Tanja E.J. Vos

Centro de Investigación en Métodos de Producción de Software
Universidad Politecnica de Valencia
Valencia, Spain
tvos@pros.upv.es

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Overview

- EU projects (with SBST) that I have been coordinating:
  - EvoTest (2006-2009)
  - FITTEST (2010-2013)
- What is means to coordinate them and how they are structured
- How do we evaluate their results through academia-industry projects.

EvoTest

- Evolutionary Testing for Complex Systems
- September 2006– September 2009
- Total costs: 4.300.000 euros
- Partners:
  - Universidad Politecnica de Valencia (Spain)
  - University College London (United Kingdom)
  - Daimler Chrysler (Germany)
  - Berner & Mattner (Germany)
  - Fraunhofer FIRST (Germany)
  - Motorola (UK)
  - Rila Solutions (Bulgaria)
- Website does not work anymore

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

EvoTest objectives/results

- Apply Evolutionary Search Based Testing techniques to solve testing problems from a wide spectrum of complex real world systems in an industrial context.
- Improve the power of evolutionary algorithms for searching important test scenarios, hybridising with other techniques:
  - other general-purpose search techniques,
  - other advanced software engineering techniques, such as slicing and program transformation.
- An extensible and open Automated Evolutionary Testing Architecture and Framework will be developed. This will provide general components and interfaces to facilitate the automatic generation, execution, monitoring and evaluation of effective test scenarios.

FITTEST

- Future Internet Testing
- September 2010 – December 2013
- Total costs: 5.845.000 euros
- Partners:
  - Universidad Politecnica de Valencia (Spain)
  - University College London (United Kingdom)
  - Berner & Mattner (Germany)
  - IBM (Israel)
  - Fondazione Bruno Kessler (Italy)
  - Universiteit Utrecht (The Netherlands)
  - Softteam (France)
- http://www.pros.upv.es/fittest/

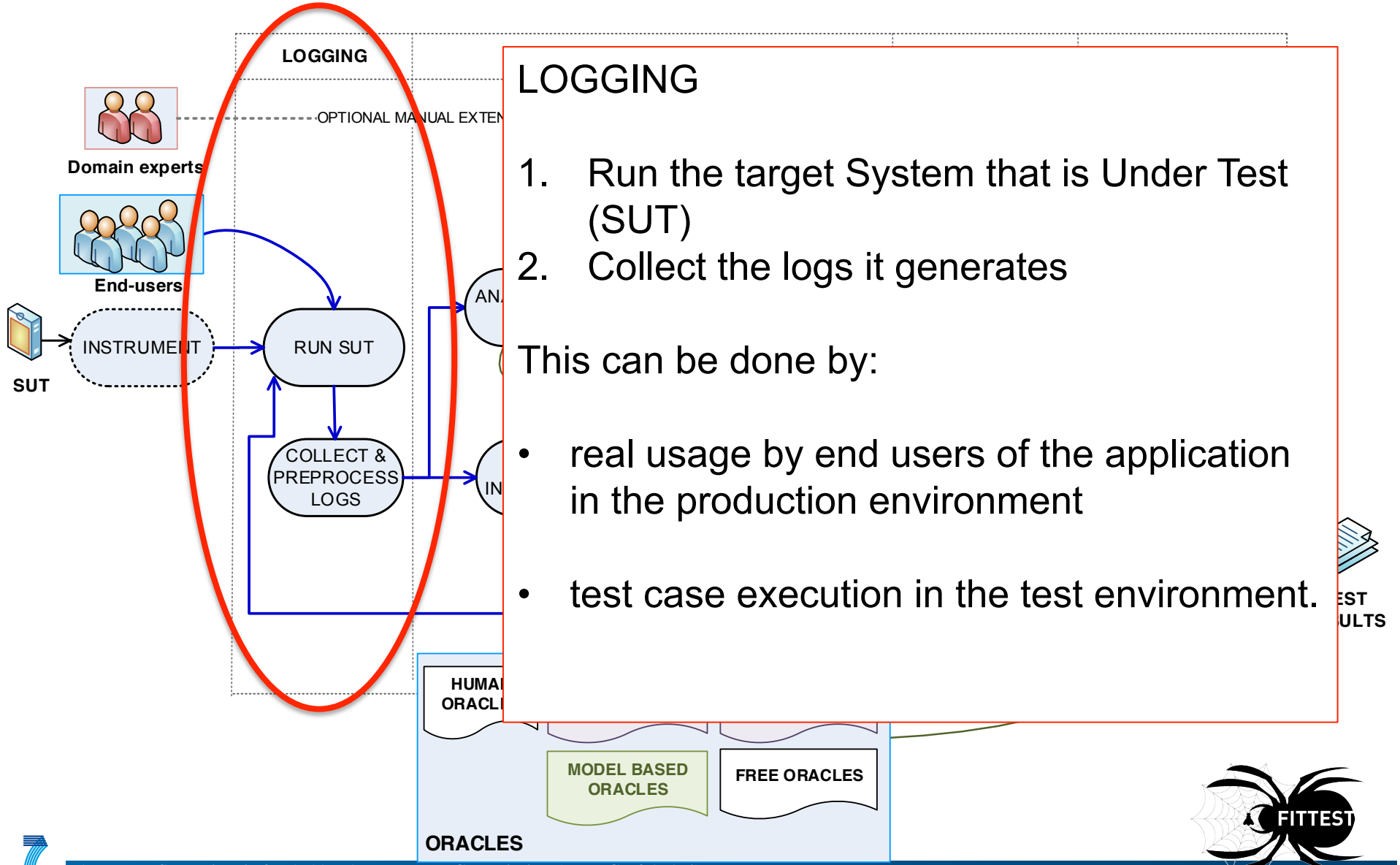
The FITTEST project is funded by the European Commission (FP7-ICT-257574)

FITTEST objectives/results

- **Future Internet Applications**
  - Characterized by an extreme high level of dynamism
  - Adaptation to usage context (context awareness)
  - Dynamic discovery and composition of services
  - Etc..
- **Testing of these applications gets extremely important**
  - Society depends more and more on them
  - Critical activities such as social services, learning, finance, business.
- **Traditional testing is not enough**
  - Testwares are fixed
- **Continuous testing is needed**
  - Testwares that automatically adapt to the dynamic behavior of the Future Internet application
  - This is the objective of FITTEST

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

How does it work?



LOGGING

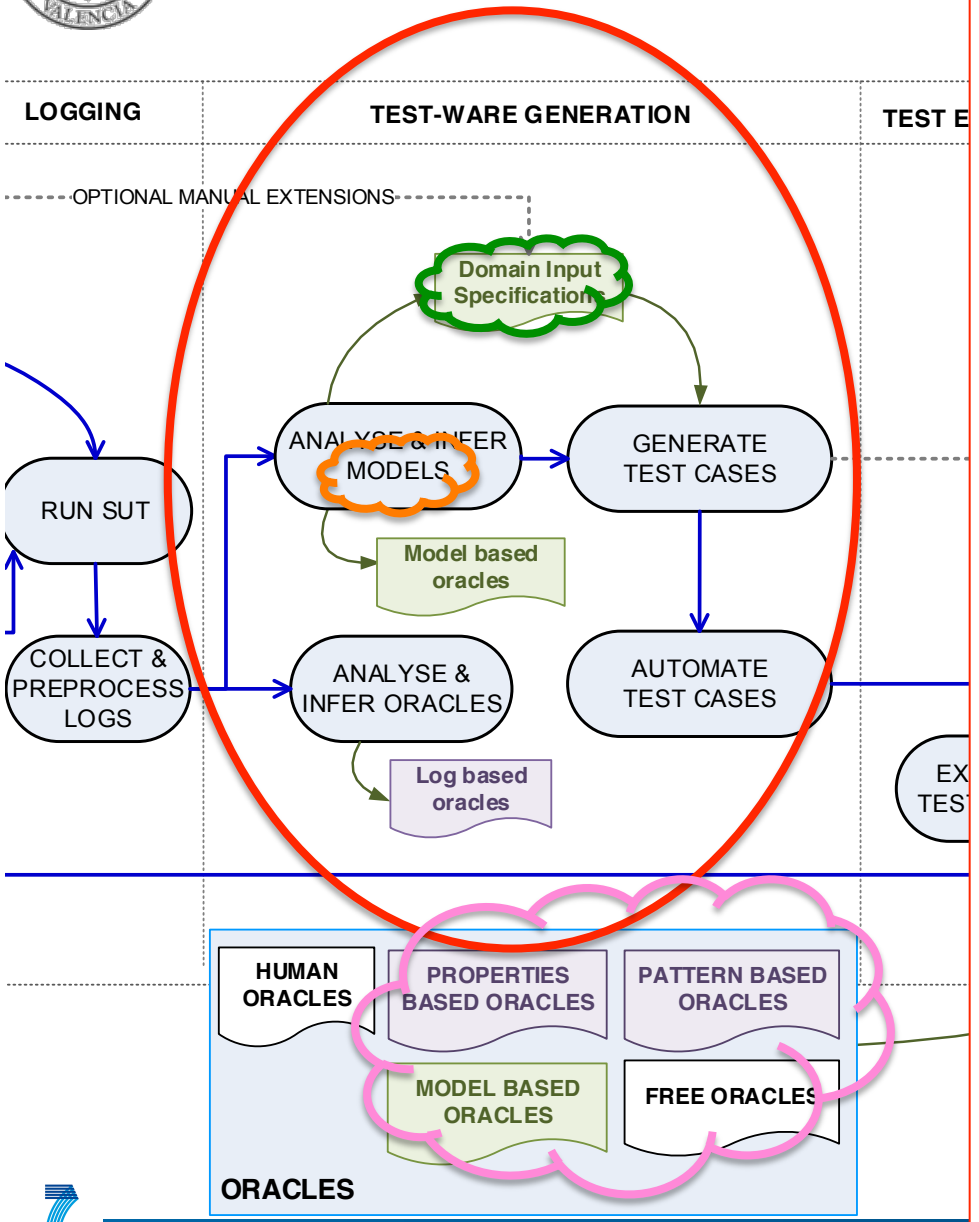
1. Run the target System that is Under Test (SUT)
2. Collect the logs it generates

This can be done by:

- real usage by end users of the application in the production environment
- test case execution in the test environment.

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

How does it work?



GENERATION

1. Analyse the logs

2. Generate different testwares:

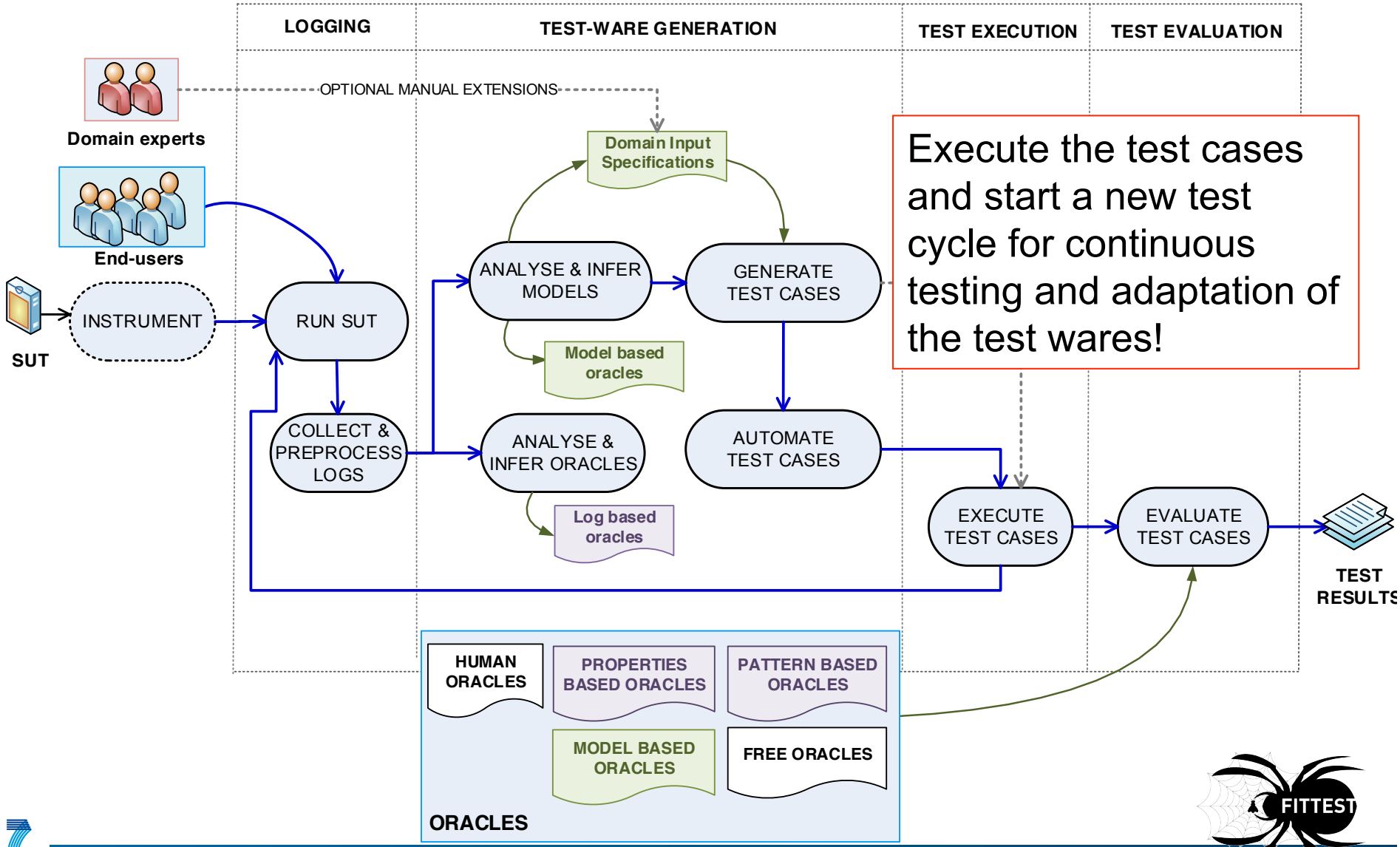
   - **Models**
   - **Domain Input Specification**
   - **Oracles**

3. Use these to generate and automate a test suite consisting off:

   - **Abstract test cases**
   - **Concrete test cases**
   - **Pass/Fail Evaluation criteria**

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

How does it work?



