3rd UCAAT User Conference on Advanced Automated Testing

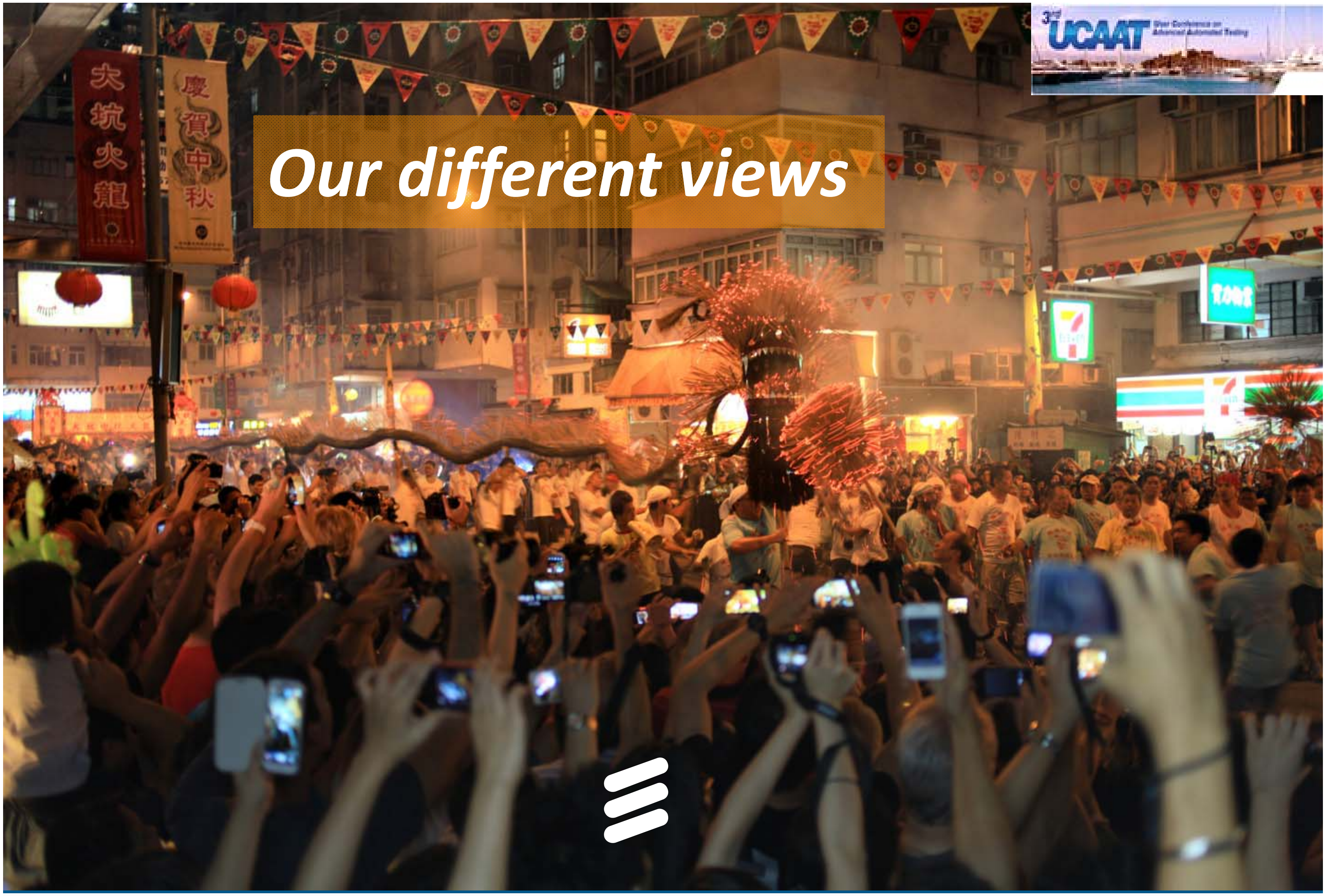
ETSI World Class Standards

Sophia Antipolis, French Riviera
