



Modelling Multi-tier Enterprise Applications Behaviour with Design of Experiments Technique

Tatiana Ustinova, Pooyan Jamshidi
Imperial College London

DICE

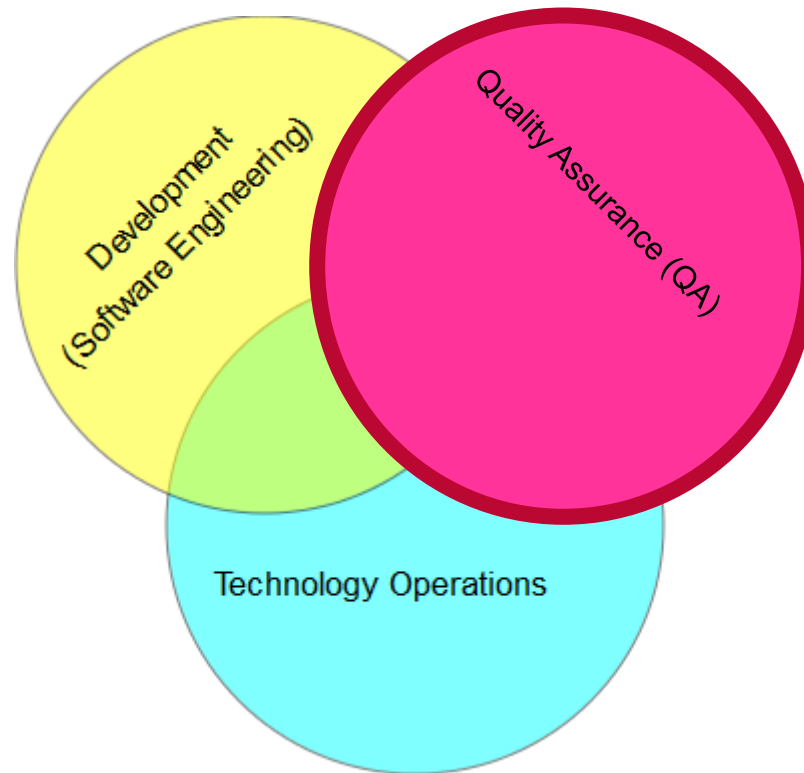
Horizon 2020 Research & Innovation Action

Grant Agreement no. 644869

<http://www.dice-h2020.eu>



Funded by the Horizon 2020
Framework Programme of the European Union



<https://upload.wikimedia.org/wikipedia/commons/b/b5/Devops.svg>

Research Aim and Objectives



Aim:

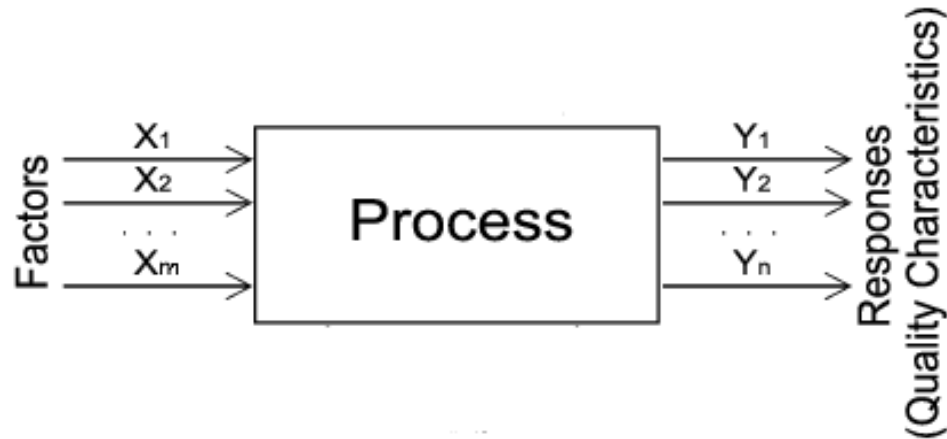
Is DoE good to model and predict application performance?

Objectives:

1. Introduce DoE.
2. Build app performance model.
3. Model prediction accuracy.

Test environment: 3-tier web-based enterprise application

Design of Experiments: Introduction



Linear Regression model: $\mathbf{Y} = \mathbf{I} + \mathbf{A} * \mathbf{X} + \boldsymbol{\varepsilon}$

where

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \quad \mathbf{I} = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{bmatrix} \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

