

Multi-Objective test case prioritization in highly configurable systems: A case study

José A. Parejo, Ana B. Sánchez, Sergio Segura, Antonio Ruiz-Cortés,
Roberto E. López-Herrejón, and Alexander Egyed

Grupo de Ingeniería del Software Aplicada
Universidad de Sevilla

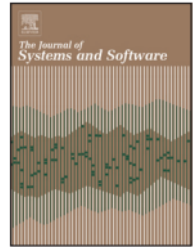
XXII Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2017)
La Laguna – 21/07/2017



Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss



Multi-objective test case prioritization in highly configurable systems: A case study



José A. Parejo^{a,*}, Ana B. Sánchez^a, Sergio Segura^a, Antonio Ruiz-Cortés^a,
Roberto E. Lopez-Herrejon^b, Alexander Egyed^b

^aETS Ingeniería Informática, Universidad de Sevilla, Spain

^bInstitute for Software System Engineering, Johannes Kepler University, Austria

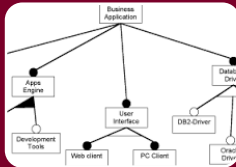
- Collaboration:



- Journal of Systems and Software: IF 2.4
CS/SE: 22/106 (Q1)



Test Case Prioritization



Highly-Configurable Systems



Approach



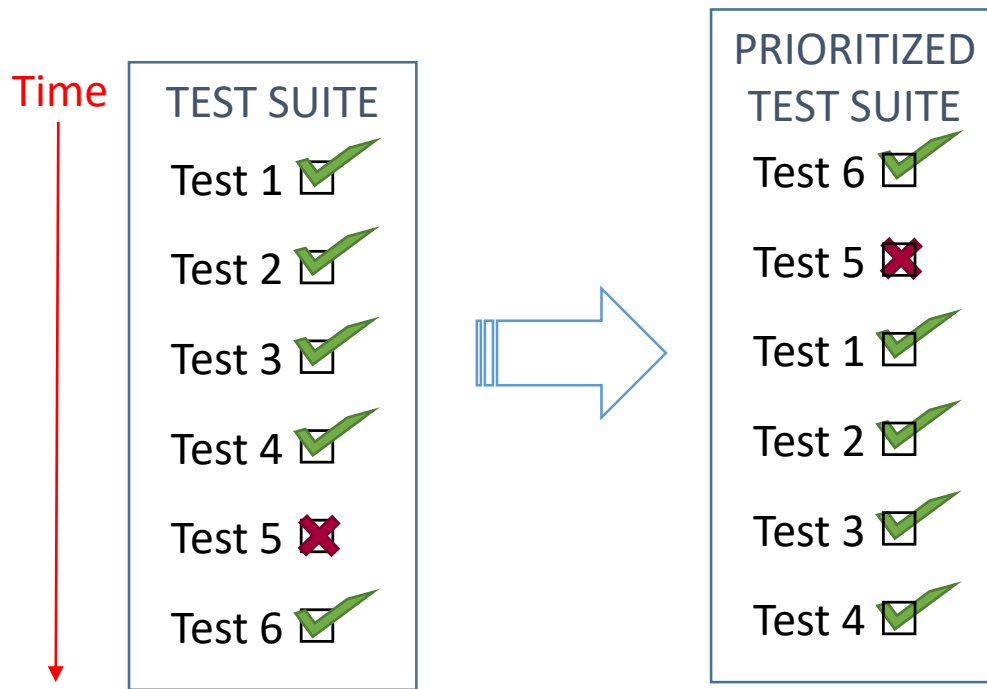
Results & Conclusions

On the importance of (fast) testing

- “The bitterness of poor quality remains long after the sweetness of low price is forgotten.” — **Benjamin Franklin**
- “Quality is not an act, it is a habit.” — **Aristotle**
- “Tests don't prove the absence of bugs. However imperfect tests, run frequently, are much better than perfect tests that are never written at all” —**Martin Fowler**
- “Keep the build (including automated tests) fast.”
—**Martin Fowler**

Test Case Prioritization

- Test case prioritization schedules test cases for execution in an order that attempts to accelerate the detection of faults.



