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

Announcements

- Course website
  - Up and running
  - http://www.engr.uvic.ca/~seng371
- Assignment 1
  - Posted; due Jan 28
- Reading assignments
  - IBM Corporation: An Architectural Blueprint for Autonomic Computing, Fourth Edition (2006) http://people.cs.kuleuven.be/~danny.weyns/csds/IBM06.pdf
- Labs instructors
  - Lorena Castaneda
  - Pratik Jain
  - Przemek Lach

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

Calendar and deadlines

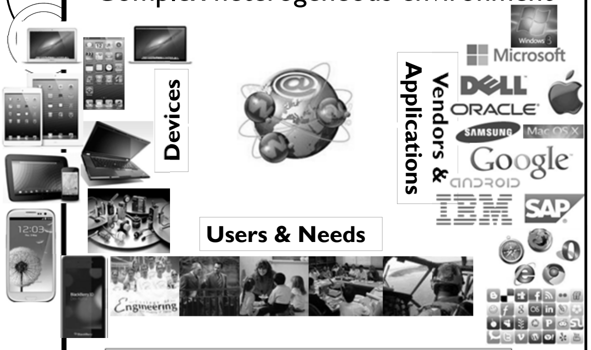
- Assignment 1
  - Due Mon, Jan 28
- Assignment 2
  - Due Thu, Feb 28
- Assignment 3
  - Due Thu, March 28
- Breaks
  - Reading Feb 18-22
  - Easter April 1
- Midterm
  - Thu, Feb 14
  - In class, closed books, closed notes
- Final
  - April 2013 to be scheduled by university
  - 3 hours, closed books,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

Course requirements

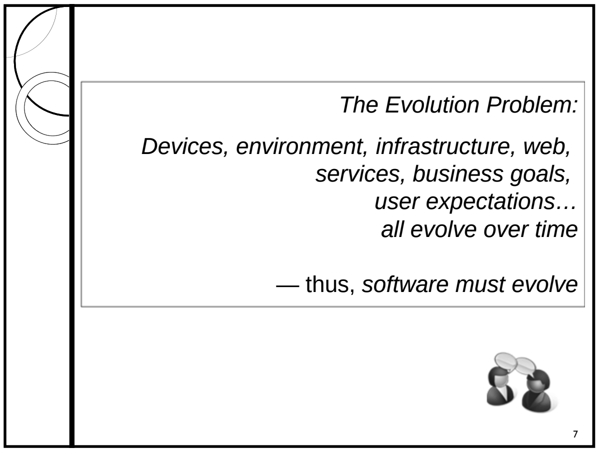
- Three assignments 45%
- Midterm 15%
- Final 40%
- Class participation +/-10%

- 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

Complex heterogeneous environment



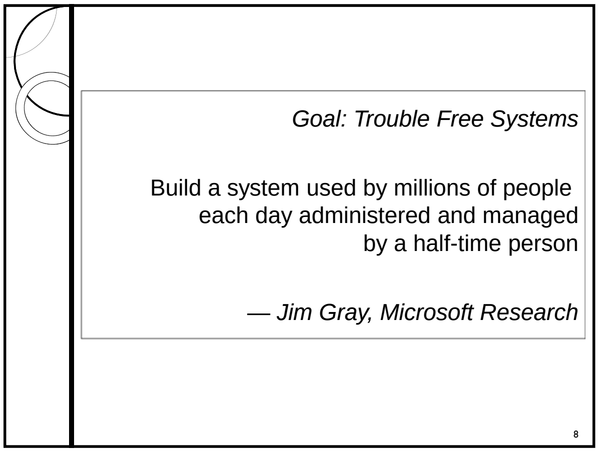
Alan Ganek, Vice President, IBM Autonomic Computing Initiative



The Evolution Problem:

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

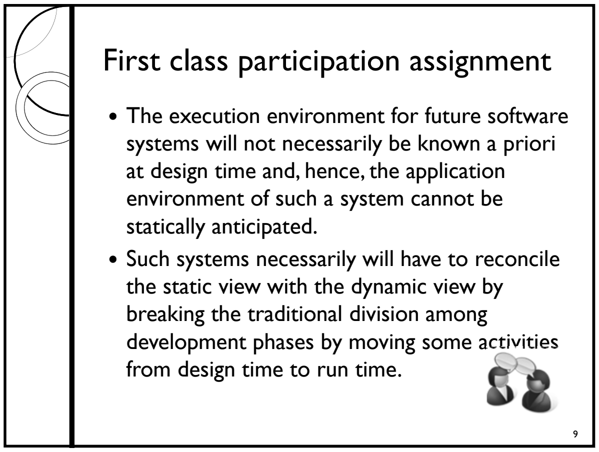
— thus, *software must evolve*



Goal: Trouble Free Systems

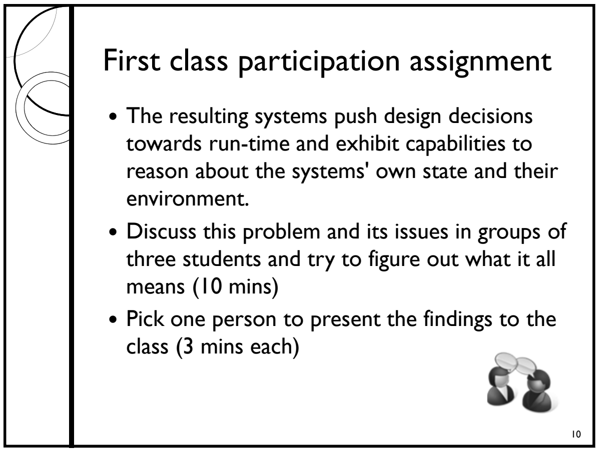
Build a system used by millions of people
each day administered and managed
by a half-time person

— Jim Gray, Microsoft Research



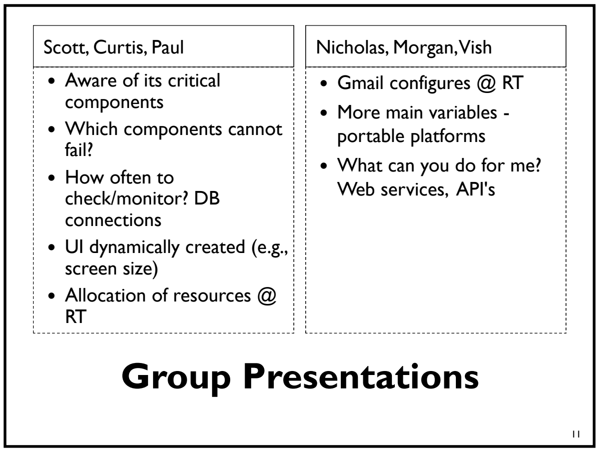
First class participation assignment

- 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.



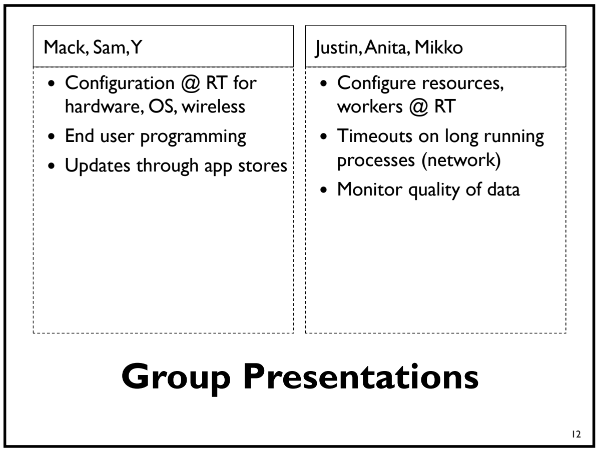
First class participation assignment

- 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)



Scott, Curtis, Paul	Nicholas, Morgan, Vish
• Aware of its critical components • Which components cannot fail? • How often to check/monitor? DB connections • UI dynamically created (e.g., screen size) • Allocation of resources @ RT	• Gmail configures @ RT • More main variables - portable platforms • What can you do for me? Web services, API's

Group Presentations



Mack, Sam, Y	Justin, Anita, Mikko
• Configuration @ RT for hardware, OS, wireless • End user programming • Updates through app stores	• Configure resources, workers @ RT • Timeouts on long running processes (network) • Monitor quality of data

Group Presentations

Daniel, David, Brad	Jeremy, Wes, Kai
• Get apps from the "source", not stored on the device • Configuring UI design/visual design (prototyping) — accessibility support	• Live previews • Load problems • Testability problems • Complicates design process • Support for platform independence • Service stacks

Group Presentations

Jeff, Michael, Adam	Brandon, Rob
• Not knowing your OS/libs • Dynamic building, linking, scripting • Pull in components @RT • UI screen size • Unavailability of services • Stateless interfaces/services • Transaction management	• Adapt to platforms • Java • Virtual Machines (VMs)

Group Presentations

Jordan, Ian, Saleh	Mark, Alan, George
• UI screen size — what is available? • Context based — adapt to context	• Multiple versions • How does user select it • Multiple app files within the same download/install

Group Presentations

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?