I – intercept
 ε – error term

Design of Experiments: Introduction

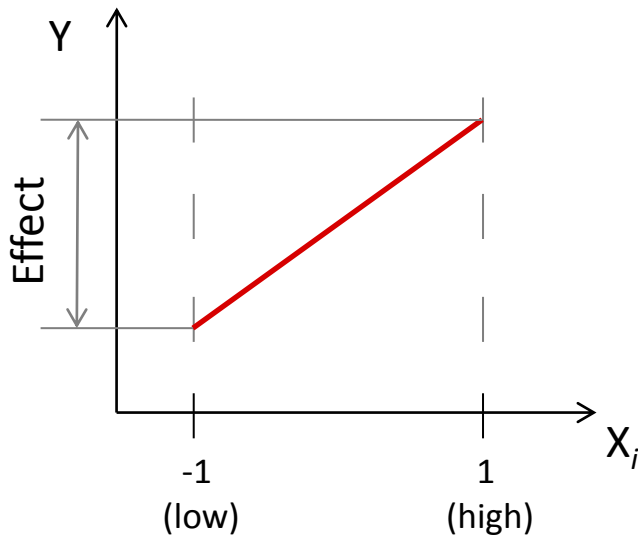


1. How to choose values for factors?
2. How many experiments to fit the model?
3. What if there are too many factors?

DoE: Screening Procedure



Two levels
for each factor



	Levels	
	Low (-1)	High (1)
Number of users	3	20
User think time, s	10	1
Execution time, min (steady state)	10	30
Workload mix (user class)	I	III

$2^4 = 16$ 5 h 20 min execution time

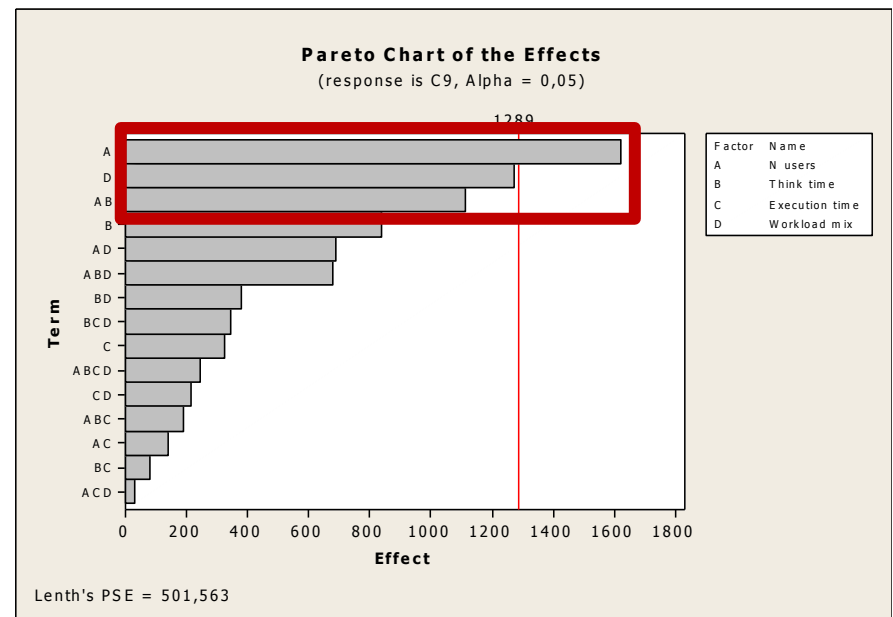
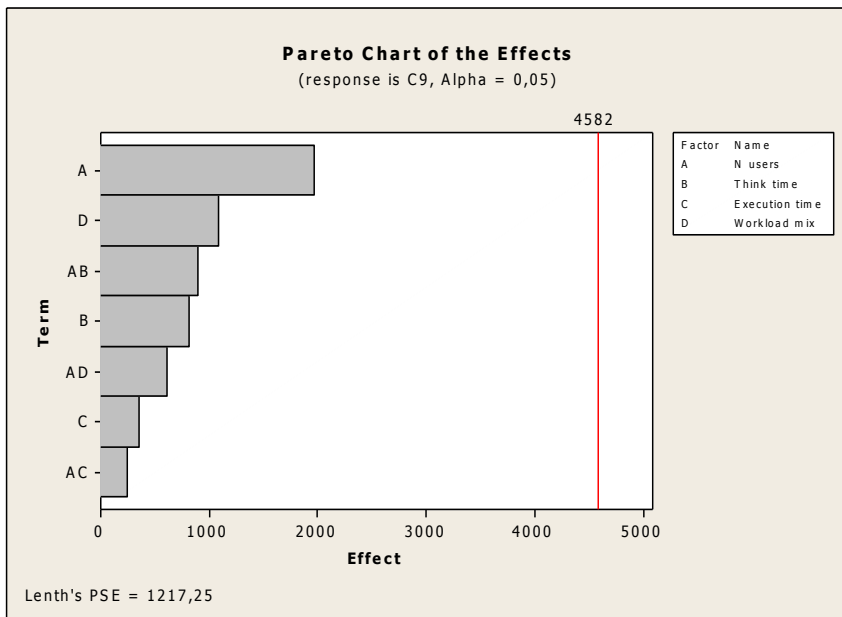
Full Factorial
Design

2^k runs, where
 k – number of
factors

Fractional Factorial Design for 4 factors

$2^{4-1} = 8$ runs

DoE: Screening procedure (contd.)



