

Panel discussion on

TOP CHALLENGING ISSUES

FOR SOFTWARE DEVELOPMENT

Roy Oberhauser
Aalen University

Importance of SW Development

- Society increasingly depends on software
 - SW impacts us all, thus SW development issues too
 - Value has shifted to software and data
- Impacts of development issues
 - Direct costs of defects and catastrophes
 - Indirect costs in lack of satisfaction and capabilities
- New trends/situations bring their own challenges:
 - GSD, Crowdsourcing, Cloud Computing, DevOps, etc.
 - Security: malware, hacking, cyber-espionage and -warfare
 - Reuse via composability and resulting issues

Yet in some cases these may be running into some common underlying challenges too...

Challenging Issues for SW Development

- Technical Issues
 - Brooks' Essential Difficulties [Br86]
 - Complexity, Conformity, Changeability, Invisibility
 - Requirements refinement [Br86]
 - "Deciding precisely what to build is hardest part of the conceptual work: establishing the detailed technical requirements, including all the interfaces to people..."
 - UI design [Bro3]
 - An art, not systematic or an engineering discipline
 - Productivity via reuse: compositionality (avoid building)

Challenging Issues for SW Development

- Technical Issues
- Process Issues
 - Agile vs. plan-driven approaches
 - Agile Manifesto with explicit emphasis on:
 - Explicit values, ownership, trust, working SW
 - CHAOS Report 2012 on Waterfall vs. Agile [CH12]:
 - Success 14 vs. 42%; Failed 29 vs. 9%;
 - But half are still Challenged in either case (57 vs. 49%)
 - Technical debt - long-term?
 - Team self-organization and role ambiguity
 - Maintaining conceptual integrity across multiple minds?
 - Forking “binges”; lack of documentation

Challenging Issues for SW Development

- Technical Issues
- Process Issues
- People-related Issues
 - Human issues: fallibility, communication issues, social skills, psychological effects
 - Adequate competency, training, and education
 - Dynamic technological landscape
 - Handling & valuing developers: Peopleware [DeLi87]

Challenging Issues for SW Development

- Technical Issues
- Process Issues
- People Issues
- Project Issues
 - SW estimation [Bro3]
 - Predictable schedule and predictable amount of work

Challenging Issues for SW Development

- Technical Issues
- Process Issues
- People Issues
- Project Issues
- Quality and other Holistic Issues
 - Less tolerance for errors?
Beyond single-point-of-failures, chain events

In Summary

Software development is really about:

Multiple fallible humans
collaborating via some lossful natural language
to precisely program essentially invisible systems
based on unclear and imperfect specifications,
thereby
creating highly complex defect-prone systems
without definitive schedule or work predictability,
on which society (gratefully and wishfully) relies...

