Back from ICST 2011

Program

- Workshops (Regression 2011)
- Keynotes
- Research papers
- Industry papers
- PhD symposium
- Tools and Services Session
- Posters

Regression 2011

- Keynote
- Position papers
- Research Papers

Regression 2011

- Empirically Evaluating Regressing Testing Techniques: Challenges, Solutions and a Potential Way Forward (position paper)
- Gregory Kapfhammer, Allegheny College, USA
- Main message: make all research material available so that the work may be reproduced

Regression 2011

- Making the Case for MORTO: Multi Objective Regression Test Optimization (position paper)
- Mark Harman, University College London, UK
- Main message: in testing research there are multiple goals (eg. coverage, test run time). Don't attempt to optimize a single objective function.
- eg. Use Pareto graphs, look for "elbow"

Keynotes

- **Wolfram Schulte, Microsoft Research, USA:** *Software engineering and testing at Microsoft : A research perspective*
- **Ian Sommerville, St. Andrews University, Scotland:** *Designing for Failure: Challenges for developing and testing complex systems of systems*
- **Walter Tichy, Karlsruhe Institute of Technology, Germany:** *Tunable Architectures or How to Get the Most out of Your Multicore*
- **Bernd Leukert, SAP AG:** *Customers as Integral Part of SAP's Quality Strategy*

Keynote: Ian Sommerville,
St. Andrews University, Scotland

- Designing for failure: Challenges for developing and testing complex systems of systems

Panel: **Software testing research: looking back to understand what is ahead of us.**

- Organizer: Benoit Baudry
- Jürgen Allgayer, Google
- Lionel Briand, Simula Research Lab
- Maximilian Fuchs, BMW
- Alessandro Orso, Georgia Tech
- Mauro Pezzè, U. Milano + Lugano
- Brian Robinson, ABB
- Jan Tretmans, Embedded Systems Inst.

Panel: **Software testing research: looking back to understand what is ahead of us.**

Three questions:

1. Main contribution of Testing research
2. Challenges
3. Target for next 10 years

Panel: **Software testing research**

Main contribution of Testing research:

- Model Based Testing
- Diversity of techniques
- Security checking (done be security community)
- Code Coverage
- Combining Static and Dynamic
- Quality process, test automation, testing as a systematic measurable process

Panel: **Software testing research**

Challenges:

- Scalable, practical, validated test strategies
- limited adoption of the techniques
- apply to real systems, embed in existing test practices
- Improve education
- handle non functional/implicit requirements
- Scale out solutions

Panel: **Software testing research**

Challenges (cont.):

- Bridge gap between ideas and tools for real world
- Integrate different approches
- Move from deploy to run time
- Automate the hard step

Panel: **Software testing research**
Maximilian Fuchs, BMW

- Electronics in cars since 1979
- Now 4GB, soon 40 GB
- Integrate over 50 Electr. Control Units
- Distributed Functionality
- Huge variety of cars
- All need to be tested, fault free

Panel: **Software testing research**

Target for next 10 years:

- Heuristics for stress testing
- Tradeoff between dependability and cost
- More empirical studies
- Stronger collaboration: 10Yr gap between research and practice may grow to 20
- Integrated quality approach
- Evolvability
- Deal with : cloud, multicore, mobile

Panel: **Software testing research**

Target for next 10 years (cont.):

- Collaborate on open source
- Mainstream test curriculum
- From toy solution to proof of concept
- Stronger interaction between researchers and practitioners
- Increase industry participation in testing conferences
- Increase experimantal validation

Keynote: Bernd Leukert, SAP AG

Executive vice president of Quality Governance and Production at SAP

- Customer focus
  - From requirements
  - To validation
  - Transparency
- LEAN
  - Borrowed manufacturing methodology
  - Continuous improvement processes
    - Each ‘tact’ (sprint): Deliverable product
    - Moving testing from QA to Dev
    - Test-Driven Development: Preventing bugs instead of fixing
  - Focus on people
    - Design Thinking
    - Give developers time → Creativity + Quality
    - Better to cut in scope

Factors Limiting Industrial Adoption of Test-Driven Development A Systematic Review

Adnan Causevic, Daniel Sundmark, Sasikumar Punnekkat
Mälardalen University, Västerås, Sweden

- Industry perceives TDD as not enough used
- Detailed analysis of 48 papers on TDD
- Top limiting factors:
  1. Increased developer time
  2. Lack of TDD experience
  3. Lack of design
  4. Lack of testing skills / knowledge
  5. Lack of TDD adherence
  6. Domain & Tool specific
  7. Legay code

Dealing with imperfections in Google-scale systems

Robert Nilsson, Google Zürich

- Regression workshop keynote
- Engineering productivity tools
- Give developers tools to run regression tests and get early feedback
  - Test prioritization based on multi-objective optimization, which also considers
    - test flakiness
    - past fault history
    - test execution time
- Statistical regression testing
  - Find important regressions
  - Cover most common/critical problems

