Welcome to SENG 371
Software Evolution
Spring 2013

A Core Course of the BSEng Program

Hausi A. Müller, PhD PEng
Professor, Department of Computer Science
Associate Dean Research, Faculty of Engineering
University of Victoria

Learning objectives

- Define and introduce the topics of software evolution and maintenance
- Discuss how these concepts fit within the wider context of software engineering
- Motivate why maintenance and evolution are important topics to consider
- Give a flavour of the theoretical background and key skills required to implement effective change

Why study this topic?

- Increasing reliance on systems, everywhere, everything, every minute...
- Critical systems—safety, life and death, financial
- Cost of change estimated at 40-70% of total life-cycle costs
  - Fred Brooks, in his seminal book *The Mythical Man-Month*, states that over 90% of the costs of a typical system arise in the maintenance phase, and that any successful piece of software will inevitably be maintained.
- Software maintenance experts and professionals are in high demand
- Few jobs are in green field development, even such jobs require extensive reuse or integration of other components

Review of last lecture

- Lot of development experience in this class
- People depend on software
- Some basic definitions
- Maintenance versus evolution
- Disappearing boundary between development time and runtime
- Out front: Models at runtime
- Back to basics: Waterfall model
- Level of indirection
- Class participation
- Opportunity to hone your communications skills

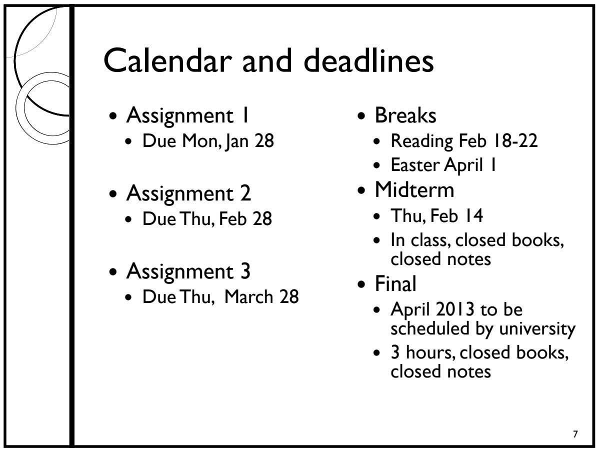
Course web sites

- Course outline
  - http://courses.seng.uvic.ca/courses/2013/spring/seng/371
- UVic Calendar Course Description
  - http://web.uvic.ca/calendar2012/CDs/SENG/371.html
- Course website
  - http://www.engr.uvic.ca/~seng371
  - Syllabus
  - Lecture slides (pdf)
  - Lab slides (pdf)
  - Assignments
  - Materials for reading assignments
  - Everything else you need to know about the course

Optional Textbooks

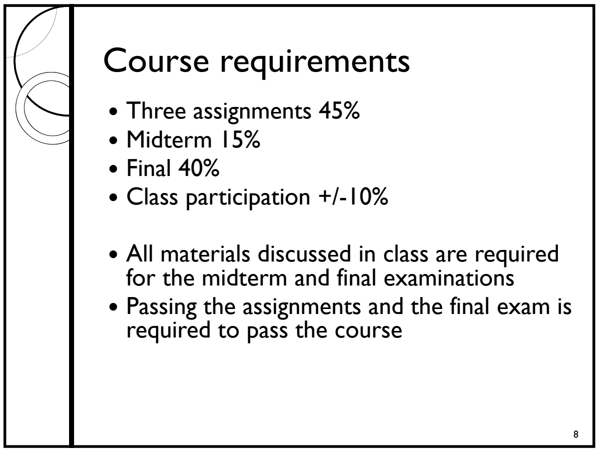
- Grubb and Takang: *Software Maintenance,* 2nd Edition, World Scientific, 2003 — ISBN: 978-981-238-426-3
- Mens and Demeyer: *Software Evolution,* Springer, 2008 — ISBN: 978-3-540-76439-7 (Print) 978-3-540-76440-3 (Online)

- There will be additional readings assigned during the term.