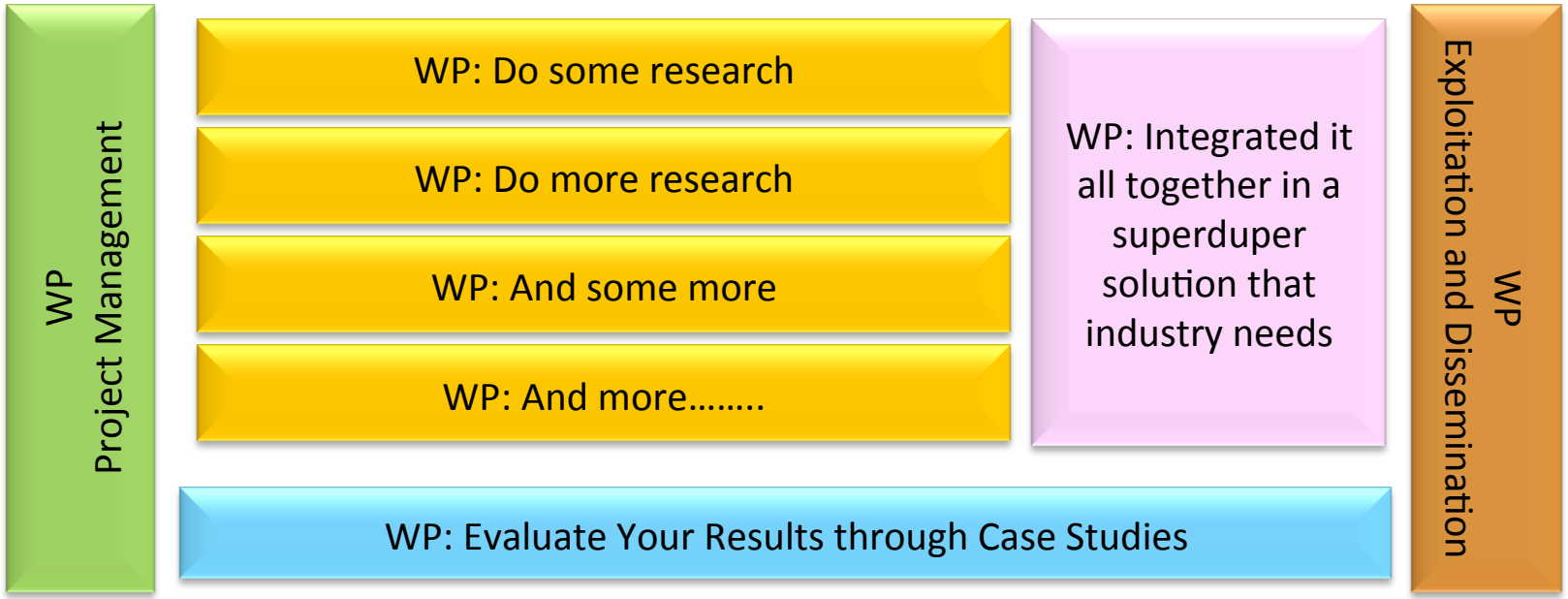
The FITTEST project is funded by the European Commission (FP7-ICT-257574)

What does it mean to be an EU project coordinator

- Understand what the project is about, what needs to be done and what is most important.
- Do NOT be afraid to get your hands dirty.
- Do NOT assume people are working as hard on the project as you are (or are as enthusiastic as you ;-)
- Do NOT assume that the people that ARE responsible for some tasks TAKE this responsibility
- Get CC-ed in all emails and deal with it
- Stalk people (email, sms, whatsapp, skype, voicemail messages)
- Be patient when explaining the same things over and over again

EU project structures

- We have: **WorkPackages** (WP)
- These are composed of: **Tasks**
- These result in: **Deliverables**

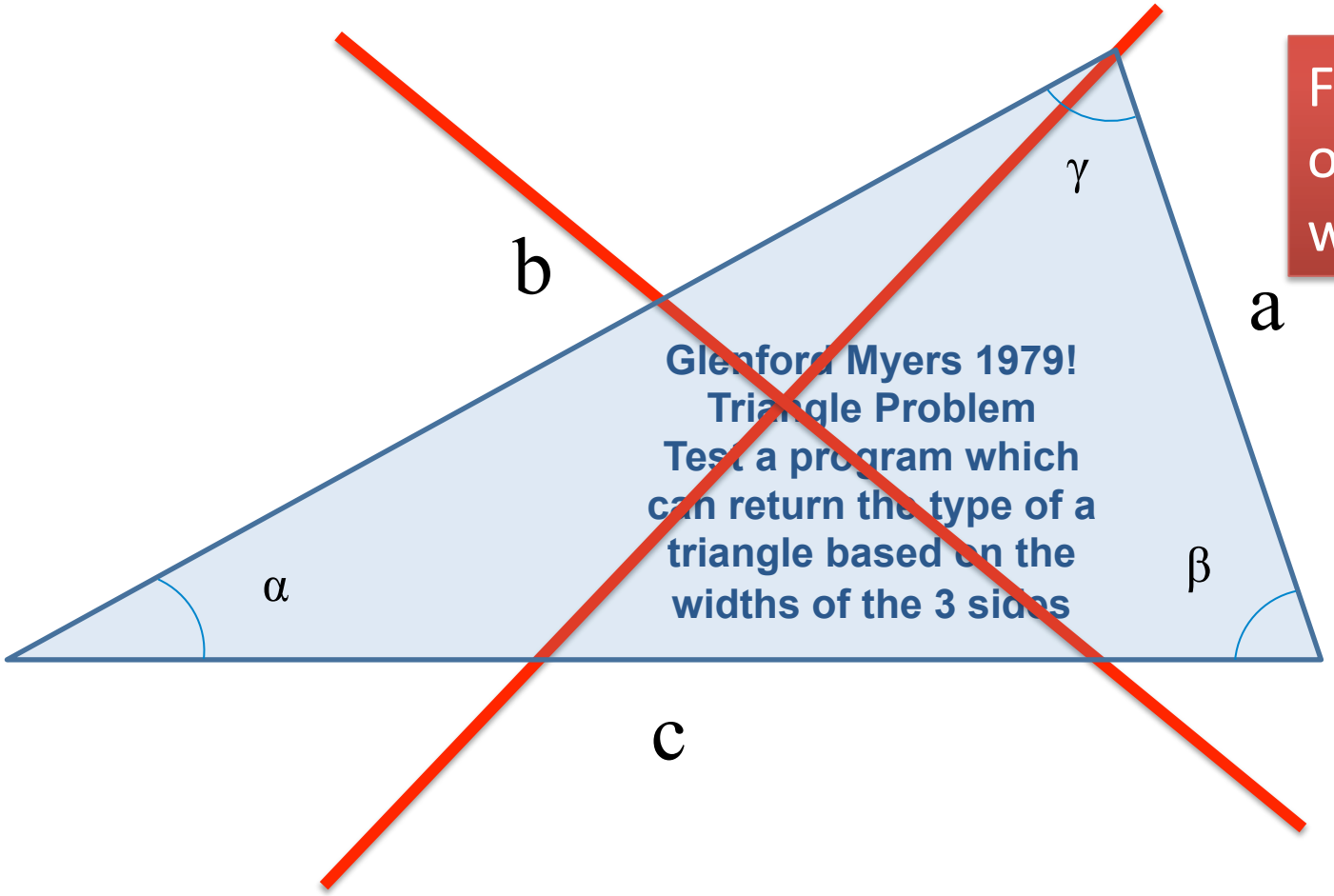


The FITTEST project is funded by the European Commission (FP7-ICT-257574)

EU project
how to evaluate your results

- You need to do studies that evaluate the resulting testing tools/techniques within a real industrial environment

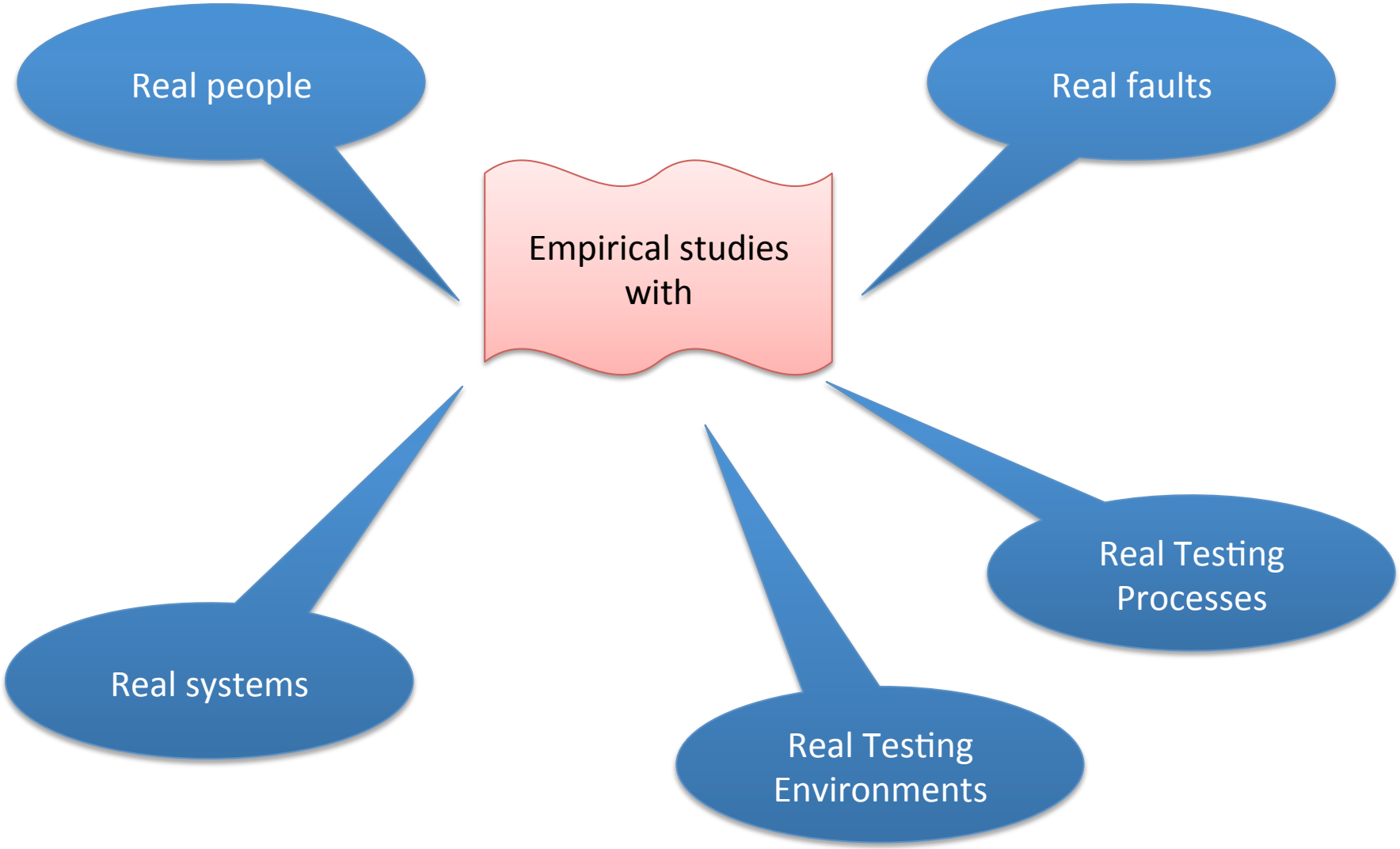
WE NEED MORE THAN …



For evaluation of testing tools we need more!

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

WE NEED….



The FITTEST project is funded by the European Commission (FP7-ICT-257574)

What are empirical studies

- [Wikipedia] *Empirical research is a way of gaining knowledge by means of direct and indirect observation or experience.*
- [PPV00] *An empirical study is really just a test that compares what we believe to what we observe. Such tests when wisely constructed and executed play a fundamental role in software engineering, helping us understand how and why things work.*
- Collecting data:
  - Quantitative data -> numeric data
  - Qualitative data -> observations, interviews, opinions, diaries, etc.
- Different kinds are distinguished in literature [WRH+00, RH09]
  - Controlled experiments
  - Surveys
  - Case Studies
  - Action research

Controlled experiments

What

Experimental investigation of hypothesis in a ***laboratory*** setting, in which conditions are set up to isolate the variables of interest ("independent variables") and test how they affect certain measurable outcomes (the "dependent variables")

Good for

- Quantitative analysis of benefits of a testing tool or technique
- We can use methods showing statistical significance
- We can demonstrate how scientific we are! [EA06]

Disadvantages

- Limited confidence that laboratory set-up reflects the real situation
- ignores contextual factors (e.g. social/organizational/political factors)
- extremely time-consuming

See: [BSH86, CWH05, PPV00, Ple95]

Surveys

What

Collecting information to describe, compare or explain knowledge, attitudes and behaviour over large populations using ***interviews*** or ***questionnaires***.

Good for

- Quantitative and qualitative data
- Investigating the nature of a large population
- Testing theories where there is little control over the variables

Disadvantages

- Difficulties of sampling and selection of participants
- Collected information tends to subjective opinion

See: [PK01-03]

Case Studies

What

A technique for detailed exploratory investigations that attempt to understand and explain phenomenon or test theories ***within their context***

Good for

- Quantitative and qualitative data
- Investigating capability of a tool within a specific real context
- Gaining insights into chains of cause and effect
- Testing theories in complex settings where there is little control over the variables (Companies!)

Disadvantages

- Hard to find good appropriate case studies
- Hard to quantify findings
- Hard to build generalizations (only context)

See: [RH09, EA06, Fly06]

Action Research

What

research initiated to ***solve an immediate problem*** (or a reflective process of progressive problem solving) involving a process of actively participating in an organization's ***change*** situation whilst conducting research

Good for

- Quantitative and qualitative data
- When the goal is solving a problem.
- When the goal of the study is change.

Disadvantages

- Hard to quantify findings
- Hard to build generalizations (only context)

See: [RH09, EA06, Fly06, Rob02]

EU project
how to evaluate your results

- You need to do empirical studies that evaluate the resulting testing tools/techniques within a real industrial environment

- The empirical study that best fits our purposes is Case Study
  - Evaluate the capability of our testing techiques/tools
  - In a real industrial context
  - Comparing to current testing practice

Case studies: not only for justifying research projects

- We **need to apply our results in industry** to understand the problems they have and if we are going the right direction to solve them.
- Real need in software engineering industry to have **general guidelines** on what testing techniques and tools to use for different testing objectives, and how usable these techniques are.
- Up to date these guidelines do not exist.
- If we would have a **body of documented experiences** and knowledge from which the needed guidelines can be extracted
- With these guidelines, testing practicioners might make **informed decisions** about which techniques to use and estimate the time/effort that is needed.

The challenge

To create a **body of evidence** consisting of evaluative studies of testing techniques and tools that can be used to understand the needs of industry and derive general guidelines about their usability and applicability in industry.

TO DO THIS WE HAVE TO:

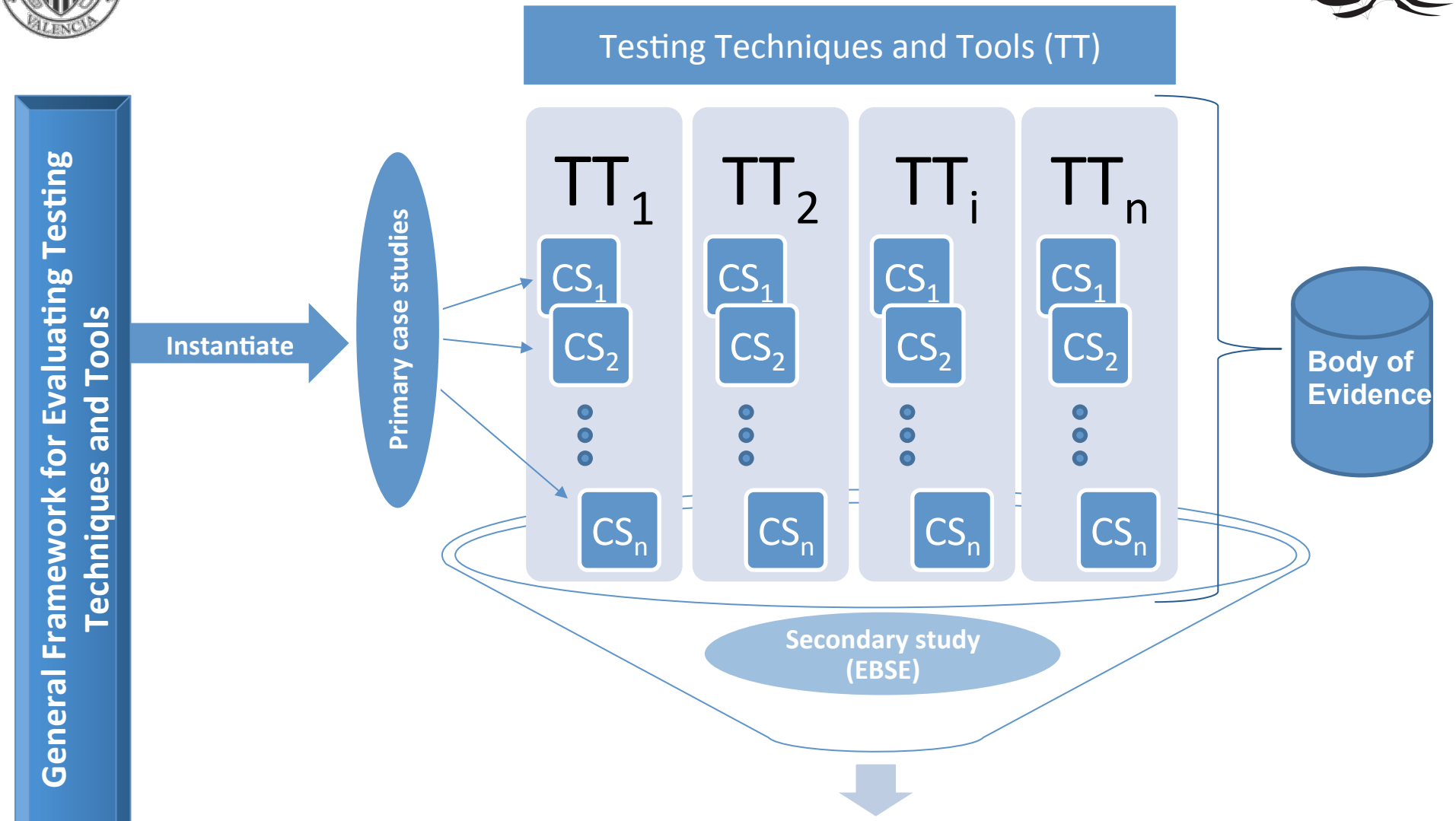
- Perform **more** evaluative empirical case studies in industrial environments
- Carry out these studies by following the **same methodology**, to enhance the comparison among the testing techniques and tools,
- Involve **realistic systems, environments and subjects** (and not toy-programs and students as is the case in most current work).
- Do the studies **thoroughly** to ensure that any benefit identified during the evaluation study is clearly derived from the testing technique studied, and also to ensure that different studies can be compared.