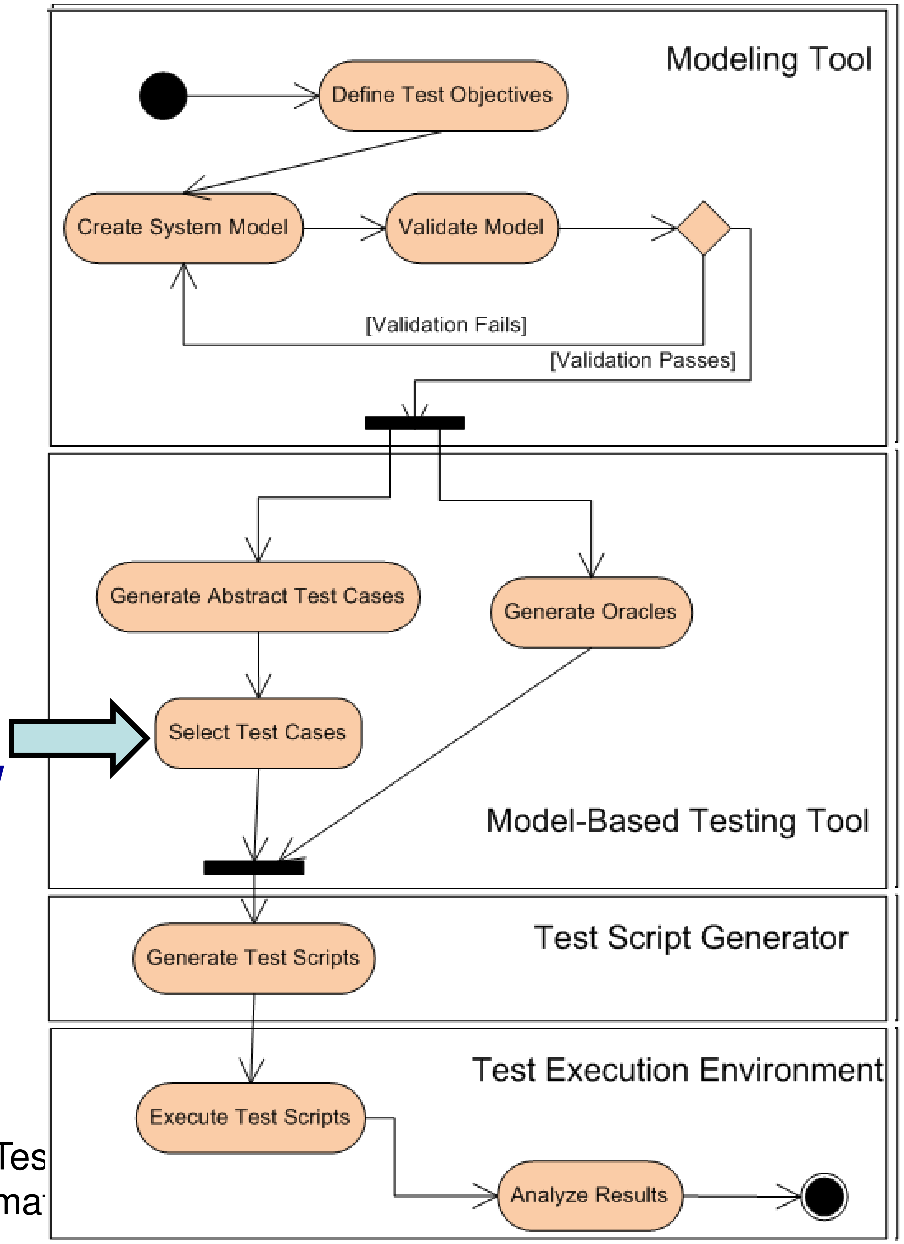
Empirical Investigation of the Effects of Test Suite Properties on Similarity-Based Test Case Selection

Hadi Hemmati, Andrea Arcuri, Lionel Briand
Simula Research Laboratory

- Model Based Testing
  - Test generation: lots
  - Very slow tests
- Test Suite Reduction
  - Cluster tests by their design similarity
  - Tests as sequences of states, transitions, …
- Works better than coverage based

Model-Based Testing (MBT)

- ❑ Systematic Test Generation
- ❑ Automation
- ❑ Scalability Issue
  - ❖ Large test suites
    - ❖ 300 test cases
  - ❖ Expensive tests
    - ❖ Time: each 10 min -> 50 hours
    - ❖ Resources-> network, HW
- ❑ Test Case Selection
  - ❖ Given a maximum budget
  - ❖ Maximum fault detection rate



