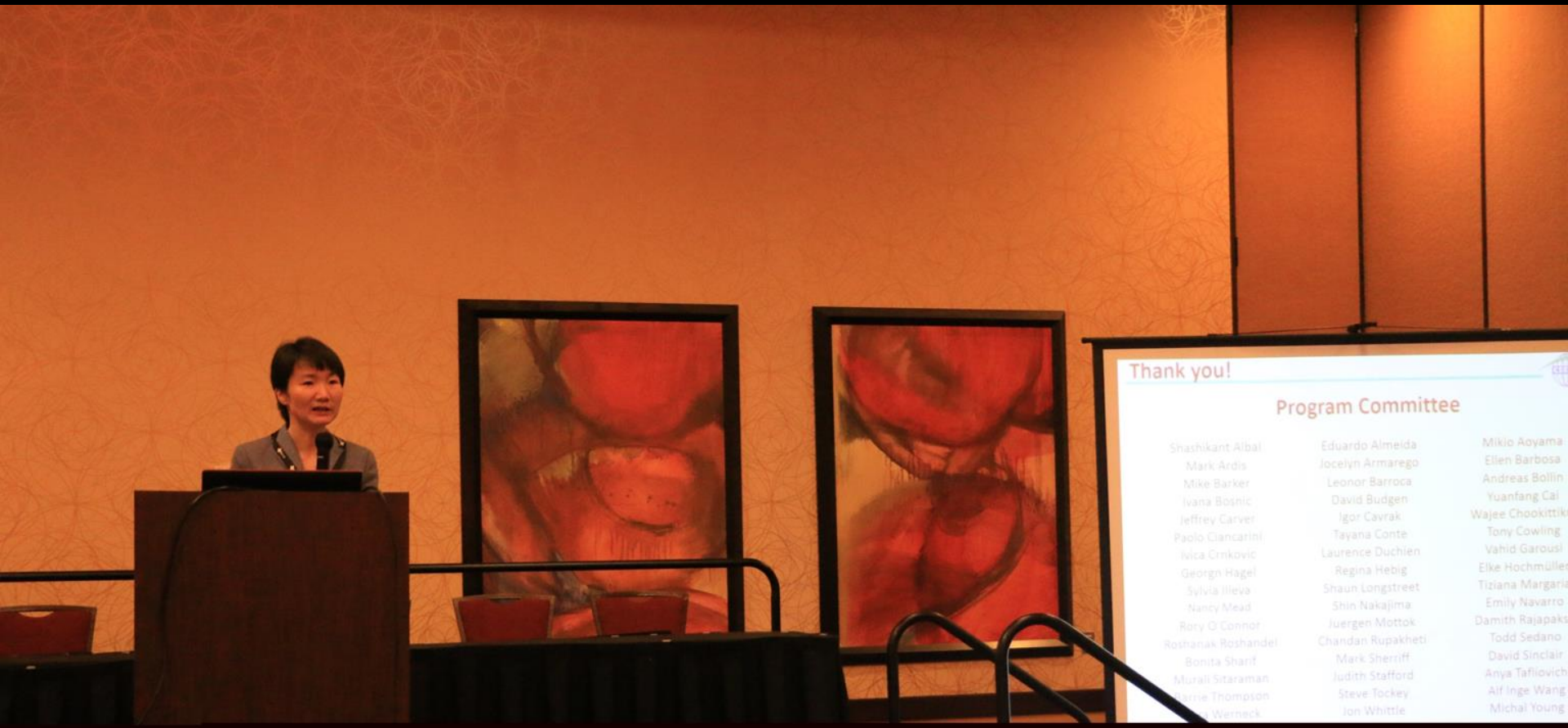
Pictures at CSEE&T 2016

Keynotes, Panels & Presentations

Day 1, April 5


EXIT
EXIT


Xuelin
Li


Thank you!
Program Committee
Shashikant Albal
Mark Ardis
Mike Barker
Ivana Bosnic
Jeffrey Carver
Paolo Ciancarini
Ivica Crnkovic
Georgn Hagel
Sylvia Ilieva
Nancy Mead
Rory O Connor
Rothanak Roshandel
Bonita Sharif
Murali Sitaraman
Eduardo Almeida
Jocelyn Armarego
Leonor Barroca
David Budgen
Igor Cavrak
Tayana Conte
Laurence Duchien
Regina Hebig
Shaun Longstreet
Shin Nakajima
Juergen Mottok
Chandan Rupakheti
Mark Sherriff
Judith Stafford
Steve Tockey
Jon Whittle
Mikio Aoyama
Ellen Barbosa
Andreas Bollin
Yuanfang Cai
Tony Cowling
Vahid Garousi
Elke Hochmüller
Emily Navarro
Damith Rajapaks
Todd Sedano
David Sinclair
Anya Tafliovich
Alf Inge Wang
Michal Young


Program Committee Geographical Distribution
• From 4 continents, 19 countries
• Total 53 reviewers
– Delivered 127 reviews
– 2 or 3 reviews per paper (average: 2.4)


Michael
Barker
Nara Institute of Science and
Technology


Michael
Barker


Daniel
Port


Daniel
Port


CSEET 2016
Software Engineering Core in an Age of Specialization
Anthony J. Lattanze


Carnegie Mellon


CSEET 2016
Software Engineering Core
in an Age of Specialization
Anthony J. Lattanze
Director, Professional Software Engineering Programs
Carnegie Mellon University, Pittsburgh PA, USA


Skedemiks


SWE Industrial Landscape
(Circa 1999)


SWE Industrial Landscape
(Circa 1969)