FOR THIS WE NEED:

- A general **methodological evaluation framework** that can simplify the design of case studies for comparing software testing techniques and make the results more precise, reliable, and easy to compare.

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

The idea



Obtain answers to general questions about adopting different software testing techniques and or tools.

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Very brief…what is EBSE

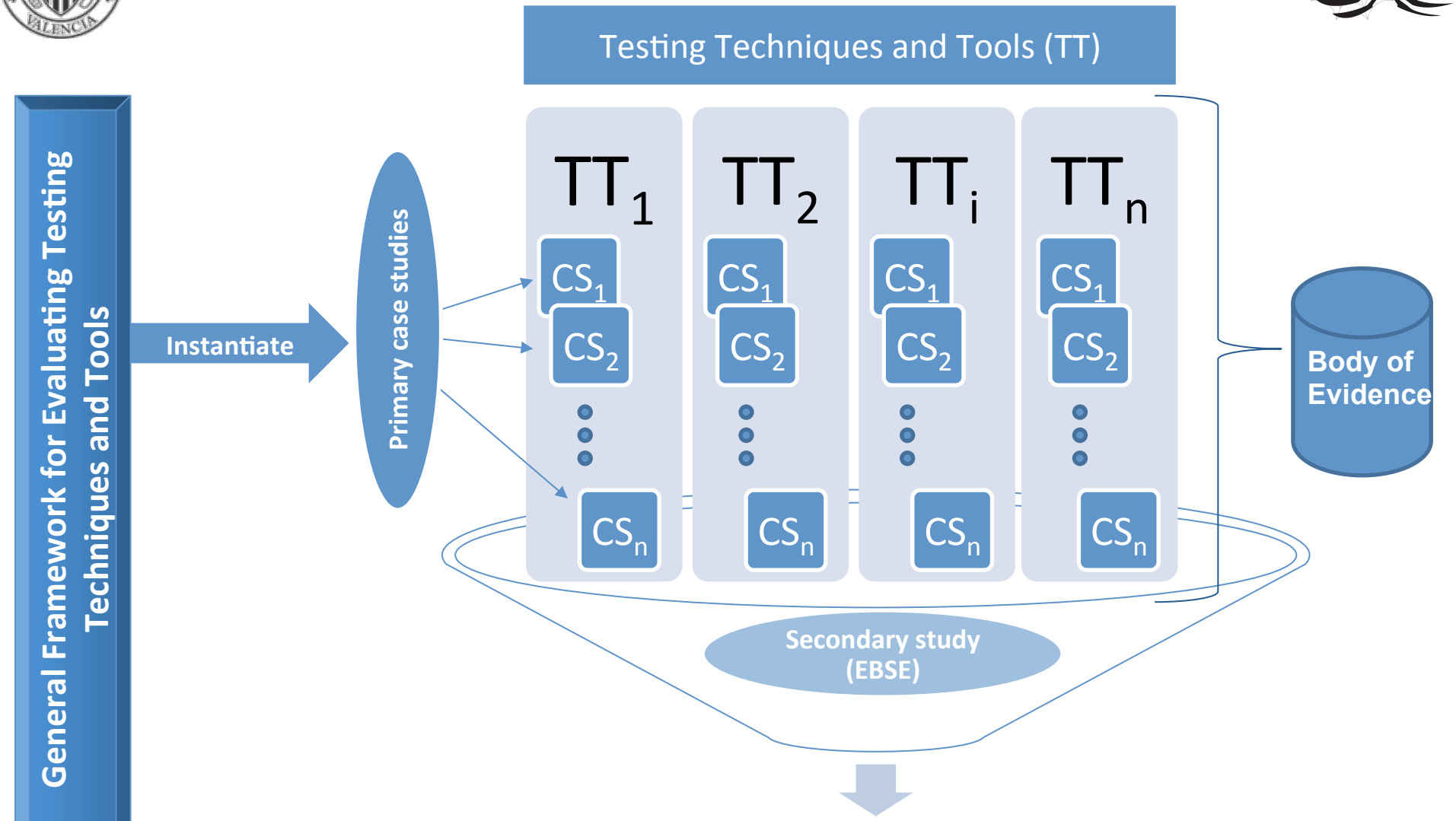
Evidence Based Software Engineering

- The essence of the evidence-based paradigm is that of systematically ***collecting and analysing all of the available empirical data*** about a given phenomenon in order to obtain a much wider and more complete perspective than would be obtained from an individual study, not least because each study takes place within a particular context and involves a specific set of participants.

- The core tool of the evidence-based paradigm is the Systematic Literature Review (SLR)
  - Secondary study
  - Gathering an analysing primary stydies.

- See: http://www.dur.ac.uk/ebse/about.php

The idea

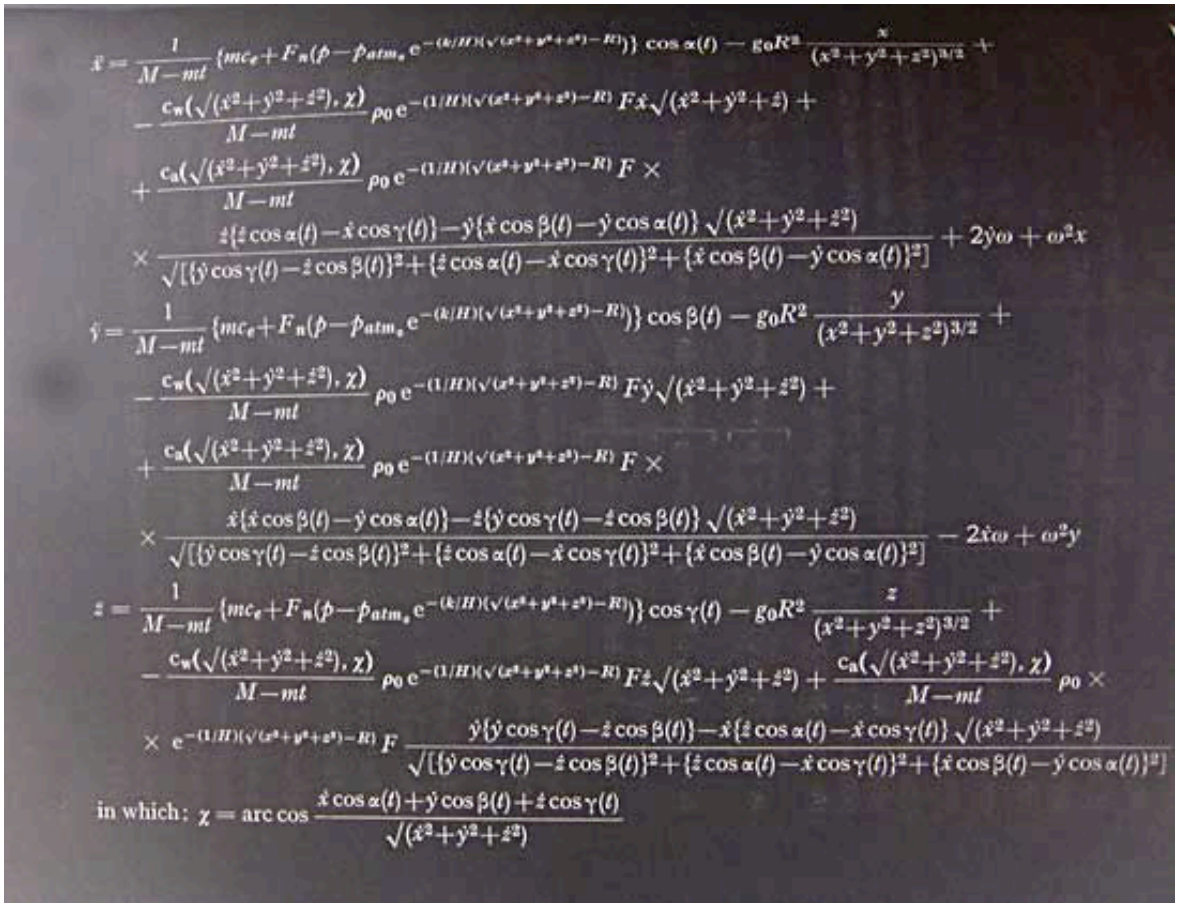


Obtain answers to general questions about adopting different software testing techniques and or tools.

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Empirical research with industry = difficult

- Not “rocket science”-difficult



$$\ddot{x} = \frac{1}{M-mt}\{mc_e + F_n(p - p_{atm_e}\mathrm{e}^{-(k/H)(\sqrt{(x^2+y^2+z^2)}-R)})\}\cos\alpha(t) - g_0R^2\frac{x}{(x^2+y^2+z^2)^{3/2}} +$$
$$-\frac{c_n(\sqrt{(\dot{x}^2+\dot{y}^2+\dot{z}^2)},\chi)}{M-mt}\rho_0\,\mathrm{e}^{-(1/H)(\sqrt{(x^2+y^2+z^2)}-R)}F\dot{x}\sqrt{(\dot{x}^2+\dot{y}^2+\dot{z}^2)} +$$
$$+\frac{c_a(\sqrt{(\dot{x}^2+\dot{y}^2+\dot{z}^2)},\chi)}{M-mt}\rho_0\,\mathrm{e}^{-(1/H)(\sqrt{(x^2+y^2+z^2)}-R)}F\times$$
$$\times\frac{\dot{z}\{\dot{z}\cos\alpha(t)-\dot{x}\cos\gamma(t)\}-\dot{y}\{\dot{x}\cos\beta(t)-\dot{y}\cos\alpha(t)\}\sqrt{(\dot{x}^2+\dot{y}^2+\dot{z}^2)}}{\sqrt{[\{\dot{y}\cos\gamma(t)-\dot{z}\cos\beta(t)\}^2+\{\dot{z}\cos\alpha(t)-\dot{x}\cos\gamma(t)\}^2+\{\dot{x}\cos\beta(t)-\dot{y}\cos\alpha(t)\}^2]}} + 2\dot{y}\omega + \omega^2x$$

$$\ddot{y} = \frac{1}{M-mt}\{mc_e + F_n(p - p_{atm_e}\mathrm{e}^{-(k/H)(\sqrt{(x^2+y^2+z^2)}-R)})\}\cos\beta(t) - g_0R^2\frac{y}{(x^2+y^2+z^2)^{3/2}} +$$
$$-\frac{c_n(\sqrt{(\dot{x}^2+\dot{y}^2+\dot{z}^2)},\chi)}{M-mt}\rho_0\,\mathrm{e}^{-(1/H)(\sqrt{(x^2+y^2+z^2)}-R)}F\dot{y}\sqrt{(\dot{x}^2+\dot{y}^2+\dot{z}^2)} +$$
$$+\frac{c_a(\sqrt{(\dot{x}^2+\dot{y}^2+\dot{z}^2)},\chi)}{M-mt}\rho_0\,\mathrm{e}^{-(1/H)(\sqrt{(x^2+y^2+z^2)}-R)}F\times$$
$$\times\frac{\dot{x}\{\dot{x}\cos\beta(t)-\dot{y}\cos\alpha(t)\}-\dot{z}\{\dot{y}\cos\gamma(t)-\dot{z}\cos\beta(t)\}\sqrt{(\dot{x}^2+\dot{y}^2+\dot{z}^2)}}{\sqrt{[\{\dot{y}\cos\gamma(t)-\dot{z}\cos\beta(t)\}^2+\{\dot{z}\cos\alpha(t)-\dot{x}\cos\gamma(t)\}^2+\{\dot{x}\cos\beta(t)-\dot{y}\cos\alpha(t)\}^2]}} - 2\dot{x}\omega + \omega^2y$$

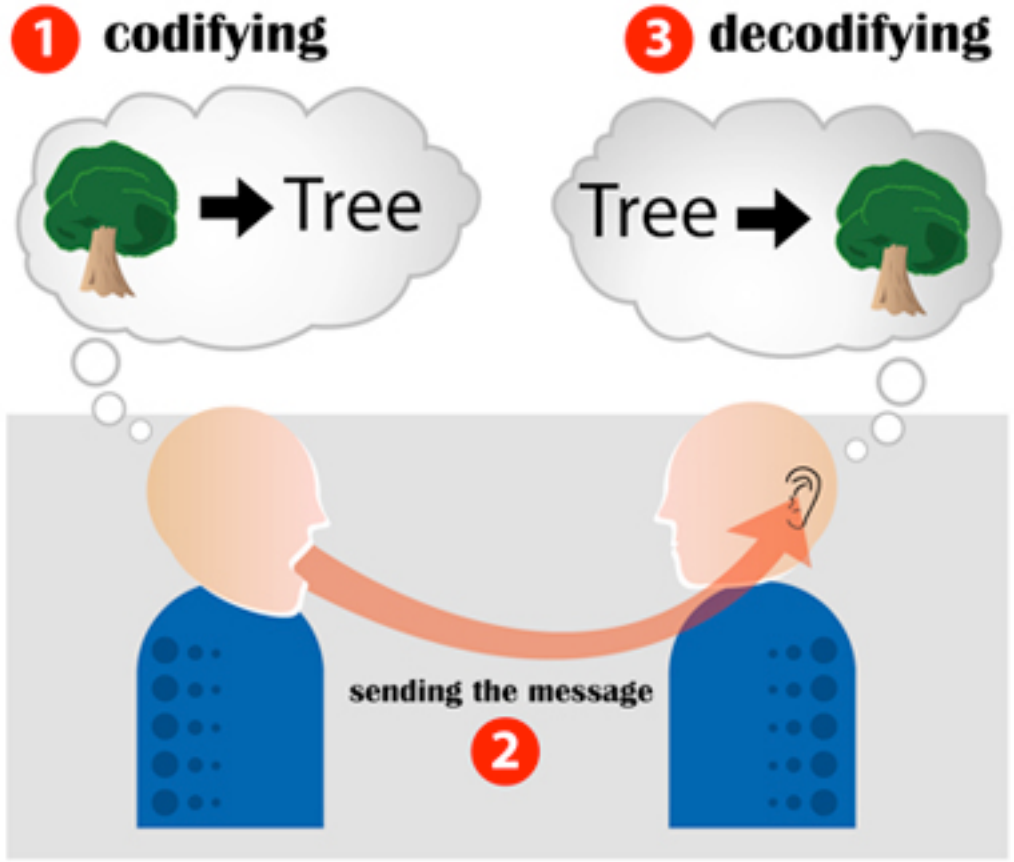
$$\ddot{z} = \frac{1}{M-mt}\{mc_e + F_n(p - p_{atm_e}\mathrm{e}^{-(k/H)(\sqrt{(x^2+y^2+z^2)}-R)})\}\cos\gamma(t) - g_0R^2\frac{z}{(x^2+y^2+z^2)^{3/2}} +$$
$$-\frac{c_n(\sqrt{(\dot{x}^2+\dot{y}^2+\dot{z}^2)},\chi)}{M-mt}\rho_0\,\mathrm{e}^{-(1/H)(\sqrt{(x^2+y^2+z^2)}-R)}F\dot{z}\sqrt{(\dot{x}^2+\dot{y}^2+\dot{z}^2)} + \frac{c_a(\sqrt{(\dot{x}^2+\dot{y}^2+\dot{z}^2)},\chi)}{M-mt}\rho_0\times$$
$$\times\,\mathrm{e}^{-(1/H)(\sqrt{(x^2+y^2+z^2)}-R)}F\frac{\dot{y}\{\dot{y}\cos\gamma(t)-\dot{z}\cos\beta(t)\}-\dot{x}\{\dot{z}\cos\alpha(t)-\dot{x}\cos\gamma(t)\}\sqrt{(\dot{x}^2+\dot{y}^2+\dot{z}^2)}}{\sqrt{[\{\dot{y}\cos\gamma(t)-\dot{z}\cos\beta(t)\}^2+\{\dot{z}\cos\alpha(t)-\dot{x}\cos\gamma(t)\}^2+\{\dot{x}\cos\beta(t)-\dot{y}\cos\alpha(t)\}^2]}}$$

in which: $\chi = \arccos\dfrac{\dot{x}\cos\alpha(t)+\dot{y}\cos\beta(t)+\dot{z}\cos\gamma(t)}{\sqrt{(\dot{x}^2+\dot{y}^2+\dot{z}^2)}}$