STCS Steps

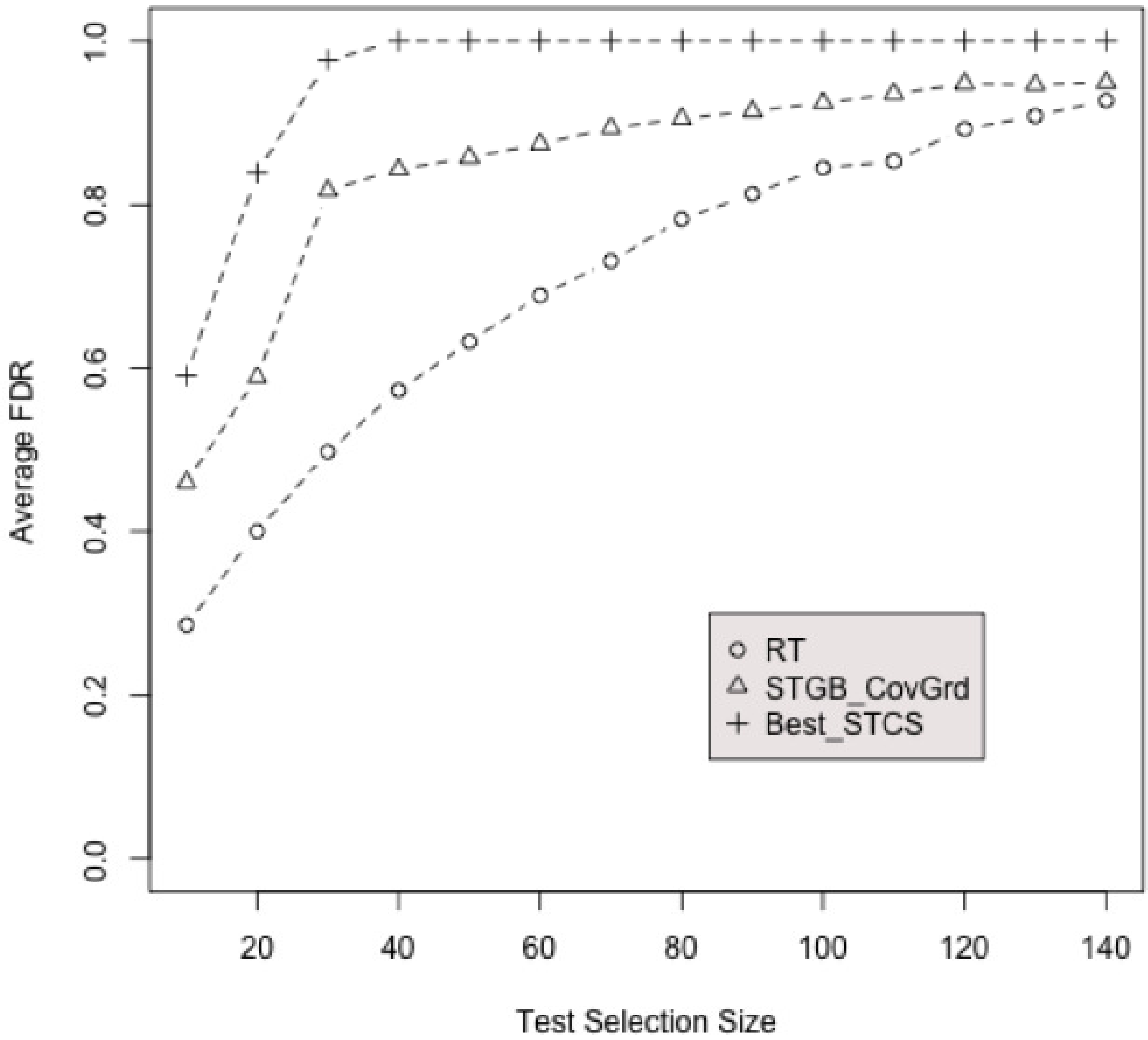
1. **Encoding of abstract test cases based on sequence of**
   - ❖ **States, Transitions, Trigger-guards, etc.**

2. **Similarity function definition**
   - ❖ **Sequence-based (e.g. Needleman-Wunsch)**

3. **Minimizing the similarity measure**
   - ❖ **Clustering, Adaptive Random Testing**
   - ❖ **Search-based: Greedy Search, Genetic Algorithms, Simulated Annealing, etc.**

Similarity-Based Test Case
.Selection, Hemmati et al

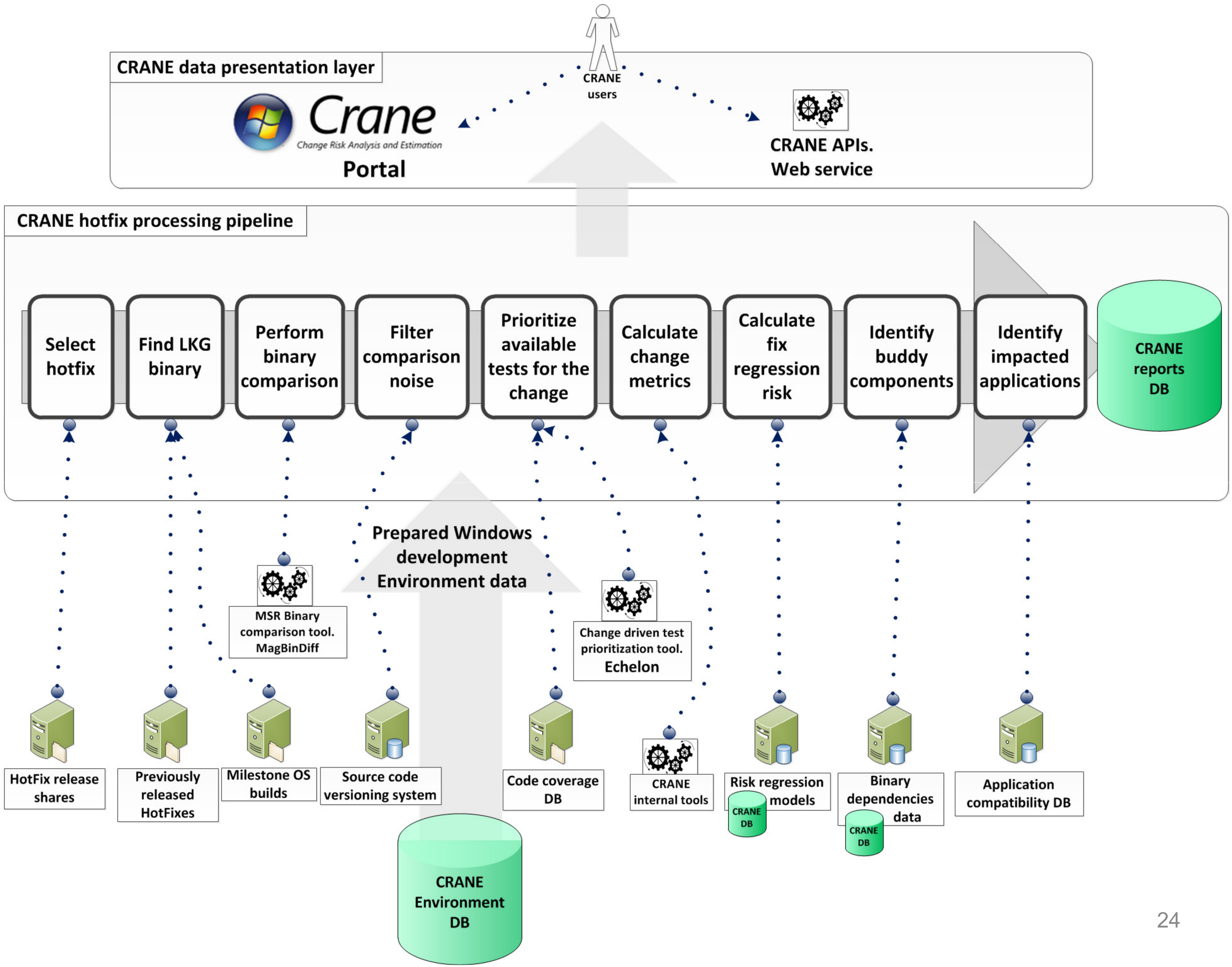
Comparing STCS Results with Random and Coverage-based Selections



