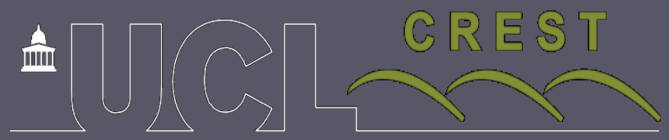


The 25th CREST Open Workshop
Requirements and Test Optimisation

Optimising overtime planning

Federica Sarro, PhD

Research Associate, CREST centre
Department of Computer Science
University College London

f.sarro@ucl.ac.uk

February 11, 2013

Motivation and Contribution of the Work

How many of you have ever worked overtime?

Image from http://raised-guides.blogspot.co.uk/2011/06/libraries-everywhere.html

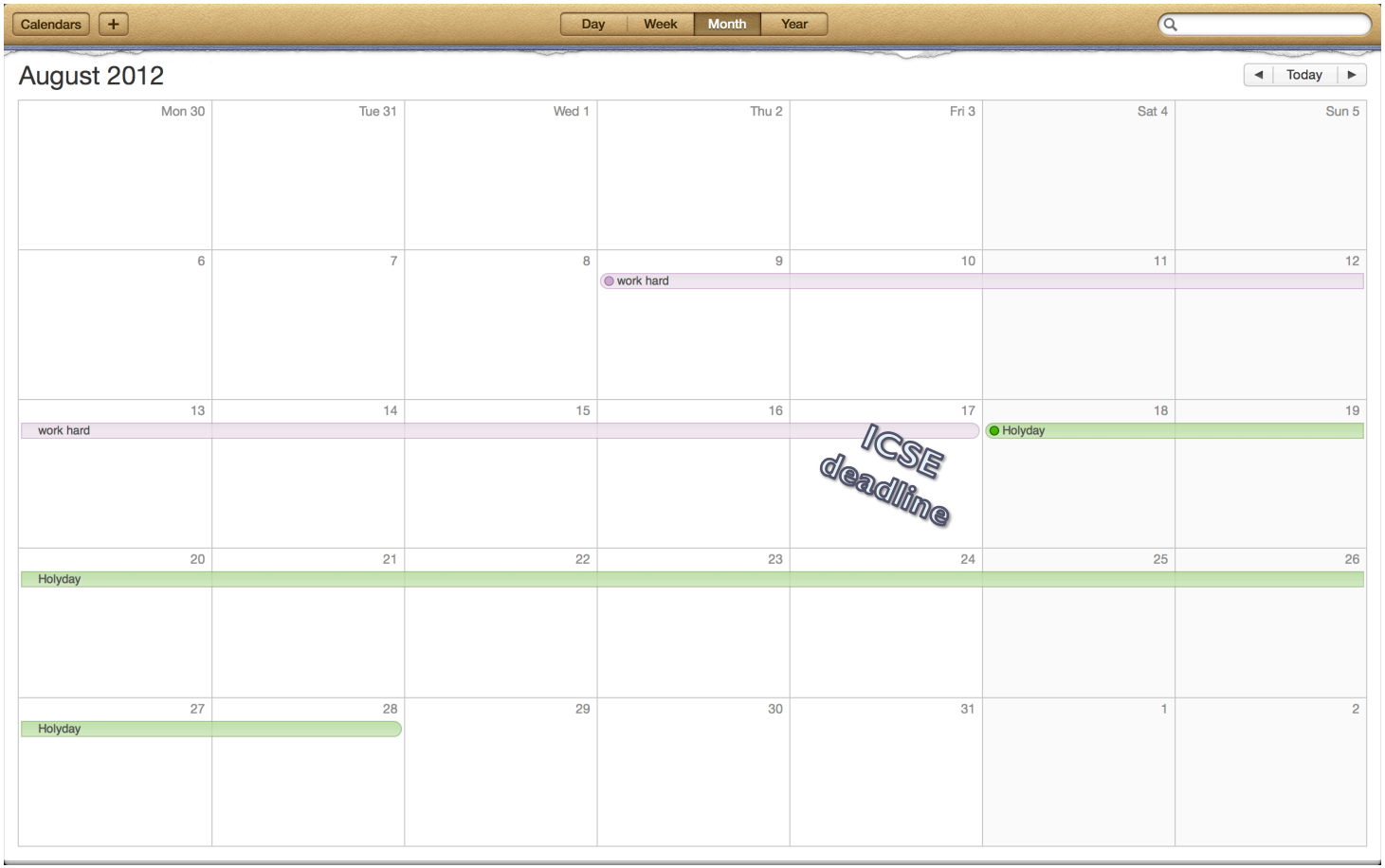
Motivation and Contribution of the Work

How many of you have ever *planned* to work overtime?