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

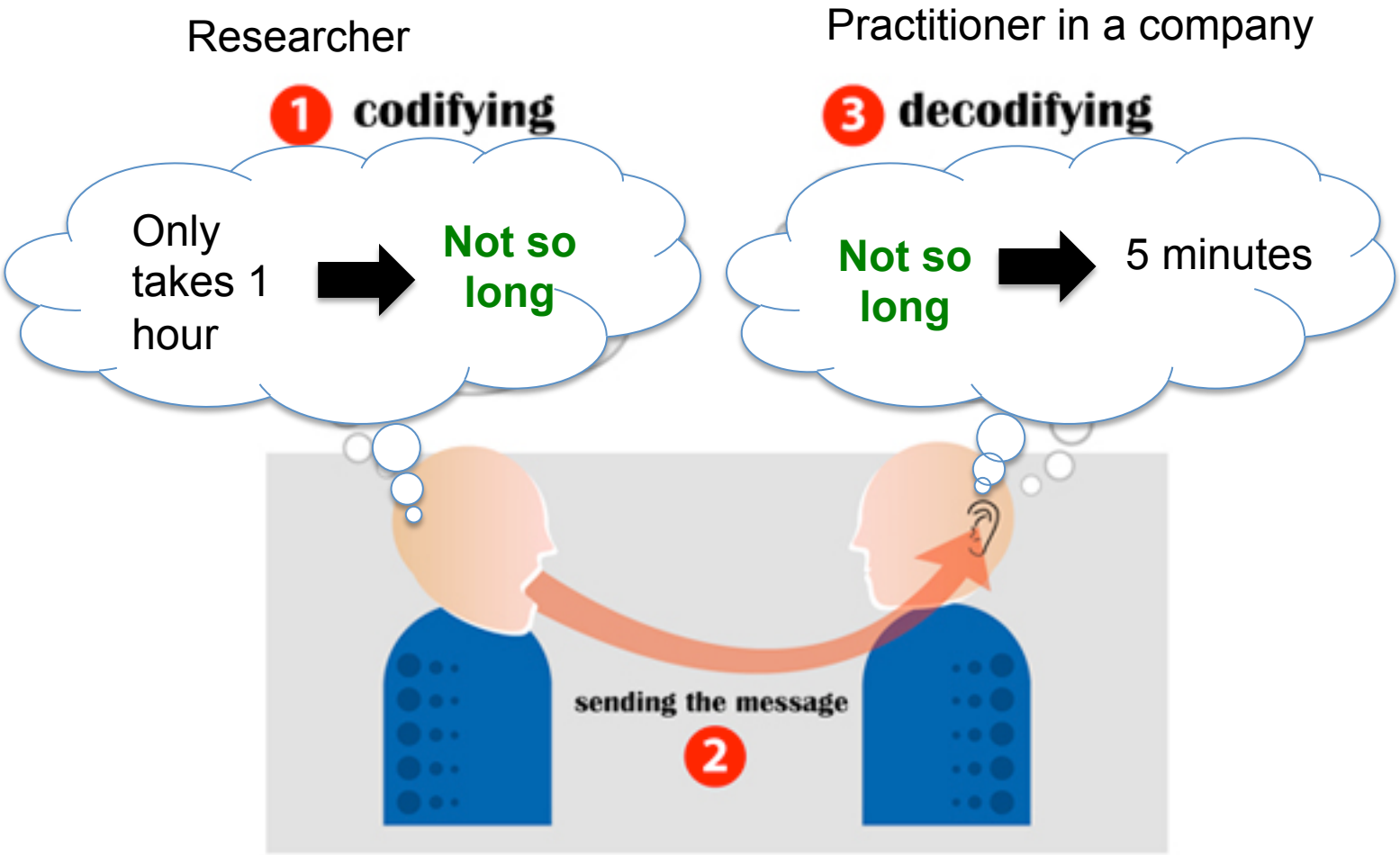
Empirical research with industry = difficult

- But “communication science”-difficult

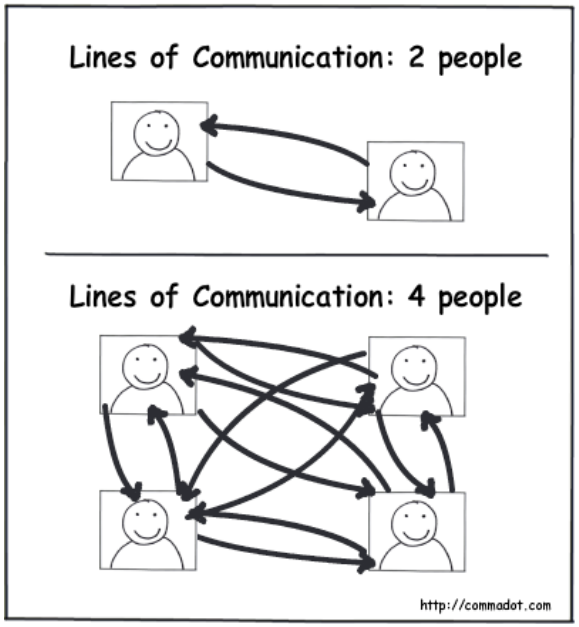


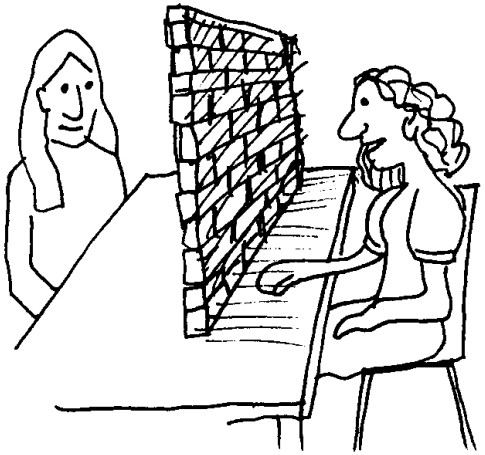
The FITTEST project is funded by the European Commission (FP7-ICT-257574)

“communication science”-difficult

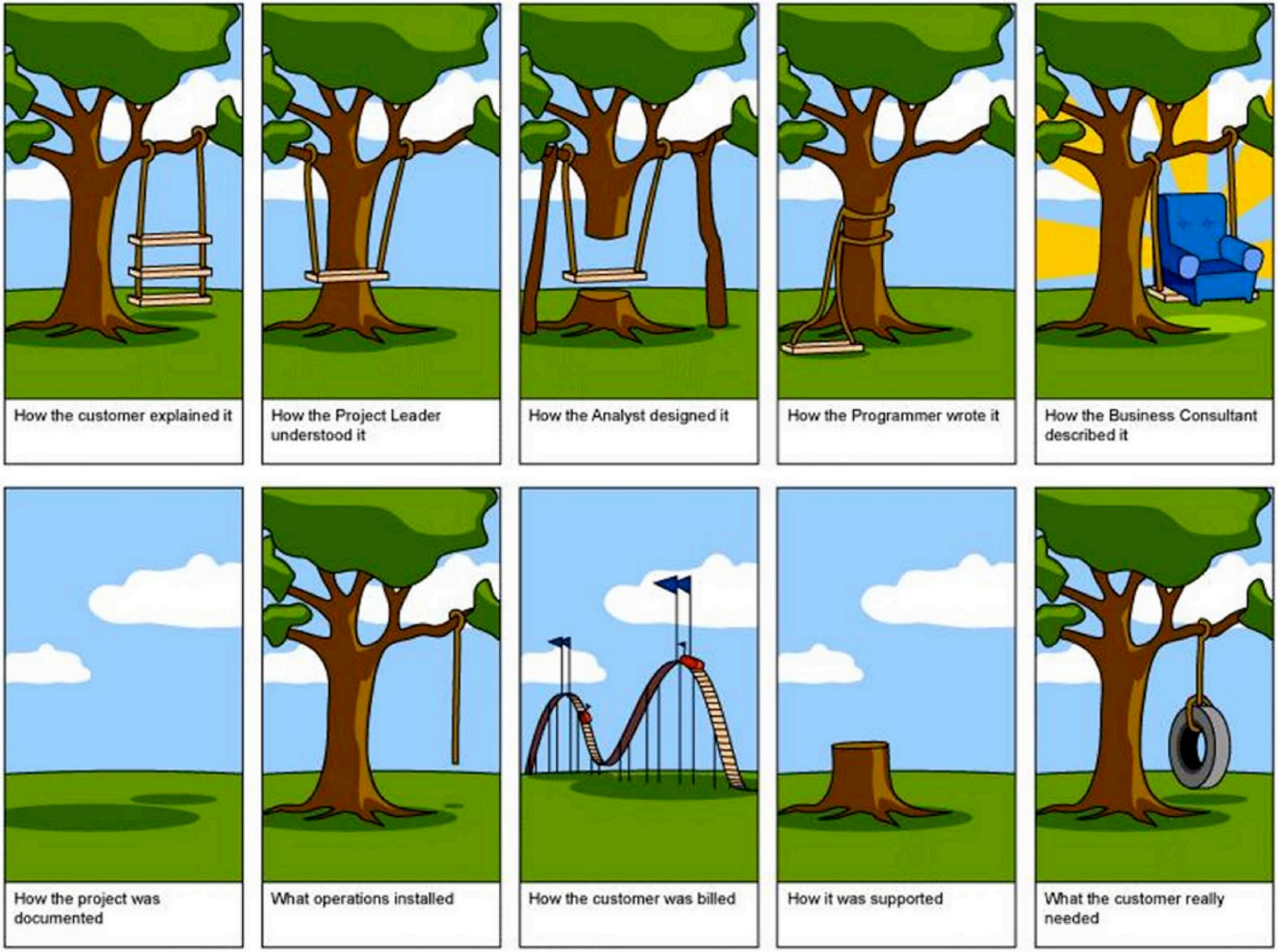


Communication is extremely difficult





The FITTEST project is funded by the European Commission (FP7-ICT-257574)


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Examples…..

Academia

- Wants to empirically evaluate T
- What techniques/tools can we compare with?
- Why don´t you know that?
- That does not take so much time!
- Finding real faults would be great!
- Can we then inject faults?
- How many people can use it?
- Is there historical data?
- But you do have that information?

Industry

- Wants to execute and use T to see what happens.
- We use intuition!
- You want me to know all that?
- That much time!?
- We cannot give this information.
- Not artificial ones, we really need to know if this would work for real faults.
- We can assign 1 person.
- That is confidential.
- Oh.., I thought you did not need that.

With the objective to improve and reduce some barriers

- Use a general methodological framework.
- To use as a vehicle of communication
- To simplify the design
- To make sure that studies can be compared and aggregated

Existing work

- By Lott & Rombach, Eldh, Basili, Do et al, Kitchenham et al
- Describe *organizational* frameworks, i.e.:
  - General steps
  - Warnings when designing
  - Confounding factors that should be minimized
- We pretended to define a methodological framework that defined *how* to evaluate software testing techiques, i.e.:
  - The research questions that can be posed
  - The variables that can be measured
  - Etc.

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

The methodological framework

- Imagine a company *C* wants to evaluate *T* to see whether it is useful and worthwhile to incorporate in its company.

- Components of the Framework (each case study will be an instantiation)
  - **Objectives**: effectiveness, efficiency, satisfaction
  - **Cases** or treatments (= the testing techniques/tools)
  - The **subjects** (= practitioners that will do the study)
  - The **objects** or pilot projects: selection criteria.
  - The **variables** and metrics: which data to collect?
  - **Protocol** that defines how to execute and collect data.
  - How to **analyse** the data
  - **Threats** to validity
  - **Toolbox**

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Definition of the framework – research questions

- **RQ1**: How does *T* contribute to the **effectiveness** of testing when it is being used in real testing environments of *C* and compared to the current practices of *C*?

- **RQ2**: How does *T* contribute to the **efficiency** of testing when it is being used in real testing environments of *C* and compared to the current practices of *C*?

- **RQ3**: How satisfied (**subjective satisfaction**) are testing practitioners of *C* during the learning, installing, configuring and usage of *T* when used in real testing environments?

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Definition of the framework – the cases

- Reused a taxonomy from Vegas and Basili adapted to software testing tools and augmented with results from Tonella
- The case or the testing technique or tool should be described by:
  - **Prerequisites**: type, life-cycle, environment (platform and languages), scalability, input, knowledge needed, experience needed.
  - **Results**: output, completeness, effectiveness, defect types, number of generated test cases.
  - **Operation**: Interaction modes, user guidance, maturity, etc.
  - **Obtaining the tool**: License, cost, support.