CRANE: Failure Prediction, Change Analysis and Test Prioritization in Practice

Jacek Czerwonka, Rajiv Das, Nachiappan Nagappan, Alex Tarvo, Alex Teterev
Microsoft Core OS Division, Microsoft Research

- Maintenance of Windows
- Post-production hotfixes
- Used extensively in Vista SP2, Win7 SP1
- Complex decision support system
- Change Analysis: what changed, coverage, history, …
- Failure Prediction: risk level
- Test Prioritization: Echelon



CRANE users
CRANE data presentation layer
Crane
Change Risk Analysis and Estimation
Portal
CRANE APIs.
Web service
CRANE hotfix processing pipeline
Select hotfix
Find LKG binary
Perform binary comparison
Filter comparison noise
Prioritize available tests for the change
Calculate change metrics
Calculate fix regression risk
Identify buddy components
Identify impacted applications
CRANE reports DB
Prepared Windows development Environment data
MSR Binary comparison tool. MagBinDiff
Change driven test prioritization tool. Echelon
HotFix release shares
Previously released HotFixes
Milestone OS builds
Source code versioning system
Code coverage DB
CRANE internal tools
Risk regression models
CRANE DB
Binary dependencies data
CRANE DB
Application compatibility DB
CRANE Environment DB

Vista SP1 QFE View Details and Notes

Change Summary

Fix Regression Risk: Very High (Probability of regression > 50%)

	Binary	Arch Layer	File Type	Regressions	Changed Block Covered
▾	p.dll	44 (range:0-64)	AMD64Native64	0/0	58/72 (80%)

	Source File	Changed Block Covered
▸	base\diagnosis\pdata.c	21/32 (65%)
▾	base\diagnosis\putil.c	2/2 (100%)

Function	Complexity	Blocks Covered	Changed Blocks Covered
Connect	30->32	76/80 (95%)	2/2 (100%)

	Source File	Changed Block Covered
▸	base\diagnosis\query.c	35/38 (92%)

	Binary	Arch Layer	File Type	Regressions	Changed Block Covered
▸	s.ocx	45 (range:0-64)	AMD64Native64	0/0	5/9 (55%)
▸	r.dll	11 (range:0-64)	X86Native32	0/0	5/5 (100%)

Buddy Components (?) View Whitelist

	Component	Source	Owner	
▸	base technologies\diagnostic framework\diagnostic scenarios	bug history	car	Remove
▸	base technologies\performance counters	trace	prav	Remove
▸	base technologies\file systems (remote)\frs2	trace	seb	Remove
▸	tools\client platform\lab	static analysis	dly	Remove
▸	networking\wireless services\wlan end to end	static analysis	ama	Remove

Impacted Applications (?)

AppId	Name	Version	Vendor
2048	visual studio 2005 german	2005 sp1	microsoft
2049	visual studio 2005 japanese	2005 sp1	microsoft

Minimum Tests (?) Export All

Pri 1 (Must run) | Pri 2 (Good to run) | Pri 3 (Optional)

Trace	Blocks Covered	Component	Owner
networking\qos\qos-core$619483 [pacerperf]	51	networking\qos	sun
server technologies\wsrm\accounting\remotejobs(lms)$365	48	server technologies\wsrm	ama
tools\performance management\performance tools$9592 [cltregress typeperf]	45	tools\performance management	prav
server technologies\wsrm\service\common\apppool$163	38	server technologies\wsrm	ama
multimedia\media foundation_common jobs\mf\config$71344	11	multimedia\media foundation	var
tools\utilities\triage$1016344 [code coverage - configuration]	8	tools\utilities\triage	and