References

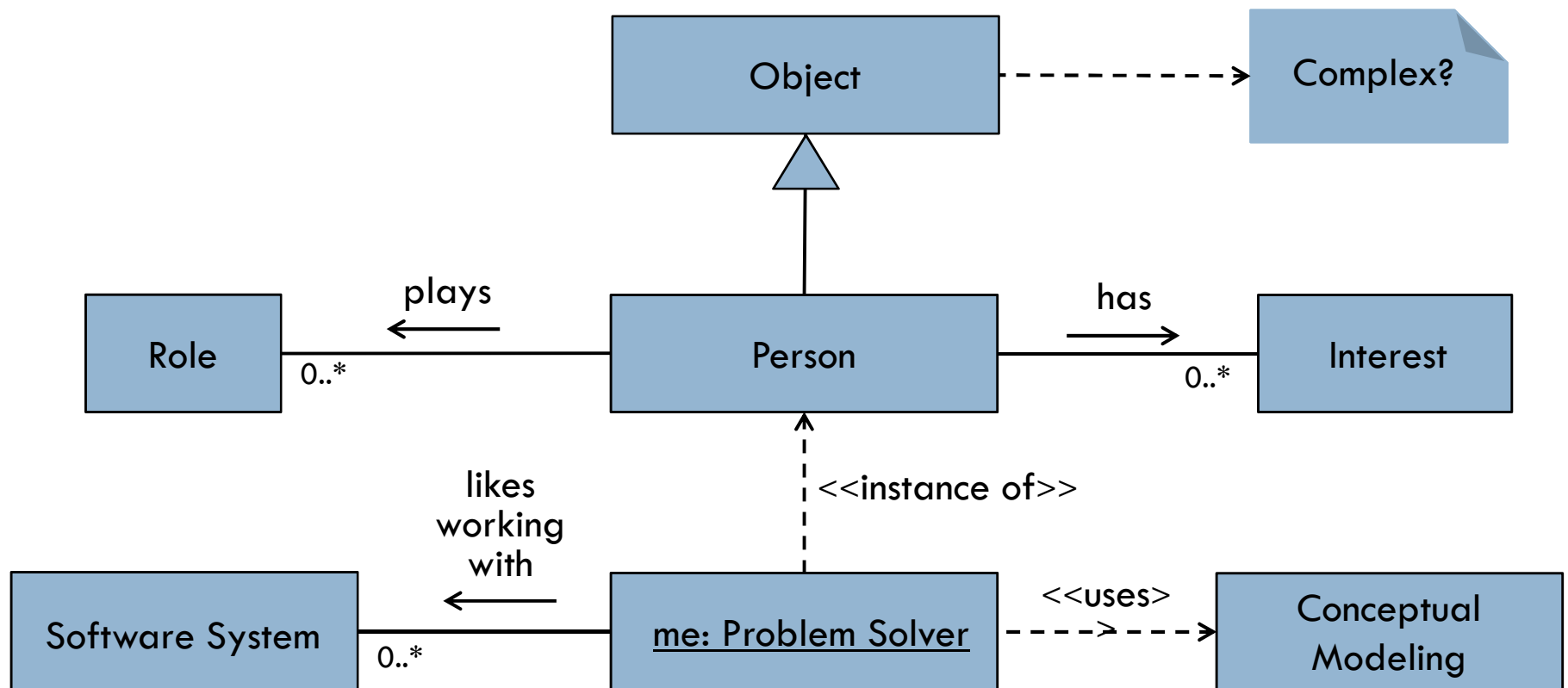
[Br86] Frederick P. Brooks, Jr. "No Silver Bullet — Essence and Accident in Software Engineering". Proceedings of the IFIP Tenth World Computing Conference: 1069–1076, 1986.

[DeLi87] Tom DeMarco and Timothy Lister. Peopleware: Productive Projects and Teams. ISBN 0-932633-43-9. 1987.