Calendar and deadlines

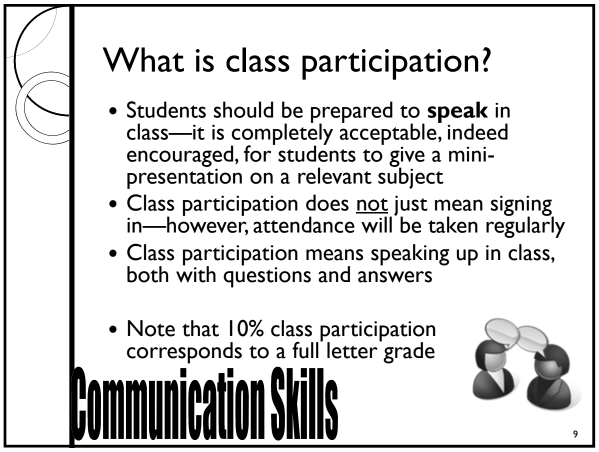
- Assignment 1
  - Due Mon, Jan 28
- Assignment 2
  - Due Thu, Feb 28
- Assignment 3
  - Due Thu, March 28
- Breaks
  - Reading Feb 18-22
  - Easter April 1
- Midterm
  - Thu, Feb 14
  - In class, closed books, closed notes
- Final
  - April 2013 to be scheduled by university
  - 3 hours, closed books, closed notes



Course requirements

- Three assignments 45%
- Midterm 15%
- Final 40%
- Class participation +/-10%

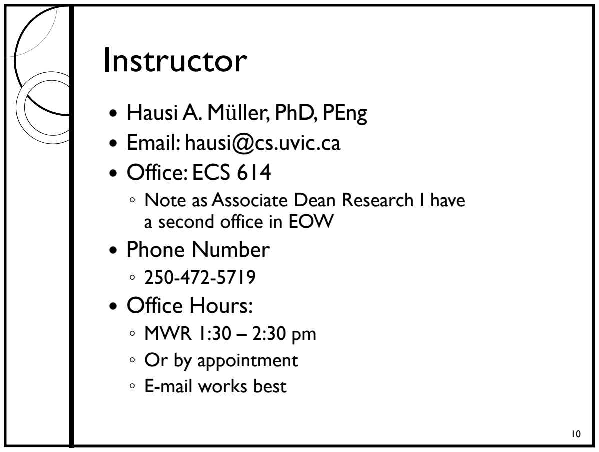
- All materials discussed in class are required for the midterm and final examinations
- Passing the assignments and the final exam is required to pass the course



What is class participation?

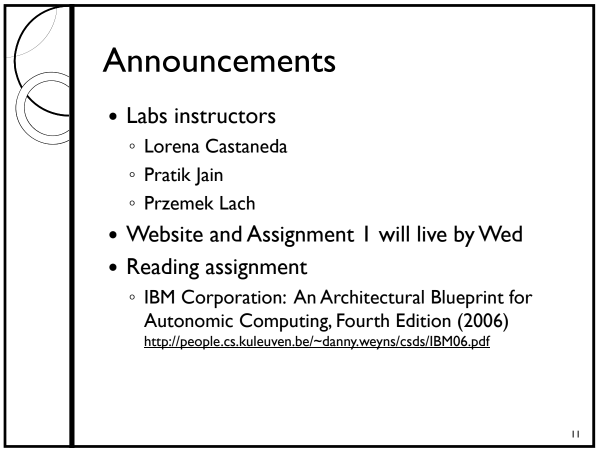
- Students should be prepared to **speak** in class—it is completely acceptable, indeed encouraged, for students to give a mini-presentation on a relevant subject
- Class participation does not just mean signing in—however, attendance will be taken regularly
- Class participation means speaking up in class, both with questions and answers

- Note that 10% class participation corresponds to a full letter grade



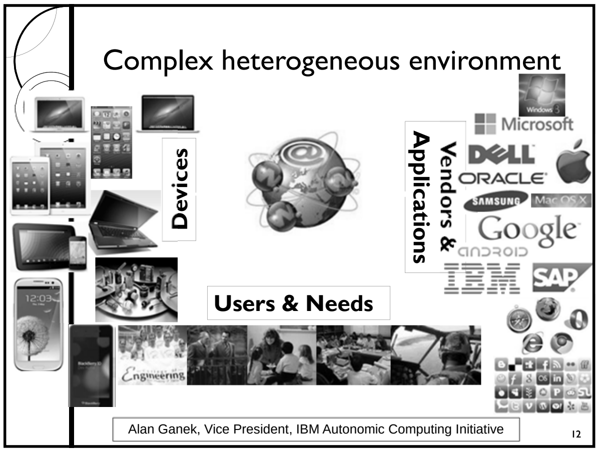
Instructor

- Hausi A. Müller, PhD, PEng
- Email: hausi@cs.uvic.ca
- Office: ECS 614
  - Note as Associate Dean Research I have a second office in EOW
- Phone Number
  - 250-472-5719
- Office Hours:
  - MWR 1:30 – 2:30 pm
  - Or by appointment
  - E-mail works best



