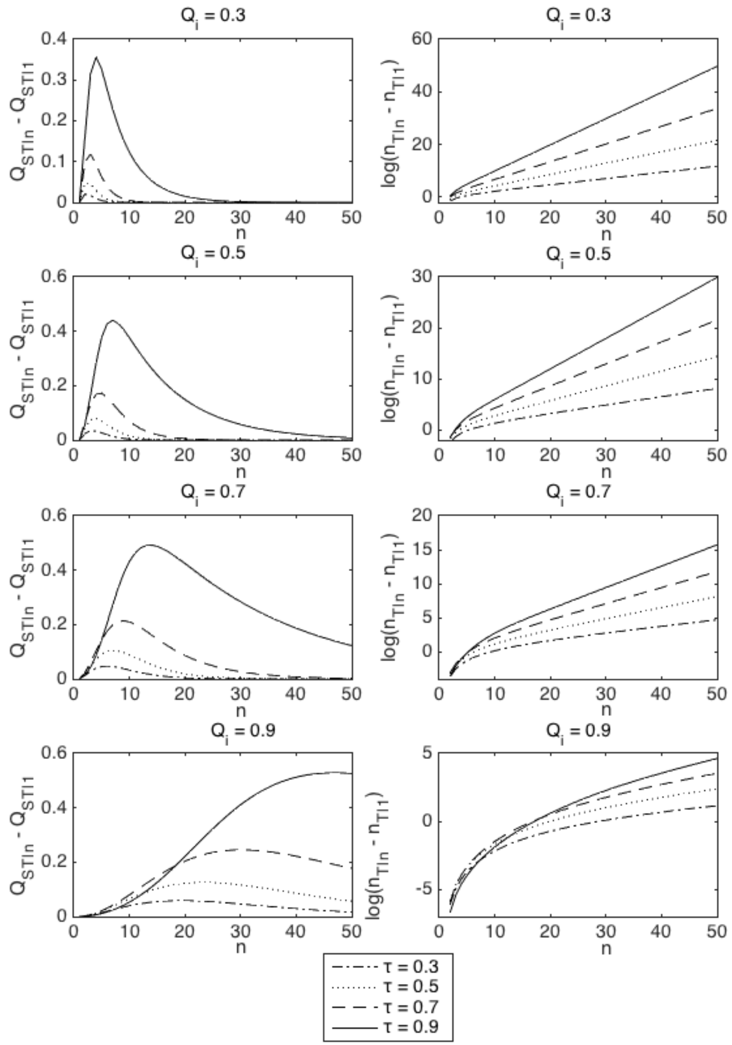
Heuristics for Single-level Test - 2

- With high quality tests and low quality components:
  - When number of components is low-medium (*n*<*30*) choose *n*-test over *1*-test because of large difference in SQaT.
  - When number of components is high (*n*>*30*) choose *1*-test which makes little difference in SQaT relative to *n*-test, but leads to relatively large savings in ENT.



Heuristics for Single-level Test - 3

- With high quality test and high quality components, regardless of the number of components, choose *n*-test.
- This leads to relatively superior results in SQaT and not much difference in test cost/time.