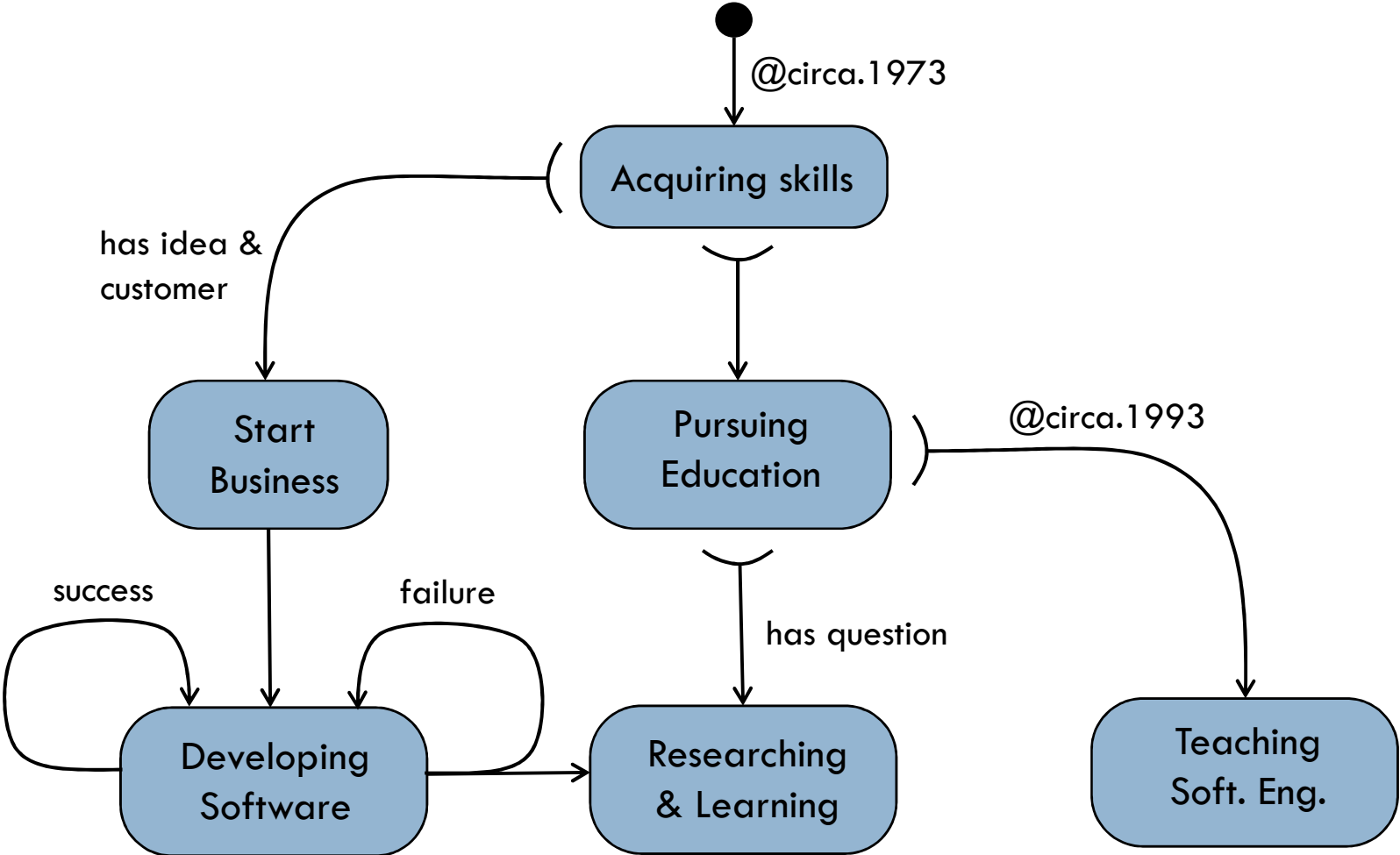
[Bro3] Frederick P. Brooks, Jr. "Three great challenges for half-century-old computer science." J. ACM 50, 1 (January 2003), 25-26.

[CH12] The CHAOS Report by the Standish Group 2012.

Stephen Clyde? Who's he?



Transitions and States?



Responsibilities and Collaborations?

In the 1980's

Problem Solver

CRC-Card-like model
[Cunningham, Beck, Wirfs-Brock, etc.]

Average Software Engineer

Responsible for ...

- *Programming (coding) core application logic*
- *Detailed design*
- *Some levels and types of design*
- *Some testing*
- *Some integration*
- *A little analysis of application problems and domains*
- *and more*

Collaborates with ...

- *Managers*
- *Customers*
- *End users*
- *Testers*
- *Graphic artists*
- *Documenters*
- *and more*

Responsibilities and Collaborations?

Today

Problem Solver

Average Software Engineer

Responsible for ...

- *Programming the “glue” between frameworks, SDK’s, and existing components*
- *All kinds of integration*
- *All kinds of testing*
- *All levels and kinds of design*
- *Analysis of application problems and domains*
- *Most of the development process*
- *and more*

Collaborates with ...

