

# Introduction to Software Development Principles and Practices (SWPP)

M1522.000100

September 1, 2022  
Byung-Gon Chun

(Credits of some slides: George Canea, EPFL)

All You Ever Wanted to Know about  
How to Build Large-Scale Software ☺

# SWPP

Before taking SWPP



After taking SWPP

- Better developer
- More beautiful code
- Cleaner code
- More robust code
- Better at writing large-scale software
- Better collaborator

...

# Who am I?

- **Professor** Byung-Gon Chun (전병곤)
  - Software Platform Lab (SPL) (<https://spl.snu.ac.kr>)
- **CEO**, FriendliAI (<https://friendli.ai>)
- Associate Director, SNU AI Institute (<https://aiis.snu.ac.kr>)
- I love solving large-scale AI problems!
- Many papers published in top conference venues (number of citations ~ 15500)
- Education: UC Berkeley PhD, Stanford MS, SNU MS/BS
- Work experience: Facebook, Microsoft, Yahoo!, Intel, ICSI
- Honors: 2011 EuroSys Test-of-Time Award, 2010 ACM SIGOPS Hall of Fame Award, Awards from Google, Amazon, Facebook, Microsoft, and Naver

# Teaching Staff

- Instructor: Byung-Gon Chun
  - Email: bgchun AT snu DOT ac DOT kr
  - Office: Bldg. 302, Rm. 322
  - Office hours: TTH 12:15PM-1:15PM, by appointment
- TAs
  - 맹재우 (email: [jwmaeng@snu.ac.kr](mailto:jwmaeng@snu.ac.kr), office: 302-420, office hour: TBD)
  - 유준열 (email: [gajagajago@snu.ac.kr](mailto:gajagajago@snu.ac.kr), office: 302-420, office hour: TBD)
  - 윤종선 (email: [titieiti@snu.ac.kr](mailto:titieiti@snu.ac.kr), office: 302-420, office hour: TBD)
  - 김민재 (email: [aingo03304@snu.ac.kr](mailto:aingo03304@snu.ac.kr), office: 302-420)  
(Online office hours in the beginning)
- Staff email:
  - [swpp.22.staff@spl.snu.ac.kr](mailto:swpp.22.staff@spl.snu.ac.kr) (instructor + TAs)
  - [swpp.22.tas@spl.snu.ac.kr](mailto:swpp.22.tas@spl.snu.ac.kr) (TAs)

# Goals for Today

- What is this course about?
- How does this class operate?
- Interactive is important!
  - Ask questions!

# **THE ROLE OF SOFTWARE IN MODERN LIFE**



## Toyota Entune

The revolutionary in-car experience that keeps you connected

Register your Entune™ account and link your existing apps to Entune™, then download the Entune™ app from your favorite App Store, such as iTunes®, Android™ Market or Blackberry® Appworld™.

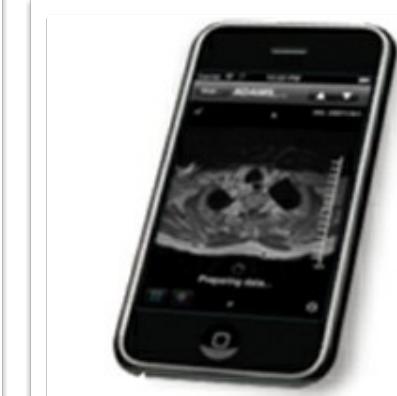


## IBM's Watson supercomputer to diagnose cancer patients

By Lucas Mearian

September 12, 2011 03:13 PM ET

Watson, IBM's game-show-playing supercomputer, will soon be used to help physicians diagnose and treat cancer patients. IBM announced that healthcare would be the first commercial application for the computer, which defeated two human champions on the popular television game show *Jeopardy!* in February.



## Centricity Radiology Mobile

Efficient, secure healthcare delivery. Analyze images and reports on your Apple or Android mobile device.



# Safety Security Reliability Performance Manageability



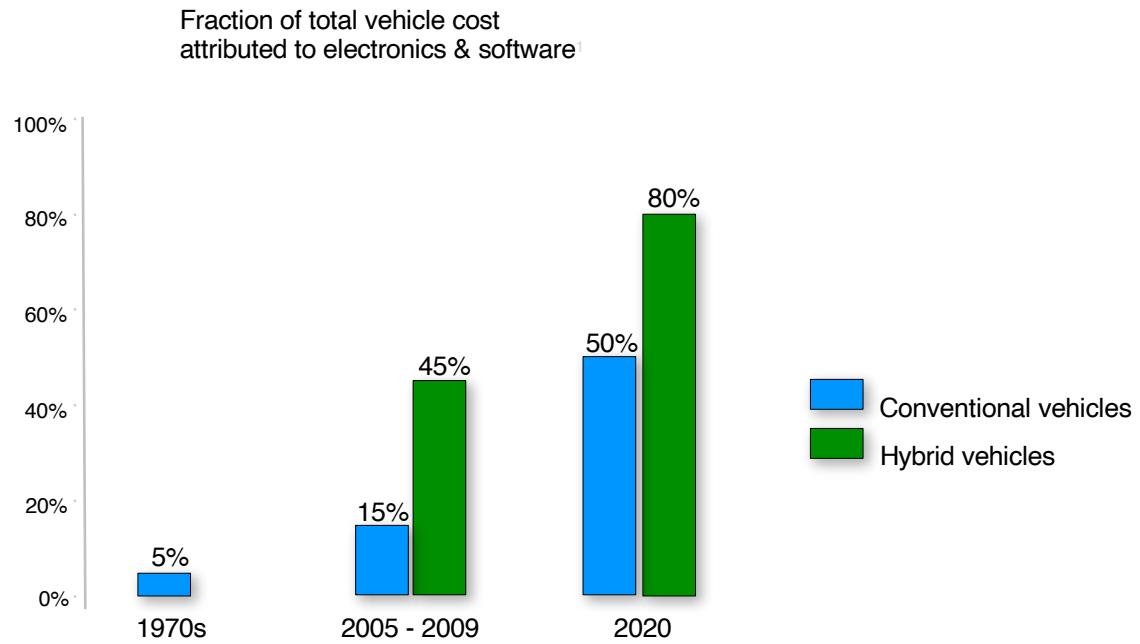
*Software CEO:*

If the automobile industry had developed like the software industry, we would all be driving \$25 cars that get 1,000 miles to the gallon\*.

