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

2

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

3

People Depend on Software

Software is central to our lives

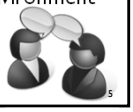
We interact daily with software

- **At home—computer games**
- **At the office—on-line services**
- **In the car—embedded control systems**

We expect software to be reliable, efficient and effective in safety-critical systems as well as desktop computers

Some basic definitions

- *Software* — the programs, documentation, and operating procedures by which computers can be made useful to humans
- *Software evolution* — a process of continuous change from a lower, simpler to a higher, more complex, or better state
- *Software maintenance* — modification of a software product after delivery, to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment
- *Maintainability* — the ease with which maintenance can be carried out



5

Boundary between development-time and runtime is disappearing

- Maintenance at runtime
- Evolution at runtime
- Requirements at runtime
- Models at runtime

- Bencomo: Workshop Series on Models@run-time, http://www.comp.lancs.ac.uk/~bencomo/WorkshopMRT.html
- Bencomo: Workshop Series on Requirements@run.time, http://www.comp.lancs.ac.uk/~bencomo/RRT/
- Dagstuhl Seminar: Models@run.time, 2011 http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=11481

6

Parts of a software system

Software Components	Examples		
Program	1 Source code 2 Object code		
Documentation	1 Analysis / specification:	(a) (b) (c)	Formal specification Context diagram Data flow diagrams
	2 Design:	(a) (b)	Flowcharts Entity-relationship charts
	3 Implementation:	(a) (b)	Source code listings Cross-reference listings
	4 Testing:	(a) (b)	Test data Test results
Operating procedures	1 Instructions to set up and use the software system 2 Instructions on how to react to system failures		

7

Maintenance versus evolution... (1)

- The term *Software Engineering* was coined in 1968 at a NATO meeting to address the upcoming "software crisis"
- Maintenance was considered to be something that was done after delivery — as in the waterfall model
- Evolution captures the more realistic evolutionary model of software — it is never "done"
- Manny Lehman in the 70's proposed laws of software evolution, after studying the IBM OS 360 operating system — findings later confirmed in other studies, especially of proprietary systems
- Term gaining more acceptance since the 90's

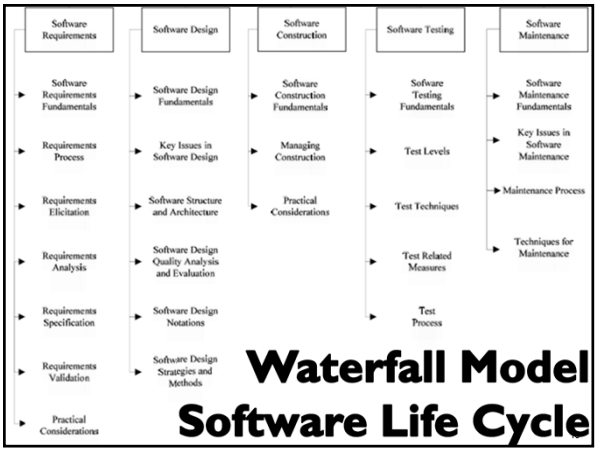
8

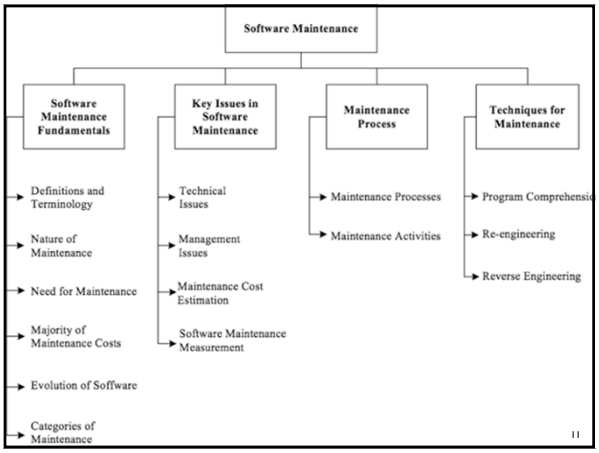
Maintenance versus evolution... (2)

- Software that is used in the real world, will need to adapt as the world continually changes
- Software that doesn't permit change is said to suffer from decay – a poorly degraded system will have to be phased out (sometimes called a legacy system)
- Software evolution is also fundamental in agile development which recognizes the need to continually adapt to changing requirements in a lightweight and agile manner
- Nowadays the terms software evolution and software maintenance are considered synonyms
- Prefer the term evolution, because maintenance may imply that the software has deteriorated in some way

Mens and Demeyer: Software Evolution, Springer, 2008

9





11

Course description

- Introduces problems and solutions of long-term software maintenance/evolution and large-scale, long-lived software systems.
- Topics include software engineering techniques for programming-in-the-large, programming-in-the-many, legacy software systems, software architecture, software evolution, software maintenance, reverse engineering, program understanding, software visualization, advanced issues in object-oriented programming, design patterns, antipatterns, and client-server computing.

12

Course description

- Large software systems form the backbone of much of the computing world; modern clients and servers rely on operating systems, database management systems, office productivity suites, web servers, and a variety of other large-scale, non-trivial software packages. Such packages can easily contain many millions of lines of source code, developed by thousands of individuals over the course of many years, often with different versions and revisions across the life of the product.
- This course introduces the problems and solutions inherent in developing such large scale software systems.

13

Course topics

- Software maintenance and evolution
- Build environment management
- Software installation and configuration
- Fundamentals of software change
- Maintenance processes
- Program understanding
- Management and measurement
- Human side of software maintenance
- Reverse engineering
- Software visualization
- Testing
- Patterns and anti-patterns
- Software Safety
- Reuse, reusability and reengineering
- Maintenance tools
- Documentation, code and API guidelines
- Open source development
- Legal aspects in maintenance

14

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

15

Prerequisites and Related Courses

- Prerequisites
  - SENG 271 Software Model Engineering
  - Basics of software life cycle
  - Basics of software architecture
- Co-requisites
  - SENG 321 Requirements Engineering

16

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.

17

Please note

- Your e-mail domain name
  - Send e-mail and submit assignments only using uvic.ca domain name
  - Messages from hotmail and yahoo in particular are filtered by lab servers and most professors and end up in the tar pit
- Assignment 1
  - A1 will be posted by Monday

18

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

19

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

20

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 <u>not</u> 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

Communication Skills

21

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

22

Questions?

- Organization of the course?
- Evaluation scheme?

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

23

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 ☺ !!

24