- *More Managers*
- *Sophisticated Customers & End Users*
- *Quality Assurance*
- *Security Officers*
- *Graphic artists*
- *Documenters*
- *and more*

What's change and what hasn't?



- **Essence** → more complexity
- **Tools** → lots of improvements, still focus primarily on reducing accidents
- **Accidental complexity** → different, but more frequent and subtle
- **Principles** → some advancements and more evidence, but still lacking maturity
- **The need for great designers** → the need for great analysts, designers, integrators, testers, etc.



Università degli Studi dell'Insubria

Top Challenging Issues for Software Development
Governance issues

Luigi Lavazza

Dipartimento di Scienze Teoriche e Applicate

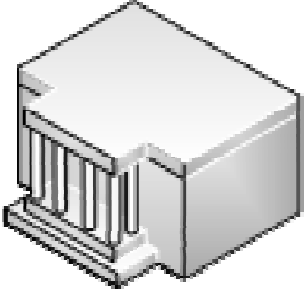
Università degli Studi dell'Insubria

Varese, Italy



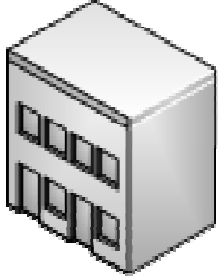
The software lifecycle “in the large”

A P.A. or company
Including business
people and business
analysts



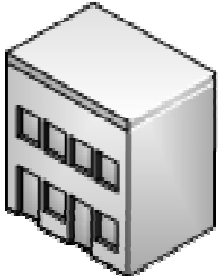
Here needs, money,
priorities are known

A company acting as
software procurer
and integrator



Here needs are
mapped onto software
requirements

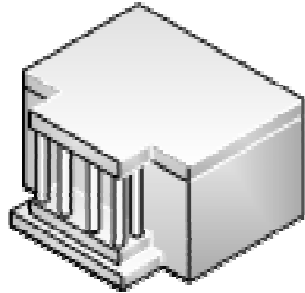
A company acting as
software developer



Here bids are
specified, on the basis
of software
requirements

Here actual software is
developed

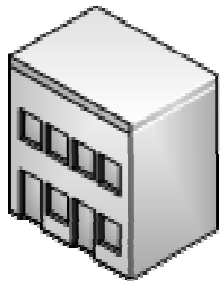
The goals at the different levels



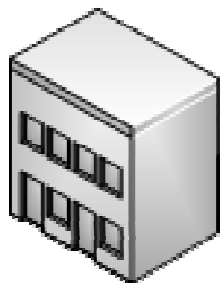
Satisfying business needs
Managing the budget and
the software portfolio,



Translating business needs into
software requirements
Roughly estimating costs to assess
requirements viability and supply costs



Translating software
requirements into bid-oriented
specifications (including
functionality, quality, cost, etc.)



Software development (of
the required quality, within
budget and in time)



The problems

- Communication “impedance mismatch”
 - How can analysts understand business people (or PA administrators)? and vice versa ...
 - How can analysts and technical people communicate with no misunderstandings?
- Dealing with invisibility
 - What are the actual needs?
 - What are the requirements?
 - What are the specifications of the software solution?
- What is the cost and value
 - Of satisfying needs?
 - Of implementing requirements?
 - Of developing software?

