Construction of User Interfaces (SE/ComS 319)

Jinu Susan Kabala

Department of Computer Science

Iowa State University, Fall 2021

INTRODUCTION

Software Construction (Engineering)

- Set of concepts & practices
- An engineering discipline about all aspects of software production
  - Gathering requirements
  - Design
  - Development
  - Testing & debugging
  - Maintenance

What is the goal of software engineering?

- Produce good and functional software, within the given time schedule and resource budget
- Good programmers ➔ good software
- Good Programmers?
  - Write good programs on-time that work correctly (functional, **user-friendly**, time-to-market, budget and resources)

Criteria for good software?

- Reliable/correct (few bugs)
- Efficient (run fast)
- Easy for maintenance
- **Good usability** (**user interfaces/interaction**)
- Good security
- ➔ **Software quality**

What are the challenges?

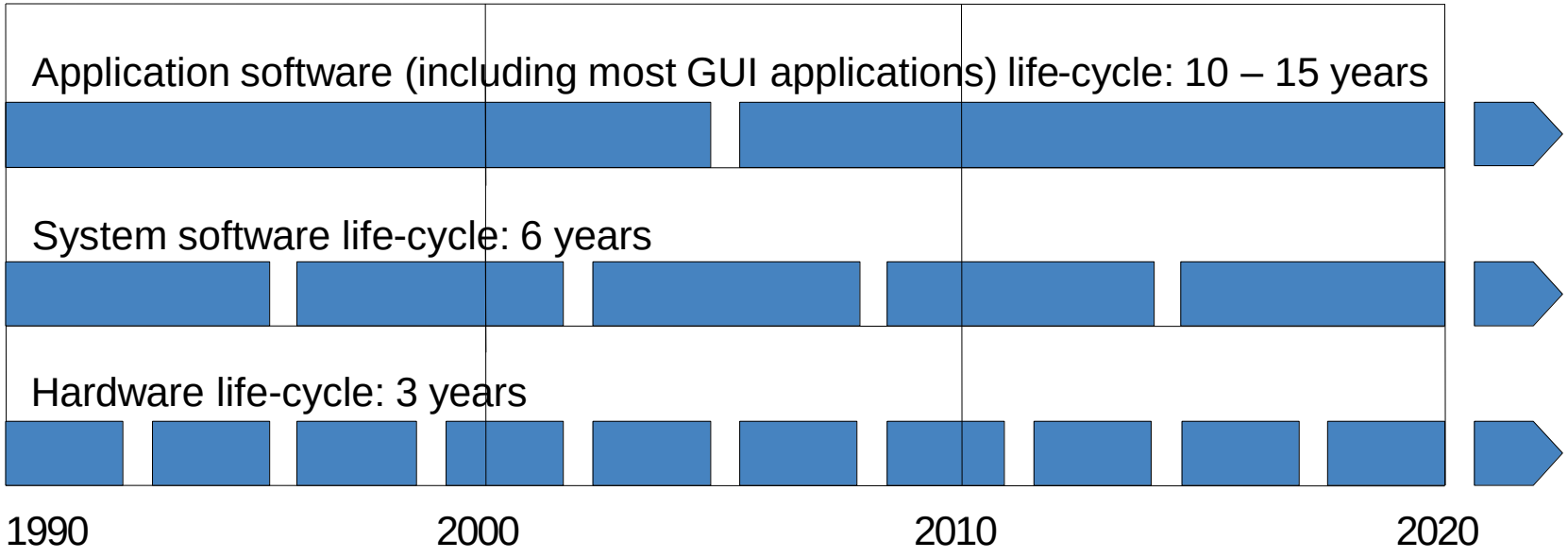
- Large code sizes (not easy to debug)
  - Linux kernel 1.0.0 (1994): 100K+ ➔ Linux kernel 2.6.0 (2003): 5M+
  - Boeing 787: 5M+
  - Chrome? Firefox? Mac OS X?
    - See: https://informationisbeautiful.net/visualizations/million-lines-of-code
- Changing requirements
  - User, hardware, multiple platforms (portability), new GUI, new Interfaces (web interface, speech control, app), …
- Large development teams distributed in different offices around the globe
- Time-to-market

What are the challenges?

Example @ Google:

- 15K+ developers in 40+ offices
- 4K+ projects under active development
- 5K+ submissions per day on average
- Single monolithic code tree with mixed language code
- Development on one branch - submissions at head
- All builds from source
- 20+ sustained code changes per minute with 60+ peaks
- 50% of code changes monthly
- 75+ million test cases run per day

Problems in creating time-to-market software



Problems in creating time-to-market software

If you develop application software, then it means that ...

- During the lifetime of the application software at least once the underlying system software and at least twice the hardware is replaced
- The target systems for which software is developed may not yet be available at development time
- Software products distributed throughout the world may need country-specific variants (e.g. native-language user interfaces, etc.)

