Qualitative Models for Requirements Engineering

tim@menzies.us
WVU, USA

Julian Richardson, RIACS, AMES
julianr@riacs.edu

(and others)

http://unbox.org/promise/2006 Go promise 2006

2nd International PROMISE Workshop

September 24, 2006, Philadelphia, Pennsylvania USA.
Co-located with the IEEE Conference on Software Maintenance

home

calls for papers
important dates
program commitee
special issue

public datasets
data format

registration
program
hotel

promise@unbox.org

Welcome to PROMISE 2006!

NEWS:

Papers accepted to PROMISE 2006 will be eligible for submission to a special issue of the Journal of Empirical Software Engineering on repeatable experiments in software engineering.

What is PROMISE?

PROMISE = **PR**edict**O**r **M**odels **I**n **S**oftware **E**ngineering

Software management decisions should be based on well-understood and well-supported predictive models.

A good model should be a generalization of real-world data. But where does the data come from?

Collecting data from real world software engineering projects is problematic. Software projects are notoriously difficult to control and corporations are often reluctant to expose their own software development record to public scrutiny.

Sound bites

- Early, we only have some
- But a little can be enough
- Finding the key issues is not hard
- Decision making = use the keys (and the rest will follow)

“design as search”

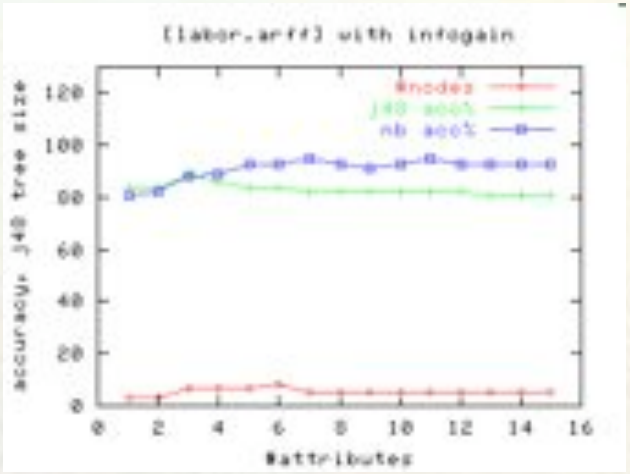
- Herbert Simon:
  - “Design = quintessential human activity”
- Allen Newell:
  - Cognition is a search for operators which we believe will take us towards our desired goals

- Q: what if our beliefs are approximate?
  - I don’t believe that you can always get rid of subjective judgments in these kinds of studies. -- Rick Kazman, Jan 6, 2006,10:53:47
- A: “Design” means doing lots of what-ifs.
  - Find consistent set(s) of beliefs a.k.a. “worlds”
  - What selects for worlds with results we want?

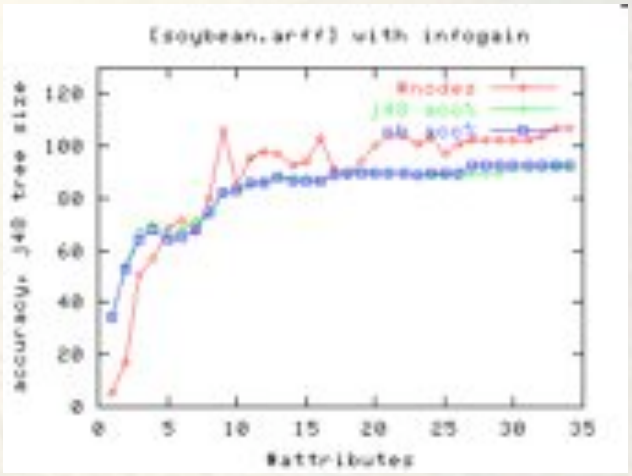
Surprisingly, don't need to explore all settings to all variables

If sort attributes on "infogain" and learn using first N attributes
then good theories with low N