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Definition of the framework – subjects

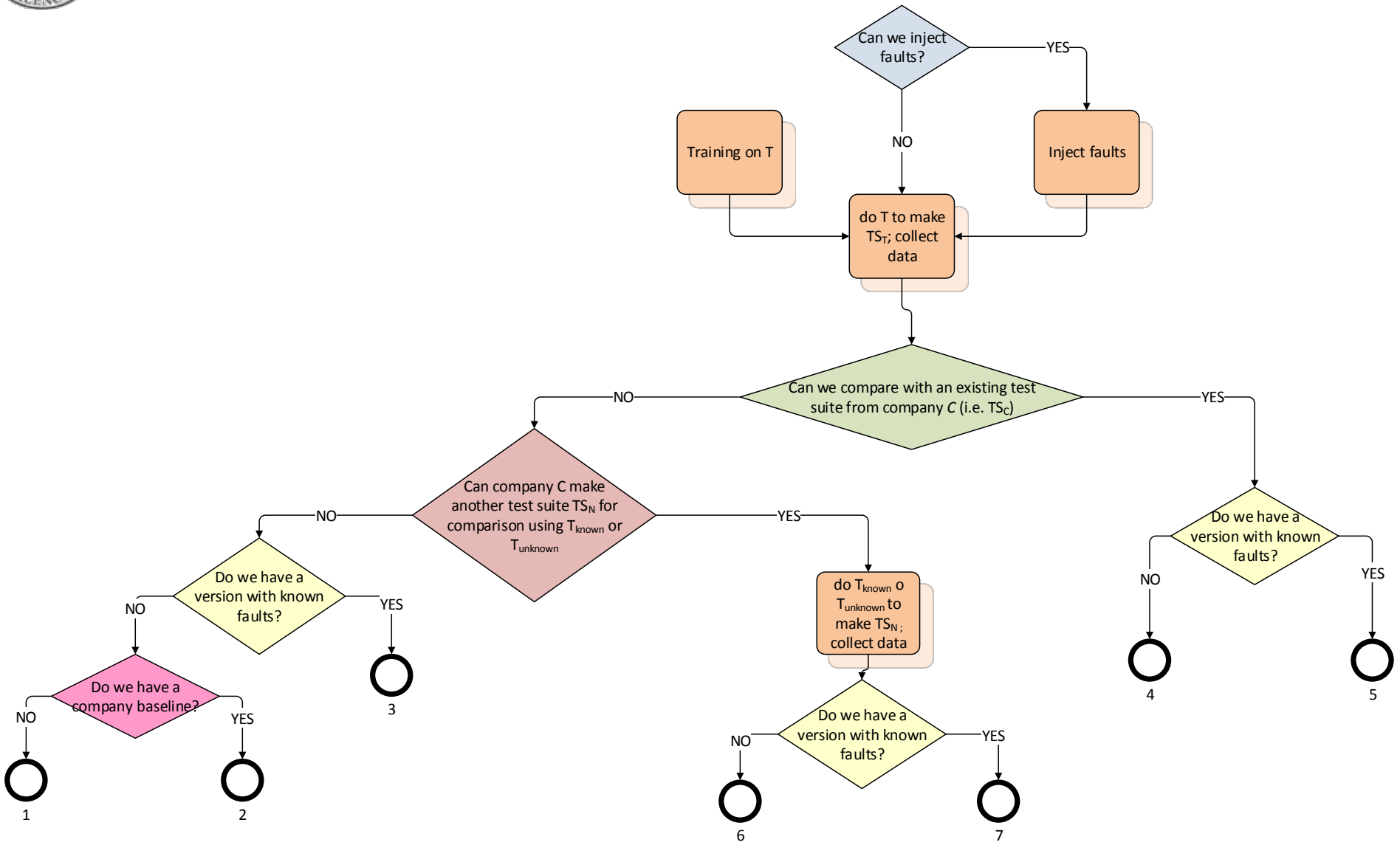
- Workers of C that normally use the techniques and tools with which T is being compared.

Definition of the framework – the objects/pilot projects

- In order to see how we can compare and what type of study we will do, we need answers to the following questions:
  A. Will we have access to a system with known faults? What information is present about these faults?
  B. Are we allowed/able to inject faults into the system?
  C. Does your company gather data from projects as standard practice? What data is this? Can this data be made available for comparison? Do you have a company baseline? Do we have access to a sister project?
  D. Does company C have enough time and resources to execute various rounds of tests?, or more concrete:
     - Is company C willing to make a new testsuite TS_{na} with some technique/tool T_a already used in the company C?
     - Is company C is willing to make a new testsuite TS_{nn} with some technique/ tool T_n that is also new to company C?
     - Can we use an existing testsuite TS_e that we can use to compare? (Do we know the techniques that were used to create that test suite, and how much time it took?)

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Definition of the framework - Protocol



The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Definition of the framework - Scenarios

- Remember:
  - Does company C have enough time and resources to execute various rounds of tests?, or more concrete:
    - Is company C willing to make a new testsuite TS_{na} with some technique/tool T_{al} already used in the company C?
    - Is company C is willing to make a new testsuite TS_{nn} with some technique/tool T_n that is also new to company C?
    - Can we use an existing testsuite TS_e that we can use to compare? (Do we know the techniques that were used to create that test suite, and how much time it took?)
- **Scenario 1** (qualitative assessment only) (Qualitative Effects Analysis)
- **Scenario 2** (Scenario 1 /\ quantitative analysis based on company baseline)
- **Scenario 3** ((Scenario 1 \/ Scenario 2) /\ quantitative analysis of FDR)
- **Scenario 4** ((Scenario 1 \/ Scenario 2) /\ quantitative comparison of T and TS_e)
- **Scenario 5** (Scenario 4 /\ FDR of T and TS_e)
- **Scenario 6** ((Scenario 1 \/ Scenario 2) /\ quantitative comparison of T and (T_a or T_n))
- **Scenario 7** (Scenario 6 /\ FDR of T and (T_a or T_n))

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

If we can inject faults, take care that

1. The artificially seeded faults are ***similar to real faults*** that naturally occur in real programs due to mistakes made by developers.
   - To identify realistic fault types, a history-based approach can be used, i.e. "real" faults can be fetched from the bug tracking system and made sure that these reported faults are an excellent representative of faults that are introduced by developers during implementation.
2. The faults should be injected in code that is ***covered by an adequate number of test cases***
   - e.g., they may be seeded in code that is executed by more than 20 percent and less than 80 percent of the test cases.
3. The faults should be injected "***fairly***," i.e., an adequate number of instances of each fault type is seeded.

Definition of the framework – data to collect

- **Effectiveness**
  - Number of test cases designed or generated.
  - How many invalid test cases are generated.
  - How many repeated test cases are generated
  - Number of failures observed.
  - Number of faults found.
  - Number of false positives (The test is marked as Failed, when the functionality is working).
  - Number of false negatives (The test is marked as Passed, when the functionality is not working).
  - Type and cause of the faults that were found.
  - Estimation (or when possible measured) of coverage reached.
- Efficiency
- Subjective satisfaction

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Definition of the framework – data to collect

- Effectiveness

- **Efficiency**
  - Time needed to **learn** the testing method.
  - Time needed to **design** or generate the test cases.
  - Time needed to **set up** the testing infrastructure (install, configure, develop test drivers, etc.) (quantitative). (Note: if software is to be developed or other mayor configuration/ installation efforts, it might be a good idea to maintain working diaries).
  - Time needed to **test** the system and observe the failure (i.e. planning, implementation and execution) in hours (quantitative).
  - Time needed to **identify** the fault type and cause for each observed failure (i.e. time to isolate) (quantitative).

- Subjective satisfaction

Definition of the framework – data to collect

- Effectiveness
- Efficiency
- **Subjective satisfaction**
  - SUS score (10 question **questionnaire** with 5 likert-scale and a total score)
  - 5 reactions (through **reaction cards**) that will be used to create a word cloud and ven diagrams)
  - Emotional face reactions during **semi-structured interviews** (faces will be evaluated on a 5 likert-scale from “not at all like this” to “very much like this”).
  - Subjective opinions about the tool.

A mixture of methods for evaluating subjective satisfaction

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

System Usability Scale (SUS)

www.usability.serco.com/trump/documents/Suschapt.doc

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

Why SUS

- studies (e.g. [TS04, BKM08]) have shown that this simple questionnaire gives most reliable results.
- SUS is technology agnostic, making it flexible enough to assess a wide range of interface technologies.
- The survey is relatively quick and easy to use by both study participants and administrators.
- The survey provides a single score on a scale that is easily understood by the wide range of people (from project managers to computer programmers) who are typically involved in the development of products and services and who may have little or no experience in human factors and usability.
- The survey is nonproprietary, making it a cost effective tool as well.

SUS Scoring

- SUS yields a single number representing a composite measure of the overall usability of the system being studied. Note that scores for individual items are not meaningful on their own.
- To calculate the SUS score, first sum the score contributions from each item.
  - Each item's score contribution will range from 0 to 4.
  - For items 1, 3, 5, 7 and 9 the score contribution is the scale position minus 1.
  - For items 2,4,6,8 and 10, the contribution is 5 minus the scale position.
  - Multiply the sum of the scores by 2.5 to obtain the overall value of SU.
- SUS scores have a range of 0 to 100.

SUS is not enough…

- In a literature review of 180 published usability studies, Hornbaek [Horn06] concludes that measures of satisfaction should be extended beyond questionnaires.

- So we add more….

Reaction Cards

The complete set of 118 Product Reaction Cards				
Accessible	Creative	Fast	Meaningful	Slow
Advanced	Customizable	Flexible	Motivating	Sophisticated
Annoying	Cutting edge	Fragile	Not Secure	Stable
Appealing	Dated	Fresh	Not Valuable	Sterile
Approachable	Desirable	Friendly	Novel	Stimulating
Attractive	Difficult	Frustrating	Old	Straight Forward
Boring	Disconnected	Fun	Optimistic	Stressful
Business-like	Disruptive	Gets in the way	Ordinary	Time-consuming
Busy	Distracting	Hard to Use	Organized	Time-Saving
Calm	Dull	Helpful	Overbearing	Too Technical
Clean	Easy to use	High quality	Overwhelming	Trustworthy
Clear	Effective	Impersonal	Patronizing	Unapproachable
Collaborative	Efficient	Impressive	Personal	Unattractive
Comfortable	Effortless	Incomprehensible	Poor quality	Uncontrollable
Compatible	Empowering	Inconsistent	Powerful	Unconventional
Compelling	Energetic	Ineffective	Predictable	Understandable
Complex	Engaging	Innovative	Professional	Undesirable
Comprehensive	Entertaining	Inspiring	Relevant	Unpredictable
Confident	Enthusiastic	Integrated	Reliable	Unrefined
Confusing	Essential	Intimidating	Responsive	Usable
Connected	Exceptional	Intuitive	Rigid	Useful
Consistent	Exciting	Inviting	Satisfying	Valuable
Controllable	Expected	Irrelevant	Secure	
Convenient	Familiar	Low Maintenance	Simplistic	

Emotional face reactions

- The idea is to elicit feedback about the product, particularly emotions that arose for the participants while talking about the product (e.g. frustration, happiness).
- We will video tape the users when they respond to the following two questions during a semi-structured interview.
  - Would you recommend this tool it to other colleagues?
    - If not why
    - If yes what arguments would you use
  - Do you think you can persuade your management to invest in a tool like this?
    - If not why
    - If yes what arguments would you use

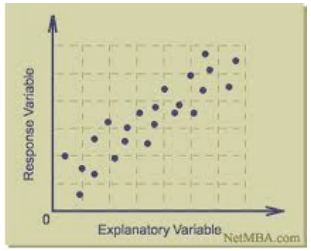
Not at all like this						Very much like this
1	2	3	4	5	6	7

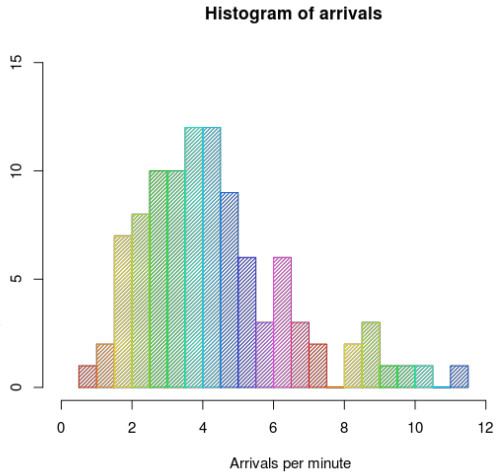
Analysing and interpreting the data

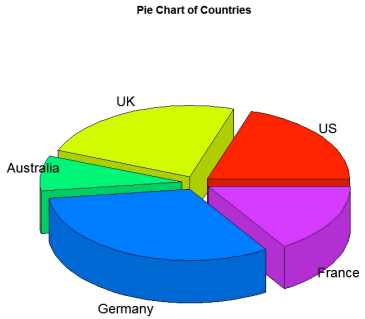
- Depends on the amount of data we have.
- If we only have 1 value for each variable, no analysis techniques are available and we just present and interpret the data.
- If we have sets of values for a variable then we need to use statistical methods
  - Descriptive statistics
  - Statistical tests (or significance testing). In statistics a result is called statistically significant if it has been predicted as unlikely to have occurred by chance alone, according to a pre-determined threshold probability, the significance level.
- Evaluating SBST tools, we will always have sets of values for most of the variables to deal with randomness

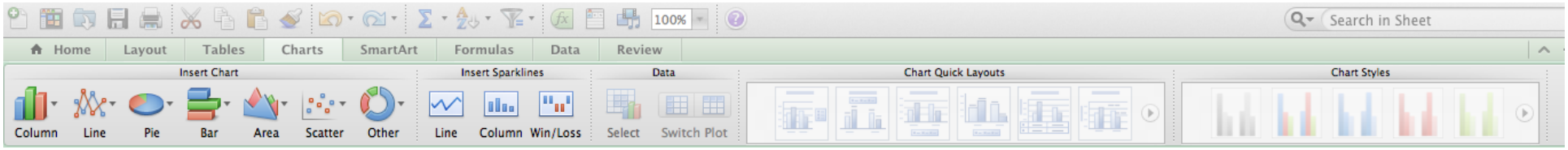
Descriptive statistics

- Mean, median, middle, standard deviation, frequency, correlation, etc
- Graphical visualisation: scatter plot, box plot, histogram, pie charts



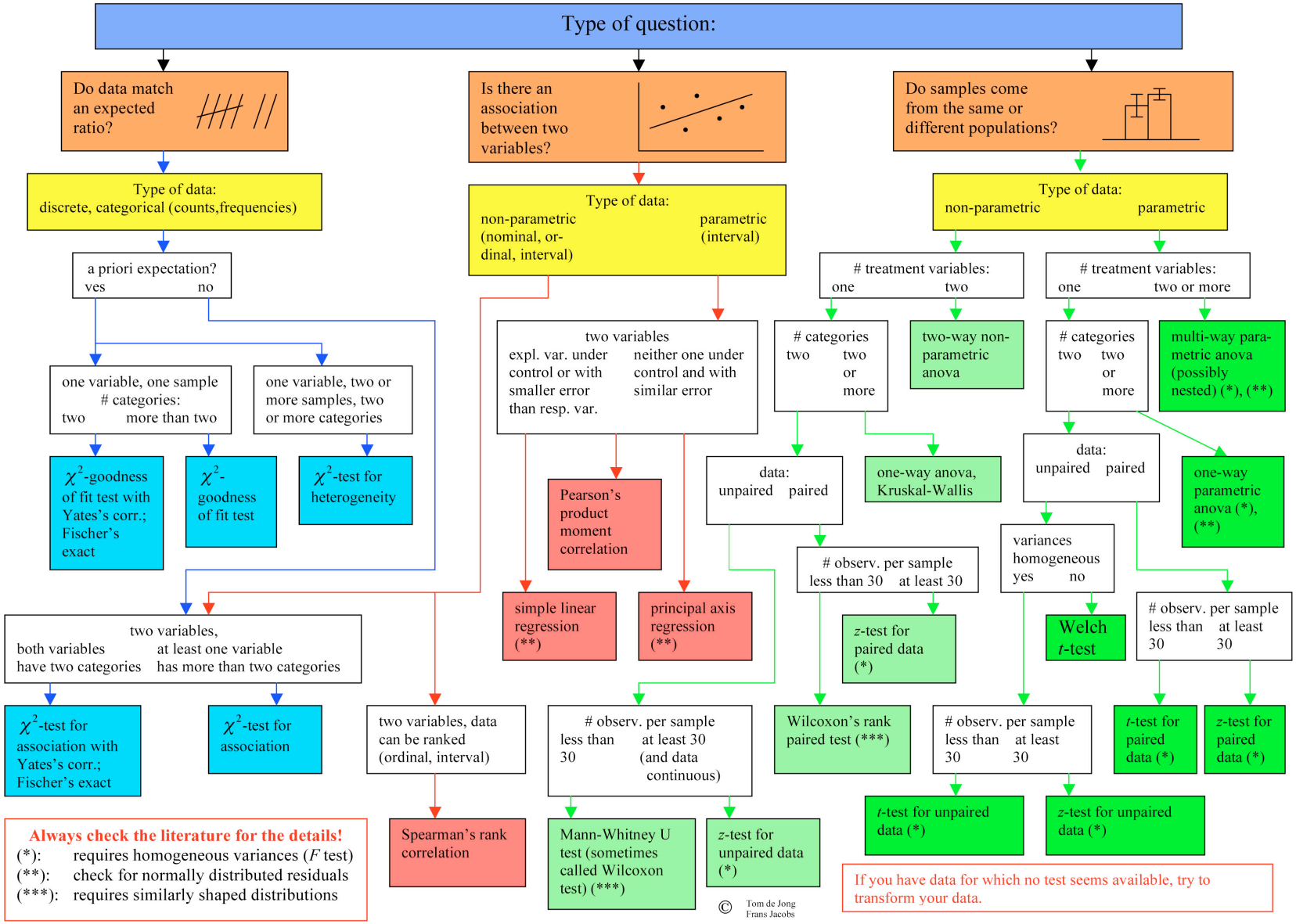






Statistical test

Most important is finding te right method



http://science.leidenuniv.nl/index.php/ibl/pep/people/Tom_de_Jong/Teaching

Definition of the framework - Threats

- Threats to validity (of confounding factors) have to be minimized
- These are the effects or situations that might jeopardize the validity of your results….
- Those that cannot be prevented have to be reported
- When working with people we have to consider many sociological effects:
- Let us just look at a couple well known ones to give you an idea….
  - The learning curve effect
  - The Hawthorne effect
  - The Placebo effect
  - Etc….

