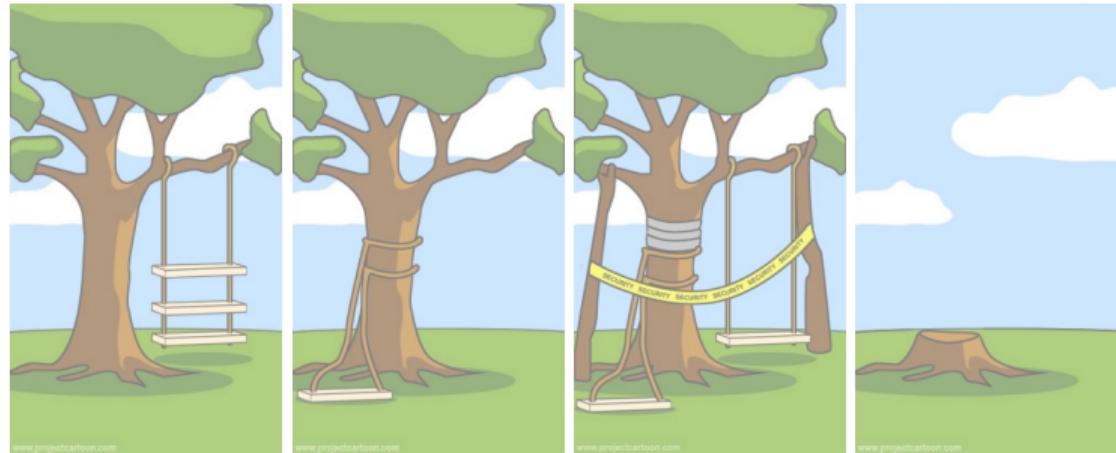




Software Engineering

12. Software Maintenance | Thomas Thüm | May 3, 2022

Software Maintenance (Software-Wartung)



how the customer
explained it

how the
programmer
implemented it

how patches were
applied

how it was
supported

**software
evolution**

**software
maintenance**

Lessons Learned

- Smells (anti patterns): structures to avoid
- Examples: mysterious name, duplicated code, long method
- Refactorings: clean-up of structures before and after every change
- Examples: rename X refactoring, extract/pull-up method refactoring
- Further Reading: [Sommerville](#), 3.2.2 Refactoring
- Further Reading: [Fowler's Refactoring](#)
- Further Reading: [Uncle Bob's Clean Code](#)

Practice

- Research another smell and report it (incl. source) to Moodle
- Find a suitable refactoring for an anti-pattern of your colleagues (answer to their post)
- Will be discussed in next lecture
- Deadline: May 2 at 12 noon



Lecture Overview

1. Software Maintenance in Action
2. Introduction to Software Maintenance
3. Migration of Legacy Software

Lecture Contents

1. Software Maintenance in Action

New Year's Eve: Party Time?!?

New Year's Eve for Maintainers

Maintenance at Work

Wisdom on Debugging

The One-Character-Fix

Leap Seconds

Wisdom on Maintenance

Lessons Learned

2. Introduction to Software Maintenance

3. Migration of Legacy Software

New Year's Eve: Party Time?!?



London Eye on January 1st, 2008 2017

New Year's Eve for Maintainers

Meanwhile in the Internet



Oops! Website currently not available.

Nice of you to come by, but currently this web page is feeling a bit under the weather.

Why not check back later?

Your DNS Zone File

Export your DNS zone file - Append a zone file

Type	Name	Value	TTL	Active
A	theburritobot.com	points to 205.178.189.129	Automatic	<input checked="" type="checkbox"/>
A	direct	points to 205.178.189.129	Automatic	<input checked="" type="checkbox"/>
A	*	points to 205.178.189.129	Automatic	<input checked="" type="checkbox"/>
CNAME	www	is an alias of theburritobot.herokuapp.com	Automatic	<input checked="" type="checkbox"/>
MX	theburritobot.com	mail handled by theburritobot.com with priority 0	Automatic	<input checked="" type="checkbox"/>

Meanwhile in San Francisco

[cloudflare.com]

"At midnight UTC on New Year's Day, deep inside Cloudflare's custom RRDNS software, a number went negative when it should always have been, at worst, zero. [...] At peak approximately 0.2% of DNS queries to Cloudflare were affected and less than 1% of all HTTP requests to Cloudflare encountered an error."