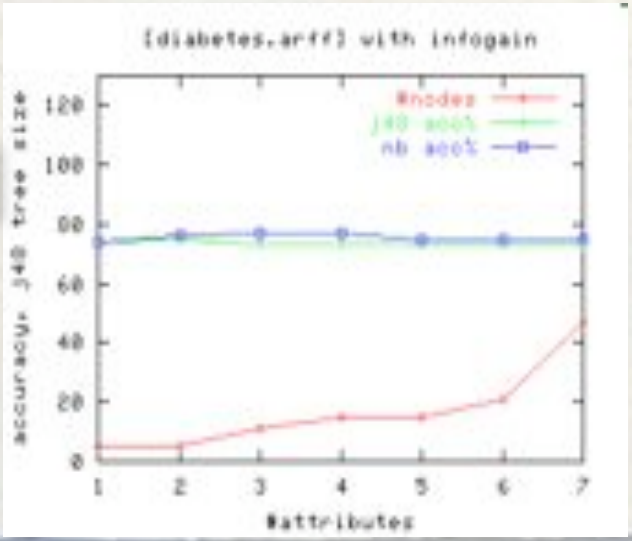
labor



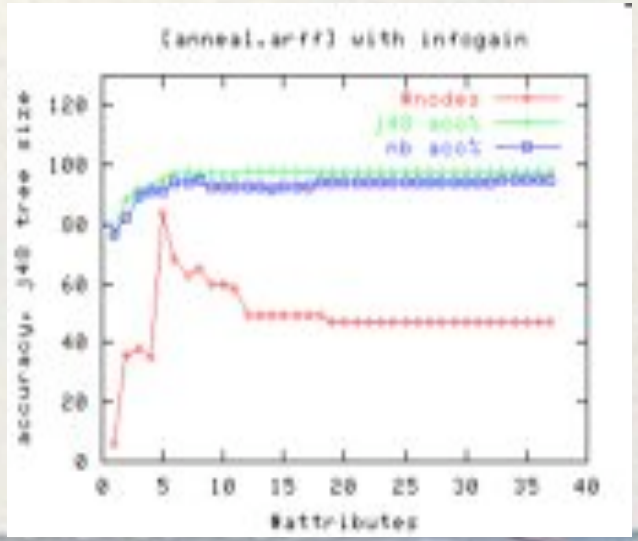
soybean



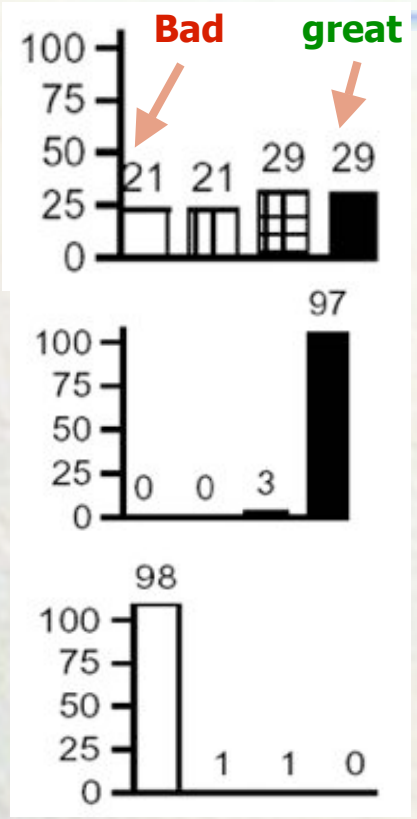
diabetes



anneal



So, we can “cheat”



Housing, Baseline (% of housing types)

6.7 <= rm < 9.8 and 12.6 <= ptratio < 15.9

0.6 <= NOX < 1.9 and 17.16 <= lstat < 39

A few variables are (often) enough

Method:

1. Stochastic sampling of lightweight notations
   - ❄ Explore all the what-ifs
2. Data mining to find the master variables

- Treatment” = policy
  - what to do
  - what to watch for
- TAR3
  - Seek attribute ranges that are often seen in “good”
  - and rarely seen in “bad”.
- Treatment= constraints that changes baseline frequencies

Related Work

- Pelanek, Druzdzel
  - State spaces clump
- DeKleer, Hall & Holmes, Williams, Clark
  - Sate spaces have collars

So a few variables will control the rest

- Selman:
  - Stochastic propositional search
- Bay& Pazzini:
  - Contrast set learning
- Kakas:
  - Abduction
- DeKleer, Poole:
  - model-based diagnosis
- Reiter:
  - default logic
- Easterbrook, Callahan:
  - lightweight formal methods
- Chung, Mylopoulos et.al.
  - “soft goal” graphs
- Shaw, Garlan:
  - qualitative functional dependencies
- MacLean:
  - QOC graphs