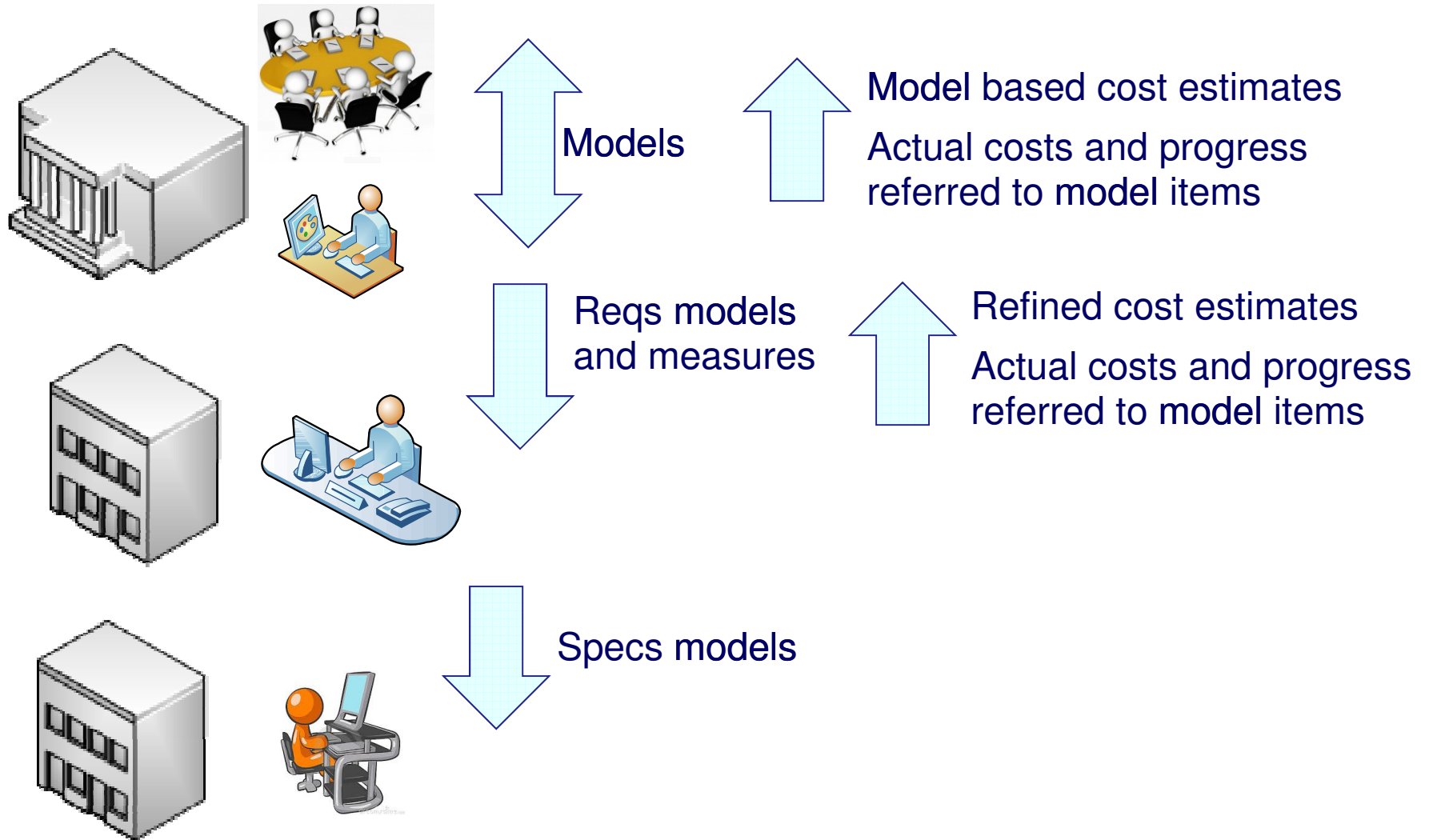


A step towards a solution

- A common language, to express
 - business processes
 - Software requirements
 - Software specs, architecture, design
- The common language should support measurement. Objective quantitative measures support
 - Cost estimation models
 - Value models
 - Evaluation of progress



Virtuous communication flows



Software Development Challenges Panel

8th International Conference on Software Engineering Advances (ICSEA 2013)

Venice, October 28th, 2013

Global Software Development (GSD) Challenges

Jameleddine Hassine

Department of Information and Computer Science
KFUPM, Kingdom of Saudi Arabia
jhassine@kfupm.edu.sa



What is Global Software Development (GSD)