Maintenance at Work

[cloudflare.com]

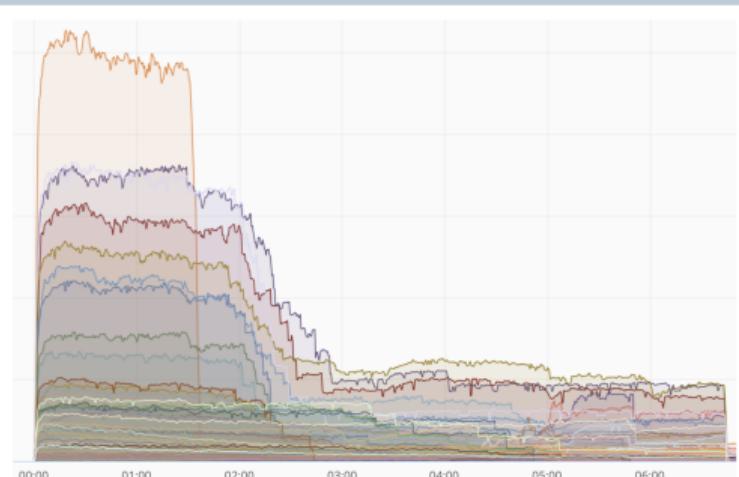
Meanwhile in San Francisco

"This problem was quickly identified. The most affected machines were **patched in 90 minutes** and the fix was **rolled out worldwide by 06:45 UTC**. We are sorry that our customers were affected, but we thought it was worth writing up the root cause for others to understand."

The Timeline

- 2017-01-01 00:00 UTC Impact starts
- 2017-01-01 00:10 UTC Escalated to engineers
- 2017-01-01 00:34 UTC Issue confirmed
- 2017-01-01 00:55 UTC Mitigation deployed to one canary node and confirmed
- 2017-01-01 01:03 UTC Mitigation deployed to canary data center and confirmed
- 2017-01-01 01:23 UTC Fix deployed in most impacted data center
- 2017-01-01 01:45 UTC Fix being deployed to major data centers
- 2017-01-01 01:48 UTC Fix being deployed everywhere
- 2017-01-01 02:50 UTC Fix rolled out to most of the affected data centers
- 2017-01-01 06:45 UTC Impact ends

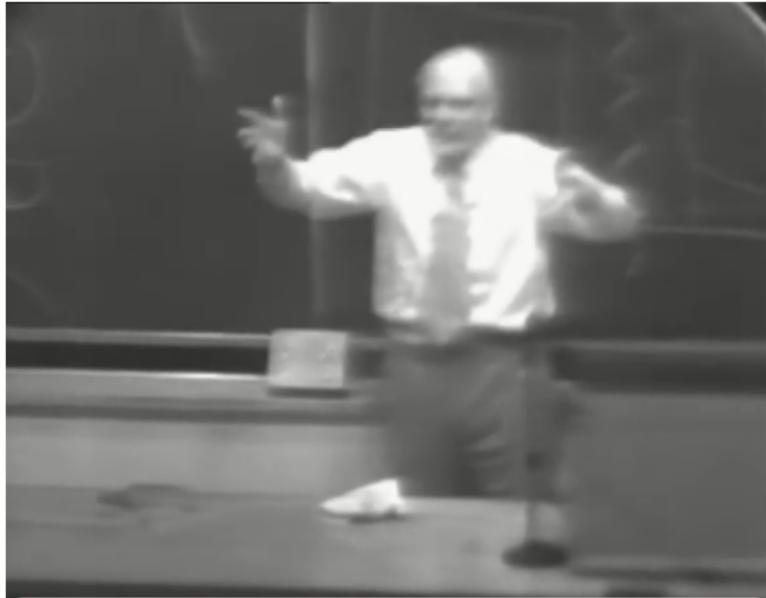
Monitoring



Fun Fact

San Francisco's standard time zone is UTC-8

Wisdom on Debugging



Gordon Glegg

[[twitter.com](#)]

"Sometimes the problem is to discover what the problem is."



Jessica Joy Kerr (2020)

[[jessitron.com](#)]

"In programming, it's dangerous to work near your working memory threshold. You get more mistakes and more complicated code."