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

The learning curve effect

- When using new methods/tools people gain familiarity with their application over time (= learning curve).
  - Initially they are likely to them more ineffectively than they might after a period of familiarisation/learning.
- Thus the learning curve effect will tend to counteract any positive effects inherent in the new method/tool.
- In the context of evaluating methods/tools there are two basic strategies to minimise the learning curve effect (that are not mutually exclusive):
  1. Provide appropriate training before undertaking an evaluation exercise;
  2. Separate pilot projects aimed at gaining experience of using a method/ tool from pilot projects that are part of an evaluation exercise.

The Hawthorn effect

- When an evaluation is performed, staff working on the pilot project(s) may have the perception that they are working under more management scrutiny than normal and may therefore work more conscientiously.
- Name comes from Hawthorne aircraft factory (lights low or high?).
- The Hawthorn effect would tend to exaggerate positive effects inherent in a new method/tool.
- A strategy to minimise the Hawthorne effect is to ensure that a similar level of management scrutiny is applied to control projects in your case study (i.e. project(s) using the current method/tool) as is applied to the projects that are using the new method/tool.

The placebo effect

- In medical research, patients who are deliberately given ineffectual treatments recover if they believe that the treatment will cure them.
- Also a software engineer who believes that adopting some practice (i.e., wearing a pink t-shirt) will improve the reliability of his code may succeed in producing more reliable code.
- Such a result could not be generalised.
- In medicine, placebo effect is minimized by not informing the subjects.
- This cannot be done in the context of testing tool evaluations.
- When evaluating methods and tools the best you can do is to:
  - Assign staff to pilot projects using your normal project staffing methods and hope that the actual selection of staff includes the normal mix of enthusiasts, cynics and no-hopers that normally comprise your project teams.
  - Make a special effort to avoid staffing pilot projects with staff who have a vested interest in the method/tool (i.e. staff who developed or championed it) or a vested interest in seeing it fail (i.e. staff who really hate change rather than just resent it).
- This is a bit like selecting a jury. Initially the selection of potential jurors is at random, but there is additional screening to avoid jurors with identifiable bias.

Definition of the framework - Threats

- There are many more factors that all need to be identified
- Not only the people but also the technology:
  - Did the measurement tools (i.e. Coverage) really measure what we thoughts
  - Are the injected faults really representative?
  - Were the faults injected fairly?
  - Is the pilot project and software representative?
  - Were the used oracles reliable?
  - Etc….

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Definition of the framework - Toolbox

- Toolbox
  - **Demographic questionnaire**: This questionnaire must be answered by the testers before performing the test. This questionnaire aims to obtain the features of the testers: level of experience in using the tool, years, job, knowledge of similar tools,
  - **Satisfaction questionnaire SUS**: Questionnaire in order to extract the testers' satisfaction when they perform the evaluation.
  - **Reaction cards**
  - **Questions for (taped) semi-structured interviews**
  - **Process for investigating the face reactions in videos**
  - An **fault taxonomy** to classify the found fault.
  - A **software testing technique and tools classification/taxonomy**
  - **Working diaries**
  - **A fault template** to classify each fault detected by means of the test. The template contains the following information:
    - Time spent to detect the fault
    - Test case that found the fault
    - Cause of the fault: mistake in the implementation, mistake in the design, mistake in the analysis.
    - Manifestation of the fault in the code

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Applying or instantiating the framework

- Search Based Structural Testing tool [Vos et.al. 2012]
- Search Based Functional Testing tool [Vos et. Al. 2013]
- Web testing techniques for AJAX applications [MRT08]
- Commercial combinatorial testing tool at Sulake (to be presented at ISSTA workshop next week)
- Automated Test Case Generation at IBM (has been send to ESEM)

- We have finished other instances:
  - Commercial combinatorial testing tool at SOFTEAM (has been send to ESEM)

- Currently we are working on more instantiations:
  - Regression testing priorization technique
  - Continuous testing tool at SOFTEAM
  - Rogue User Testing at SOFTEAM
  - …

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

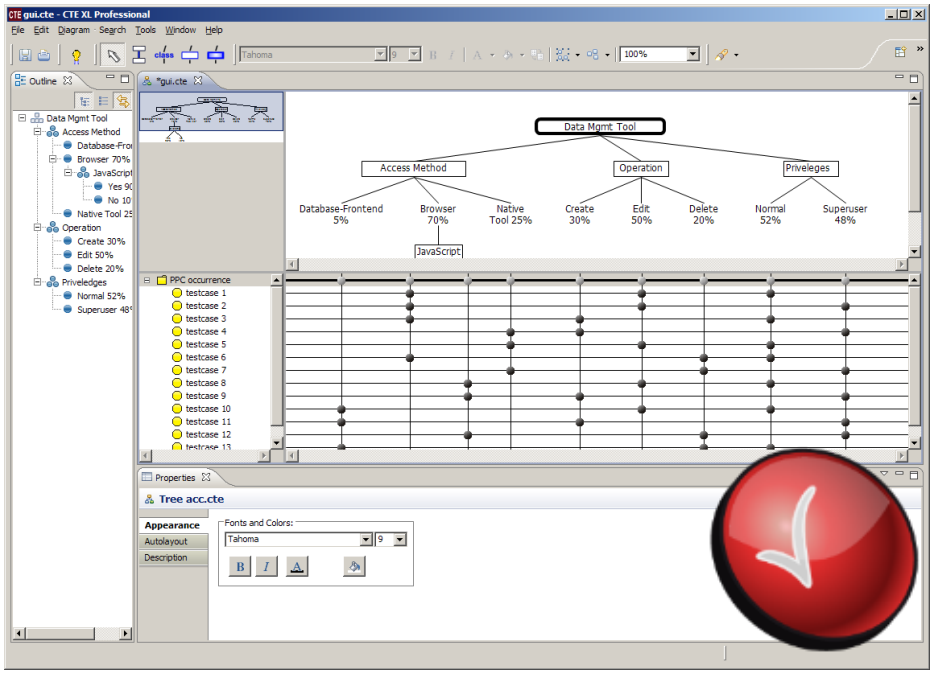
Combinatorial Testing example case study Sulake: Context

- Sulake is a Finish company
- Develops social entertainment games
- Main product: Habbo hotel
  - World's largest virtual community for teenagers
  - Millions of teenagaers a week all over the world (ages 13-18)
  - Access direct through the browser or facebook
  - 11 languages available
  - 218.000.000 registered users
  - 11.000.000 visitors / month
- System can be accessed through wide variety of browsers, flashplayers (and their versions) than run on different operating systems!
- Which combinations to use when testing the system?!

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Can a tool help?

- What about the CTE XL Profesional or CTE for short?
- Combinatorial Tree Editor:
  - Model your combinatorial problem in a tree
  - Indicate the priorities of the combinations
  - The tool automatically generated the best test cases!



Combinatorial Testing
example case study Sulake: Research Questions

- What do we want to find out?
- Research questions:
  - **RQ1**: Compared to the current test suites used for testing in Sulake, can the test cases generated by the CTE contribute to the effectiveness of testing when it is used in real testing environments at Sulake?
  - **RQ2**: How much effort would be required to introduce the CTE into the testing processes currently implanted at Sulake?
  - **RQ3**: How much effort would be required to add the generated test cases into the testing infrastructure currently used at Sulake?
  - **RQ4**: How satisfied are Sulake testing practitioners during the learning, installing, configuring and usage of CTE when it is used in real testing environment

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

The testing tools evaluated
(the cases or treatments)

- Current combinatorial testing practice at Sulake
  - Exploratory testing with feature coverage as objective
  - Based on real user information(i.e. browsers, OS, Flash, etc.) combinatorial aspects are taken into account

COMPARED TO

- Classification Tree Editor (CTE)
  - Classify the combinatorial aspects as a classification tree
  - Generate prioritized test cases and select

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Who is doing the study (the subjects)

- Subjects: 1 senior tester from Sulake (6 years sw devel, 8 years testing experience)

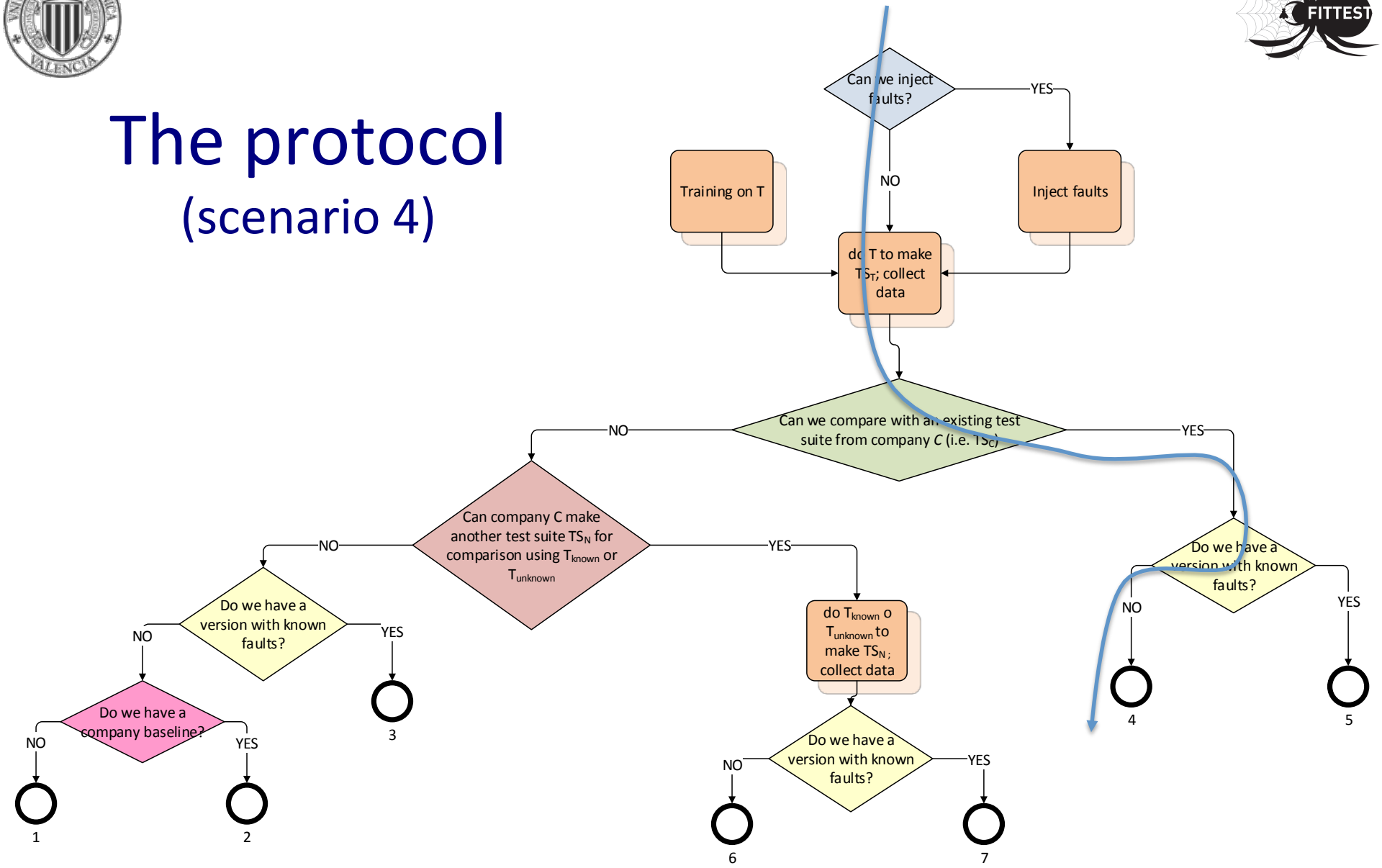
The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Systems Under Test
(objects or pilot projects)

- Objects:
  - 2 nightly builds from Habbo
  - Existing test suite that Sulake uses (TS_{sulake}) with 42 automated test cases
  - No known faults, no injection of faults

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

The protocol
(scenario 4)



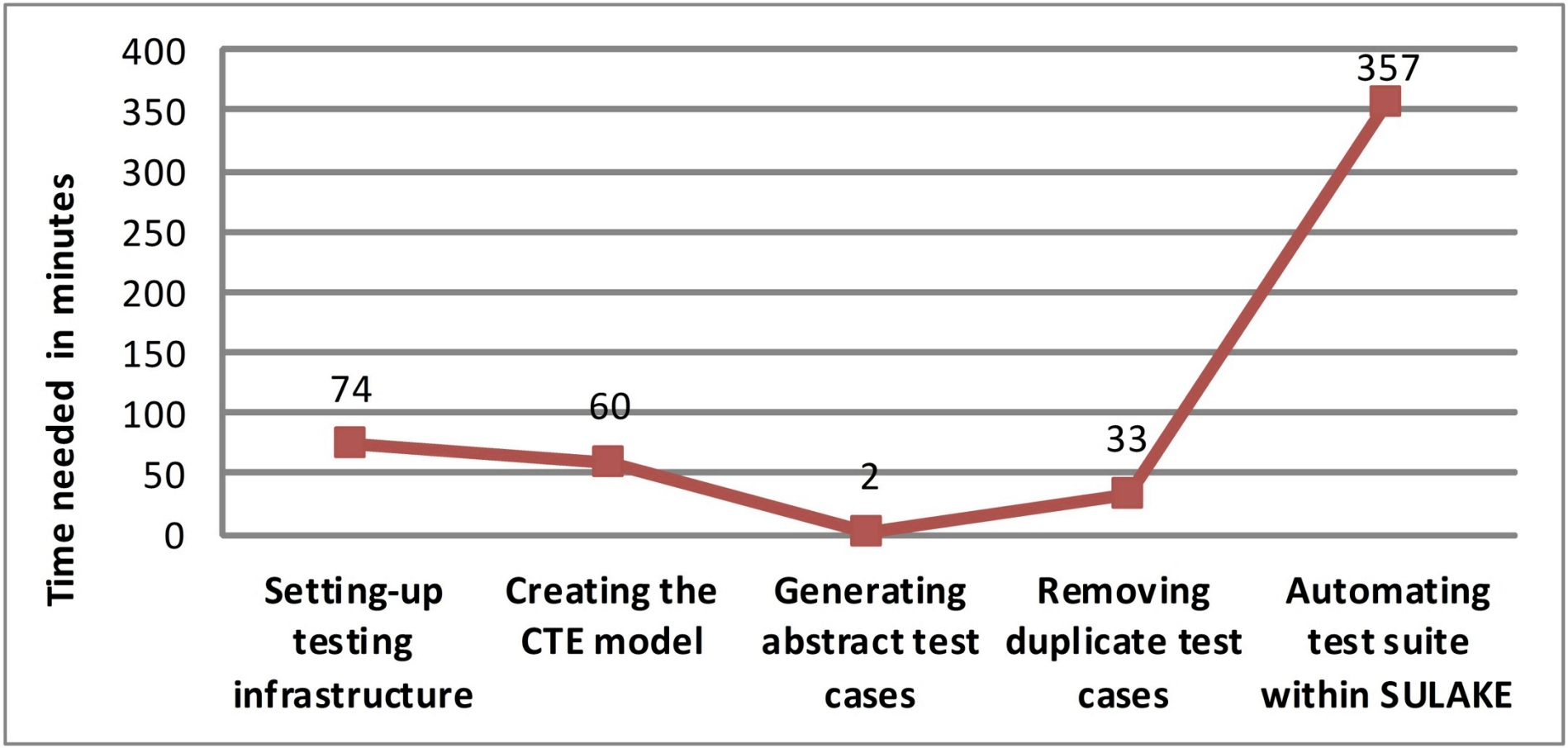
The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Collected data