Michael
Barker


Carnegie Mellon


Rogardt
Heldal
Chalmers University of
Technology


Len
Bass
Professional Education
Consortium

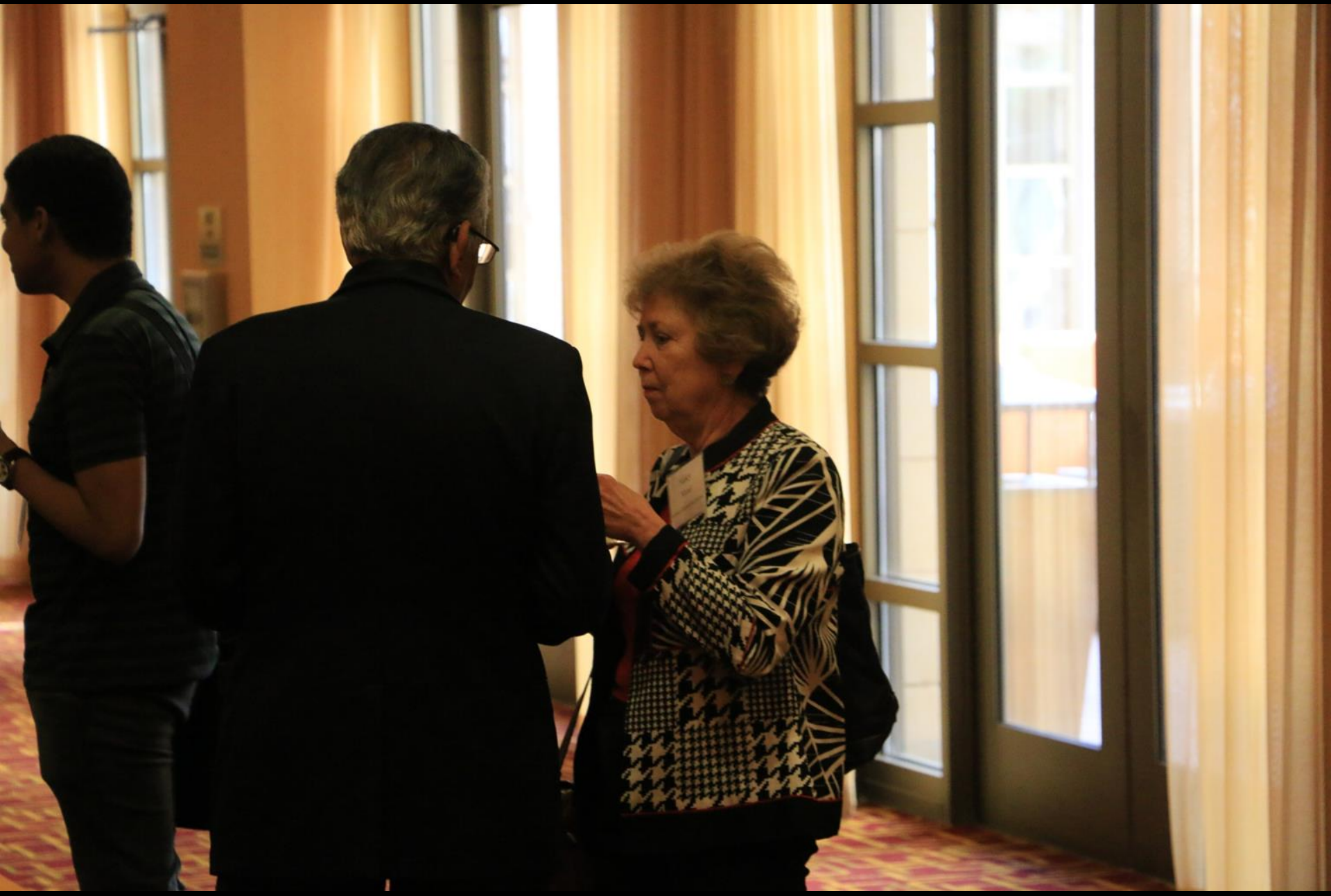

FIRE


EXIT
Topics


EXIT


Topics
What is continuous delivery and what problem does it solve?
What skills do the students need and what is the industry need for such skills?
What do we teach in our courses and what supporting material do we use?
How advanced is the material and what level of students should be taught this material?