- Software work undertaken at geographically separated locations across national boundaries in a coordinated fashion involving real time (synchronous) and asynchronous interaction.
 - **Communication** for information exchange.
 - **Coordination** of teams and activities (adhering to goals and policies)

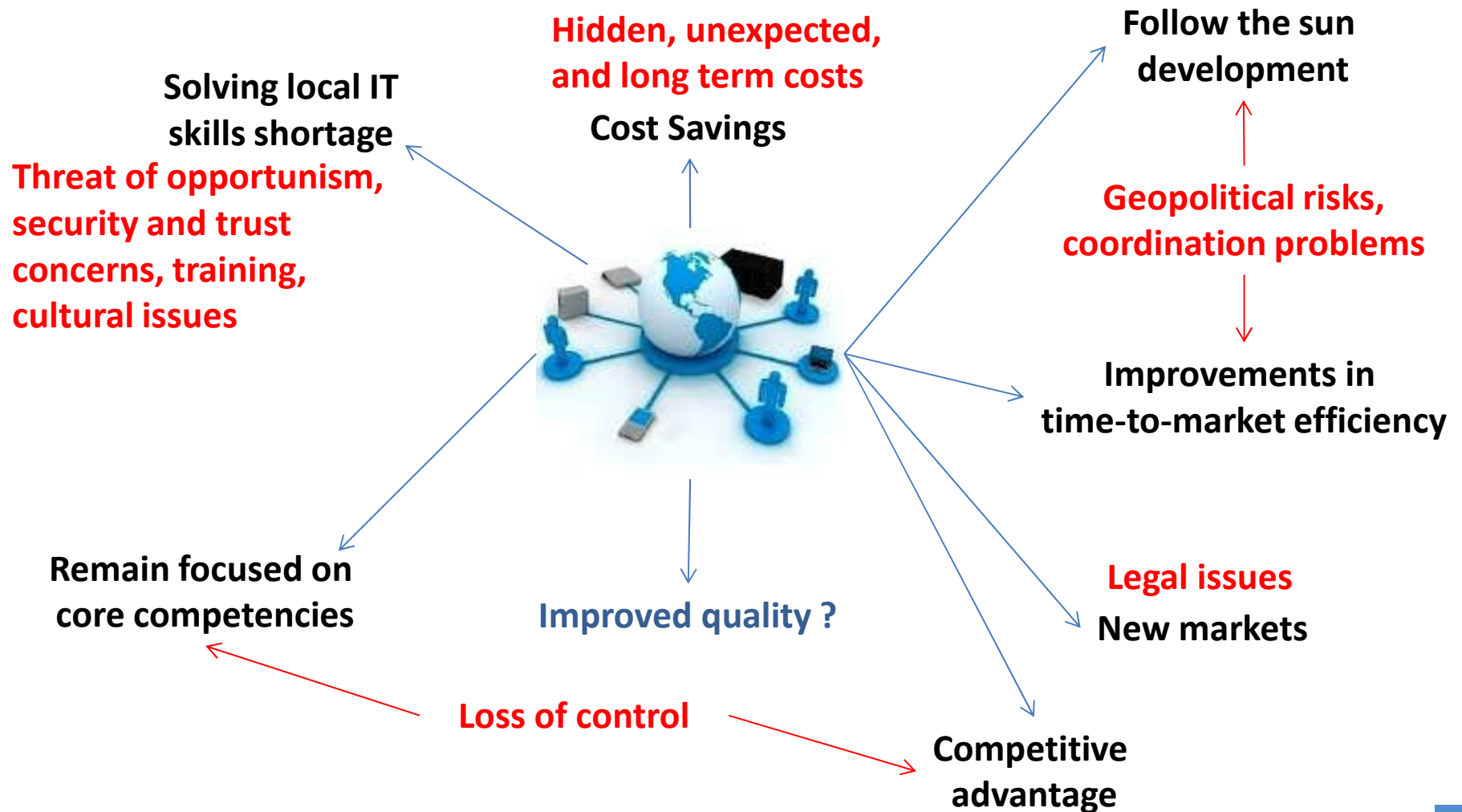
Types of GSD Organizations

- Organizations shift all or part of their software development to other countries (referred to as off shoring). Independent client companies who outsource their software development to a vendor or software supplier.
- Multinational organizations distribute their software development activities across multiple subsidiary sites, located in different countries, e.g., Cisco, IBM, Alcatel-Lucent, Siemens, etc.