Bugs are detected faster with the prioritized test plan!

Average Percentage of Faults Detected (APFD)

- APFD is a metric that ranges from 0 to 1
- It measures the average of the percentage of faults detected by a test suite weighted by the speed of detection.

$$APFD = 1 - \frac{TF_1 + TF_2 + \dots + TF_n}{n \times m} + \frac{1}{2n}$$

TEST SUITE

Test 1

Test 2

Test 3

Test 4

Test 5

Test 6

$$\begin{aligned} APFD &= 1 - \frac{5}{1 \times 6} + \frac{1}{2 \cdot 6} \\ &= 1 - 5/6 + 1/12 = \\ &= 1 - 0.8\overline{3} + 0.08\overline{3} = 1 - 0.75 = \\ &= 0.25 \end{aligned}$$

PRIORITIZED TEST SUITE

Test 6

Test 5

Test 1

Test 2

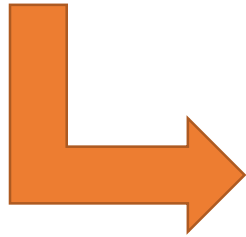
Test 3

Test 4

$$\begin{aligned} APFD &= 1 - \frac{2}{1 \times 6} + \frac{1}{2 \cdot 6} \\ &= 1 - 2/6 + 1/12 = \\ &= 1 - 0.1\overline{6} + 0.08\overline{3} = 1 - 0.08\overline{3} = \\ &= 0.91\overline{6} \end{aligned}$$

The problem with APFD

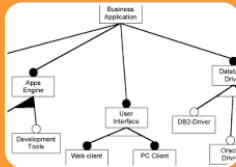
- In order to compute it, you need to execute the test suite, and find which tests detect each bug



We need an alternative objective to prioritize tests “a priori”!



Test Case Prioritization



Highly-Configurable Systems



Approach



Results & Conclusions

Highly-Configurable Systems (HCS)

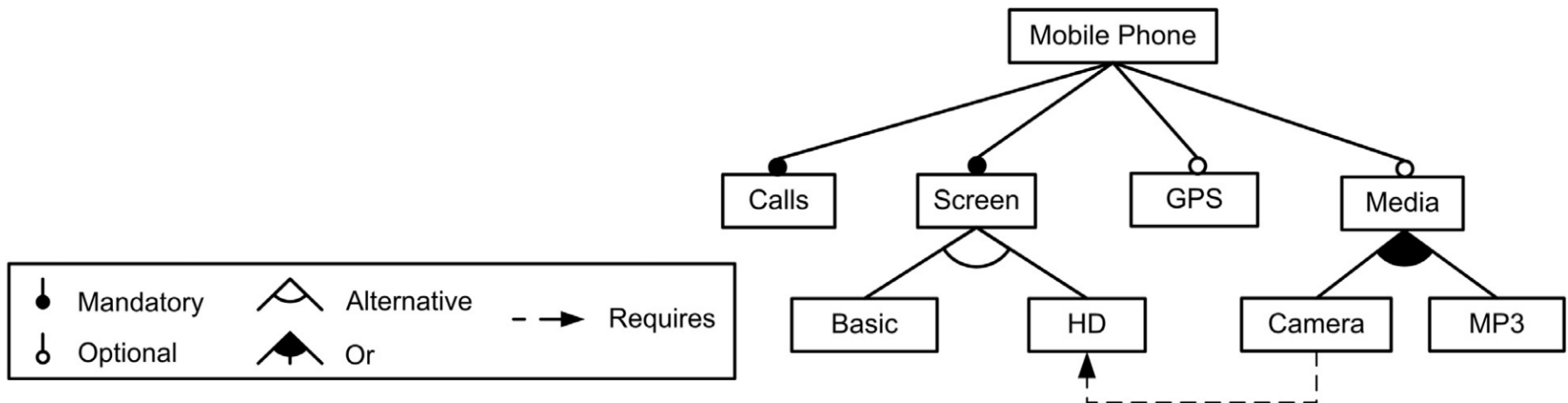
- **Highly-Configurable Systems (HCSs)** provide a common *core functionality* and a set of *optional features to tailor variants of the system* according to a given set of requirements.