20-22 October 2015

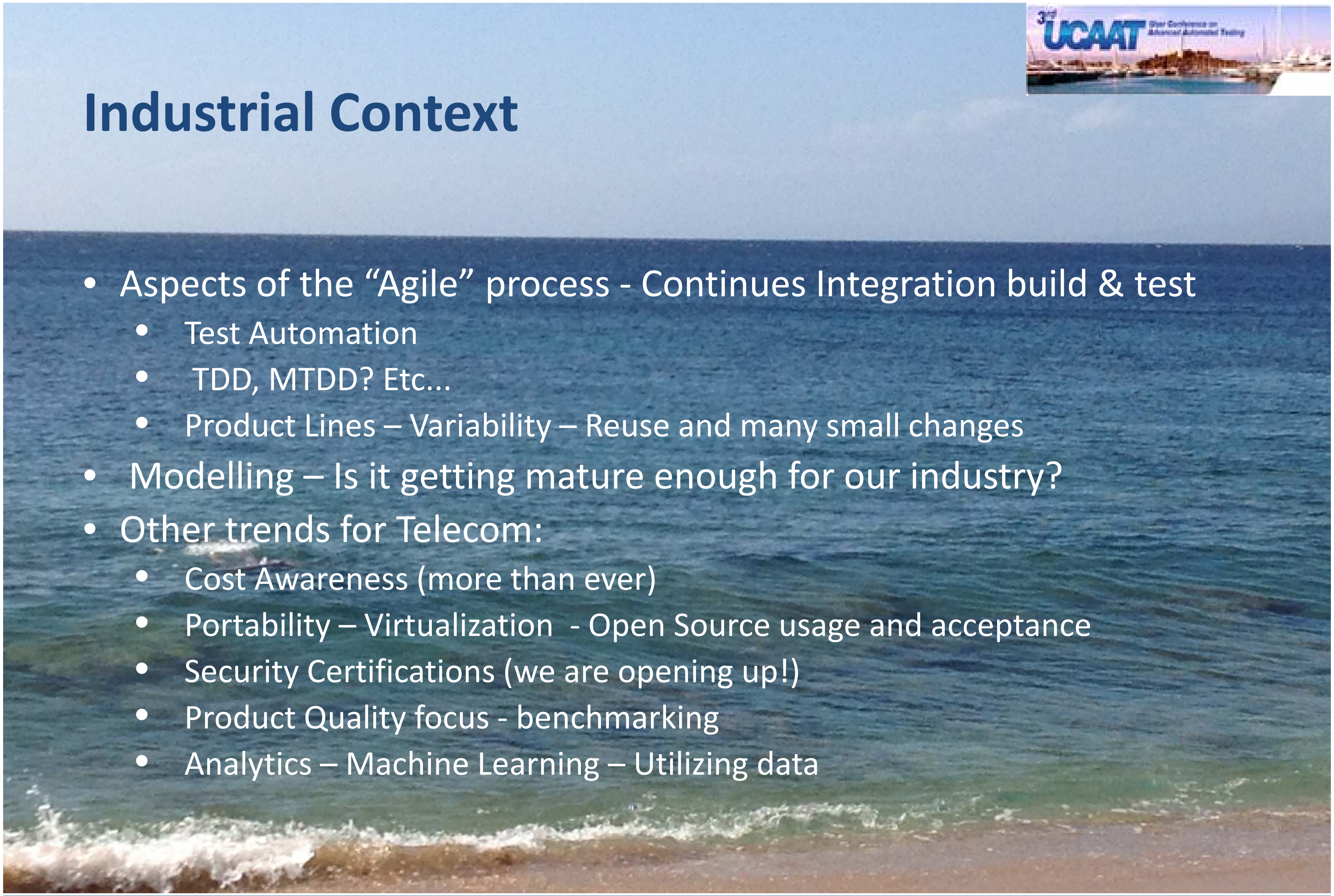
REQUIREMENTS FOR AND CHALLENGES WITH ADVANCED TEST AUTOMATION - THE INDUSTRY PERSPECTIVE