FIRE
Christopher
Parson




Emily
Hill
Daniel
Pratt

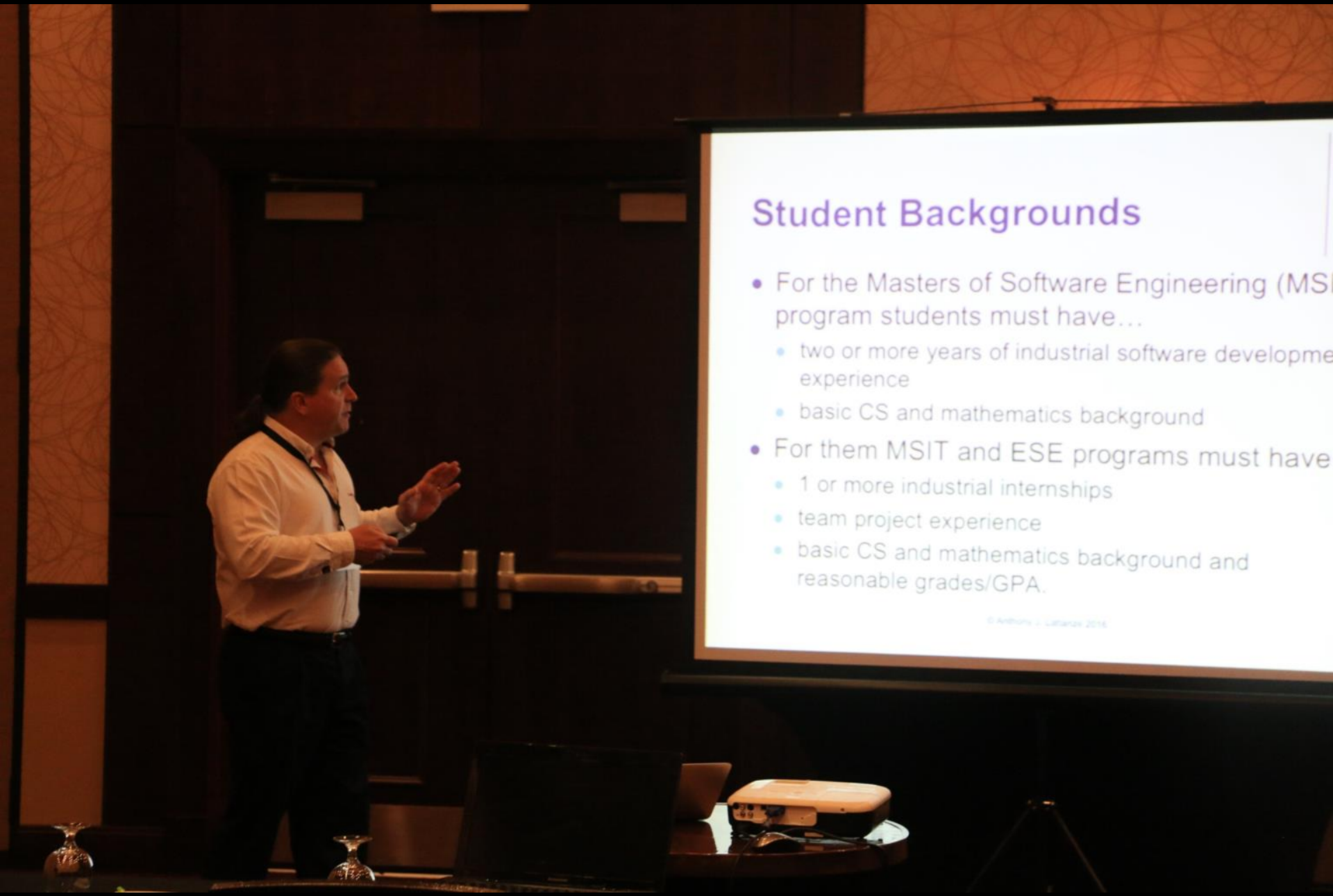

Andreas
Meier

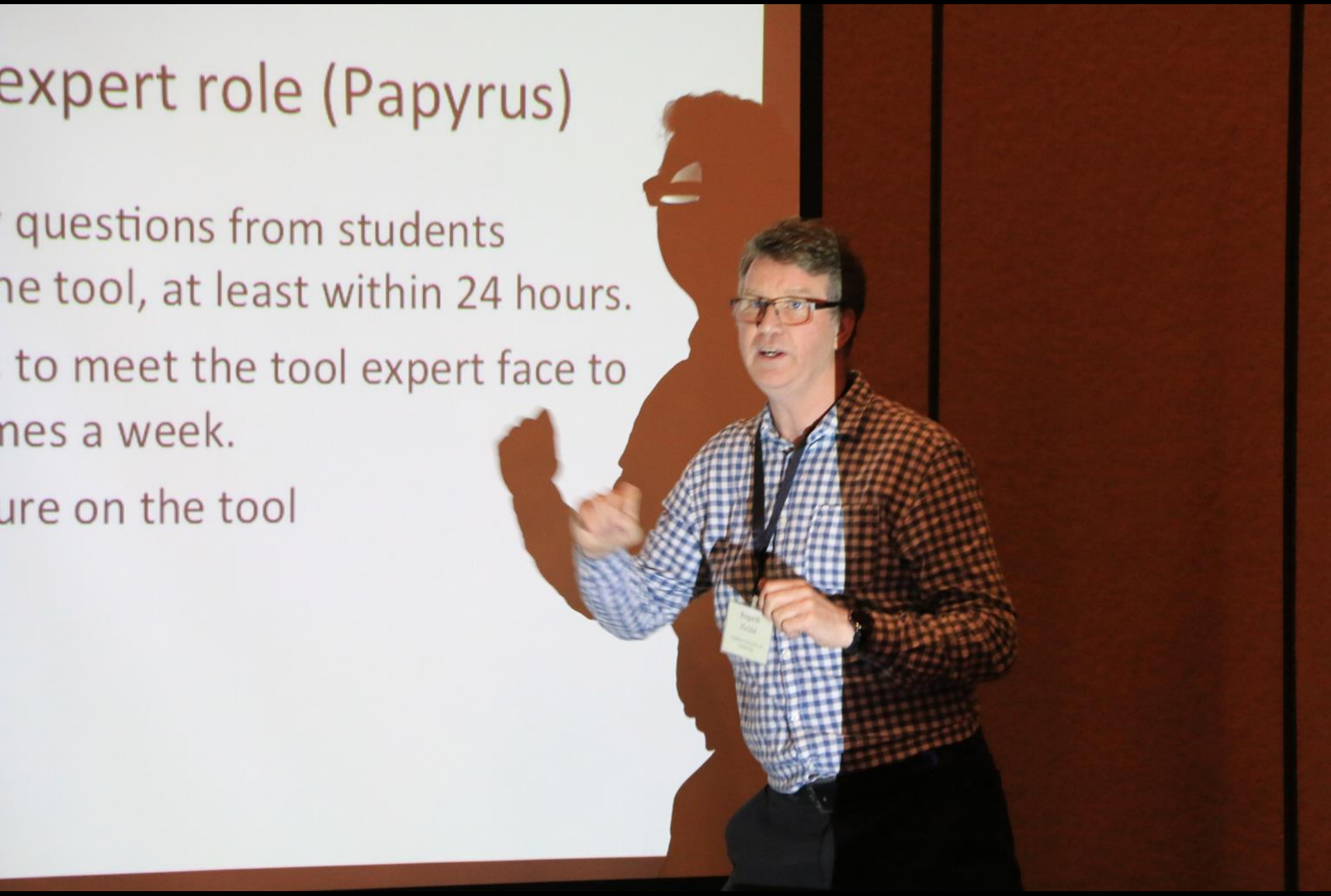

Patrick
Letouze


CHALMERS
UNIVERSITY OF GOTHENBURG
Impact of the Use of Industrial Modelling Tools on Modelling Education
Grischa Liebel, Rogardt Heldal, and Jan-Philipp Steghöfer


r tools?
The students can teach themselves the tools, that should be trivial


Teach the ideas!
Want to teach the ideas:
Do not want to teach the tools:


Student Backgrounds
• For the Masters of Software Engineering (MSI
program students must have…
• two or more years of industrial software developme
experience
• basic CS and mathematics background
• For them MSIT and ESE programs must have
• 1 or more industrial internships
• team project experience
• basic CS and mathematics background and
reasonable grades/GPA.


expert role (Papyrus)
questions from students
ne tool, at least within 24 hours.
to meet the tool expert face to
nes a week.
ure on the tool


Much Direction – 3
Directed process framework:
specified that all teams use Team Software Process (TSP) - training provided by the SEI
provided structure and consistency
Strengths and weaknesses:
students did not think, did not practice judgment in how to select, instantiate, and tailor processes
after a time, teams merely went "through the motions" – do something to make mentors happy
if teams are bent on hacking, they will hack… issuing a fixed process framework won't help
HOW?
© Anthony J. Lattanze 2016
19


leeting Industry Needs
eturn on Investment
Conclusions and Lessons Learned
Nancy
Mead
3
Software Engineering Institute


until recently
Nancy
Mead


WHAT IS CROSSFIT?
And what does it have to do with SE Ed?