Image from http://raised-guides.blogspot.co.uk/2011/06/libraries-everywhere.html

Motivation and Contribution of the Work

- People recognize the need to plan their work and also their "extra work"!



Motivation and Contribution of the Work

- Software engineers are often pushed into high levels of unplanned overtime
- Project managers often rely on overtime to meet deadlines
  - in some areas of software development crunch periods of overtime were reported as common by 60% of programmers
    - 47% said they were not compensated

Olson, B., & Swenson, D "Overtime effects on project team effectiveness." Midwest Instruction and Computing Symposium, Duluth, Minnesota., 2011

Motivation and Contribution of the Work

- Previous studies highlighted several side effects of *unplanned overtime* on software engineering projects…
  - positive correlations between unplanned overtime and stress/ depression indicators
  - increased software defect counts

- …but also evidence that *proper overtime planning* leads to
  - greater software engineer job satisfaction
  - improved customer satisfaction
  - few of the side-effects that accompany unplanned overtime

Olson, B., & Swenson, D "Overtime effects on project team effectiveness." Midwest Instruction and Computing Symposium, Duluth, Minnesota., 2011
M. Nishikitani, M. Nakao, K. Karita, K. Nomura, and E. Yano, "Influence of overtime work, sleep duration, and perceived job characteristics on the physical and mental status of software engineers", 2005.
B. Akula and J. Cusick, "Impact of overtime and stress on software quality," in 4th International Symposium on Management, Engineering, and Informatics, 2008.
C. Mann and F. Maurer, "A case study on the impact of scrum on overtimes and customer satisfaction," in Agile Development 2005.
D. G. Beckers, D. van der Linden, P. G. Smulders, M. A. Kompier, T. W. Taris, and S. A. Geurts, "Voluntary or involuntary? control over overtime and rewards for overtime in relation to fatigue and work satisfaction,". 33–50, 2008.

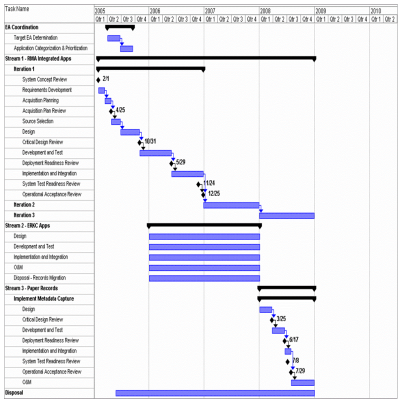
Motivation and Contribution of the Work

- Proper overtime planning on SE projects faces with human issues and other crucial decisions
  - When should you start project overtime?
    - in the early part of the project? only in case of project overrun? …
  - Which activities should you speed up?
    - those on the critical path (*cp*)?
      - speeding up one activity on the *cp*, the *cp* itself may change
    - which activity on the *cp* should you speed up?
      - some may be more expensive than others
    - how much should you speed it up?
      - …

Motivation and Contribution of the Work

- There has been no research aimed at providing support to software engineers in their attempts to plan for overtime
- We introduced an approach to support software engineers in better planning for overtime while managing risk
- Contribution of our work
  - multi-objective formulation of the project overtime planning problem
  - empirical study on 6 real world software projects
  - analysis of different risk assessment models
  - actionable insights into project planning tradeoffs using Pareto fronts obtained by our approach

Problem Formulation



- Work Breakdown Schedule (WBS) produced by a software engineer
- WBS modeled as an acyclic directed graph
  - nodes = work packages (+effort and duration)
  - edges = dependencies beetween wps

- Analyse the effects of choices of *overtime* assignments on *project duration* and *risk of overrun*

Problem Formulation

- Candidate solution: assignment of overtime to work packages that seeks to minimise
  - Overtime (O) $$O = \sum_{i=0}^{n} Overtime(wp_i)$$
  - Project Duration (D) $$D = \sum_{wp \in CP} Duration(wp)$$
  - Risk of Overrun (R)

$$R = R_{AvgRisk} = \frac{\sum_{p \in \Pi} risk_p}{|\ \Pi\ |}$$

$$R = R_{MaxRisk} = max_{p \in \Pi - CP} risk_p$$

$$R = R_{TrsRisk}(L) = \frac{|\ \{p \cdot p \in \Pi \wedge risk_p > L\}\ |}{|\ \Pi\ |} \cdot 100$$