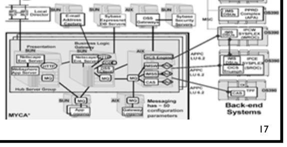
Complexity of configurations

- 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 …

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

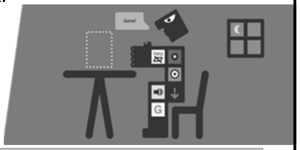
x 2-5 parameters

2^{150} settings



The Conquest of Complexity

- There has never been anything quite like information technology before, but there have certainly been other complex technologies that needed simplifying
- To be truly successful, a complex technology needs to "disappear"



Source: A. Kluth. Information Technology. *The Economist*, Oct 28, 2004

19th Century Technology

- Mechanical Clocks and Sewing machines
  - 1820’s Long 40 page manuals of usage
  - 1880’s Are simple and widely used
- Phonograph
  - Edison’s 1877 cylinder version was unusable
  - Berliner’s simplified disc version became gramophones, Victrolas, and record players

19th Century Technology

- Automobile
  - 1900s: mostly burden and challenge
    - Required skill in lubricating moving parts
    - Sending oil manually to the transmission
    - Frequent breakdowns
    - Mechanic hired as chauffeur
  - 1930s: usable and ready for mass market
    - Infrastucture: roads, gas stations, repair shops
    - Hiding technology from drivers
    - Highly more complex on the inside, because most of the tasks that had previously been carried out by drivers now had to be done automatically
    - Greatly simplified interface, more reliable

Model T Ford

20th Century Technology

- Electricity and power distribution
  - First generation
    - Households and firms have own generators
    - Full time job to keep the generators going
    - Vice President of Electricity (VPE)
      - like CIO or CTO today
  - Only one generation later
    - Power grid
    - Simplified, ubiquitous power plug
    - VPE disappeared
      - will CIOs or CTOs disappear?

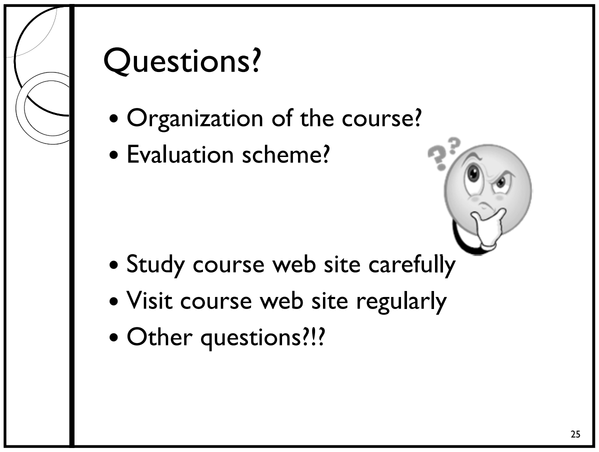
Predictable evolutionary path of technology

- Early stages
  - Technology needs lots of human involvment
  - New inventions are typically “geeky”, requiring significant expertise to install and maintain
  - In general, the “default” seems to be human work, due to its flexibility and adaptivity
  - At an early stage human involvement is always superior to alternatives
  - Culling of features is futile
- Push the complexity to the back end to make the front end very simple
  - Consumers don’t know when the Power Company upgrades its technology

Predictable evolutionary path of technology

- Mature stage
  - Need for human expertise is greatly reduced due to technology becoming simple and standardized
  - To increase adoption and sales (electricity, cars)
  - To decrease cost (industrial revolution, agriculture)
  - To allow super-human performance (space aviation)
- Simplicity of usage often means increased overall system complexity
  - For every mouse click we take out of the user experience, 20 things have to happen in the software behind the scenes

Given this historical perspective, maybe there is hope for the information technology sector?

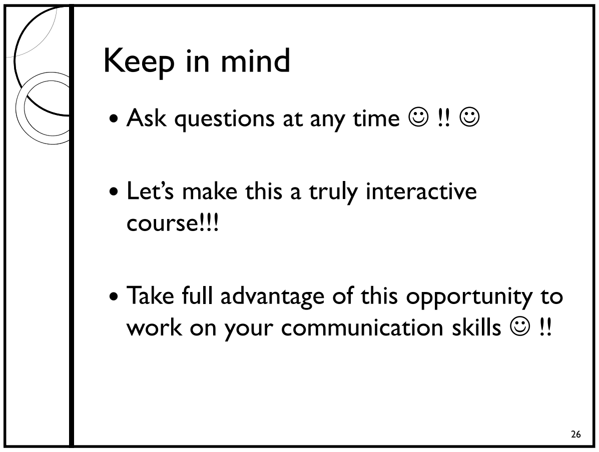


Questions?

- Organization of the course?
- Evaluation scheme?

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

25



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 ☺ !!

26