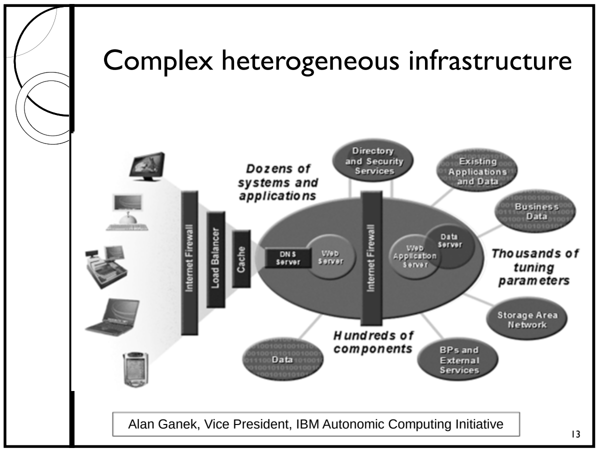
Announcements

- Labs instructors
  - Lorena Castaneda
  - Pratik Jain
  - Przemek Lach
- Website and Assignment 1 will live by Wed
- Reading assignment
  - IBM Corporation: An Architectural Blueprint for Autonomic Computing, Fourth Edition (2006)
    http://people.cs.kuleuven.be/~danny.weyns/csds/IBM06.pdf



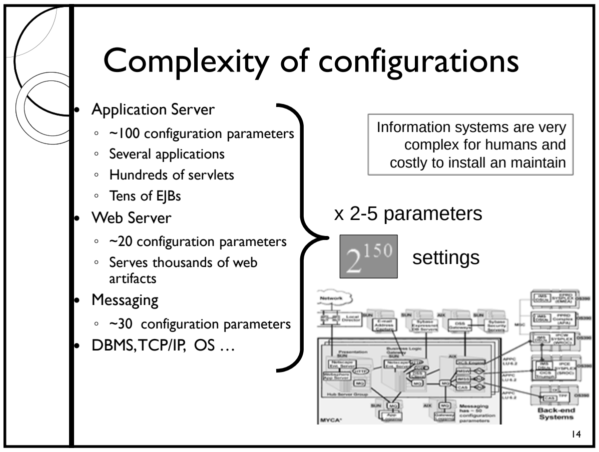
Complex heterogeneous environment

Alan Ganek, Vice President, IBM Autonomic Computing Initiative



Complex heterogeneous infrastructure

Alan Ganek, Vice President, IBM Autonomic Computing Initiative



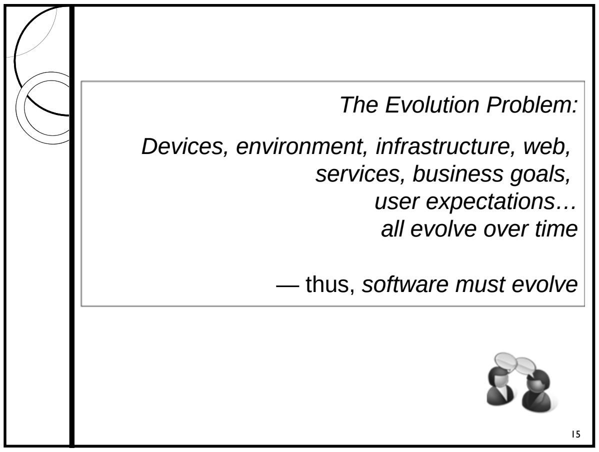
Complexity of configurations

- Application Server
  - ~100 configuration parameters
  - Several applications
  - Hundreds of servlets
  - Tens of EJBs
- Web Server
  - ~20 configuration parameters
  - Serves thousands of web artifacts
- Messaging
  - ~30 configuration parameters
- DBMS,TCP/IP, OS …