The Solution Approach: Computational Search

- Non dominated Sort Genetic Algorithm-II (NSGAII)
  - widely used Multi-Objective Evolutionary Algorithm
  - an objective vector is considered <O, D, R>
  - the fitness assignment is based on the concepts of non-dominance and crowding distance
- NSGAIIv
  - same characteristics as the standard NSGAII but…
  - …exploits a new crossover operator that aims to preserve genes shared by the fittest overtime assignments
    - avoiding the well-known disruptive effects of crossover

The Solution Approach: Representation

- Each solution is encoded as a chromosome
  - each gene represents the amount of overtime assigned to a given work package (wp)

	wp_1	wp_2	wp_3	…	wp_n
Assigned overtime	2	1	0	…	3

The Solution Approach: Fitness Function

- To evaluate the fitness of each chromosome we employed a multi-objective function to simultaneously minimise
  - Overtime (O), Project Duration (D) and Risk of Overrun (R)

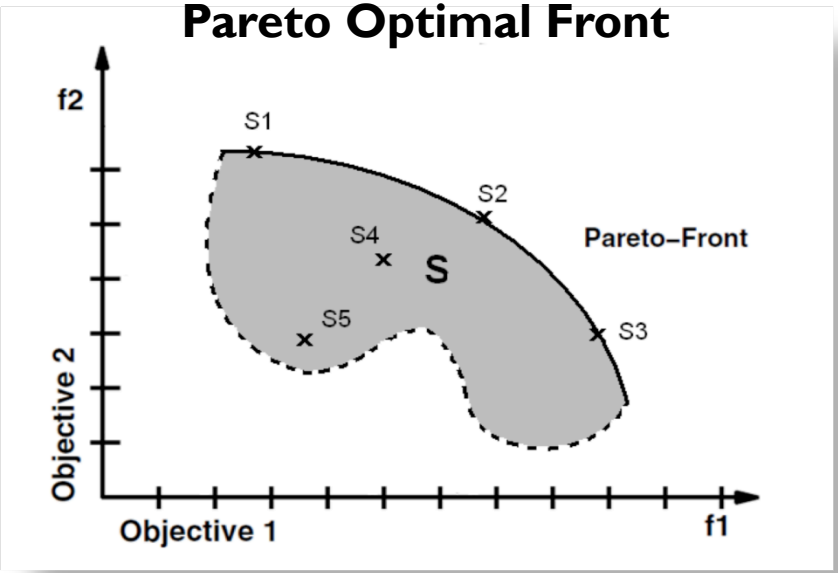
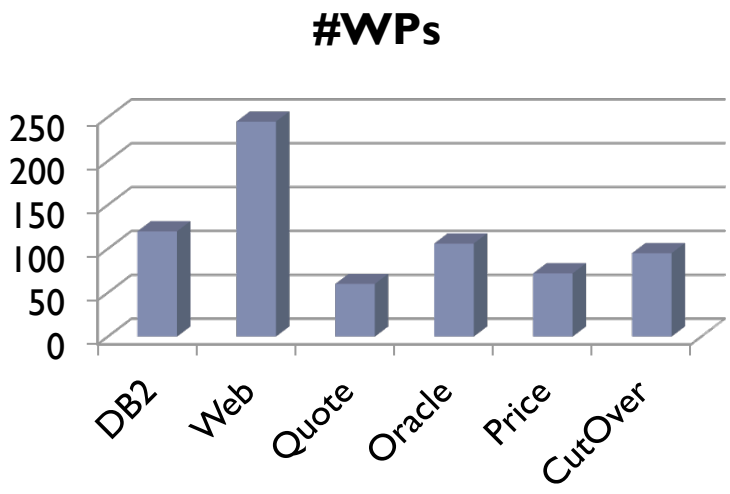