The One-Character-Fix

[cloudflare.com]

The Patch

```
121 121      rttMax := servers[server_count-1].GetRTT()  
122 -       if rttMax == 0 {  
122 +     if rttMax <= 0 {  
123 123         rttMax = DefaultTimeout  
124 124     }
```

A Falsehood Programmers Believe About Time

“The root cause of the bug that affected our DNS service was the belief that **time cannot go backwards**. In our case, some code assumed that the difference between two times would always be, at worst, zero.

RRDNS is written in Go and uses Go’s `time.Now()` function to get the time. Unfortunately, this function does not guarantee monotonicity.”

Leap Seconds (Schaltsekunden)

23:59:60

Leap Second 37

[spinellis.gr]

	Linux NTP	FreeBSD NTP	Linux NTP Google	Linux TAI
SI	64.677	64.648	64.641	64.652
Unix	1483228803.685	1483228804.649	1483228803.663	1483228830.882
Human	2017-01-01 00:00:03	2017-01-01 00:00:04	2017-01-01 00:00:03	2017-01-01 00:00:03
SI				
58	1483228798	1483228798	1483228798	1483228824
	2016-12-31 23:59:58	2016-12-31 23:59:58	2016-12-31 23:59:58	2016-12-31 23:59:58
59	1483228799	1483228799	1483228799	1483228825
	2016-12-31 23:59:59	2016-12-31 23:59:59	2016-12-31 23:59:59	2016-12-31 23:59:59
60	1483228799	1483228800	1483228799	1483228826
	2016-12-31 23:59:59	2017-01-01 00:00:00	2016-12-31 23:59:59	2016-12-31 23:59:60
61	1483228800	1483228801	1483228800	1483228827
	2017-01-01 00:00:00	2017-01-01 00:00:01	2017-01-01 00:00:00	2017-01-01 00:00:00
62	1483228801	1483228802	1483228801	1483228828
	2017-01-01 00:00:01	2017-01-01 00:00:02	2017-01-01 00:00:01	2017-01-01 00:00:01
63	1483228802	1483228803	1483228802	1483228829
	2017-01-01 00:00:02	2017-01-01 00:00:03	2017-01-01 00:00:02	2017-01-01 00:00:02

27 Leap Seconds since 1972

[wikipedia.org]

Year	Jun 30	Dec 31	
1991	0	0	
1992	+1	0	
1993	+1	0	
1994	+1	0	
1995	0	+1	
1996	0	0	
1997	+1	0	
1998	0	+1	
1999	0	0	
2000	0	0	
2001	+1	0	
2002	+1	0	
2003	0	0	
2004	0	0	
2005	+1	0	
2006	0	0	
2007	0	+1	
2008	0	0	
2009	+1	0	
2010	0	0	
2011	0	0	
2012	+1	0	
2013	0	0	
2014	0	0	
2015	+1	0	
2016	0	+1	
2017	0	0	
2018	0	0	
2019	0	0	
2020	0	0	
2021	0	TBA	
Year	Jun 30	Dec 31	
Total	11	16	
		27	
		Current TAI – UTC	
		37	

WHY DO THE CLOCKS SAY IT'S 3AM?

ADDING AN EXTRA DAY CREATES
TOO MANY GLITCHES. INSTEAD,
WE'RE JUST RUNNING OUR CLOCKS
3.4% SLOWER DURING FEBRUARY
TO AVOID THE IRREGULARITY.



THIS YEAR, GOOGLE HAS EXPANDED
THEIR LEAP SECOND "SMEARING" TO
COVER LEAP DAYS AS WELL.

Wisdom on Maintenance



Karolina Szczur

[[twitter.com](#)]

“Writing software as if we are the only person that ever has to comprehend it is one of the biggest mistakes and false assumptions that can be made.”



John F. Woods

[[google.com](#)]

“Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live.”