- Examples:



HCSs and Feature Models

- *Feature*: a choice to include a certain functionality in a system configuration
- It is common that not all combinations of features are allowed or meaningful
- *Feature Model*: graph representation of all the possible configurations of the HCS in terms of features and constraints among them.

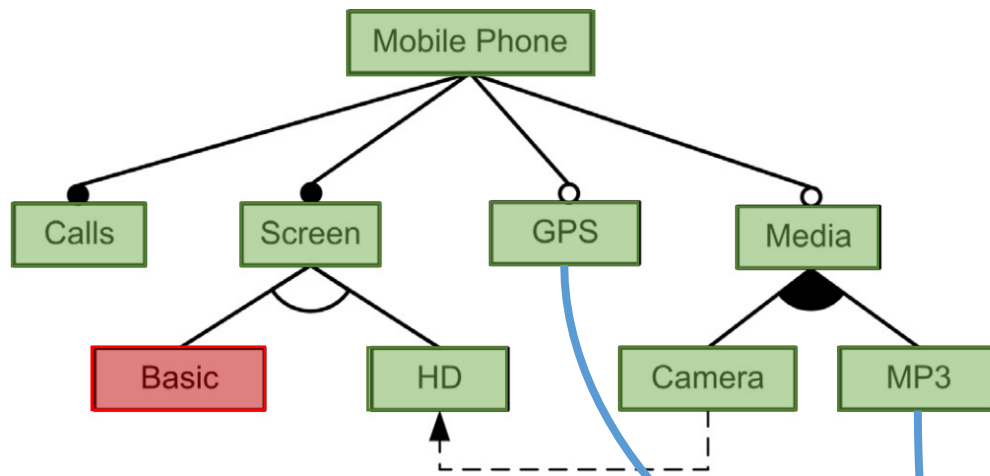


HCS Testing

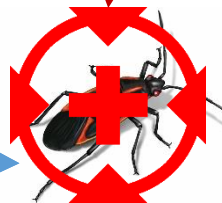
- Tests on HCSs are performed for an specific configuration

➔ **test case** is defined as:

- *a configuration* of the HCS under test (set of features)
- *a test suite* (set of test cases)



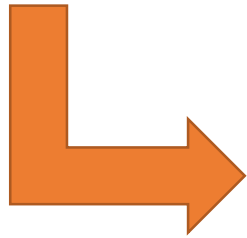
TEST SUITE	
Test 1	<input checked="" type="checkbox"/>
Test 2	<input checked="" type="checkbox"/>
Test 3	<input checked="" type="checkbox"/>
Test 4	<input checked="" type="checkbox"/>
Test 5	<input checked="" type="checkbox"/>
Test 6	<input checked="" type="checkbox"/>



The clips of the voice guiding are listed as songs in the MP3 app

The Drupal Case Study

- Exhaustive testing of HCSs is infeasible



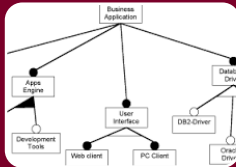
Test Case Prioritization! But...

What Objectives should we use to prioritize?

- **Functional objectives** (based on the features of the system and their interactions) such as complexity metrics, configuration dissimilarity.
- **Non-functional objectives:** consider extra-functional information such as user-preferences, cost, memory.



Test Case Prioritization



Highly-Configurable Systems



Approach



Results & Conclusions

In this work we take into account...