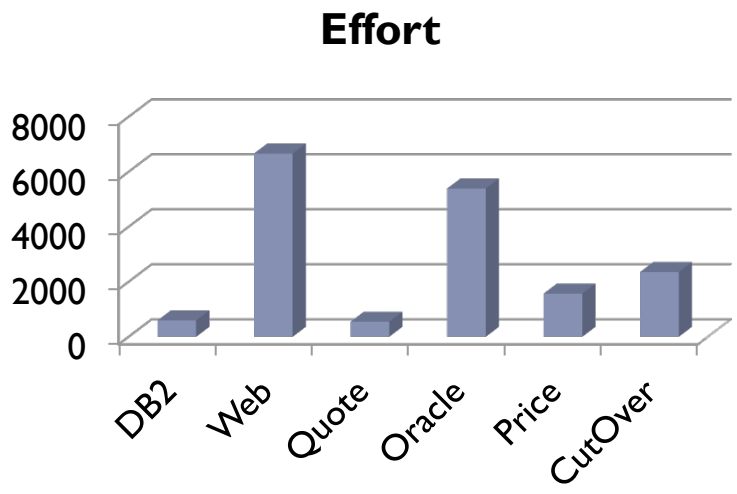
Figure by Yuanyuan Zhang, Multi-Stakeholder Tensioning Analysis in Requirements Optimisation

"a solution A is said to dominate a solution B if and only if A is no worse than B in all objectives, and A is strictly better than B in at least one objective"

Software Projects Employed in the Empirical Study

- WBS of 6 real software projects coming from three different organisations
  - different kinds of software engineering development
  - different size: from 60 to 245 work packages
  - different duration: from a few person weeks to several person years





Research Questions

- RQ1 (SBSE Validation): How do NSGAII and NSGAII*v* perform compared to random search?
- RQ2.1 (Comparison to State of the Art Search): How does NSGAII*v* perform compared to NSGAII?
- RQ2.2 (Usefulness): How does NSGAII*v* perform compared to currently used overtime planning approaches?
- RQ3 (Insight): Can our approach yield useful insights into the trade offs between objectives for real world software projects?
- RQ4 (Impact of Risk Assessment Models): What is the difference between the three approaches to risk measurement?

Research Questions

- RQ1 (SBSE Validation): How do NSGAII and NSGAIIv perform compared to random search?
- RQ2.1 (Comparison to State of the Art Search): How does NSGAIIv perform compared to NSGAII?
- RQ2.2 (Usefulness): How does NSGAIIv perform compared to currently used overtime planning approaches?
- RQ3 (Insight): Can our approach yield useful insights into the trade offs between objectives for real world software projects?
- RQ4 (Impact of Risk Assessment Models): What is the difference between the three approaches to risk measurement?

Analysis of Results: RQ1 and RQ2.1

- RQ1 (SBSE Validation): How do NSGAII and NSGAIIv perform compared to random search?
- RQ2.1 (Comparison to State of the Art Search): How does NSGAIIv perform compared to NSGAII?

- Answers
  - RQ1: NSGAII and NSGAIIv achieved significantly superior results compared to random search with an 'high' effect size
  - RQ2.1: NSGAIIv outperformed the standard NSGAII in 41 out of 54 (76%) experiments
    - in 35 of these 41 (85%) it does so with a Cohen effect size 'high'
    - NSGAII did not outperform NSGAIIv in any of the experiments