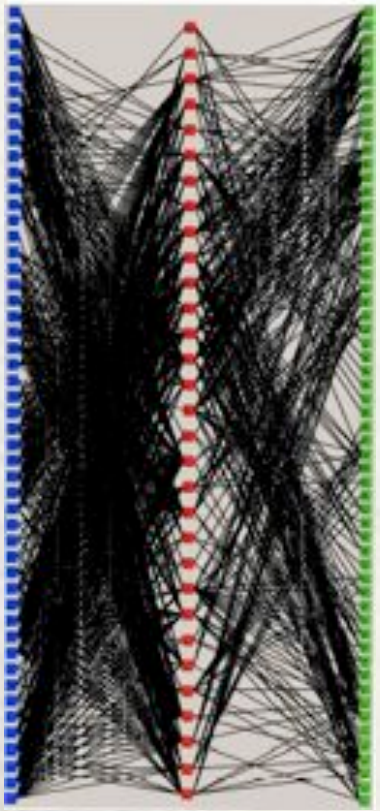
DDP @ JPL

SILAP @ IV&V

NEAR @ APL

XOMO @ AMES

DDP@JPL (with Martin Feather)

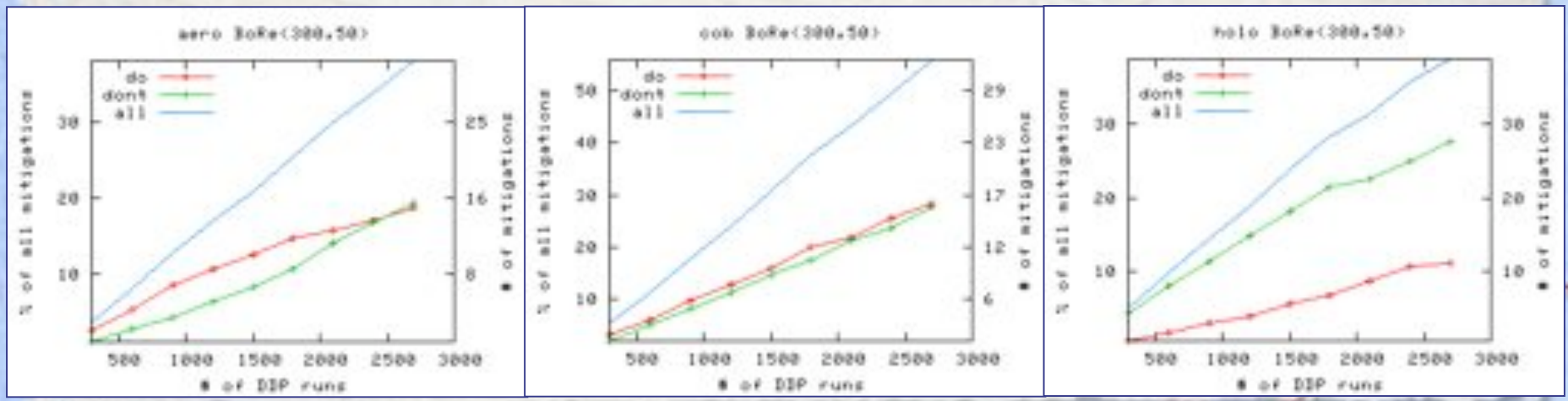


analyze this

- Cornford and Feather [3]
  - Visual tool for "group think"
  - RISKS hurt REQUIREMENTS
  - MITIGATIONS remove risks, cost money.
  - Seek cheap mitigations resolving risks that hurt the important requirements
- Has been used for:
  - Starlight, Deep Space 1&2, X2000 electronics packages; Interferometry design; Mars Globa Surveyor extended missions, Technology Infusion/Maturity assessments, ...
- Being used for:
  - SCrover: University of Southern California's autonomous rover
  - used for
    - Cost and risk models for autonomous systems

DDP@JPL + Surfer

Currently, under the hood, SURFER calls treatment learning. This may change…..