- Functional Objectives:
 - Coefficient of Connectivity-Density (CoC)
 - Dissimilarity
 - Pairwise coverage.
 - Variability coverage and cyclomatic complexity

- Non-Functional Objectives:
 - Number of changes
 - Number of faults
 - Feature size

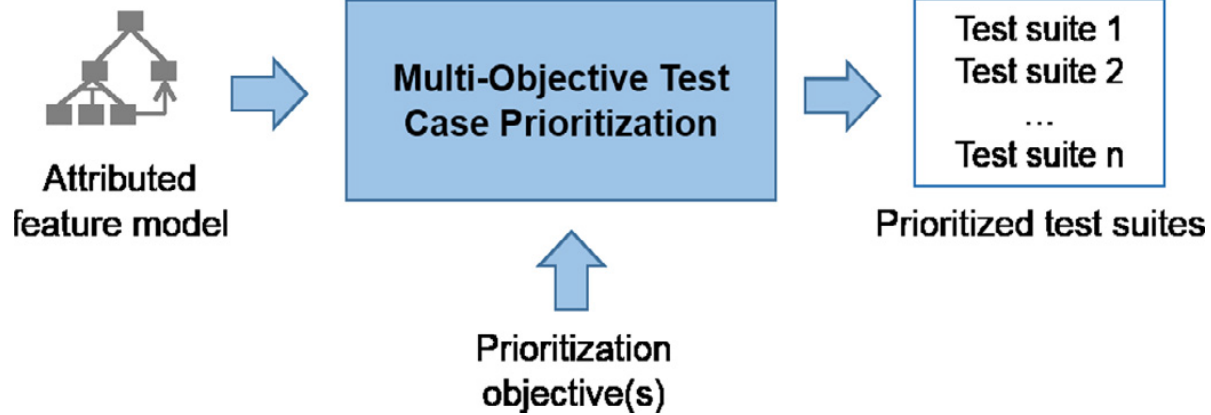
The drupal case study



- Drupal is a highly modular open source web content management framework written in PHP
 - Used to build internet portals, e-commerce applications and online newspapers
 - > 30,000 modules
 - > 630,000 users and developers
- We analyzed versions 7.22 and 7.23
- Non-functional data as feature attributes obtained from the Drupal website, the Drupal Git repository and the Drupal issue tracking system
 - *Feature size*
 - *Number of changes* .
 - *Single faults*
 - *Integration faults*

Our Approach

- *General overview:*



- *Aim:* Find the **combination of objective functions** that provides better APFD



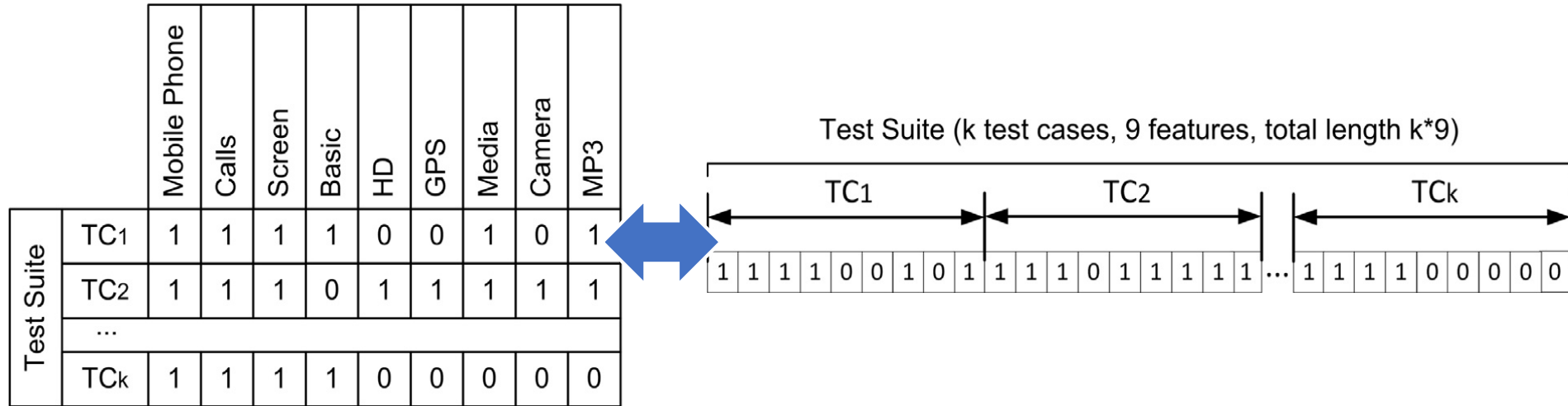
- *Solutions:* A sequence of configurations