Software Maintenance in Action

Lessons Learned

- Maintenance problems caused by leap seconds
- Solutions: consider negative intervals, leap second smearing
- Time pressure for maintenance
- Develop software for maintenance
- Further Reading: see references above

Practice

- Form groups of 2–3 students
- Think (2 min): which software maintenance has affected you? how long did it to fix it?
- Pair (5 min): discuss the example with your colleagues
- Share (3 min): share one example in Moodle and vote for other examples



Lecture Contents

1. Software Maintenance in Action
2. Introduction to Software Maintenance
 - Software Maintenance
 - Kinds of Maintenance
 - Maintenance and Evolution
 - Reengineering Tasks
 - Wisdom on Reengineering
 - Lessons Learned
3. Migration of Legacy Software

Software Maintenance

[Ludewig and Licher]

Motivation

- for software: no compensation of deterioration, repair, spare parts
- corrections (especially shortly after first delivery)
- modification and reconstruction

Operation and Maintenance Phase

[IEEE Std 610.12]

"The period of time in the software life cycle during which a software product is employed in its operational environment, monitored for satisfactory performance, and modified as necessary to correct problems or to respond to changing requirements."

Maintenance

[IEEE Std 610.12]

"The process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment."

Kinds of Maintenance

[Ludewig and Licher]

Adaptive Maintenance

[Lientz and Swanson 1980]

“Software maintenance performed to make a computer program usable in a changed environment.”

desktop application for a new version of an operating system (e.g., from Windows 8.1 to 10)

Corrective Maintenance

[Lientz and Swanson 1980]

“Maintenance performed to correct faults in software.”

Windows calculator showing wrong formulas

Perfective Maintenance

[Lientz and Swanson 1980]

“Software maintenance performed to improve the performance, maintainability, or other attributes of a computer program.”

better handling of very large files in a text editor

Preventive Maintenance

[Lientz and Swanson 1980]

“Maintenance performed for the purpose of preventing problems before they occur.”

2,000 year problem, leap seconds/years

LATEST: 10.17

UPDATE

CHANGES IN VERSION 10.17: THE CPU NO LONGER OVERHEATS WHEN YOU HOLD DOWN SPACEBAR.

COMMENTS:

LONGTIMEUSER4 WRITES:

THIS UPDATE BROKE MY WORKFLOW!
MY CONTROL KEY IS HARD TO REACH,
SO I HOLD SPACEBAR INSTEAD, AND I
CONFIGURED EMACS TO INTERPRET A
RAPID TEMPERATURE RISE AS "CONTROL".

ADMIN WRITES:

THAT'S HORRIFYING.

LONGTIMEUSER4 WRITES:

LOOK, MY SETUP WORKS FOR ME.
JUST ADD AN OPTION TO REENABLE
SPACEBAR HEATING.

EVERY CHANGE BREAKS SOMEONE'S WORKFLOW.

Maintenance and Evolution

[Ludewig and Licher]

Maintenance

- mostly corrections
- small changes
- often unforeseen changes
- results in patches and hot fixes (minor updates)
- minor release, new minor version: 2.3.0 ⇒ 2.3.1

Evolution

- new or removed functionality
- large changes
- often foreseen changes
- results in upgrades or service packs (major updates, cumulative updates)
- major release, new major version: 2.3.2 ⇒ 2.4.0

Reengineering Tasks

[Ludewig and Licher]

Reverse Engineering

[Chikofsky und Cross]

“Reverse engineering is the process of analyzing a system to identify the system’s components and their interrelationships and create representations of the system in another form or at a higher level of abstraction.”

updating UML diagrams from source code

Forward Engineering

[Chikofsky und Cross]

“Forward engineering is the traditional process of moving from high-level abstractions and logical, implementation-independent designs to the physical implementation of a system.”

see Software Engineering I

Restructuring

[Chikofsky und Cross]

“Restructuring is a transformation from one form of representation to another at the same relative level of abstraction. The new representation is meant to preserve the semantics and external behavior of the original.”

refactorings as introduced in lecture on evolution

Reengineering

[Chikofsky und Cross]

“Reengineering is the examination and alteration of a subject system to reconstitute it in a new form and the subsequent implementation of the new form.”

combination of reverse engineering, refactoring, and forward engineering

Wisdom on Reengineering



Kyle Simpson

[[twitter.com](#)]

“There's nothing more permanent than a temporary hack.” (Sprichwort: Nichts ist so beständig wie das Provisorium.)



Jeff Sickel

[[twitter.com](#)]

“Deleted code is debugged code.”