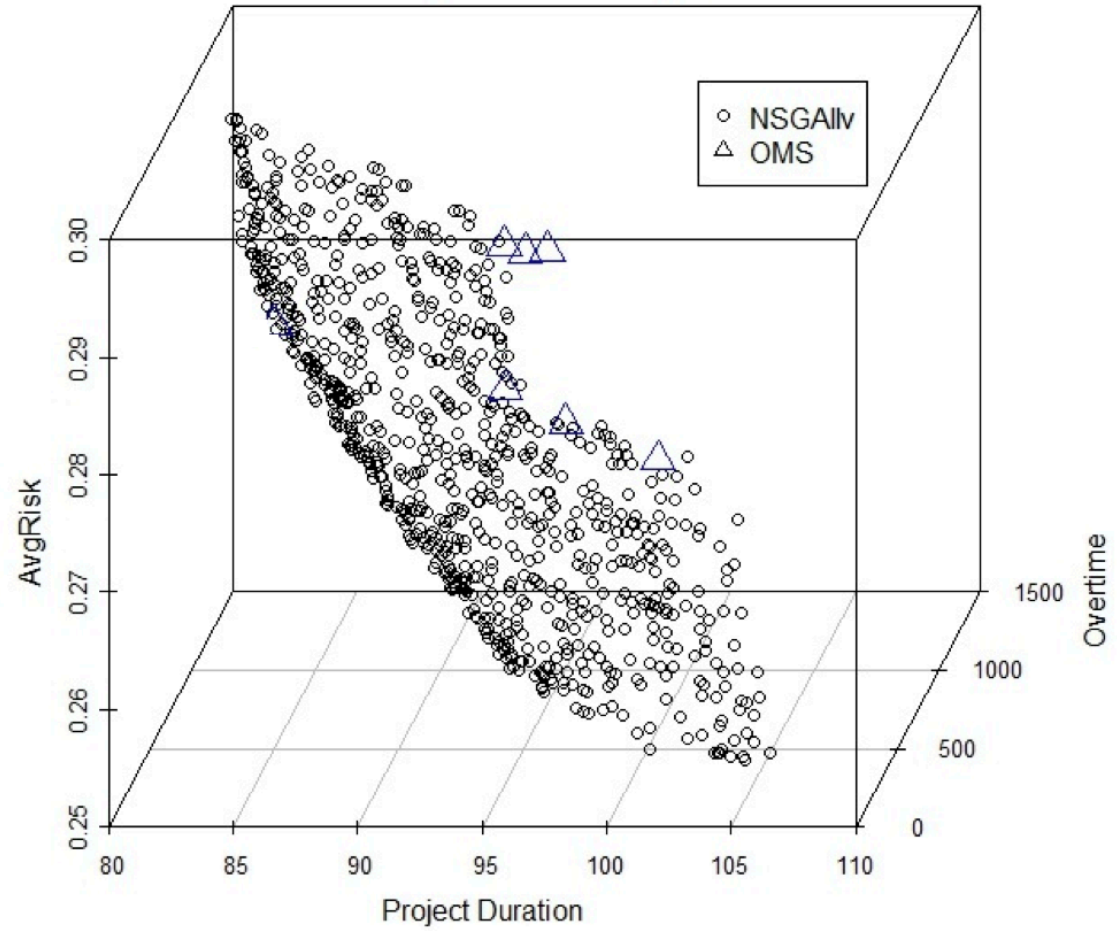
Research Questions

- RQ1 (SBSE Validation): How do NSGAII and NSGAII*v* perform compared to random search?
- RQ2.1 (Comparison to State of the Art Search): How does NSGAIIv perform compared to NSGAII?
- RQ2.2 (Usefulness): How does NSGAII*v* perform compared to currently used overtime planning approaches?
- RQ3 (Insight): Can our approach yield useful insights into the trade offs between objectives for real world software projects?
- RQ4 (Impact of Risk Assessment Models): What is the difference between the three approaches to risk measurement?

Analysis of Results: Usefulness

- RQ2.2 How does NSGAII*v* perform compared to currently used overtime planning approaches?
  - Current overtime planning practice
    - spreading the overtime over all work packages (*margarine management*)
    - loading overtime onto the critical path to reduce completion time
    - loading overtime onto the later half of the project to compensate for earlier delays

Analysis of Results: Usefulness



Pareto surfaces for NSGAIIv (circles) and for all the three Overtime Management Strategies (triangles) obtained using AvgRisk for the project Web