Extent of Global Software Development

- More than 90% of Fortune 500 companies use external resources for IT services delivery [2010]
- 31% of IT spending by companies in 2010 was on external services
- Cisco, IBM, Alcatel-Lucent, British Telecom, and General Electric have moved parts of their software development to countries like Ireland and India

Why Global Software Development?



Issues and Challenges

- **Strategic issues:** when, to whom and how, task allocation.
- **Communication issues:** distance, time zone difference, infrastructure support, distinct backgrounds, lack of informal communication.
- **Requirements dynamism:** user requirements are dynamic
- **Cultural issues:** individualism vs. collectivism, emotional vs. neutral, attitude to time, attitude to governments, etc.
- **Geographical dispersion:** coordination complexity, vendor support, access to experts, software practices that need face-to-face communications.
- **Technical issues:** information and artifact sharing, software architecture.
- **Knowledge management:** slow communication, poor documentation, management of repositories, etc.

Addressing GSD issues

- Global software development process capabilities:
 - **Rigor and Standardization:** may offset some of the negative effects of team dispersions
 - **Agility and Customizability:** may help teams cope with user requirements dynamism. Such processes can lower cost and shorten time in responding to user change requests.
- Work allocations:
 - Transfer by development stage, transfer by functionalities, product line approach
- Organizational models: local managers at each site (with common visions).
- Cultures must be understood and respected. They could not be easily changed!
- Other solutions:
 - Share your experience...

Thank You

How to handle Human Diversity at the Web?



Professor Adriana E. MARTIN

PhD in Informatics Science - Master in Software Engineering

GIFIS Research Group, National University of Austral Patagonia (UNPA), Santa Cruz, Southern Patagonia, Argentina

GIISCo Research Group, National University of Comahue (UNComa), Neuquén, Northern Patagonia, Argentina

amartin@uaco.unpa.edu.ar / adrianaelba.martin@gmail.com





*You can't just punch in
"Let there be Light"
without writing the code underlying
the user interface functions!*

J. P. Rini (1997)

*So, what's the problem? Web users are just like humans...
and there are many different areas concerned with what
we called:*

"The Human Side of the Web"



Web Accessibility promotes
"A Web for All"

by removing barriers based on standards and guidelines that help Web sites or products meet basic requirements...

It seems that we already have good proposals, so again: what's the problem?



Human-Computer Interaction (HCI) involves the study, planning and design of the *interaction between people (users) and computers...*

User-Centered Design (UCD) works on the strategy of

positioning users' needs at the center of a Web design...
(also known as Usability Engineering)



*"The Web is more of a **Social** creation than a **Technical** one..."*

Tim Berners-Lee: The Web's Brainchild by UNESCO's Courier (2000)



Web Usability promotes

"A Web site or product easy to Use"
by targeting to users who use that Web site or product to study the user's attitude towards it...

Web Accessibility

"A Web for All"

puts the focus on technical aspects

possible?

what about the human interaction aspect?

So, let's take a second look to some of these areas we mentioned before, and their proposals...



... more than 7 billion people around the world
As predicted by the National Geographic Magazine - Special Series: 7 Billion (2011)



content and frequency of tests?

representative users?

what about considering people who have disabilities?

Web Usability

"A Web site or product easy to Use"

puts early focus on identifying users and tasks

Web Accessibility

"A Web for All"

puts the focus on technical aspects



How we cope with these issues within the boundaries of these areas?