DoE: Screening procedure (contd.)



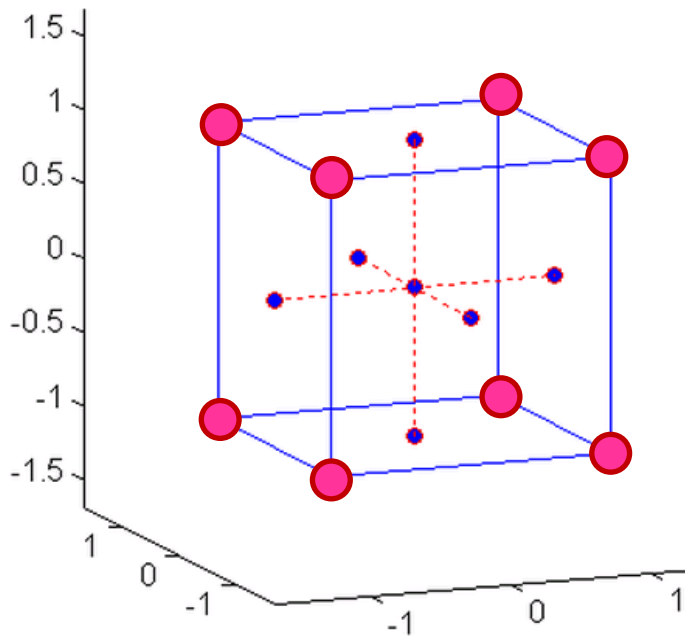
Allocation of variation, %

Effect	Response time	CPU utilisation
N_users	26.03	54.27
Think time	4.53	42.99
User class	36.25	1.14
N_users:Think time	19.13	0.59
N users>User class	6.63	7.886×10^{-6}
Think time>User class	1.5×10^{-8}	1.8917×10^{-4}
N_users:Think time: User class	5.42	0.91
Error	2.01	7.6946×10^{-4}

DoE: Constructing the Model



Box-Wilson Response Surface Design



24 runs

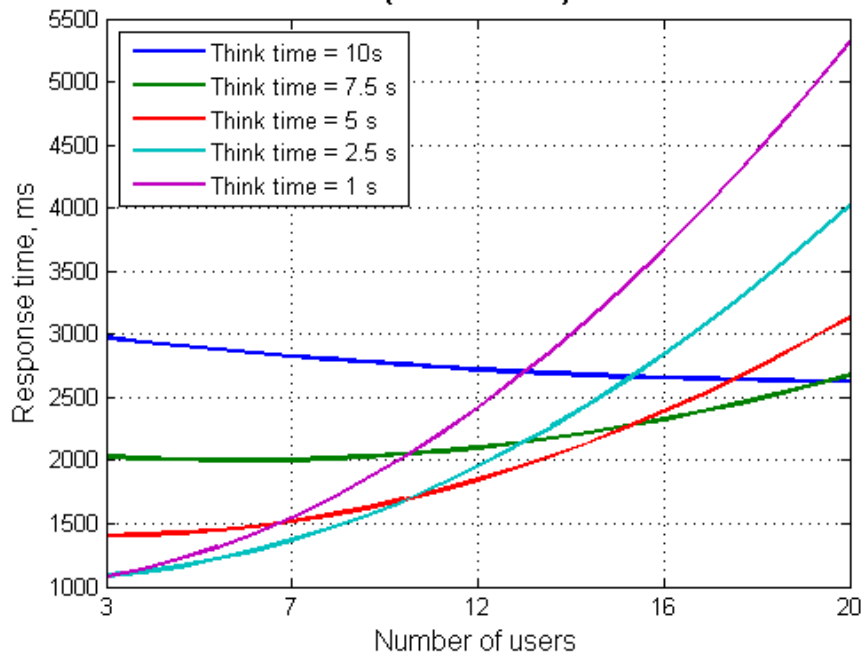
Linear Regression Models

Name	Formula
Linear	$y = I + a_1X_1 + a_2X_2 + a_3X_3$
Interactions	$y = I + a_1X_1 + a_2X_2 + a_3X_3 + a_4X_1:X_2 + a_5X_1:X_3 + a_6X_2:X_3$
Pure Quadratic	$y = I + a_1X_1 + a_2X_2 + a_3X_3 + a_4X_1^2 + a_5X_2^2 + a_6X_3^2$
Quadratic	$y = I + a_1X_1 + a_2X_2 + a_3X_3 + a_4X_1:X_2 + a_5X_1:X_3 + a_6X_2:X_3 + a_7X_1^2 + a_8X_2^2 + a_9X_3^2$
Full Polynomial	$y = I + a_1X_1 + a_2X_2 + a_3X_3 + a_4X_1:X_2 + a_5X_1:X_3 + a_6X_2:X_3 + a_7X_1:X_2:X_3 + a_8X_1^2 + a_9X_2^2 + a_{10}X_3^2 + a_{11}X_1^2:X_2 + a_{12}X_1:X_2^2 + a_{13}X_1^2:X_3$