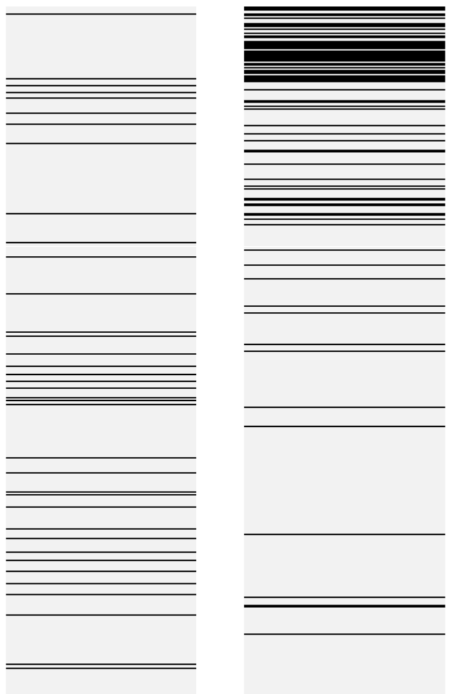
Similar Hotfixes

Branches

- WinXP SP2 QFE
- WinXP SP2 GDR
- WinXP SP3 QFE
- WinXP SP3 GDR
- Win2003 SP1 QFE
- Win2003 SP1 GDR
- Win2003 SP2 QFE
- Win2003 SP2 GDR
- Vista SP1 QFE
- Vista SP1 GDR
- Vista RTM QFE
- Vista RTM GDR

Risk prediction

- Which fixes carry more than average risk of regression?



Uniform distribution of regressions within a set of fixes

Skewed distribution of regressions within a set of fixes

.CRANE, Czerwonka et al

Human Risk prediction vs Automated

Manual human risk assessment:

•It is very hard for Dev and Test to distinguish “Low” and “Medium” risk categories.

•Dev and Test identify fix as “High risk” very very infrequently.

•Regression rate is relatively high in this manual “High risk” category.

Fix-regression proneness (automatic):

•Metrics

– Organization Structure: # of engineers (present & past), cohesive owenership,…
– Code Churn
– Code complexity, etc.

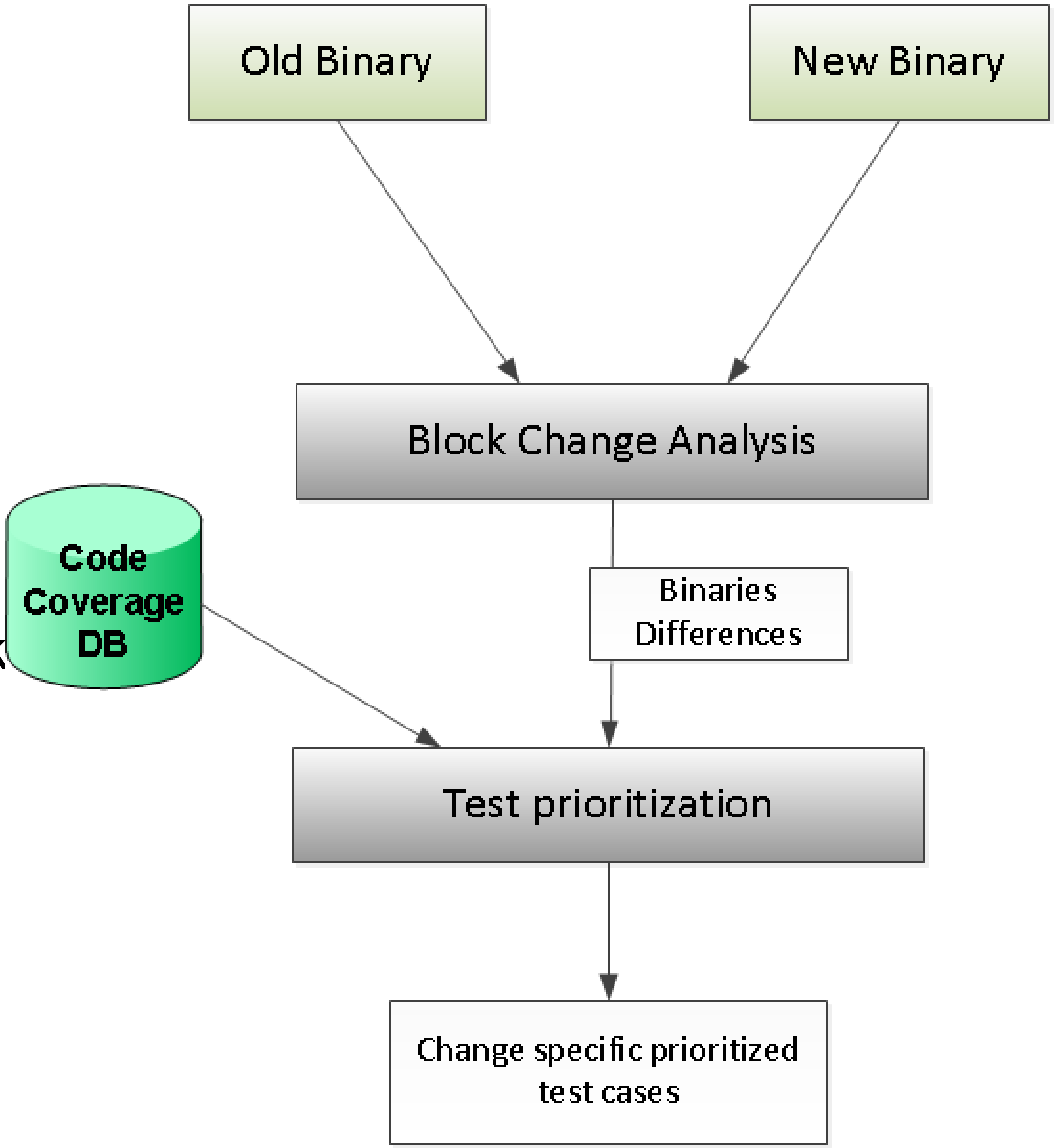
•“High” and “Very High” categories get a lot more fixes.