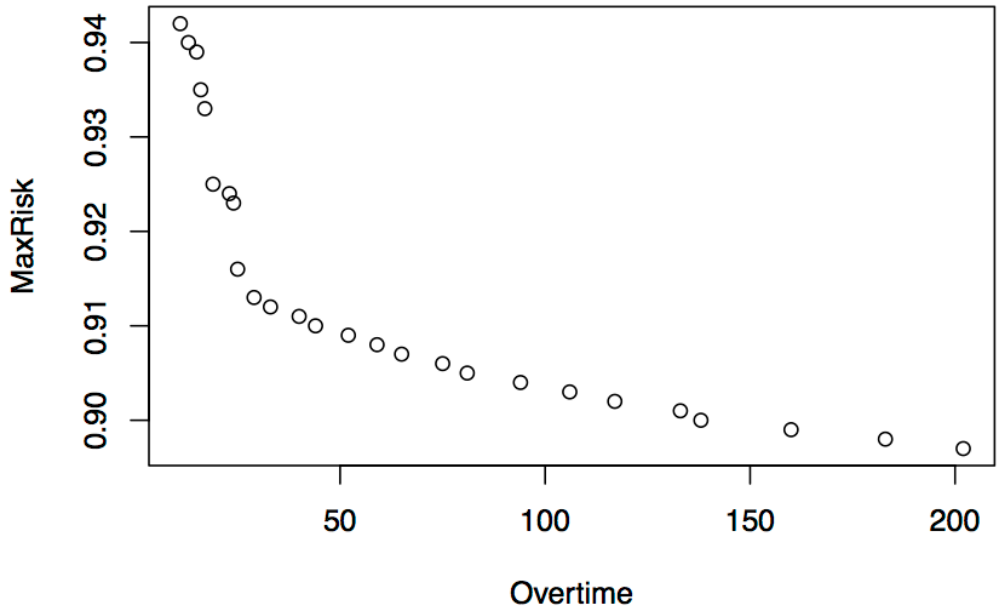
Analysis of Results: Insight

- RQ3 Can our approach yield useful insights into the trade offs between objectives for real word software projects?
  - How much spend on overtime is cost effective for my project plan?
  - What must I spend to reduce overrun risk by x%?
    - double overtime, double rewards?

It would naturally be tempting to seek the maximum overtime budget allowable to ensure that there is the largest resource available to deal with problems

Analysis of Results: Insight

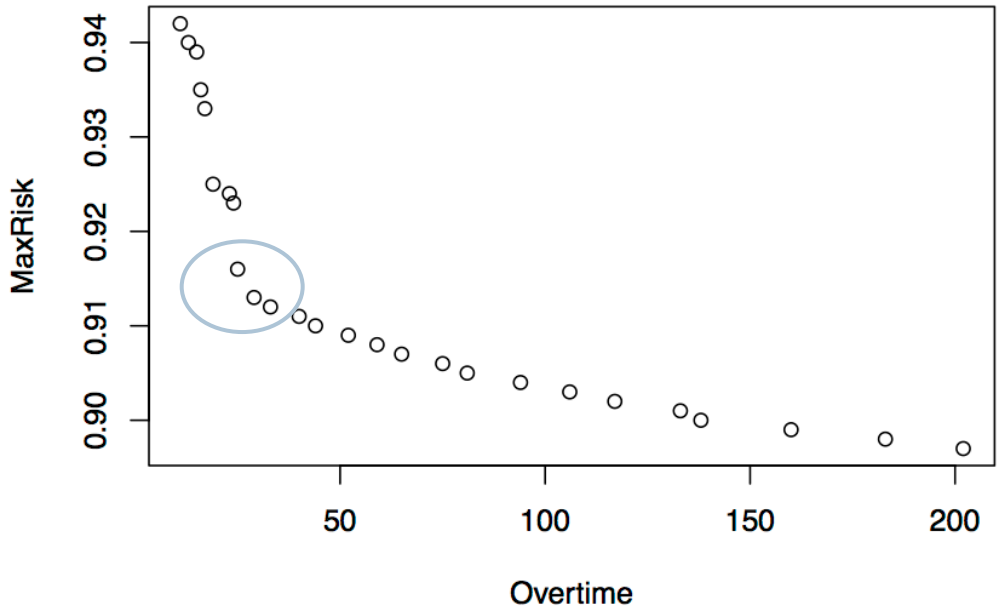
- RQ3 Can our approach yield useful insights into the trade offs between objectives for real word software projects?
  - How much spend on overtime is cost effective for my project plan?
  - What must I spend to reduce overrun risk by x%?
    - double overtime, double rewards?



2D Pareto surface projections for NSGAIIv obtained using AvgRisk for the project Price

Analysis of Results: Insight

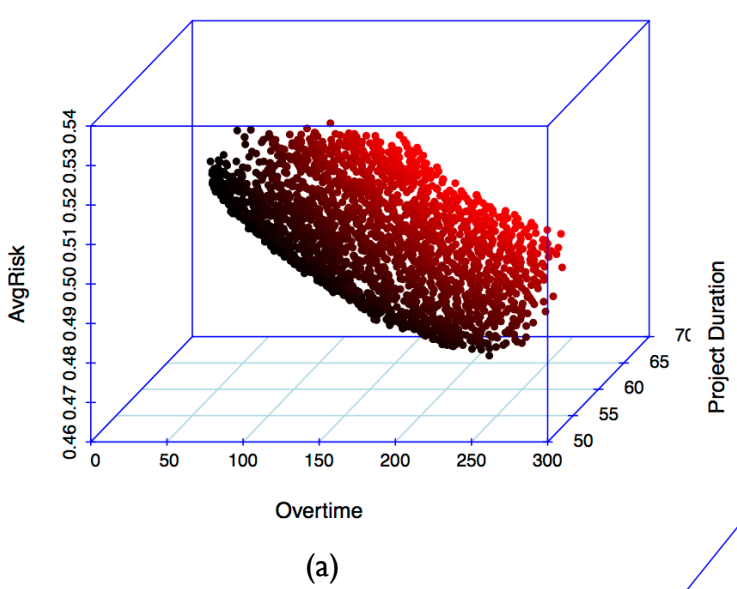
- RQ3 Can our approach yield useful insights into the trade offs between objectives for real word software projects?
  - How much spend on overtime is cost effective for my project plan?
  - What must I spend to reduce overrun risk by x%?
    - double overtime, double rewards?