Mobile phone test suite.

ID	Test case
TC1	Mobile Phone, Calls, Screen, Basic, Media, MP3
TC2	Mobile Phone, Calls, Screen, HD, GPS, Media, Camera, MP3
TC3	Mobile Phone, Calls, Screen, HD, Media, Camera
TC4	Mobile Phone, Calls, Screen, HD
TC5	Mobile Phone, Calls, Screen, Basic, GPS

Multi-Objective evolutionary algorithm (NSGA-II)

- Solution encoding:



- Initial population: CASA Algorithm [Herrejon et al 2014]

- Operators:

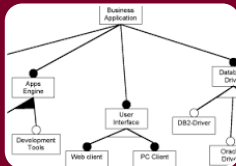
- Crossover (single cut point on suite joints)

- Mutation:

- *Test case swap*
- *Test case addition/removal*
- *Test case substitution*



Test Case Prioritization



Highly-Configurable Systems



Approach



Results & Conclusions

Combinations of 1, 2 and 3 objectives (to predict APFD)

Optimistic approach (we compare the best value in the
Pareto front)

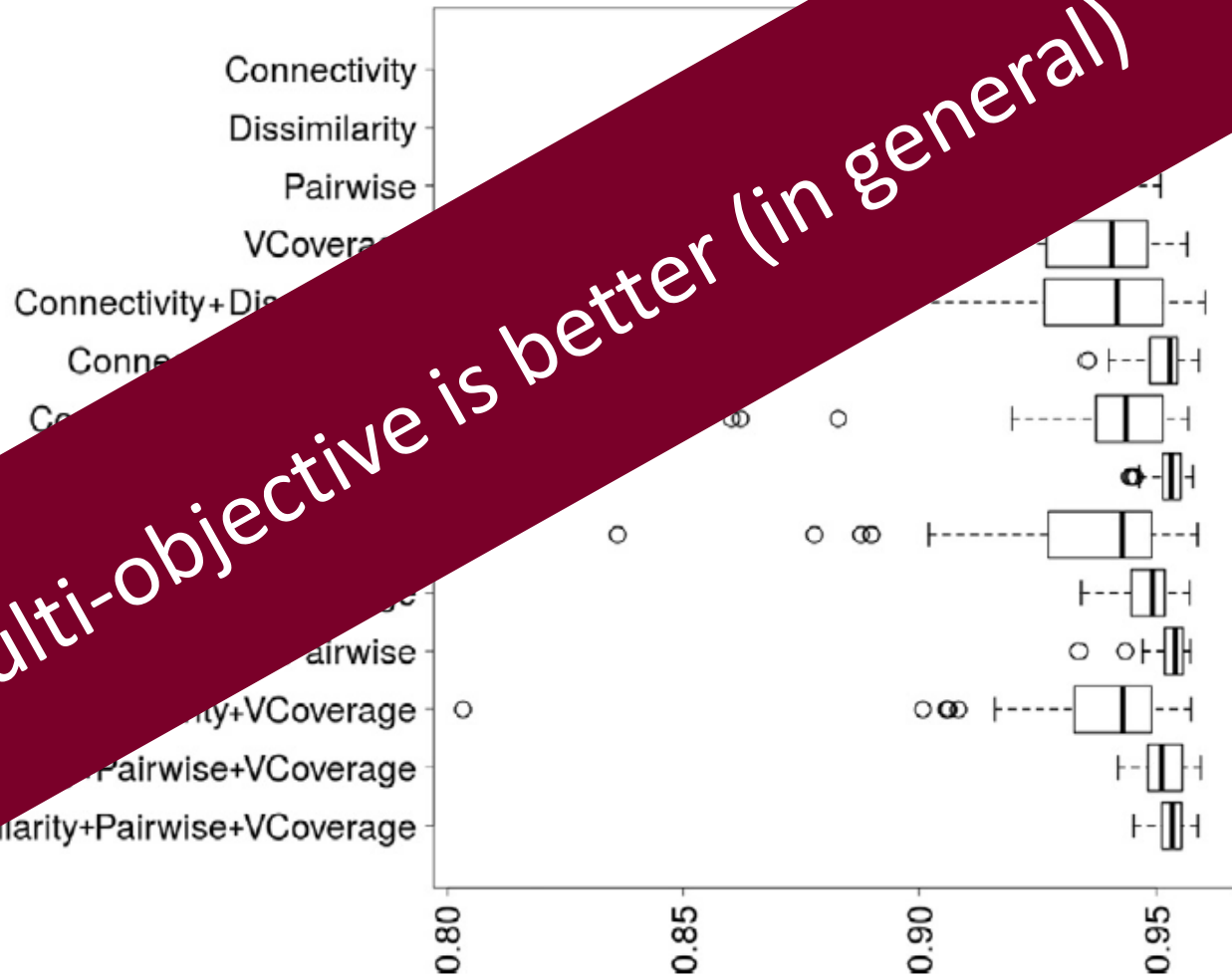
Hundreds of hours of programming and
debugging later...