Bucket	Regression probability
Very High	46.2%
High	17%
Average	4.4%
Low	3.1%
Very Low	1.6%

Czerwonka et al

Echelon tool

- Comparison of 2 versions of the same binary on binary blocks level.
- Use Code Coverage data to select tests traversing through changed code achieving maximum block (statement) coverage.
- Minimize cost to run prioritized tests (prefer automatic tests, optimize for test execution time).



.CRANE, Czerwonka et al

Effectiveness of test prioritization

- Definitions:

  Regression := Any defect found in a fix either internally or externally that causes re-creation of the fix package.

  Test selection hit := CRANE recommended an existing test for execution able to detect a defect.

  Test selection miss := A test able to detect a defect exist but CRANE did not recommend it.

	Study 1	**Study 2**	**Study 3**
A - Total number of regressed fixes	X	Y	Z
B - Number of fixes with existing tests able to find a problem	0.83 X	0.67 Y	Z
C - Number of fixes for which a suitable existing test was identified by recommendations	0.43 X	0.42 Y	0.5 Z
Effectiveness [C/B]	52%	63%	50%

- ~55% effectiveness / typically less then a hundred tests → fair trade-off

.CRANE, Czerwonka et al

Using Semi-Supervised Clustering to Improve Regression Test Selection Techniques

Songyu Chen, Zhenyu Chen, Zhihong Zhao, Baowen Xu
and Yang Feng, Nanjing University, China

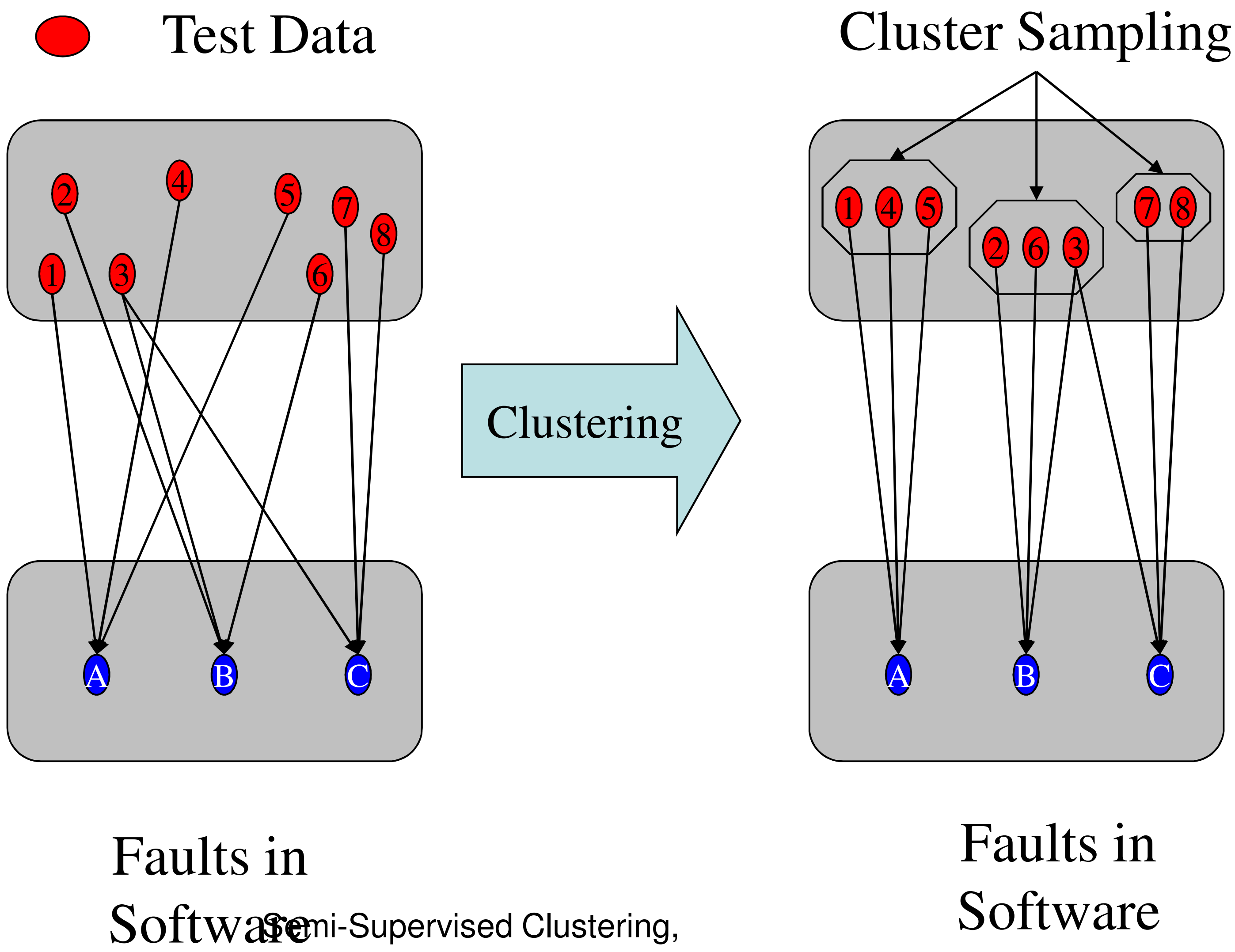
- Test Suite Reduction
  - Choose small but effective subset of test suite
  - not version specific
- Unsupervised K-means (previous work)
  - Cluster tests by their coverage vector
  - function level granularity
- Semi-supervised K-means
  - Coverage matrix X transformed to smaller dimension matrix Y using constraints
  - Cluster tests by their y vectors
- Constraints derived from previous test results
  - Must_link(t1,t2) if both always failed on same versions
  - Cannot_link(t1,t2) if both never failed on same version