Presented by Dr. Sigrid Eldh, Ericsson, Sweden

ERICSSON



Our different views



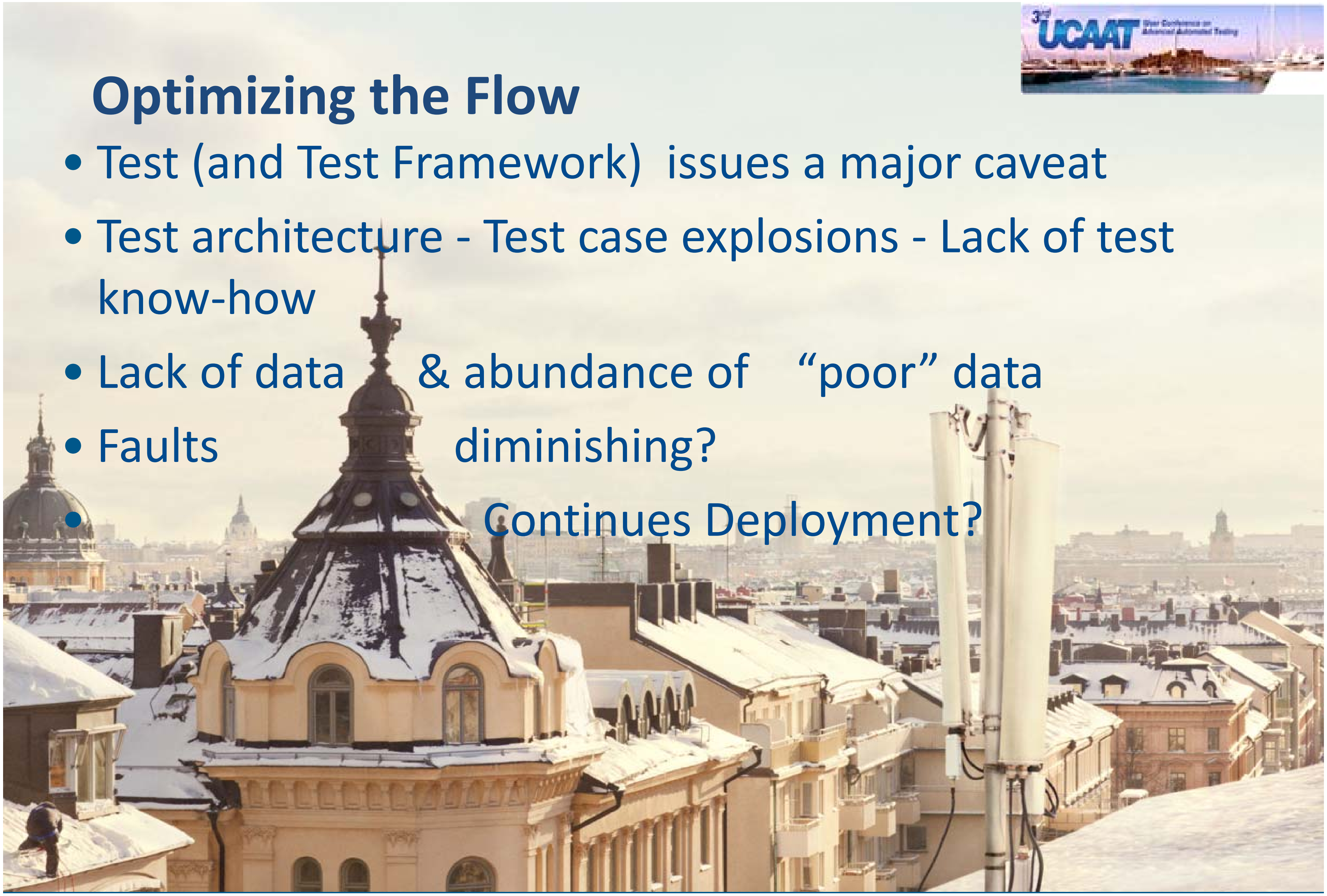
Industrial Context

- Aspects of the “Agile” process - Continues Integration build & test
  - Test Automation
  - TDD, MTDD? Etc…
  - Product Lines – Variability – Reuse and many small changes
- Modelling – Is it getting mature enough for our industry?
- Other trends for Telecom:
  - Cost Awareness (more than ever)
  - Portability – Virtualization - Open Source usage and acceptance
  - Security Certifications (we are opening up!)
  - Product Quality focus - benchmarking
  - Analytics – Machine Learning – Utilizing data

What happened with TESTING in the AGILE context?

System test (and testers role) is diminished, testing has become more of a developers task, this has consequences:

- More tests – boost and management quality engagement
- Test cases more focused on code level (not system level)
- End-to End hard in large complex (telecom) systems
- Hard to get "users in team" – Requirement
- Requirements? User Stories? Detail?
- "Quality Police"/gating and trust
- TDD?



Optimizing the Flow

- Test (and Test Framework) issues a major caveat
- Test architecture - Test case explosions - Lack of test know-how
- Lack of data & abundance of "poor" data
- Faults diminishing?
- Continues Deployment?

First and Foremost TEST Automation!

Automation in Test => Automation EVERYWHERE

- It is from capture requirements to installation & deployment
  - Analyzing usage, users – behavior of our test systems