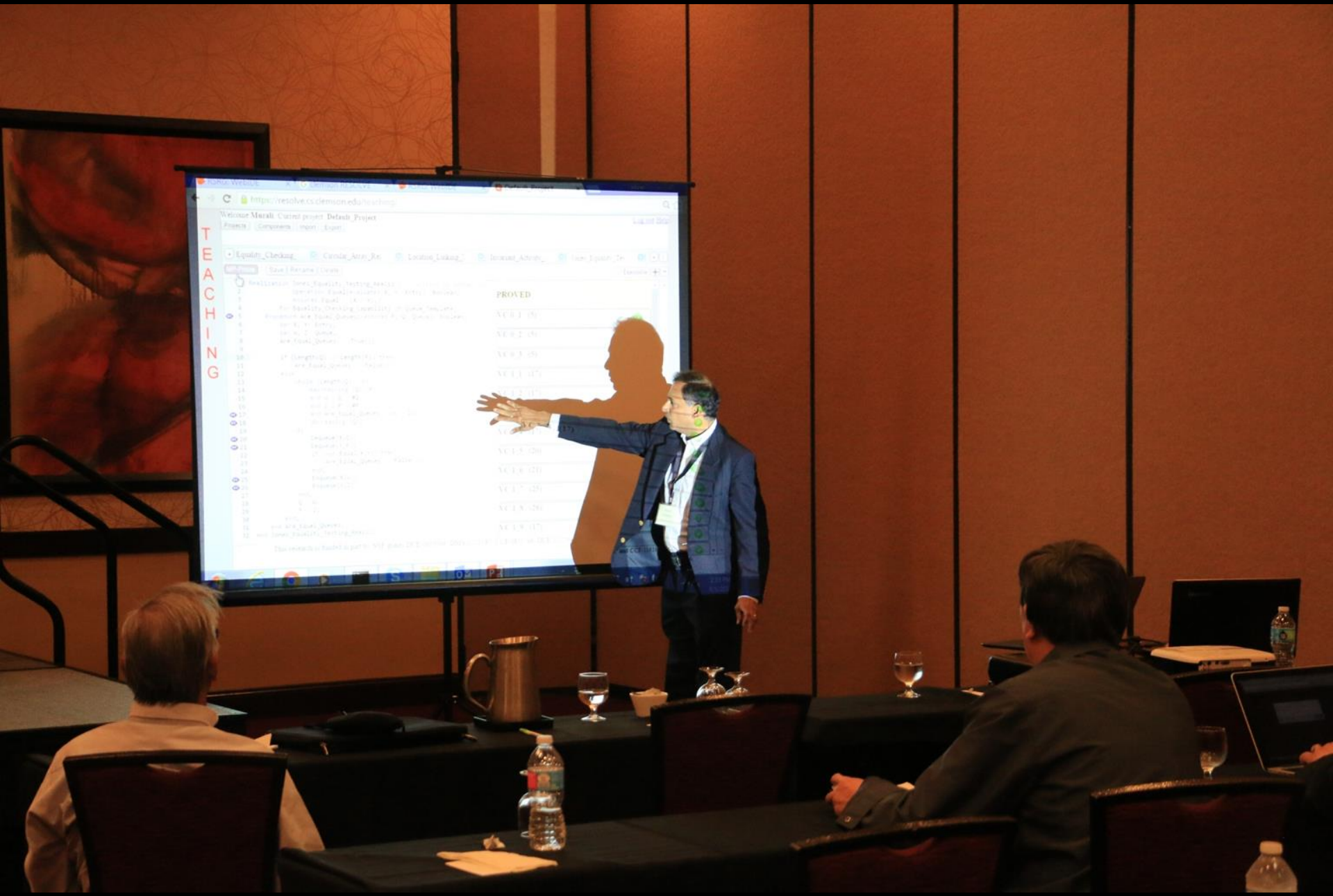

WOD
WORKOUT OF THE DAY


Timeline for the Community College and Industry Apprenticeship Program


Timeline for the Community College and Industry Apprenticeship Program


Girish
Seshagiri

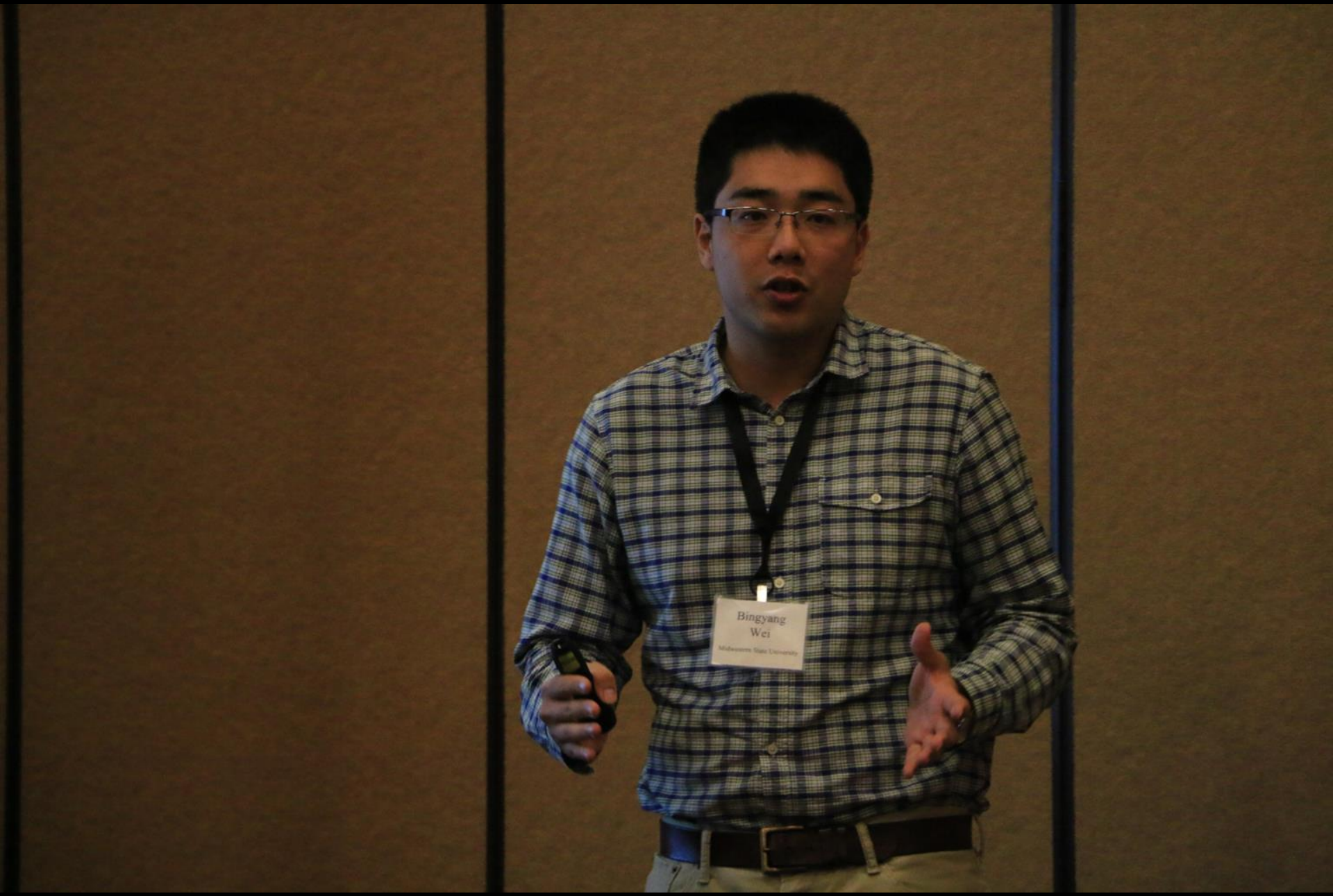

Murali
Sitaraman


TEACHING
PROVED


Tool Assisted Loop Invariant Development and Analysis
Caleb Priester, Yu-Shan Sun, and Murali Sitaraman
RESOLVE Software Research Group (RSRG)
Clemson University
http://www.cs.clemson.edu/resolve
CLEMSON UNIVERSITY
Funded in part by NSF grants CCF-1161916 and DUE-1022941


EXIT
Forming Teams is Hard with Distance Education
5


Early Atte
Research papers
too theoretical
Manage hypothetical teams
too clumsy and inconsistent
Remote teaming
too many students could not synchronize their schedules
most have "real jobs"
UT DALLAS


Bingyang
Wei
Midwestern State University


Students develop incomplete set of UML models for a system

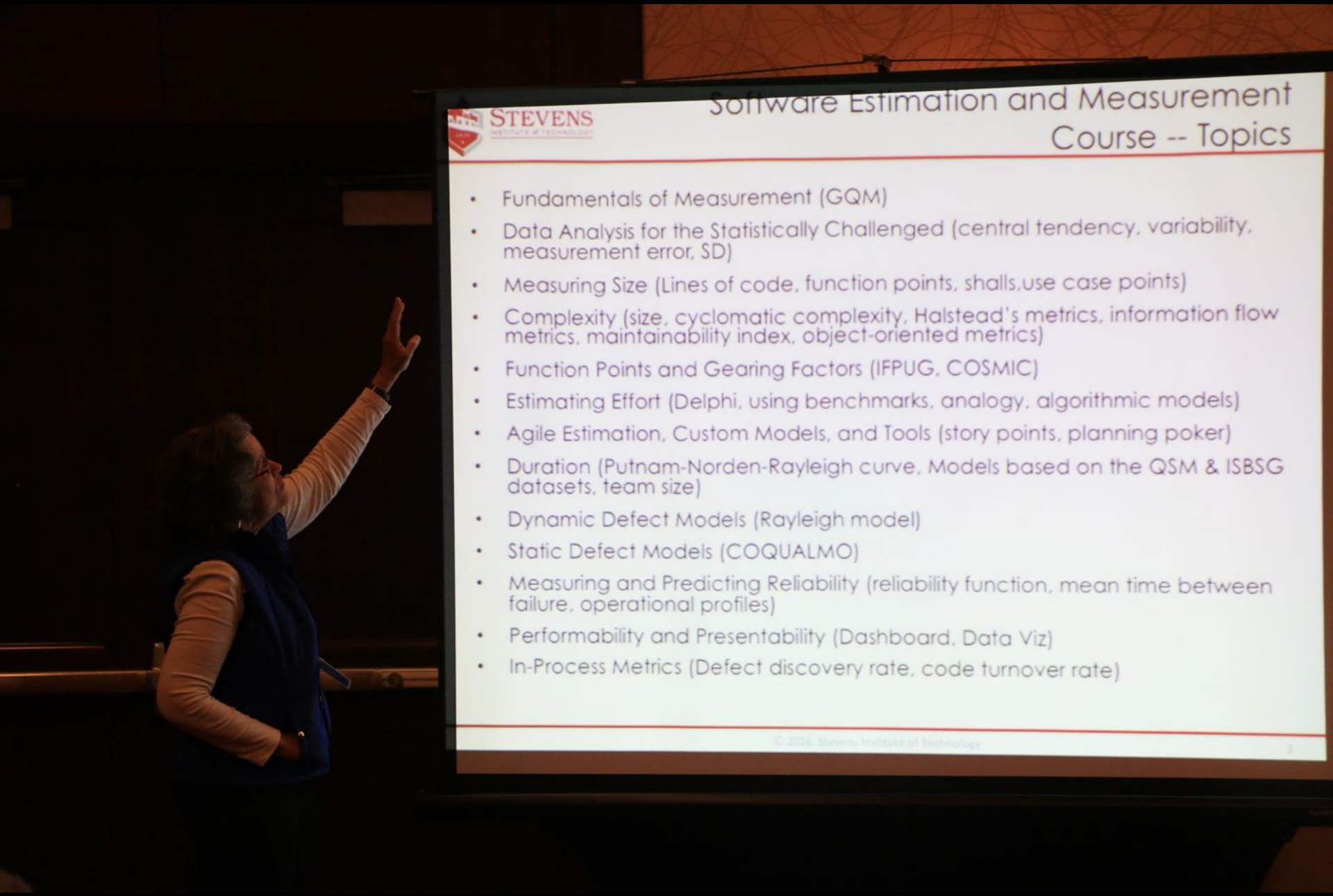

Introduction
Project Management (PM) is an important area of SE education.
It is a critical area for many software organizations
Currently, a significant amount of software projects still fail!
The absence of a PM process is pointed out as one of main causes.
Lack of proper PM process
Leads to
Unaccomplished deadlines
Budget overrun
Incomplete scope coverage
GQS
INCoD
Rafael Queiroz Gonçalves, M.Sc.
Prof. Dr. Christiane Gresse von Wangenheim, PMP


Agile
Values
laboration
ractices
al Practices
Computer
y of
ponds
0 person


Outline- Importance of Object Oriented Programming
• Object Oriented Programming is based on the concept of objects and class.
Advantages:
• Objects in OOP can be reused in various program.
• OOP is easier to program and maintain than non-object oriented programming.
• Many real time applications including mobile application can be developed using OOP.
• Many advanced computer science courses are included in curriculum which requires lot of computation problems to be solved using OOP.
• Studies have shown OOP improves the problem solving and analytical skill of students.