Variables	TS_{Sulake}	TS_{CTE}
Measuring effectiveness:		
Number of test cases	42	68 (selected 42 high priority)
Number of invalid test cases	0	0
Number of repeated test cases	0	26
Number of failures observed	0	12
Number of fauls found	0	2
Type and cause of the faults	N/A	1. critical, browser hang 2. minor, broken UI element
Feature coverage reached	100%	40%
All pairs coverage reached	N/A	80%
Measuring efficiency:		
Time needed to learn the CTE testing method	N/A	116 min
Time needed to design and generate the test suite with the CTE	N/A	95 min (62 for tree, 33 for removing duplicates)
Time needed to setup testing infrastructure specific to CTE	N/A	74 min
Time needed to automate the test suite generated by the CTE	N/A	357 min
Time needed to execute the test suite	114min	183 min (both builds)
Time needed to identify fault types and causes	0min	116 min
Measuring subjective satisfaction		
SUS	N/A	50
Reaction cards	N/A	Comprehensive, Dated, Old, Sterile, Unattractive
Informal interview	N/A	video

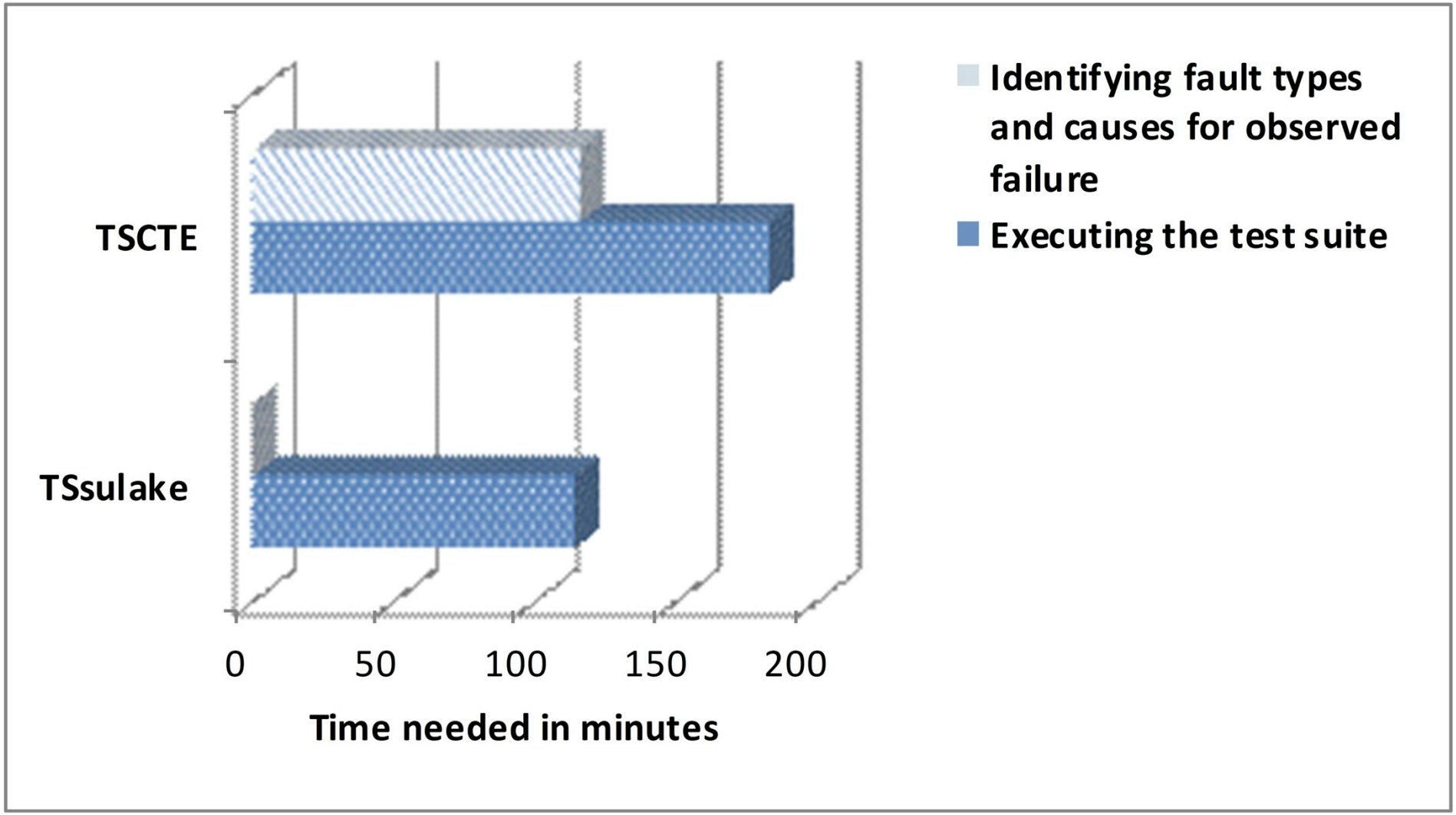
The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Descriptive statistics for efficiency (1)



The FITTEST project is funded by the European Commission (FP7-ICT-257574)

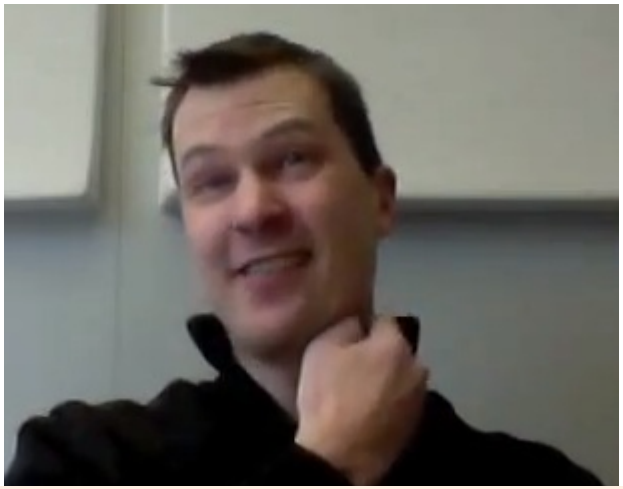
Descriptive statistics for efficiency (2)



The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Emotional face reactions

1.

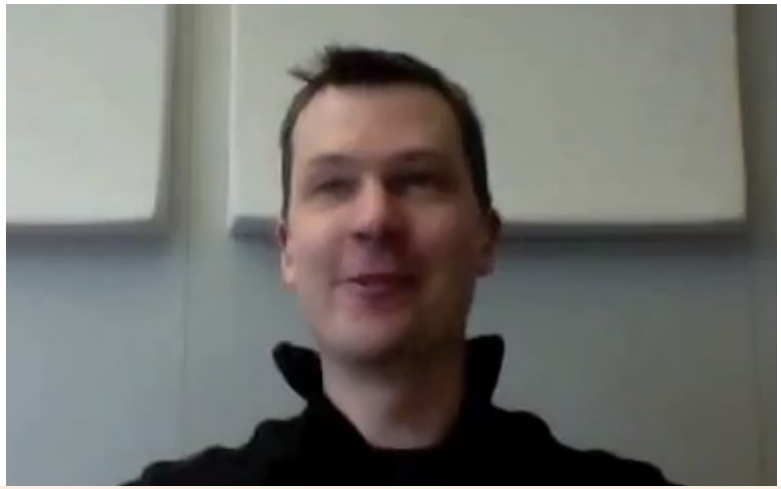


Duplicates Test Cases

2.

Readability of the CTE trees

3.



Technical support and user manuals

4.

Appearance of the tool

Combinatorial Testing example case study Sulake: Conclusions

- RQ1: Compared to the current test suites used for testing in Sulake, can the test cases generated by the CTE contribute to the effectiveness of testing when it is used in real testing environments at Sulake?
  - 2 new faults were found!!
  - The need that more structured combinatorial testing is necessary was confirmed by Sulake.
- RQ2: How much effort would be required to introduce the CTE into the testing processes currently implanted at Sulake?
  - Effort for learning and installing is medium, but can be justified within Sulake being only once
  - Designing and generating test cases suffers form duplicates that costs time to be removed. Total effort can be accepted within Sulake because critical faults were discovered.
  - Executing the test suite generated by the CTE takes 1 hour more that Sulake test suite (due to more combinatorial aspects being tested). This means that Sulake cannot include these tests in daily build, but will have to add them to nightly builds only.
- RQ3: How much effort would be required to add the generated test cases into the testing infrastructure currently used at Sulake?
  - Effort for automating the generated test cases, but can be justified within Sulake.
- RQ4: How satisfied are Sulake testing practitioners during the learning, installing, configuring and usage of CTE when it is used in real testing environment.
  - Seems to have everything that is needed but looks unattractive.

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Automated Test Case Generation example case study IBM: context

- IBM Research Labs in Haifa, Israel
- Develop a system (denoted IMP ;-) for resource management in a networked environment (servers, virtual machines, switches, storage devices, etc.)
- IBM Lab has a designated team that is responsible for testing new versions of the product.
- The testing is being done within a simulated testing environment developed by this testing team
- IBM Lab is interested in evaluating the Automated Test Case Generation tools developed in the FITTEST project!

Automated Test Case Generation example case study IBM: the research questions

- What do we want to find out?
- Research questions:
  - **RQ1**: Compared to the current test suite used for testing at IBM Research, can the FITTEST tools contribute to the effectiveness of testing when it is used in real testing environments at IBM?
  - **RQ2**: Compared to the current test suite used for testing at IBM Research, can the FITTEST tools contribute to the efficiency of testing when it is used in real testing environments at IBM?
  - **RQ3**: How much effort would be required to deploy the FITTEST tools within the testing processes currently implanted at IBM Research?

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

The testing tools evaluated
(the cases or treatments)

- Current test case design practice at IBM
  - Exploratory test case design
  - The objective of test cases is maximise the coverage of the system use-cases

COMPARED TO

- FITTEST Automated Test Case Generation tools
  - Only part of the whole continuous testing approach of FITTEST

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

FITTEST

FSM2Tests

- Takes FSMs and a Domain Input Specification (DIS) file created by a tester for the IBM Research SUT to generate concrete test cases.
- This component implements a technique that combines model-based and combinatorial testing (see [NMT12]):
  1. generate test paths from the FSM (using various simple and advanced graph visit algorithms)
  2. transform these paths into classification trees using the CTE XL format, enriched with the DIS such as data types and partitions;
  3. Generate test combinations from those trees using combinatorial criteria.

Logs2FSM

- Infers FSM models from the logs
- Applying an event-based model inference approach (read more in [MTR08]).
- The model-based oracles that also result from this tool refer to the use of the paths generated from the inferred FSM as oracles. If these paths, when transformed to test cases, cannot be fully executed, then the tester needs to inspect the failing paths to see if that is due to some faults, or the paths themselves are infeasible.

Who is doing the study (the subjects)

- Subjects:
  - 1 senior tester from IBM (10 years sw devel, 5 years testing experience of which 4 years with IMP system)
  - 1 researcher from FBK (10 years of experience with sw development, 5 years of experience with research in testing)

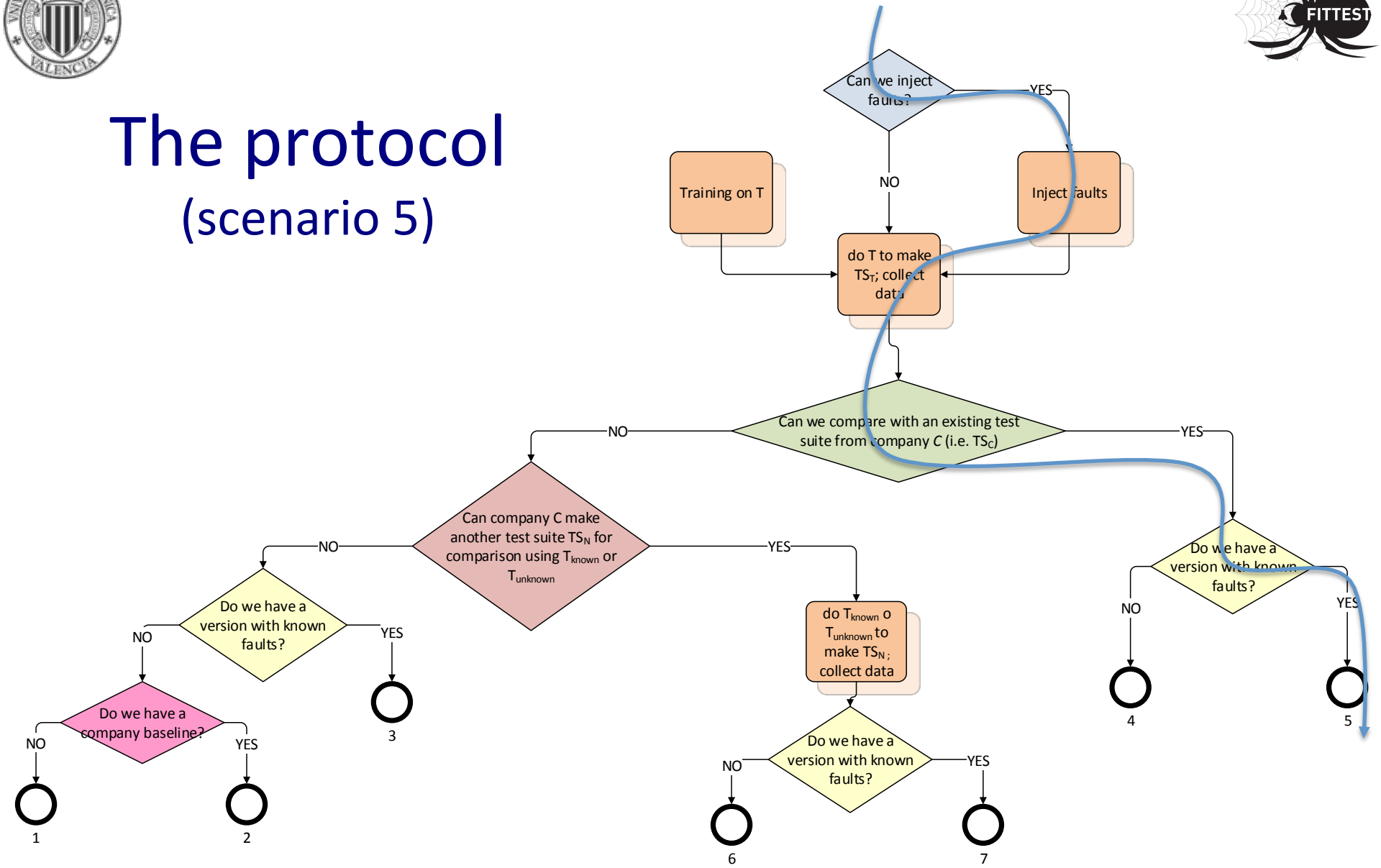
The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Pilot project