x 2-5 parameters

2^{150} settings

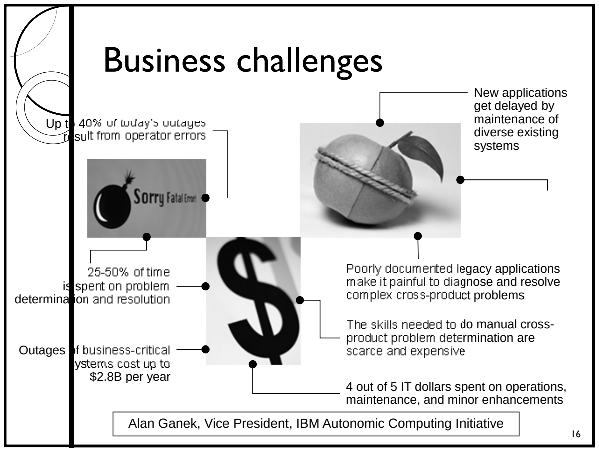
Information systems are very complex for humans and costly to install an maintain



The Evolution Problem:

Devices, environment, infrastructure, web, services, business goals, user expectations… all evolve over time

— thus, *software must evolve*



Business challenges

Sorry Fatal Error!

Up to 40% of today's outages result from operator errors

New applications get delayed by maintenance of diverse existing systems

25-50% of time is spent on problem determination and resolution

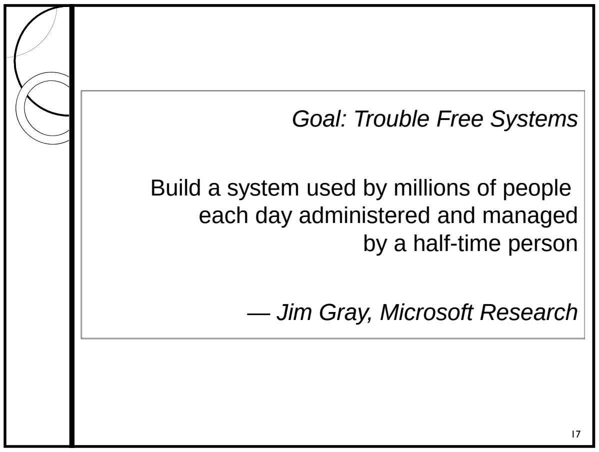
Poorly documented legacy applications make it painful to diagnose and resolve complex cross-product problems

The skills needed to do manual cross-product problem determination are scarce and expensive

Outages of business-critical systems cost up to $2.8B per year

4 out of 5 IT dollars spent on operations, maintenance, and minor enhancements

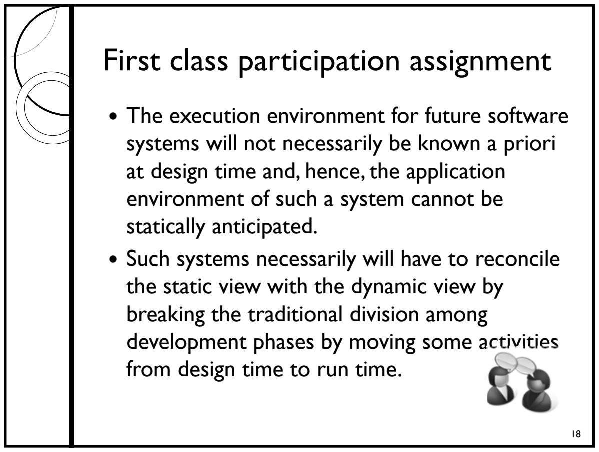
Alan Ganek, Vice President, IBM Autonomic Computing Initiative



Goal: Trouble Free Systems

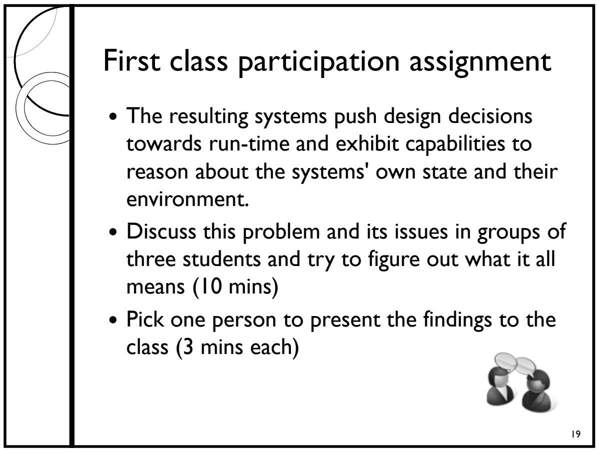
Build a system used by millions of people each day administered and managed by a half-time person