Mapa oficial
do campus da
Universidade Federal
de Santa Catarina
GQS/INC6D/INE/UFSC
www.gqs.ufsc.br
Rafael
Gonçalves

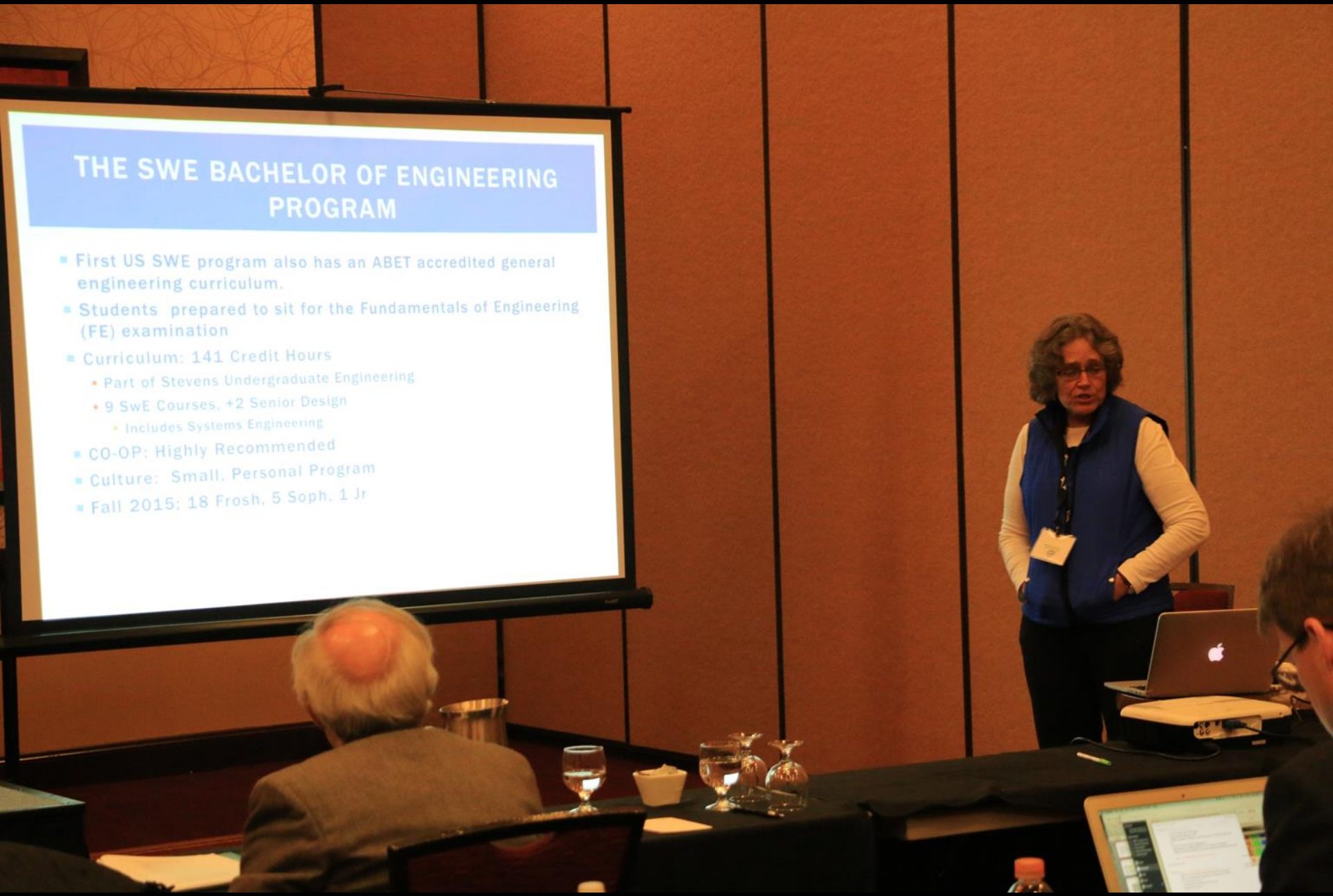

EXIT
EXIT

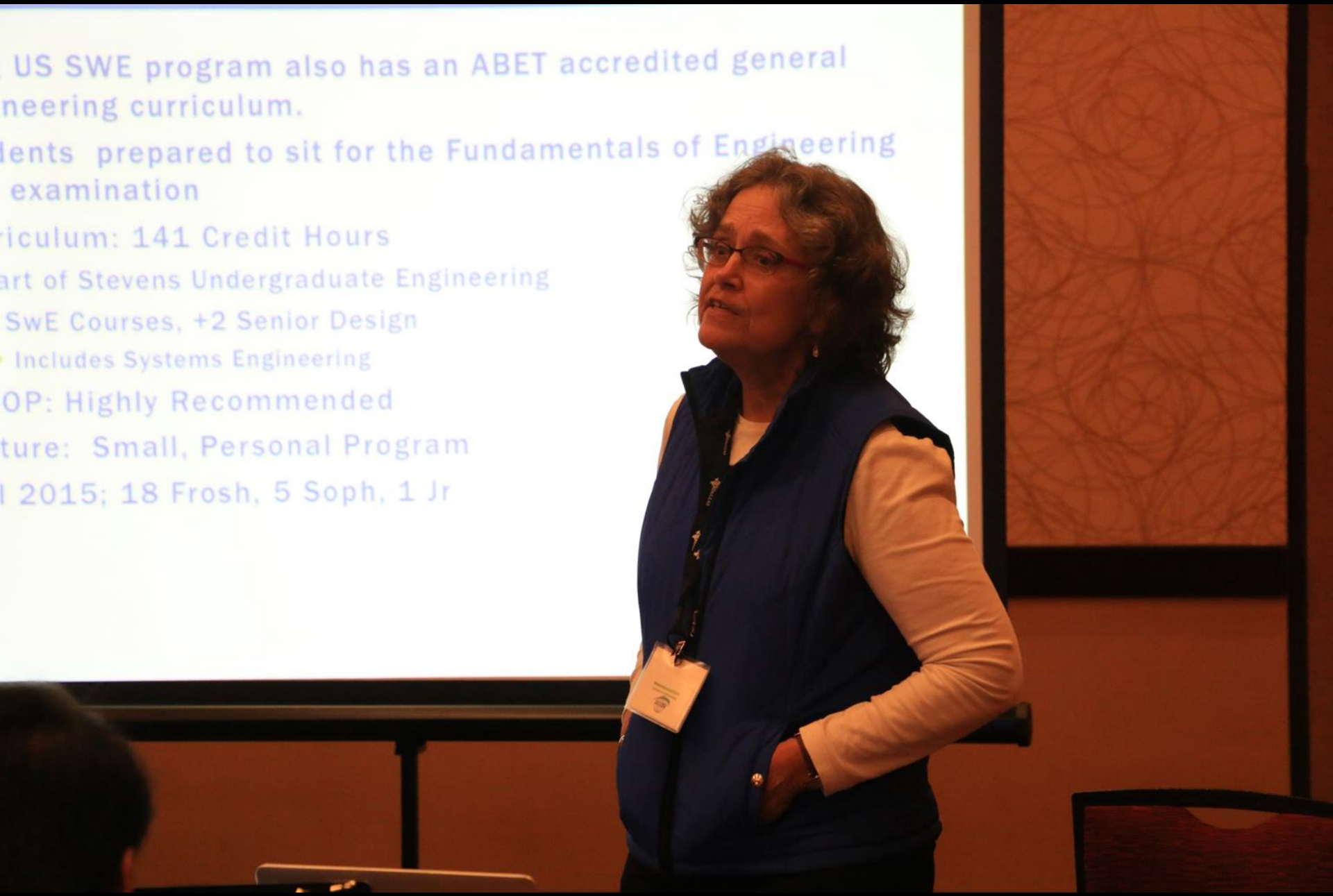

STEVENS
INSTITUTE of TECHNOLOGY
Software Estimation and Measurement
Course -- Topics
• Fundamentals of Measurement (GQM)
• Data Analysis for the Statistically Challenged (central tendency, variability, measurement error, SD)
• Measuring Size (Lines of code, function points, shalls,use case points)
• Complexity (size, cyclomatic complexity, Halstead's metrics, information flow metrics, maintainability index, object-oriented metrics)
• Function Points and Gearing Factors (IFPUG, COSMIC)
• Estimating Effort (Delphi, using benchmarks, analogy, algorithmic models)