Introduction to Software Maintenance

Lessons Learned

- Maintenance (Phase)
- Kinds: adaptive, corrective, perfective, preventive
- Maintenance vs evolution
- Reengineering = reverse engineering + restructuring + forward engineering
- Further Reading: [Ludewig and Lichter](#), Chapter 22 ([Software-Wartung](#)) and 23 ([Reengineering](#))

Practice

- Quiz'n'Disquiz
- Quiz: fill out the Moodle quiz on your own
- Disquiz: compare and discuss the results with your colleagues



Lecture Contents

1. Software Maintenance in Action
2. Introduction to Software Maintenance
3. Migration of Legacy Software
 - Legacy Software
 - Flash Player
 - Migration
 - Lessons Learned

Legacy Software (Altsysteme)

Legacy Software

[Ludewig and Licher]

“A large software system that we don't know how to cope with but that is vital to our organization.”

Winamp 5.623 on 4k Display in Windows 10



Typical Properties

[Ludewig and Licher]

- larger than 100k LOC
- older than 10 years
- original developers and architects not available anymore
- outdated programming languages and development concepts
- business-critical
- outdated or missing documentation
- based on outdated hardware and system software
- subject to numerous iterations of corrective and adaptive maintenance
- high costs for maintenance

LOADING... PLEASE
INSERT DISK INTO DRIVE A:
↳ *CLICK* THERE YOU GO.
THANK YOU. WOW, THIS
DISK IS INCREDIBLY FAST!
↳ YEAH, UH, IT'S THE NEW
MODEL FROM MEMOREX.
AMAZING. AND HOW IS
PRESIDENT REAGAN?
↳ HE'S...HE'S FINE.



I FEEL WEIRD USING OLD
SOFTWARE THAT DOESN'T
KNOW IT'S BEING EMULATED.

Flash Player

End of Life: December 31, 2020



The screenshot shows the official Adobe Flash Player End-of-Life (EOL) page. At the top, there's a navigation bar with a menu icon, the Adobe logo, and a "Sign In" link. Below the navigation is a search bar with a magnifying glass icon and the text "ADOLE FLASH PLAYER". The main content area features a large heading "Adobe Flash Player" and "EOL General Information Page". A paragraph explains that Adobe no longer supports Flash Player after December 31, 2020, and blocked Flash content from running in Flash Player beginning January 12, 2021. It advises users to immediately uninstall Flash Player to help protect their systems. Another paragraph notes that some users may continue to see reminders from Adobe to uninstall Flash Player.

Online Game Fuchstreff Discontinued



The screenshot shows the homepage of the online game Fuchstreff. The header includes the Fuchstreff logo, a user profile icon, and links for "wleefuchs", "SPIELEN!", "COMMUNITY", "LERNEN", "LIGA", "FUCHSSCHAU", "FORUM", "KALENDER", and "HILFE". A prominent yellow banner in the center states: "Leider findet der Fuchstreff ohne den Flash Player ein Ende." and "Wir möchten uns bei Euch allen für die schönen Jahre bedanken und wünschen Euch altes Gute." Below the banner, there's a link to "Der Fuchstreff-Blog • RSS-Feed". A post by Sarah dated December 17, 2020, is displayed, titled "Danke für alles", with 214 comments. The post text expresses gratitude to the community for the good years and expresses sadness over the discontinuation of the game due to the lack of Flash support.

Migration

[Ludewig and Licher]

Software Migration

- opposed to data and hardware migration
- renewing or replacing legacy software
- new software needs to be downward compatible
- new software meets current functional and non-functional requirements
- data often needs to be migrated
- outage as short as possible

Migration Strategies

- wrapping: building a new software around a stable version of the legacy software
- redevelopment: functionality implemented again and replaces legacy software
- incremental migration: stepwise renewal and replacement
- big bang migration: whole legacy software is migrated in a single step
- combinations feasible (except for incremental and big bang)

Migration of Legacy Software

Lessons Learned

- Legacy software: typical properties, examples
- Software migration
- Migration strategies: wrapping, redevelopment, incremental migration, big bang migration
- Further Reading: [Ludewig and Lichter](#), Chapter 23 ([Reengineering](#))

Practice

- 1. Post an example of software in Moodle that reached its end of life
- 2. Interpret the reasons for an example by a colleague
- Will be discussed in next lecture
- Deadline: May 9 at 12 noon