- How we build the architecture in our test systems
  - handle changes, scalability and adapt to new types of systems
  - Parallelism of test execution – cloud
- How we Automatically measure Quality of our tests as well as quality of our system

TAIM- Test Automation Improvement Model

	Level / Focus	1 Initial Metrics defined & deployed Initial	2 Repeatable Data collected Analysis	3 Defined Mechanism Statistical Validity	4 *Self*- Managed Actions and Issues "highlighted" Accuracy	5 *Self*-Optimized Cost minimization Safety-critical Fail-safe
	General	Cost, standards, Metrics				
1	**Test Management**	Planning factors Automation ++ Trend, cost etc			Self adapting Guidance	Management "redundant" in ongoing
2	**Test Requirements**	*Standards, 25010 e.g. testability* Traceability, Validation				
3	**Test Specifications**	TDT, TC Gen, Pre-process				
4	**Test Code**	Lang, templates (models), Architecture				
5	**Test Automation Process**	Context, type, level, *CR/AR mgmt*, Improve , Flow				
6	**Test Execution**	Select, Type, When (Func/non-Func, Regression)			Automatic priority schemes of what, how and when to execute (validate)	
7	**Test Verdicts**	Test Oracle, Post Process			Validity & Gap analysis	
8	**Test Environment (context)**	Set-up /Prep, Type: Simu/ Emu/Hw/Virtual, test data			Self-installing, self-configuring , self-utilization	Self-optimizing
9	**Tools**	Select, Integrate (tool-chains), Components/API				
10	**Fault/ Defect management**	CR/AR; Class, Ide/triage, Localize, Prediction			Self-healing systems	Cost optimization