• Agile Estimation, Custom Models, and Tools (story points, planning poker)
• Duration (Putnam-Norden-Rayleigh curve, Models based on the QSM & ISBSG datasets, team size)
• Dynamic Defect Models (Rayleigh model)
• Static Defect Models (COQUALMO)
• Measuring and Predicting Reliability (reliability function, mean time between failure, operational profiles)
• Performability and Presentability (Dashboard, Data Viz)
• In-Process Metrics (Defect discovery rate, code turnover rate)
© 2020 Stevens Institute of Technology
3

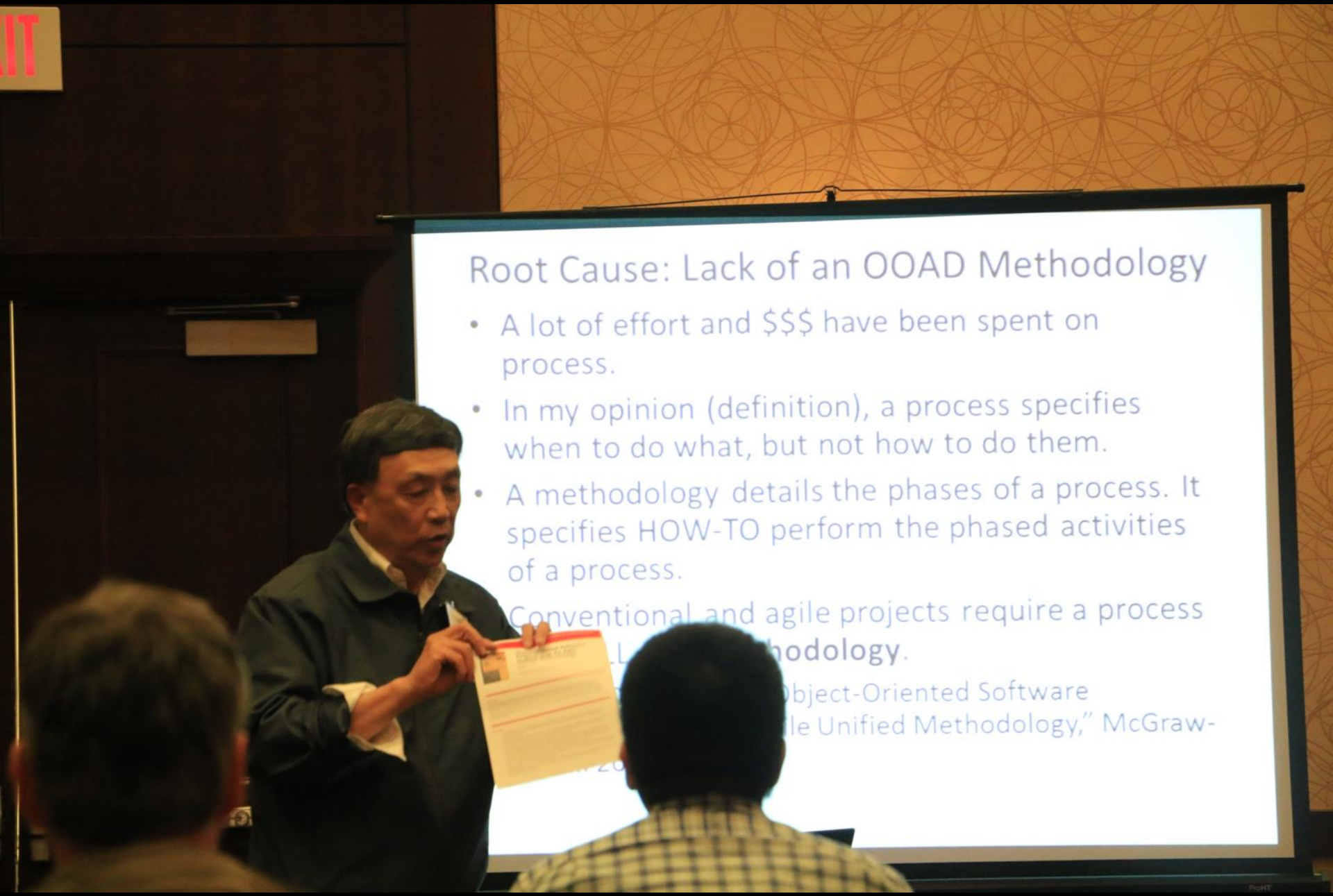

Google Drive is now full screen.
Exit full screen (Esc)
Dynamic Software Updating Techniques in Practice and Educator's Guides: A Review
Ville Ilvonen, Petri Ihantola, Tommi Mikkonen
Tampere University of Technology