DDP @ JPL

SILAP @ IV&V

NEAR @ APL

XOMO @ AMES

SILAP@IVV (with Marcus Fisher)

Q: What most increases project errorPotential?

A: SILAP

- from DELPHI sessions with experienced NASA IV&V managers
- a network of weighted project factors
- E.g.

*function the(X) { return one (X) * all(X) }*

- *One:* project data
- *All:* DEPHI knowledge

- E.g.

function development() {
return the("experience") +
the("organization") }
function software() {
return the("complexity") +
the ("innovation") +
the("softwareSize")

- Passes the "elbow test"
  - Domain experts elbow us out of the way …
  - … in their haste to fix some error.

just a notation we made up one night

SILAP@IVV
Stability Studies

- Run 5000 simulations
- Ten times: divide data into 90% train, 10% test
- Apply TAR3:
  - Only report treatments found in ≥ 7 samples
- Score treatments by what makes error potential worse
  - I.e. explore the worst case scenario
- Worst case scenarios:
  - Very poor developer experience and any one of
    - High reuse is a goal
    - Similar software has been used on prior missions
    - Software very simple; e.g. no intense numerical solutions.
    - Software being built by a team at one location

(so no one thinks to monitor these projects)

DDP @ JPL

SILAP @ IV&V

NEAR @ APL

XOMO @ AMES

NEAR@APL
(near earth autonomous rendezvous)

id	software process option	safety	dev. time	dev. cost	life cycle cost	capability
1	target critical mission phases	+	+	+	-	-
2	target critical commands	+	+	+	-	-
3	target critical events	+	+	+	-	-
4	onboard checking	+	-	-	+	0
5	reduce flight complexity	+	+	+	?	-
6	test fly prototypes	+	+	+	?	?
7	enhance safing	+	-	-	+	?
8	certification	+	?	?	?	?
9	increase vv	+	-	-	+	?
10	reduce onboard autonomy	?	+	+	-	-
11	reuse across missions	?	+	+	?	?
12	increase developer capabilities	+	+	+	?	?
13	increase developer tool use	+	+	+	?	?
14	implement optional functions after launch	?	+	?	?	?
15	reduce vv cost	0	0	+	+	0
16	increase vv speed	0	+	0	0	0
17	increase vv capabilities	+	+	+	0	+

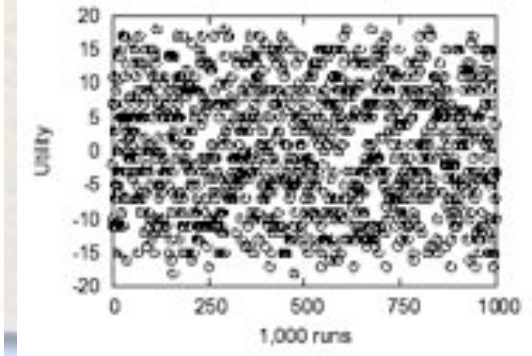
Nondeterminant: "-" and "+" = what?

Weighted Quality attributes

add

effect

Columns of qualitative influences

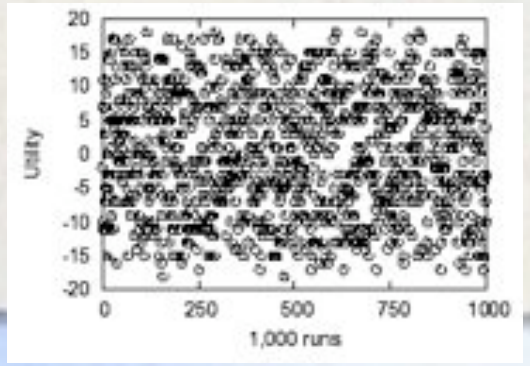


NEAR@APL
TAR3 = combine "heavy lifters"