Focus Areas in TAIM

1. Test Management
   1. Planning & Deployment
   2. Evaluation
   3. Automation analysis
   4. Technical Debt
2. Test Requirements
   1. Traceability
   2. Validation
3. Test Specifications
   1. Test Case generation
   2. Test Design Technique (TDT)
   3. Pre-process analysis
4. Test Code
   1. Language
   2. Standards/templates
   3. Architectures (within code)
5. Test Automation Process
   1. Context, type, Level
   2. CR/AR
   3. Improvements
   4. Flow, speed & workflows
6. Test Execution
   1. Selection
   2. Functional
   3. Non-Functional (Robustness,…)
   4. Regression test (legacy)
7. Test Verdicts
   1. Post-process analysis
   2. Test Oracle
8. Test Environment (context)
   1. Test case set up
   2. Type: Simulated, Emulated, limited, actual
   3. Test Data
   4. *Standards and certification suites and API's*
9. Test Tools
   1. Tool selection
   2. Integration, Context "Tools chain"
   3. Tool(s) Architecture: Classification
   4. Components, API's
10. Fault/Defect Management
    1. Change Report/Anomaly (Failure bug reports)
    2. Classifications
    3. Fault identification, triaging
    4. Fault Localization
    5. Fault Correction
    6. Fault Prediction

General : Measurements, Standards, Cost,

Requirements

- Test Tool Frameworks needs investments “out of the box” and through the lifecycle!
- No Stupid “one tool fits all” approach
  - Different tools for different problems
- PLEASE start with “components” view and open source
  - Common parts e.g. visualization (Standardizations!)

- Accurate measurements (!) of entire life-cycle
- Intuitive to learn!
- You should not need a Masters or PhD!
- Usability – “interactive guiding?”
- Think: Testing for dummies…
- Fast Feedback – now!
- *ADAPTIBILITY!*

Test Design Techniques in Industry

- A Test Case is a test case……
  - A lack of competence
  - Requirement – User story driven
  - Manual vs Automated – Usage or systematic
- Main caveat is that at some point “correct” (in detail) must be defined
  - Issues for Non-functional tests (where limits are often hard to “predefine”)
  - Coverage is easy (gating) and abundance to tools
- How you express/define your TDT
  - Tool support
  - Automatically generate TC or “man made”
  - Informal/formal (business data vs code correct)
- MBT, Mutation, Search-based to Input, Code coverage

Light in the Tunnel for Model Based Test In Industry

MBT Tools are getting more user friendly

Need for higher formalism in specifications

Modeling “natural” way to define & describe

Modeling aids to conserve architecture

Everything is a model (abstraction)

- Two ends of the scale – meet each other!