Simple Example

- Hamming Distance
  - $D(t_1,t_2)=4$
  - $D(t_1,t_3)=3$
  - $D(t_1,t_3) < D(t_1,t_2)$
  - (t_1,t_3) is more likely to be in same cluster than (t_1,t_2)
- In fact:
  - t1 and t2 reveal a same fault
  - t3 is a passing test

	t_1	t_2	t_3	t_4
f_1	1	0	1	0
f_2	1	1	0	1
f_3	0	1	0	1
f_4	1	0	1	0
f_5	1	0	1	1
f_6	0	0	1	1
f_7	1	1	0	1

Failure Proximity



Semi-Supervised Clustering, .Chen et al

Constraint-based Semi-Supervised Clustering

- Use pair-wise constraints to label partial data.
  - Must-Link: two tests must be in a same cluster.
    - Tests triggered by some same faults
    - How strict?
  - Cannot -Link: two tests cannot be in a same cluster.
    - Tests triggered by different faults

Semi-Supervised K-means

- x_i is a test, represented by feature vector.
  - For example, $x_i = (0,1, 1, 0, 0, 1)$
- w is a weight matrix for transformation.
- y_i is a test transformed from x_i by w.
  - $y_i = w^T x_i$
- Find a w to max the objective function $J(w)$.

Example of Transformation

Tests	Function Call Profile(18 functions)
x_1	1 1 0 1 1 0 1 1 1 1 1 1 1 1 0 1 0 1
x_2	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
x_3	0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
x_4	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
x_5	0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1
x_6	0 1 0 0 1 0 0 0 1 1 1 1 1 1 0 0 1 1

- Four constraints are derived from test results
  - Must-link: (x_5, x_6) and (x_3, x_4),
  - Cannot-link: (x_1, x_2) and (x_4, x_5).
- D(x_5, x_6)=6 and D(x_1, x_2)=D(x_3, x_4)=4
- (x_5, x_6) may be separated with higher probability by clustering.

SSDR

- SSDR to generate a weight matrix *W*.
- D. Zhang et al. , SDM'07

W for Transformation				
0.0275	**-1.0000**	**0.1999**	**-0.2702**	**0.0232**
-0.0010	**-0.0016**	**0.0015**	**-0.0001**	**-0.0896**
-0.8432	**-0.0377**	**0.4270**	**0.5201**	**0.1791**
-0.0060	**-0.0088**	**0.0069**	**0.1169**	**-0.2316**
-0.0112	**-0.0132**	**0.0194**	**0.0353**	**-0.9854**
-0.7132	**0.2414**	**0.3974**	**-1.0000**	**-0.0582**
-0.0081	**-0.0091**	**0.0146**	**0.0668**	**-0.4543**
-0.0082	**-0.0094**	**0.0146**	**0.0661**	**-0.4613**
-0.0009	**-0.0013**	**0.0013**	**0.0003**	**-0.0749**
-0.0009	**-0.0013**	**0.0013**	**0.0003**	**-0.0749**
0.0000	**0.0000**	**0.0000**	**0.0000**	**0.0000**
-0.0010	**-0.0016**	**0.0015**	**-0.0001**	**-0.0896**
-0.0009	**-0.0013**	**0.0013**	**0.0003**	**-0.0749**
-0.0010	**-0.0016**	**0.0015**	**-0.0001**	**-0.0896**
-1.0000	**-0.1365**	**-1.0000**	**-0.0603**	**-0.0210**
-0.0062	**-0.0087**	**0.0067**	**0.1176**	**-0.2204**
-0.8432	**-0.0364**	**0.4286**	**0.3854**	**-0.0650**
-0.0113	**-0.0126**	**0.0149**	**-0.0274**	**-1.0000**

Example of Transformation

Tests	Transformed Data
y_1	**-0.0286 -1.0594 0.2855 -1.2884 -4.8778**
y_2	**-3.4299 -1.0262 0.5371 -1.4332 -4.7485**
y_3	**-3.4566 -0.0262 0.3378 -1.1644 -4.9261**
y_4	**-3.4299 -1.0262 0.5371 -1.4332 -4.7485**
y_5	**-0.7666 0.1895 0.4824 -2.0195 -4.5652**
y_6	**-0.8728 -0.0741 0.4709 0.0394 -2.7098**