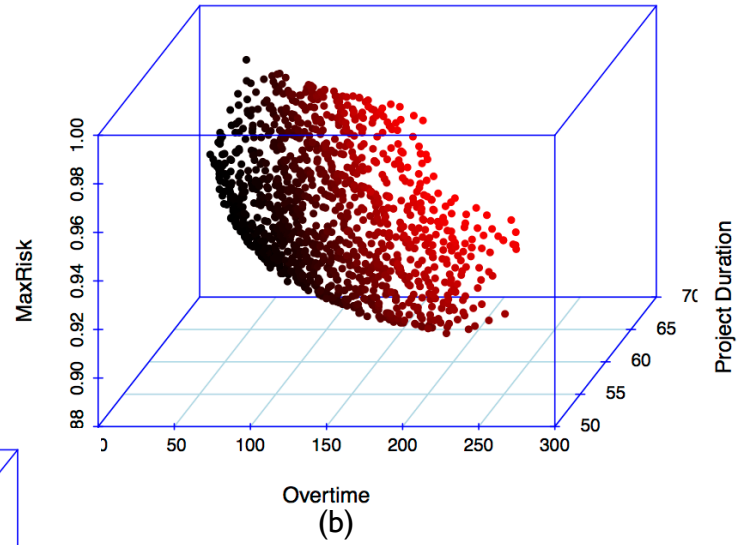
2D Pareto surface projections for NSGAIIv obtained using AvgRisk for the project Price

Analysis of Results: Impact of Risk Assessment Models

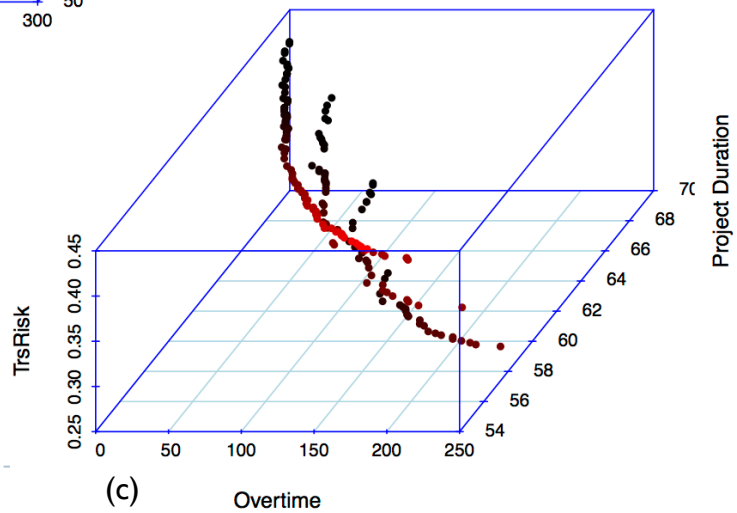
- RQ4 What is the difference between the three approaches to risk measurement?



(a)



(b)



(c)

Pareto surface for the project Price obtained using each of the three risk assessment approaches

Conclusions and Future Work

- We introduced a search based approach to overtime planning on software engineering projects
- We evaluated it on 6 real world projects
  - it performs significantly better than currently used overtime practice and than a standard multi-objective optimisation
  - it can provide actionable insights to the software engineer
- We plan to deploy a freely available, open source plugin component to popular project planning tools
  - this will allow evaluation of the interface between technical aspects and related socio-technical issues

Thanks for your attention

Not Going to Take this Anymore:
Multi-Objective Overtime Planning for Software Engineering Projects

Filomena Ferrucci[1], Mark Harman[2], Jian Ren[2], Federica Sarro[2]
[1]University of Salerno
[2]University College London

Federica Sarro
f.sarro@ucl.ac.uk