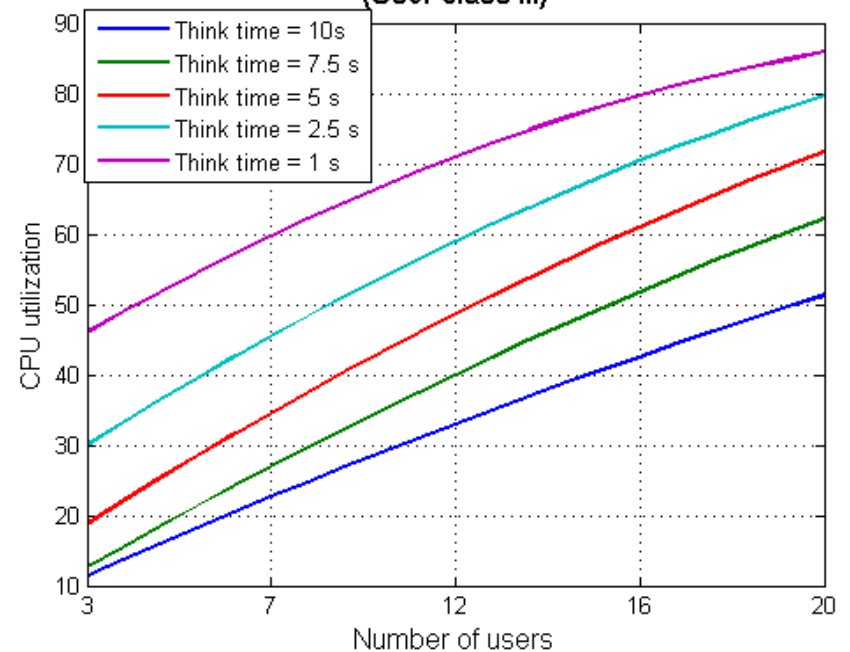
DoE: Constructing the Model (cont.)



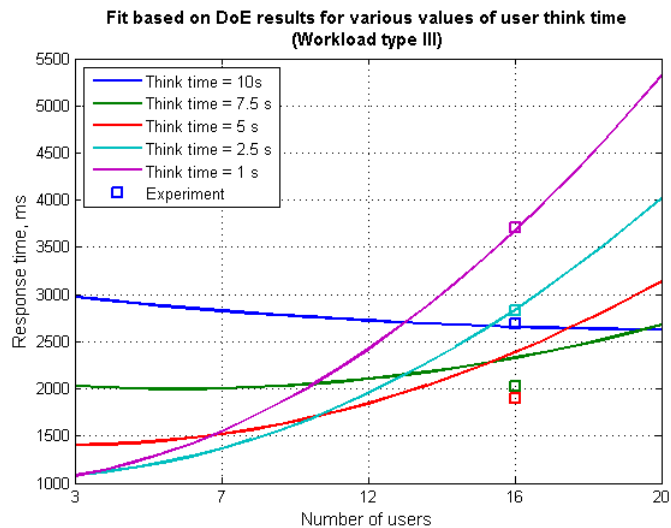
Fit based on DoE results for various values of user think time
(User class III)



Fit based on DoE results for various values of user think time
(User class III)



Model Prediction Accuracy



Prediction accuracy: error and bias

		Total prediction error σ , %		Bias, %	
		RT	CPU	RT	CPU
Response Surface models	Linear	6.51	4.3	-3.62	-0.75
	Interactions	6.32	4.09	-2.6	-0.65
	Pure quadratic	5.11	4.93	-2.02	-0.79
	Quadratic	5.42	4.09	-1.0	-0.69
	Full polynomial	5.12	4.06	-1.97	-0.96
FF		6.896	3.987	-4.96	-0.32
MVA		40.0	11.4	-234.6	7.29

Conclusions



- DoE prediction accuracy: 5-6% for RT and 4-5% for Ucpu. Out-of-the box QN algorithm - 40% and 12% respectively.
- DoE captured app's 'anomalous' behaviour w/o information about its 'insides'.
- Screening: 3 factors - 98% of variation in RT and 99.9% in Ucpu.
- Fractional factorial designs - use with care.



- 1. What is the place of DoE in Continuous Testing?**
- 2. CT presents challenges to DevOps. Design of Experiments might be a solution.**



Thank you!