Informal
Formal

The drive of autonomous – SELF*

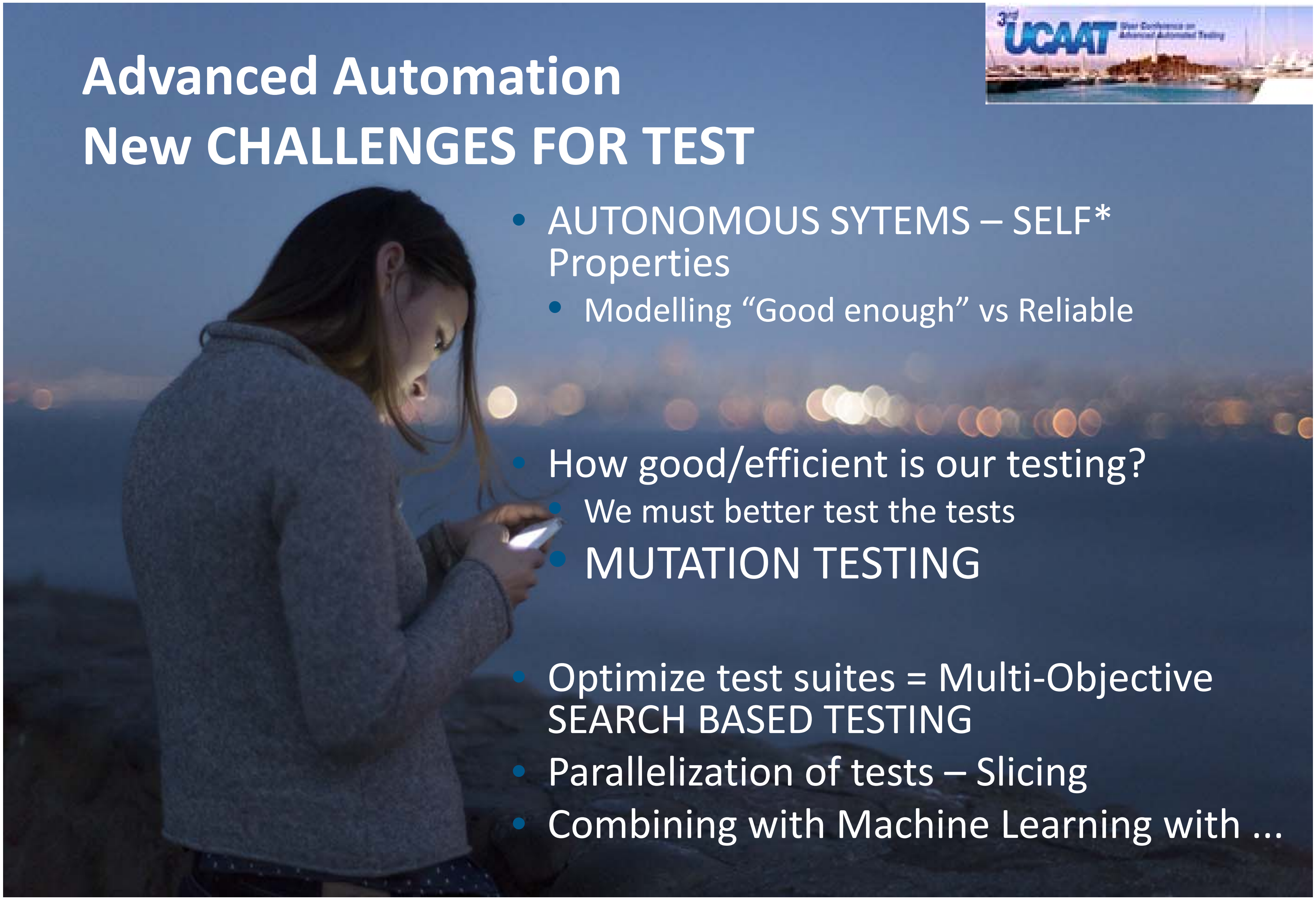
Better at “mature products”

Certifications

Constraints – model checking – formal methods

⚠ Usability/History
Model vs Code (bin)
Failure Cases
Good enough vs “sound”

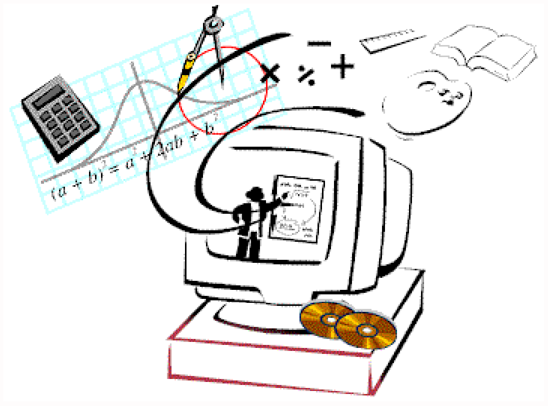
Advanced Automation New CHALLENGES FOR TEST



- AUTONOMOUS SYTEMS – SELF* Properties
  - Modelling “Good enough” vs Reliable
- How good/efficient is our testing?
  - We must better test the tests
  - MUTATION TESTING
- Optimize test suites = Multi-Objective SEARCH BASED TESTING
- Parallelization of tests – Slicing
- Combining with Machine Learning with ...

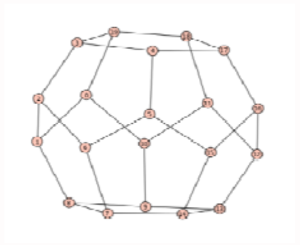
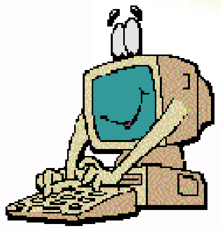
LOTS of opportunities

- Would love initiative on standardizations on test tools (components/API)
- Combine & Explore different type of Techniques!!!