- Objects:
  - SUT: Distributed application for managing system resources in a networked environment
    - A management server that communicates with multiple managed clients (= physical or virtual resources)
    - Important product for IBM with real customers
    - Case study will be performed on a new version of this system
  - Existing test suite that IBM uses (TS_{ibm})
    - Selected from what they call System Validation Tests (SVT) -> tests for high level complex costumer use-cases
    - Manually designed
    - Automatically executed through activation scripts
  - 10 representative faults to inject into the system

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

The protocol
(scenario 5)



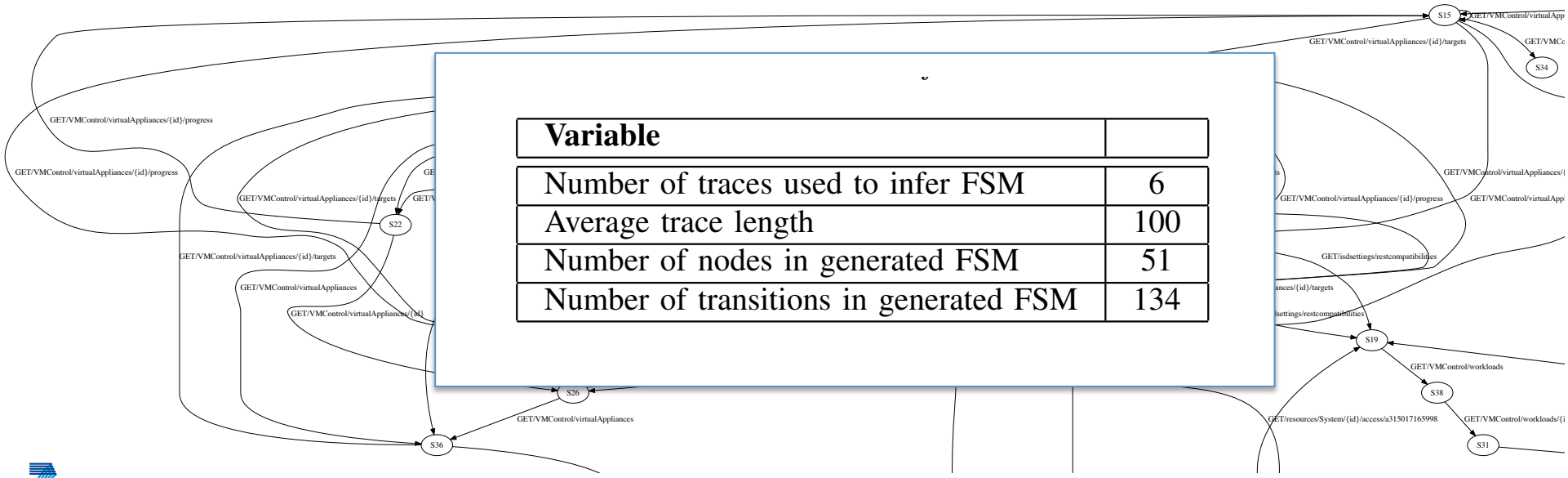
The FITTEST project is funded by the European Commission (FP7-ICT-257574)

More detailed steps

1. Configure the simulated environment and create the logs [IBM]
2. Select test suite TS_{ibm} [IBM]
3. Write activation scripts for each test case in TS_{ibm} [IBM]
4. Generate $TS_{fittest}$ [FBK subject]
   a. Instantiate FITTEST components for the IMP
   b. Generate the FSM with Logs2FSM
   c. Define the Domain Input Specification (DIS)
   d. Generate the concrete test data with FSM2Tests
5. Select and inject the faults [IBM]
6. Develop a tool that transforms the concrete test cases generated by the FITTEST tool FSM2Tests to an executable format [IBM]
7. Execute TS_{ibm} [IBM]
8. Execute $TS_{fittest}$ [IBM]

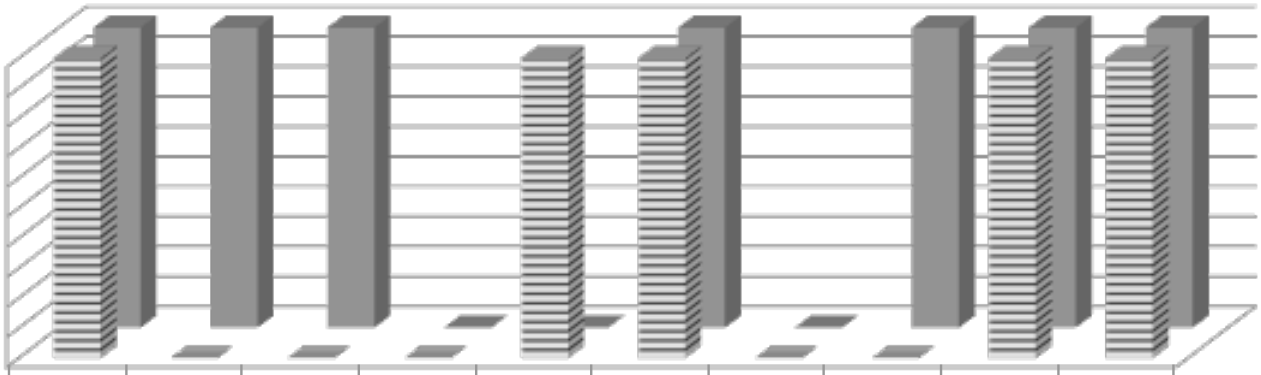
Collected Measures

	TS_{ibm}		$TS_{fittest}$	
size				
number of abstract test cases	NA		84	
number of concrete test cases	4		3054	
number of commands (or events)	1814		18520	
construction				
design of the test cases	manual cf. Section III-B1		automated cf. Section III-B2	
effort				
effort to create the test suite	design	5 hours	set up FITTEST tools	8 hours
	activation scripts	4 hours	generate the FSM	automated, less than 1 second CPU time
			specify the DIS	2 hours
			generate concrete tests	automated, less than 1 minute CPU time
			transform into executable format	20 hours



Variable	
Number of traces used to infer FSM	6
Average trace length	100
Number of nodes in generated FSM	51
Number of transitions in generated FSM	134

Collected measures



	IF1	IF2	IF3	IF4	IF5	IF6	IF7	IF8	IF9	IF10
TS_ibm	1	0	0	0	1	1	0	0	1	1
TS_fittest	1	1	1	0	0	1	0	1	1	1

Variable	**TS**$_{ibm}$	normalized by size	**TS**$_{fittest}$	normalized by size
Execution Time with fault injection	36.75	9.18	127.87	1.52
Execution Time without fault injection	27.97	6.99	50.72	0.60

Automated Test Case Generation example case study IBM: Conclusions

- **RQ1**: Compared to the current test suite used for testing at IBM Research, can the FITTEST tools contribute to the effectiveness of testing when it is used in real testing environments at IBM?
  - TS_{ibm} finds 50% of injected faults, $TS_{fittest}$ finds 70%
  - Together they find 80%! -> IBM will consider combining the techniques
- **RQ2**: Compared to the current test suite used for testing at IBM Research, can the FITTEST tools contribute to the efficiency of testing when it is used in real testing environments at IBM?
  - FITTEST test cases execute faster because they are smaller
  - Shorter tests was good for IBM -> easier to identify faults
- **RQ3**: How much effort would be required to deploy the FITTEST tools within the testing processes currently implanted at IBM Research?
  - Found reasonable by IBM considering the fact that manual tasks need to be done only once and more faults were fund.

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

Threats to validity

- The learning curve effect
- (an basically all the other human factors ;-)
- Missing information in the logs leads to weak FSMs.
- Incomplete specification of the DIS lead to weak concrete test cases.
- The representativeness of the injected faults to real faults.

Final Things ……

- As researchers, we should concentrate on future problems the industry will face. ¿no?
- How can we claim to know future needs without understanding current ones?
- Go to industry and evaluate your results!

Need any help?
More information?
Want to do an instantiation?

Contact:

Tanja E. J. Vos

email: tvos@pros.upv.es
skype: tanja_vos
web: http://tanvopol.webs.upv.es/
project: http://www.facebook.com/FITTESTproject
telephone: +34 690 917 971

The FITTEST project is funded by the European Commission (FP7-ICT-257574)

References

empirical research in software engineering

[**BSH86**] V R Basili, R W Selby, and D H Hutchens. Experimentation in software engineering. IEEE Trans. Softw. Eng., 12:733–743, July 1986.

[**BSL99**] Victor R. Basili, Forrest Shull, and Filippo Lanubile. Building knowledge through families of experiments. IEEE Trans. Softw. Eng., 25(4):456–473, 1999.

[**CWH05**] M. Host C. Wohlin and K. Henningsson. Empirical research methods in software and web engineering. In E. Mendes and N. Mosley, editors, Web Engineering - Theory and Practice of Metrics and Measurement for Web Development, pages 409– 429, 2005.

[**Fly04**] Bent Flyvbejrg. Five misunderstandings about case-study research. In Jaber F. Gubrium Clive Seale, Giampietro Gobo and David Silverman, editors, Qualitative Research Practice., pages 420–434. London and Thousand Oaks, CA., 2004.

[**KLL97**] B. Kitchenham, S. Linkman, and D. Law. Desmet: a methodology for evaluating software engineering

[**KPP+ 02**] Barbara A. Kitchenham, Shari Lawrence Pfleeger, Lesley M. Pickard, Peter W. Jones, David C. Hoaglin, Khaled El Emam, and Jarrett Rosenberg. Preliminary guidelines for empirical research in software engineering. IEEE Trans. Softw. Eng., 28(8):721–734, 2002.

[**LSS05**] Timothy C. Lethbridge, Susan Elliott Sim, and Janice Singer. Studying software engineers: Data collection techniques for software field studies. Empirical Software Engineering, 10:311– 341, 2005. 10.1007/s10664-005-1290-x.

[**TTDBS07**] Paolo Tonella, Marco Torchiano, Bart Du Bois, and Tarja Syst¨a. Empirical studies in reverse engineering: state of the art and future trends. Empirical Softw. Engg., 12(5):551–571, 2007.

[**WRH+00**] Claes Wohlin, Per Runeson, Martin Host, Magnus C. Ohls- son, Bjorn Regnell, and Anders Wesslen. Experimentation in software engineering: an introduction. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[**Rob02**] Colin Robson. Real World Research. Blackwell Publishing Limited, 2002.

[**RH09**] Per Runeson and Martin Host. Guidelines for conducting and reporting case study research in software engineering. Empirical Softw. Engg., 14(2):131–164, 2009.

[**PPV00**] Dewayne E. Perry, Adam A. Porter, and Lawrence G. Votta. 2000. Empirical studies of software engineering: a roadmap. In *Proceedings of the Conference on The Future of Software Engineering* (ICSE '00). ACM, New York, NY, USA, 345-355.

[**EA06**] http://www.cs.toronto.edu/~sme/case-studies/case_study_tutorial_slides.pdf

[**Ple95**] Pfleeger,S.L.; Experimental design and analysis in software engineering. Annals of Software Engineering 1, 219-253, 1995.

[**PK01-03**] Pfleeger, S.L. and Kitchenham, B.A.. Principles of Survey Research. Software Engineering Notes, (6 parts) Nov 2001 - Mar 2003.

[**Fly06**] Flyvbjerg, B.; Five Misunderstandings about Case Study Research. Qualitative Inquiry 12 (2) 219-245, April 2006

[**KPP95**] Kitchenham, B.; Pickard, L.; Pfleeger, Shari Lawrence, "Case studies for method and tool evaluation," *Software, IEEE* , vol.12, no.4, pp.52-62, Jul 1995

References

Empirical research in software testing

[**BS87**] Victor R. Basili and Richard W. Selby. Comparing the effectiveness of software testing strategies. IEEE Trans. Softw. Eng., 13(12):1278–1296, 1987.

[**DRE04**] Hyunsook Do, Gregg Rothermel, and Sebastian Elbaum. In- frastructure support for controlled experimentation with soft- ware testing and regression testing techniques. In Proc. Int. Symp. On Empirical Software Engineering, ISESE '04, 2004.

[**EHP+ 06**] Sigrid Eldh, Hans Hansson, Sasikumar Punnekkat, Anders Pet- tersson, and Daniel Sundmark. A framework for comparing efficiency, effectiveness and applicability of software testing tech- niques. Testing: Academic & Industrial Conference on Practice And Research Techniques (TAIC part), 0:159–170, 2006.

[**JMV04**] Natalia Juristo, Ana M. Moreno, and Sira Vegas. Reviewing 25 years of testing technique experiments. Empirical Softw. Engg., 9(1-2):7–44, 2004.

[**RAT+ 06**] Per Runeson, Carina Andersson, Thomas Thelin, Anneliese Andrews, and Tomas Berling. What do we know about de- fect detection methods? IEEE Softw., 23(3):82–90, 2006.

[**VB05**] Sira Vegas and Victor Basili. A characterisation schema for software testing Techniques. Empirical Softw. Engg., 10(4):437–466, 2005.

[**LR96**] Christopher M. Lott and H. Dieter Rombach. Repeatable software engineering experiments for comparing defect-detection techniques. Empirical Software Engineering, 1:241–277, 1996. 10.1007/BF00127447.

References

Descriptions of empirical studies done in software testing

[Vos et. Al. 2010] T. E. J. Vos, A. I. Baars, F. F. Lindlar, P. M. Kruse, A. Windisch, and J. Wegener, "Industrial scaled automated structural testing with the evolutionary testing tool," in ICST, 2010, pp. 175–184.

[Vos et. al 2012] Tanja E. J. Vos, Arthur I. Baars, Felix F. Lindlar, Andreas Windisch, Benjamin Wilmes, Hamilton Gross, Peter M. Kruse, Joachim Wegener: Industrial Case Studies for Evaluating Search Based Structural Testing. International Journal of Software Engineering and Knowledge Engineering 22(8): 1123- (2012)

[Vos et. Al. 2013] T. E. J. Vos, F. Lindlar, B. Wilmes, A. Windisch, A. Baars, P. Kruse, H. Gross, and J. Wegener, "Evolutionary functional black-box testing in an industrial setting," *Software Quality Journal, 21* (2): 259-288 (2013)

[MRT08] A. Marchetto, F. Ricca, and P. Tonella, "A case study- based comparison of web testing techniques applied to ajax web applications," International Journal on Software Tools for Technology Transfer (STTT), vol. 10, pp. 477– 492, 2008

Other references

[TS04] T.S. Tullis and J. N. Stetson. A comparison of questionnaires for assessing website usability. In Proceedings of the Usability Professionals Association Conference, 2004.

[BKM08] Aaron Bangor, Philip T. Kortum, and James T. Miller. An empirical evaluation of the system usability scale. International Journal of Human-computer Interaction, 24:574–594, 2008.

[Horn06] *Hornbæk*, K. (*2006*), "*Current Practice* in *Measuring Usability*: *Challenges* to *Usability Studies* and *Research*", *International Journal of Human-Computer Studies*, Volume *64*, Issue 2 , February *2006*, Pages *79-102*.

[MTR08] Marchetto, Alessandro; Tonella, Paolo and Ricca, Filippo., State-based testing of Ajax web applications, Software Testing, Verification, and Validation, 2008 1st International Conference on, pp121-130, IEEE, 2008.

[NMT12] C. D. Nguyen, A. Marchetto, and P. Tonella. Combining model-based and combinatorial testing for effective test case generation. In Proceedings of the 2012 International Symposium on Software Testing and Analysis, pages 100-110, ACM, 2012.