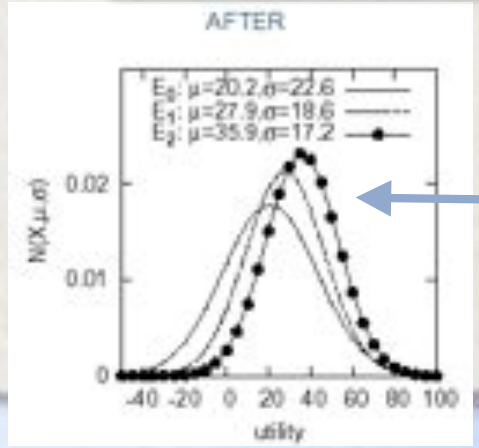
- Divide scores;
  e.g. low, medium, high = 2,4,8
- Baseline:
  e.g. (2 * # lows)
  + (4 * # mediums) + (8 * # highs)
- lift(attribute=range) =
  log((all ∩ range)/baseline)
  - Lift = 0 if useless
  - Lift > 0 if useful
  - Lift < 0 if dangerous
- TAR3: forward select search of combinations of high lifters
  - Treatment: fewest settings with most effect

Often, a few ranges with large lifts

lift1	frequency
-3:	[1]
-1:	[1]
0: ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	[59]
1: ~~~~~~~~~~	[20]
2: ~~	[5]
3: ~~~	[6]
4:	[1]



=>



Mean score doubled (20 to 35.9)

DDP @ JPL

SILAP @ IV&V

NEAR @ APL

XOMO @ AMES

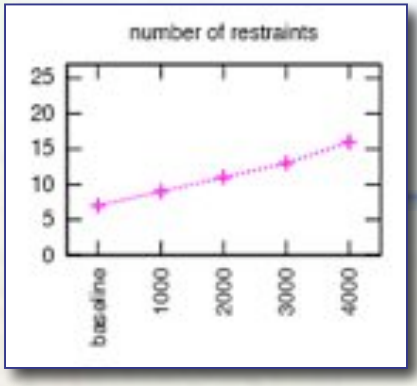
XOMO @ AMES
Optimization of COCOMO-family models

- COCOMO:
  - effort estimation
- COQUALMO:
  - bugs introduced - bugs removed
- Madachy model:
  - how many dumb things are you doing today?

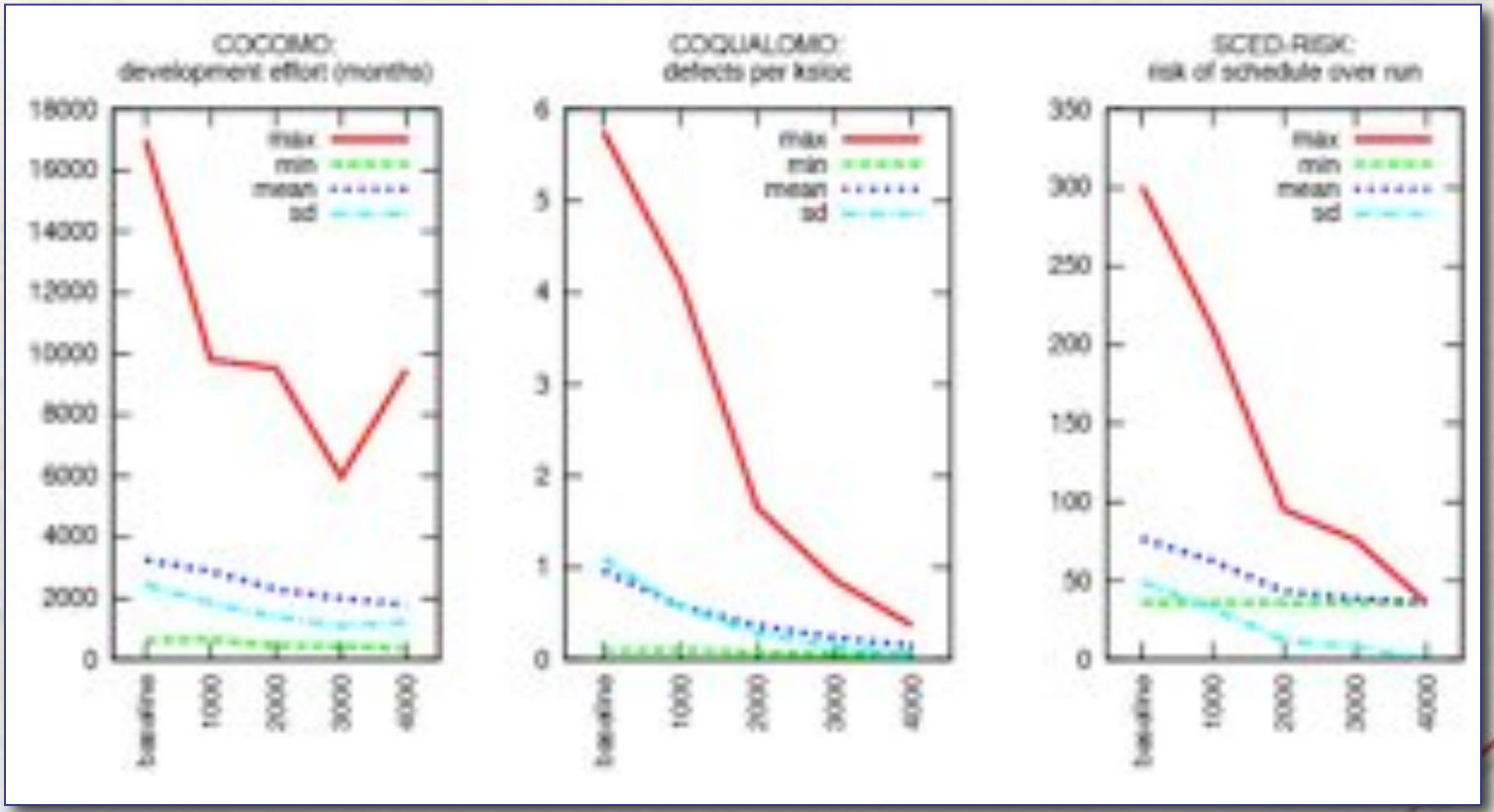
- Incremental optimization over 26 variables
  - Monte carlo simulations
  - Learn best ranges
  - More simulations, focusing on the better ranges