HI SWEETIE, HOW WAS SCHOOL TODAY?
YOU CAN READ ALL ABOUT IT ON MY BLOG, DAD.
SPORTS
Bill Porter

Plenty of Challenges

Conclusion

INDUSTRIAL CONTEXT

Aspects of the "Agile" process – Continues Integration build & test

- Test automation
- TDD, MTDD? Etc...
- ProductLines – Variability – Reuse and many small changes

Modeling – Is it getting mature enough for our industry?

Other trends for Telecom

- Cost awareness (more than ever)
- Portability – Virtualization – Open Source usage and acceptance
- Security Certifications (we are ope[illegible])
- Product Quality focus – benchmarking
- Analytics – Machine Learning – Utilizing data

WHAT HAPPENED WITH TESTING IN THE AGILE CONTEXT?

System test (and testers role) is diminished, testing has become more the developers task, this has consequences.

More tests – boost and management quality engagement

Test cases more focused on code level (not system level)

End-to End hard in large complex (telecom) systems

Hard to get "users in team" – Requirement

Requirements? User Stories? Detail?

"Quality Police" gating and trust

TDD?

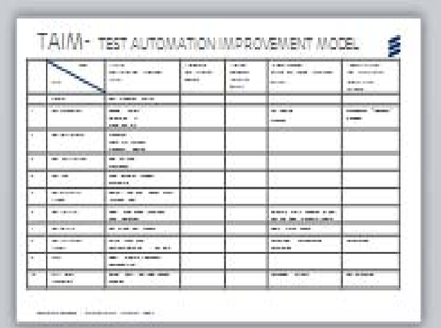
OPTIMIZING THE FLOW

- Test (and Test Framework) issues a major caveat
- Test architecture – Test case explosions – Lack of test know-how
- Lack of data & abundance of "poor" data
- Faults diminishing?
- Continues Deployment?

FIRST AND FOREMOST TEST AUTOMATION!

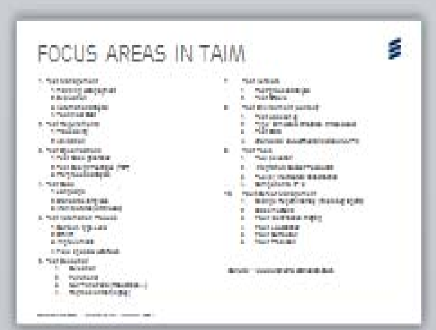
Automation In Test => Automation EVERYWHERE

- It is from capture requirements to installation & deployment
  - Analyzing usage, uses – behavior of our test [illegible]
- How we build the architecture in our test systems
  - handle changes, scalability and adapt to new types of [illegible]
  - Parallelism of test execution – cloud
- How we Automatically measure Quality of our tests as well as quality of our system



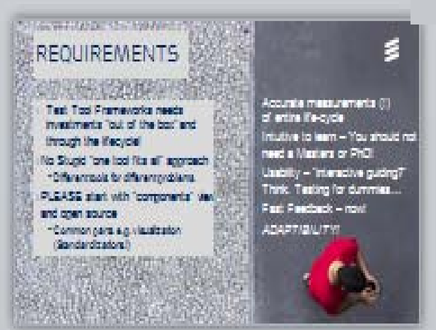
TAIM- TEST AUTOMATION IMPROVEMENT MODEL

[illegible]



FOCUS AREAS IN TAIM

[illegible]



REQUIREMENTS

Test Tool Frameworks needs investments "out of the box" and through the lifecycle

No Stupid "one tool fits all" approach

- Different tools for different problems

PLEASE start with "components" [illegible] and open source

- Common parts e.g. visualization (Standardizations)

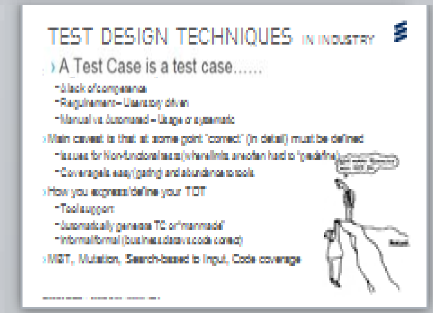
Accurate measurements [illegible] of entire life-cycle

Intuitive to learn – You should not need a Masters or PhD!