Dynamic Software Updating Techniques in Practice and Educator's Guides: A Review
Ville Ilvonen, Petri Ihantola, Tommi Mikkonen
Tampere University of Technology


David
Kung


THE SWE BACHELOR OF ENGINEERING PROGRAM
First US SWE program also has an ABET accredited general engineering curriculum.
Students prepared to sit for the Fundamentals of Engineering (FE) examination
Curriculum: 141 Credit Hours
Part of Stevens Undergraduate Engineering
9 SwE Courses, +2 Senior Design
Includes Systems Engineering
CO-OP: Highly Recommended
Culture: Small, Personal Program
Fall 2015: 18 Frosh, 5 Soph, 1 Jr


US SWE program also has an ABET accredited general
neering curriculum.
lents prepared to sit for the Fundamentals of Engineering
examination
riculum: 141 Credit Hours
art of Stevens Undergraduate Engineering
SwE Courses, +2 Senior Design
Includes Systems Engineering
OP: Highly Recommended
ture: Small, Personal Program
l 2015; 18 Frosh, 5 Soph, 1 Jr


Root Cause: Lack of an OOAD Methodology
• A lot of effort and $$$ have been spent on process.
• In my opinion (definition), a process specifies when to do what, but not how to do them.
• A methodology details the phases of a process. It specifies HOW-TO perform the phased activities of a process.
Conventional and agile projects require a process
hodology.
bject-Oriented Software
le Unified Methodology," McGraw-

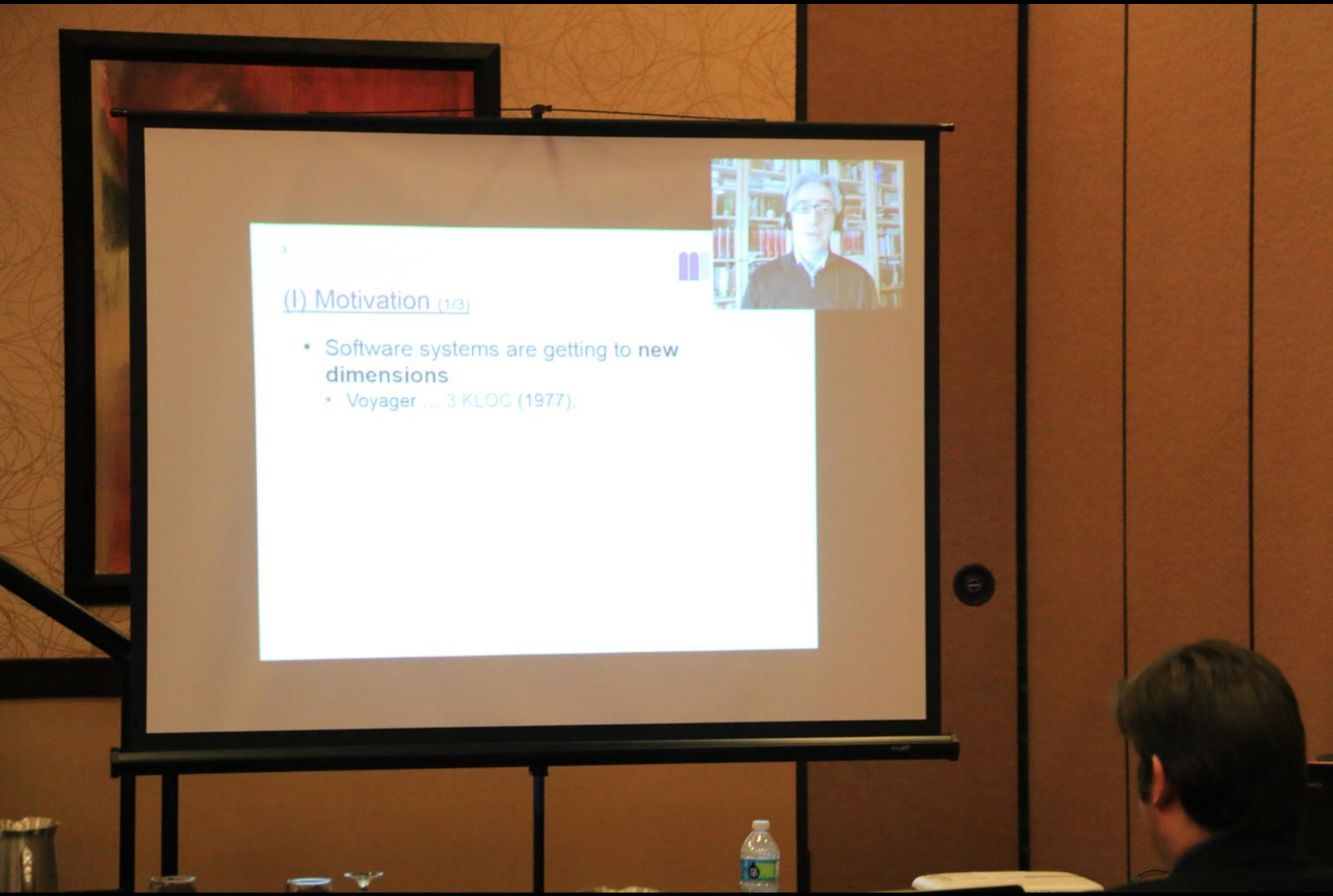

EXIT


Personality plays a important role for team/individual learning effectiveness in controlled PBL
[CSEE&T'13][CSEE&T'14][TETC'16]
Team Charactenstics for Maximizing the Educational Effectiveness of Practical Lectures on Software Intensive Syst
Development, CSEE&T 2013
The Impacts of Personal Characteristic on Educational Effectiveness in Controlled Project Based Learning on Softw
Systems Development, CSEE&T 2014
Relation between Combinations of Personal Ch and Educational Effectiveness for a Controlled Pro
Learning Course, IEEE Trans. Emerging To 2016


EXIT
Carnegie Mellon University
Silicon Valley
Code Readability Testing,
an Empirical Study
Presenter
Todd Sedano


Pivotal
Todd
Sedano
Carnegie Mellon University


Data Collection


(I) Motivation (1/3)
• Software systems are getting to new dimensions
• Voyager … 3 KLOC (1977).


EXIT
University of Applied Sciences and Arts
Northwestern Switzerland
Teaching Agile
Collaboration Skills in t
Classroom
Martin Kropp, University of Applied Sciences
Northwestern Switzerland
Andreas Meier, Zurich University of Applied Scien
Switzerland
Robert Biddle, Carleton University
Canada


Robert
Biddle
Carleton University