4 experiments and lots of analyses...

Results

- Functional objectives

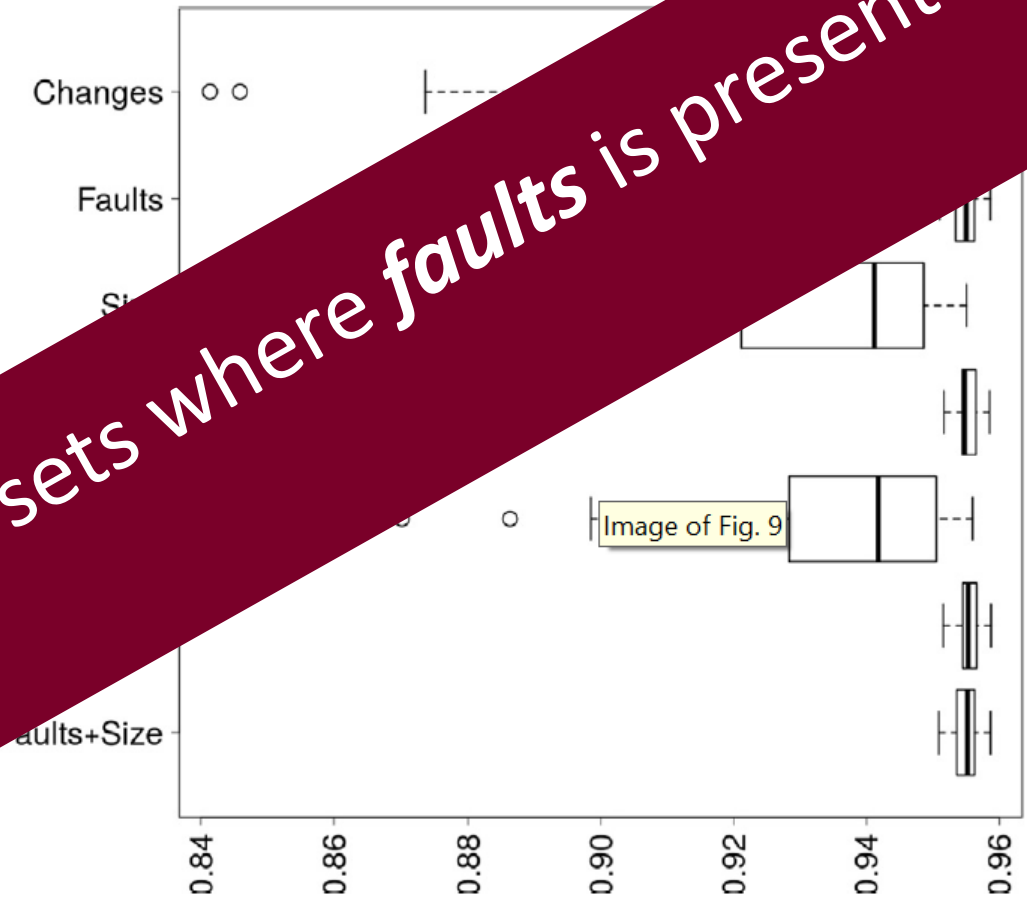
Multi-objective is better (in general)