Usability – "interactive guiding"? Think: Testing for dummies...

Fast Feedback – now!

ADAPTIBILITY!



TEST DESIGN TECHNIQUES IN INDUSTRY

› A Test Case is a test case......

- Lack of competence
- Requirement – User story driven
- Manual vs Automated – Usage or systematic

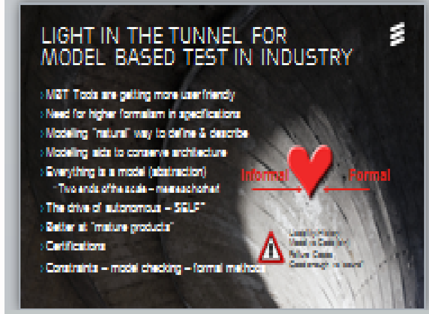
› Main caveat is that at some point "correct" (in detail) must be defined

- Issues for Non-functional issues (where limits are often hard to "predefine")
- Coverage is easy (gating) and abundance of tools

› How you express/define your TDT

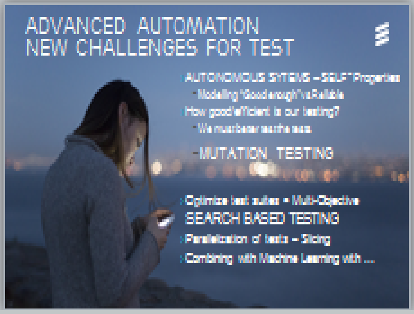
- Tool support
- Automatically generate TC or "manmade"
- Informal/formal (business data vs code correct)

› MBT, Mutation, Search-based to Input, Code coverage



LIGHT IN THE TUNNEL FOR MODEL BASED TEST IN INDUSTRY

- › MBT Tools are getting more user friendly
- › Need for higher formalism in specifications
- › Modeling "natural" way to define & describe
- › Modeling aids to conserve architecture
- › Everything is a model (abstraction)
  - Two ends of the scale – meet each other!
- › The drive of autonomous – SELF"
- › Better at "mature products"
- › Certifications
- › Constraints – model checking – formal methods



ADVANCED AUTOMATION NEW CHALLENGES FOR TEST

› AUTONOMOUS SYTEMS – SELF" Properties

- Modelling "Good enough" vs Reliable

› How good/efficient is our testing?

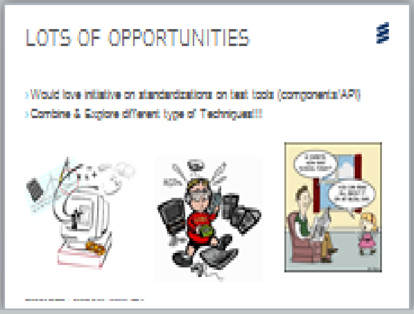
- We must be clear as the area

- MUTATION TESTING

› Optimize test suites = Multi-Objective SEARCH BASED TESTING

› Parallelization of tests – Slicing

› Combining with Machine Learning with ....



LOTS OF OPPORTUNITIES

› Would love initiative on standardizations on test tools (components/API)

› Combine & Explore different type of Techniques!!!

PLENTY OF CHALLENGES

THANK YOU FOR LISTENING!

QUESTIONS???

- *Sigrid.Eldh at Ericsson.com*
- *Twitter DrSEldh*

ERICSSON