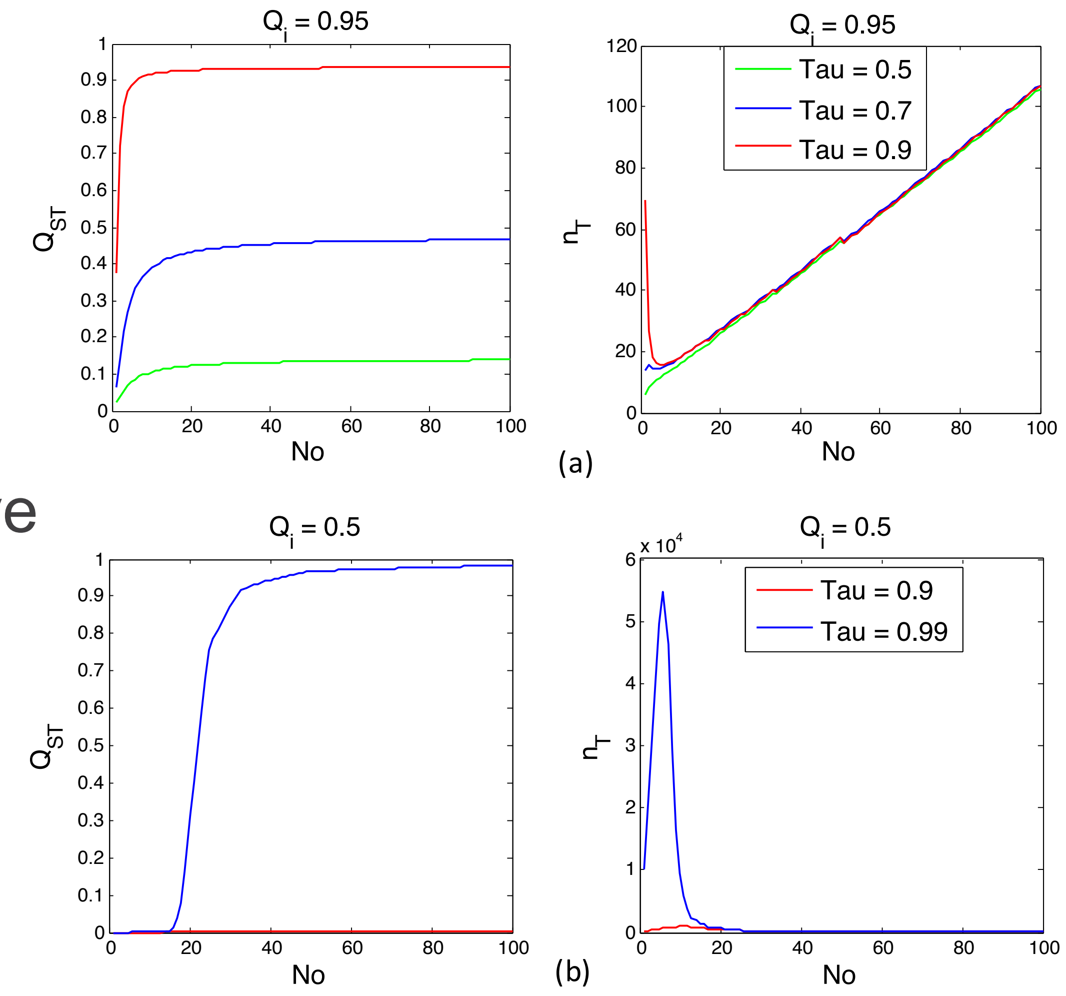


Modular System Testing

- $Q_{ST} = \frac{\prod_{i=1}^{n_M} Q_{MT_i}}{1-\tau_S(1-\prod_{i=1}^{N_M} Q_{MT_i})}$
- $n_{ST} = \sum_{i=1}^{N_M} \frac{n_{T|M_i}}{1-\tau_S(1-Q_{MT_i})} + \frac{1}{1-\tau_S(1-\prod_{i=1}^{n_M} Q_{MT_i})}$
- Q_{MT_i} quality of modules which depend on unit qualities within modules and TQs of module tests
- τ_S TQ at system level

Effect of System Size

- SQaT and ENT of an *n*-component test with *n* = 100, and with (a) all Q_i = 0.95 and (b) all Q_i = 0.5. The horizontal axis is the balanced modularization no which also corresponds to the number of modules used in the test.
  - Number 1 corresponds to two consecutive 1-tests.
  - Number 100 corresponds to a *n*-test followed by 1-test.
  - Number 2 corresponds to [[50],[50]].
  - Number 3 corresponds to [[33],[33],[34]].
  - and so on.



Pareto Optimal Test Architecture

Optimal M No		Test Quality															
		0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99
Unit Quality	0.5	100	100	100	100	100	100	100	100	100	100	100	100	100	100	51	51
	0.6	100	100	100	100	100	100	100	100	100	100	100	100	51	51	51	51
	0.7	100	100	100	100	100	100	100	100	51	51	51	51	51	51	34	34
	0.8	51	51	51	100	51	51	51	51	34	34	34	34	26	26	26	26
	0.9	13	17	21	26	26	17	13	13	13	13	13	13	13	13	13	13
	0.91	10	15	17	17	17	17	13	13	13	13	13	13	13	8	8	8
	0.92	8	13	15	17	17	13	13	8	8	8	8	8	8	8	8	8
	0.93	1	9	13	13	13	13	8	8	8	8	8	8	8	8	8	8
	0.94	1	7	8	9	9	8	6	6	6	6	6	6	6	6	6	6
	0.95	1	5	6	7	8	6	6	6	6	6	6	6	6	6	6	6
	0.96	1	3	4	5	6	5	4	4	4	4	4	4	4	4	4	4
	0.97	1	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	0.98	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2
	0.99	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Conclusion

- The simple test architecture suggested by V-model is not always optimal. More modular test architectures for specific TQ and unit qualities can have similar SQaT as the V-model test architecture with much lower expected ENTs.
- The selection of modular architecture can be highly sensitive to TQs and unit qualities. The model presented here is a useful starting point for architecture selection and analysis.
- Several heuristics were driven based on the model that might help in test planning of complex systems for highest gains in terms of quality and cost.

28th Annual **INCOSE**
international symposium

Washington, DC, USA
July 7 - 12, 2018

www.incose.org/symp2018