Results

- Non-functional objectives:

Objectives sets where faults is present are better



Results

- *Multi > mono-* (except for faults)*
- *Multi MIXED > Multi FUNCT*
- *Multi Non-FUNCT*
- *Multi Functional > Multi- MIXED*

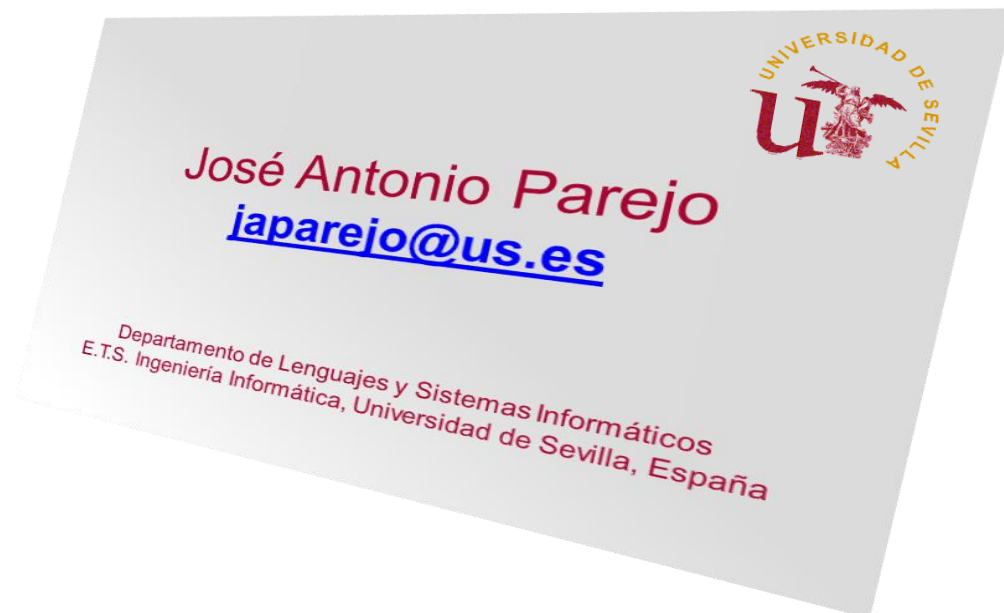
Best overall results were achieved by Dissimilarity and Faults

Thanks!

Multi-Objective test case prioritization in highly configurable systems: A case study



www.isa.us.es



21th International Systems and Software Product Line Conference (SPLC 2017)

Sevilla, Spain

September 25-29, 2017

www.splc2017.net



Resultados

- Combinaciones mixtas:
 - Funcionan mejor que las funcionales pero peor que las no funcionales (solo un poco) en general.