"An Usable Accessibility"



Web Usability

"A Web site or product easy to Use"

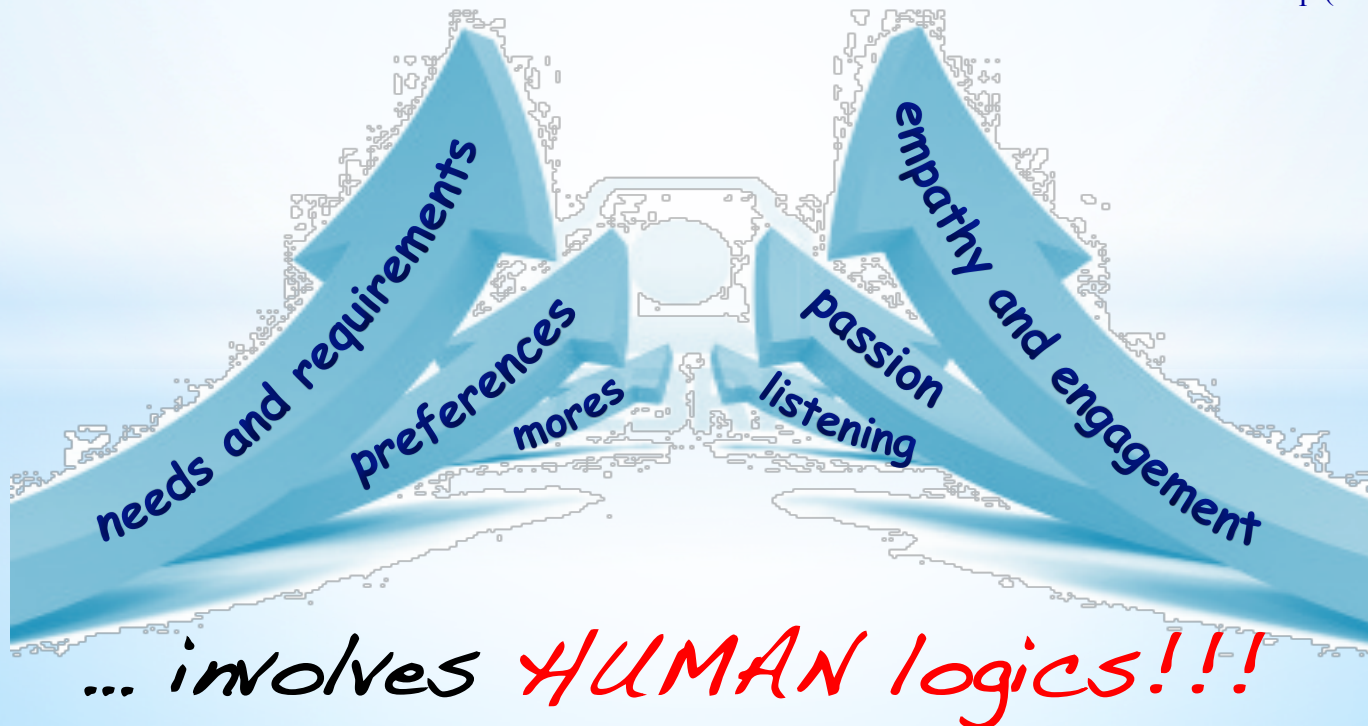
puts early focus on identifying users and tasks

... researchers and practitioners have to leave their sandboxes and work their fields with a broader view!

For sure this is a very fast decade and it's a fact that we need technology to survive, but mostly it's a SOCIAL DECADE where

"It's really about the Web User experience"

Mike Paciello of The Paciello Group (2009)



... there is no doubt that we need to improve recipes, ergo, effective and practical solutions to better handle the human nature of our Web users.



Thanks for your attention!

Professor Adriana E. MARTIN

amartin@uaco.unpa.edu.ar // adrianaelba.martin@gmail.com