— Jim Gray, Microsoft Research



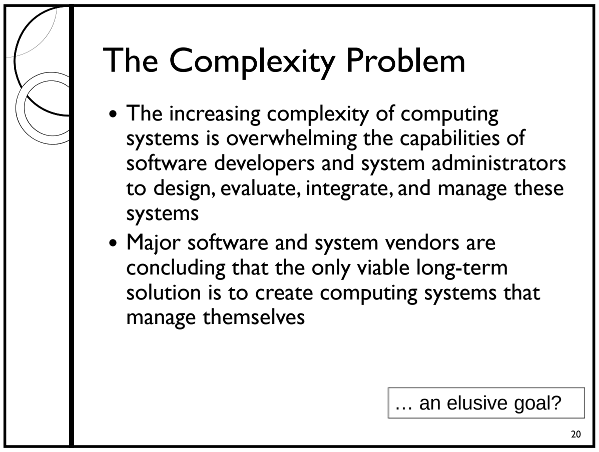
First class participation assignment

- The execution environment for future software systems will not necessarily be known a priori at design time and, hence, the application environment of such a system cannot be statically anticipated.
- Such systems necessarily will have to reconcile the static view with the dynamic view by breaking the traditional division among development phases by moving some activities from design time to run time.



First class participation assignment

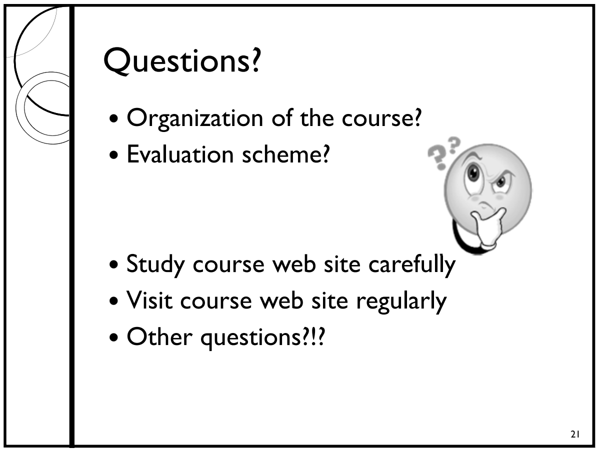
- The resulting systems push design decisions towards run-time and exhibit capabilities to reason about the systems' own state and their environment.
- Discuss this problem and its issues in groups of three students and try to figure out what it all means (10 mins)
- Pick one person to present the findings to the class (3 mins each)



The Complexity Problem

- The increasing complexity of computing systems is overwhelming the capabilities of software developers and system administrators to design, evaluate, integrate, and manage these systems
- Major software and system vendors are concluding that the only viable long-term solution is to create computing systems that manage themselves

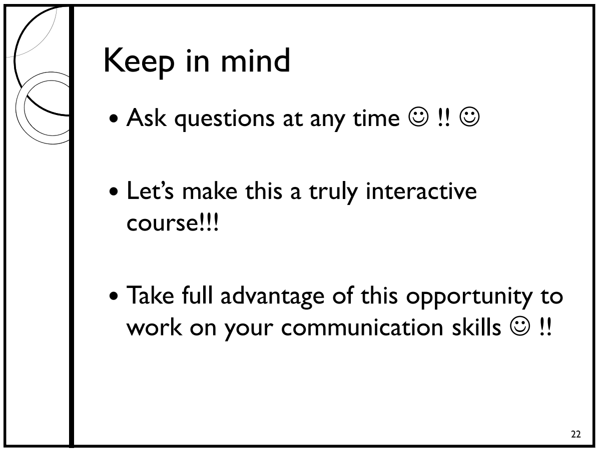
… an elusive goal?



Questions?

- Organization of the course?
- Evaluation scheme?

- Study course web site carefully
- Visit course web site regularly
- Other questions?!?



Keep in mind

- Ask questions at any time ☺ !! ☺
- Let's make this a truly interactive course!!!
- Take full advantage of this opportunity to work on your communication skills ☺ !!