Changes in the software in the last decades

- Increasing Importance – software is becoming more and more critical and important in many in-demand areas
- Growing complexity
- **Growing software in mobile devices (everywhere with different UIs)**
- Networking and geographical distribution
- Increasing quality requirements
- Increasing standard software
- Increasing ”legacy code“
- Increasing "off-the-shelf" development

Importance of software

- Give examples of where you depend on software in your daily life!

- How did you like their UI (user interfaces)?

User Interface (UI) is very important!

Software quality

- Software failures are the reason for 50% of industrial sector failures
- 0.1% defect level means:
  - per year:
    - 20,000 defective drugs and medications
    - 300 failing cardiac pacemakers
  - per week:
    - 500 errors in medical operations
  - per day:
    - 16,000 lost letters in the mail
    - 18 plane crashes
  - per hour:
    - 22,000 checks were not booked correctly

In the future: still massive quality assurance efforts needed!

Software quality problems

Example:
The Ariane 5 rocket exploded 40s after the start. (1996, $500 million damage).

Details: http://www.around.com/ariane.html

Video: http://www.youtube.com/watch?v=kYUrqdUyEpI

Ariane 5 – Navigation system (Ada code)

```
...
declare
 vertical_veloc_sensor: float;
 horizontal_veloc_sensor: float;
 vertical_veloc_bias: integer;
 horizontal_veloc_bias: integer;
 ...
begin
 declare
  pragma suppress(numeric_error, horizontal_veloc_bias);
 begin
  sensor_get(vertical_veloc_sensor);
  sensor_get(horizontal_veloc_sensor);
  vertical_veloc_bias := integer(vertical_veloc_sensor);
  horizontal_veloc_bias := integer(horizontal_veloc_sensor);
  ...
 exception
  when numeric_error => calculate_vertical_veloc();
  when others => use_irs1();
 end;
end irs2;
```

16-bit signed integer

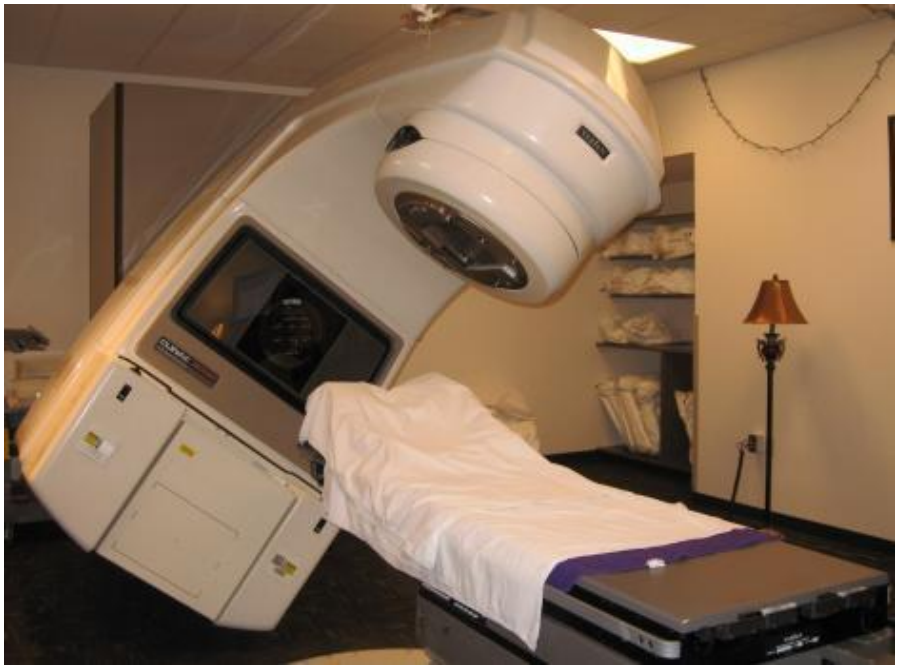
Floating point value: 32768.0

Overflow on conversion, which was not caught.
$2^{15} - 1 = 32.767$ (signed integer)

Switch to replacement computer.

Accident with Therac-25 – Poor user interface!

- Therac-25 was a medical linear accelerator used to treat cancer
- **Patients died** due to overdose of radiation causes by software bugs and poor user interactions. **User interface problems!**



Reasons for the cause of the accidents

- The system noticed that something was wrong and halted the X-ray beam
  - But merely displayed the word "MALFUNCTION" followed by a number from 1 to 64.
- The user manual did not explain or even address the error codes
  - So the operator pressed the P key to override the warning and proceed anyway
- Machine pauses treatment **but did not indicate the reason and why**!
- **Interaction with user:**
  - Monitoring input and editing changes from an operator
  - Updating the screen to show the current status of machine

Need of better software engineering!

- Advanced tools and methods for software engineering
- Software process
- All aspects of software production, including: **User interface design (UI)** or **user interface engineering**
  - Design of user interfaces for software (on various targets such as PC, mobile devices, and other electronic devices) to maximize usability and the user experience!

Summary

- What is software construction/engineering?
- Criteria for good user-friendly software, challenges, etc.
- Importance of software, user interfaces and software quality.