- Case study: building autonomous systems
  - Prec = low
  - Cplx = high
  - etc

XOMO @ AMES



baseline	learned restraints			
	1000	2000	3000	4000
75 ≤ ksloc ≤ 125 rely = 5 prec = 1 acap = 5 aexp = 1 cplx = 6 ltex = 1 ruse = 6	sced=4 peer_reviews=5	pmat=5 pcap=4	tool=4 execution_testing- _and_tools=5	team=5 resl=5 automated- analysis=5



Discussion

Counter proposals

- Won't the learning just recreate the original model?
  - No: summary much smaller
  - Finds relationships that are obscure in model.
- Why not use standard Monte Carlo methods?
  - TAR3 produces much smaller theories
- Why not model with fuzzy logic, Bayes nets, decision diagrams,..?
  - All of these impose restrictions on the modeling language
  - Funnel theory: a few master variables that set the remaining "slaves:
  - Language details less important than sampling output
  - Our goal: decisions from models written any way at all
- Why not search with genetic algorithms, neural nets, …?
  - Wasted time.
  - If master variables , master variables will be obvious
- Why not search for master variables with an ATMS?
  - ATMS' complete search takes exponential time;
  - TAR3's stochastic search takes time linear on data set size

Current work

- Massive scale up:
  - Real-time monitoring of gigabytes of data
  - From “requirements engineering” to
    - real time, run-time decision making

And so?

Sound bites

- Early, we only have some
  - Early lifecycle decision making plagued by uncertainty
- But a little can be enough
  - Within a large space of “maybes”, there may be some “key variables” that set the rest
  - So “decision making” can be just “set the keys”
- Finding the key issues is not hard
  - If they really are critical, they will reveal themselves
  - So sample a little, watch a little, try a few combinations
  - TAR3
- Decision making = use the keys (and the rest will follow)
  - Applications from JPL, IVV, APL, AMES

Questions?
Comments?

http://unbox.org/promise/2006/site.php?what=date: Go promise 2006

2nd International PROMISE Workshop

September 24, 2006, Philadelphia, Pennsylvania USA.
Co-located with the IEEE Conference on Software Maintenance

home

calls for papers
important dates
program commitee
special issue

public datasets
data format

registration
program
hotel

promise@unbox.org

Important dates

Submission of workshop papers : June 19, 2006
Notification of workshop papers : July 14, 2006
Publication ready copy : August 14, 2006
Workshop : Sunday, September 24, 2006 : see program