Constraints	Original Distance	New Distance
(x_5, x_6)	6	7.7625
(x_3, x_4)	1	1.1442
(x_1, x_2)	4	11.6709
(x_4, x_5)	4	8.9514

Objective Function

- The objective function *maximum J(w)*:

$$J(w) = \frac{1}{2n^2}\sum_{i,j}(w^T x_i - w^T x_j)^2$$

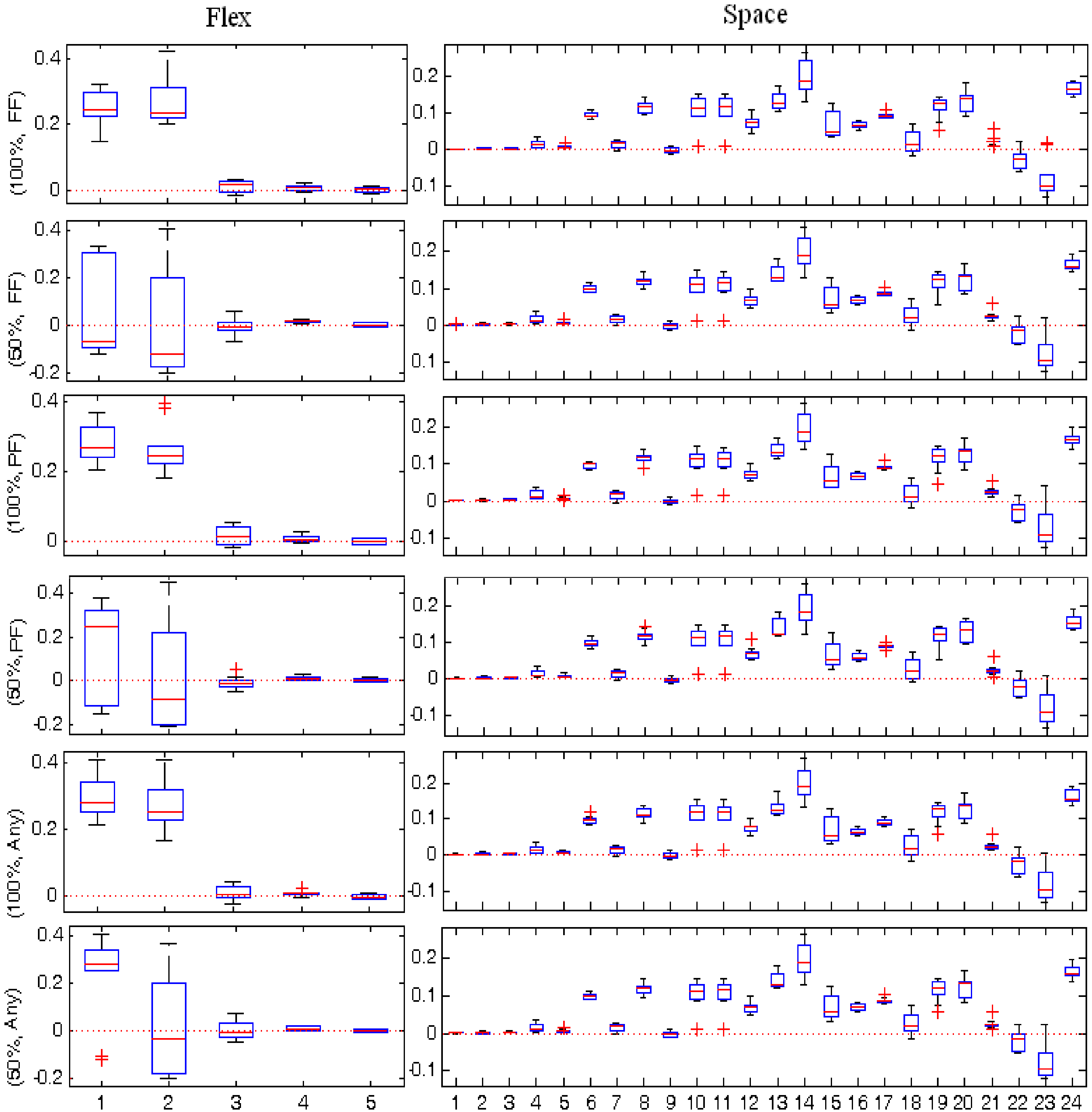
$$+ \frac{\alpha}{2n_C}\sum_{(x_i,x_j)\in C}(w^T x_i - w^T x_j)^2$$

$$- \frac{\beta}{2n_M}\sum_{(x_i,x_j)\in M}(w^T x_i - w^T x_j)^2$$

Evaluation Metric

$$F-measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$Precision = \frac{|T_F^{'}|}{|T^{'}|} \qquad Recall = \frac{|T_F^{'}|}{|T_F|}$$


Flex
Space
(100%, FF)
(50%, FF)
(100%, PF)
(50%,PF)
(100%, Any)
(50%, Any)

Conclusion

- SSKM can improve test selection in most cases.
- Two useful observations:
  - (1) Better effectiveness when the failed tests are in a medium proportion.
  - (2) A strict definition of pairwise constraint can improve the effectiveness of cluster test selection.