*Car company CEO:*

If cars were like software, they would crash twice a day for no reason, and when you called for service, they would tell you to reinstall the engine.

\* 1,000 miles/gallon ≈  
0.2 liters / 100 km



“Operating like a technology company, Ford leverages software to continuously update and upgrade the driver experience, moving at a pace similar to that seen in the consumer electronics industry.”

*Ford PR spokesperson*<sup>2</sup>

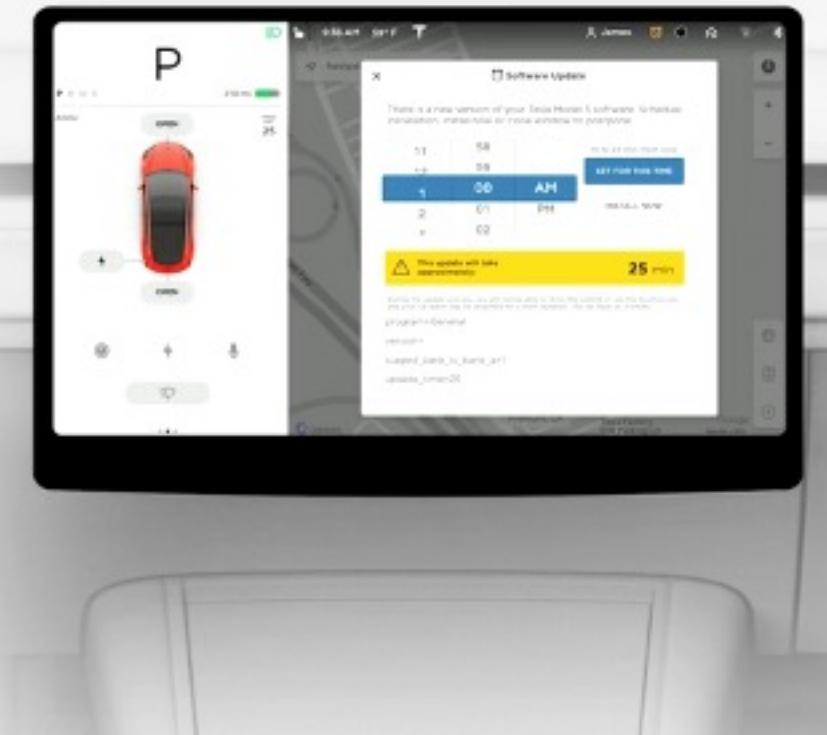
<sup>1</sup> R.N. Charette, *This Car Runs on Code*, IEEE Spectrum (February 2009)

<sup>2</sup> D. Zax, *A Software Update for Your Car?*, MIT Technology Review Blog (March 2012)



TESLA over-the-air software update

Schedule your update  
for your convenience



July 2011 - Honda recalls Accord passenger cars: inappropriate **software** setting of the Engine Control Unit can cause engine oil to enter the combustion chamber during driving.

Apr 2011 - Ford recalls Galaxy, S-MAX, Mondeo cars: **software errors** can affect the Body Control Module which could cause electrical malfunctions, including headlamps switching off.

Volvo recalls V70 and XC70 vehicles: side/curtain airbag system may not work properly due to a **malfunction in the software**.

Jan 2010 - Toyota recalls several million vehicles because of a **software problem** in the stability control software that can cause braking to be delayed, which led to accidents and injuries.

May 2010 - Fiat recalls Ulysse vans: due to a **software problem**, the headlamps could switch off while driving, which could lead to a road accident.

Feb 2010 - Chrysler, Dodge and Dodge cars recalled due to **software** in the Totally Integrated Power Module may stall unexpectedly, potentially causing a crash.

Oct 2010 - Citroën recalls C2, C3, C8 passenger cars: a **software problem** could cause vehicle lights to randomly go out when not required, which could lead to a road accident.

## Amazon Web Services: The hidden bugs that made AWS' outage worse

Although an electrical storm caused the Amazon Web Services outage, it was a slew of previously unseen bugs in the cloud's technology that extended the downtime

# Software bug led to death in Uber's self-driving crash



Sensors detected Elaine Herzberg, but software reportedly decided to ignore her.

<https://arstechnica.com/tech-policy/2018/05/report-software-bug-led-to-death-in-ubers-self-driving-crash/>



# Why Study Software Engineering?

- How to write software that is
  - Safe
  - Secure
  - Reliable
  - High-performance
  - Manageable
- Improve developers' productivity
- Stimulate programmers' creativity
- Make it cheaper to build good software

# **WHAT IS THIS COURSE ABOUT?**

# This Course is About

- Principles + Practices  
of building large-scale software systems
- A hands-on course on large-scale software systems  
Project-oriented: build a web service of your own for a semester

# This Course is About

- Building large software systems that actually work is hard.  
This course covers **techniques for dealing with the complexity of software systems**
- We will focus on the technology of software development principles and software engineering for the individual and small team
  - Coding conventions, specifications, software design, software architecture, testing, abstraction, modularity, design patterns, software development process, version control, code review, tooling, etc.

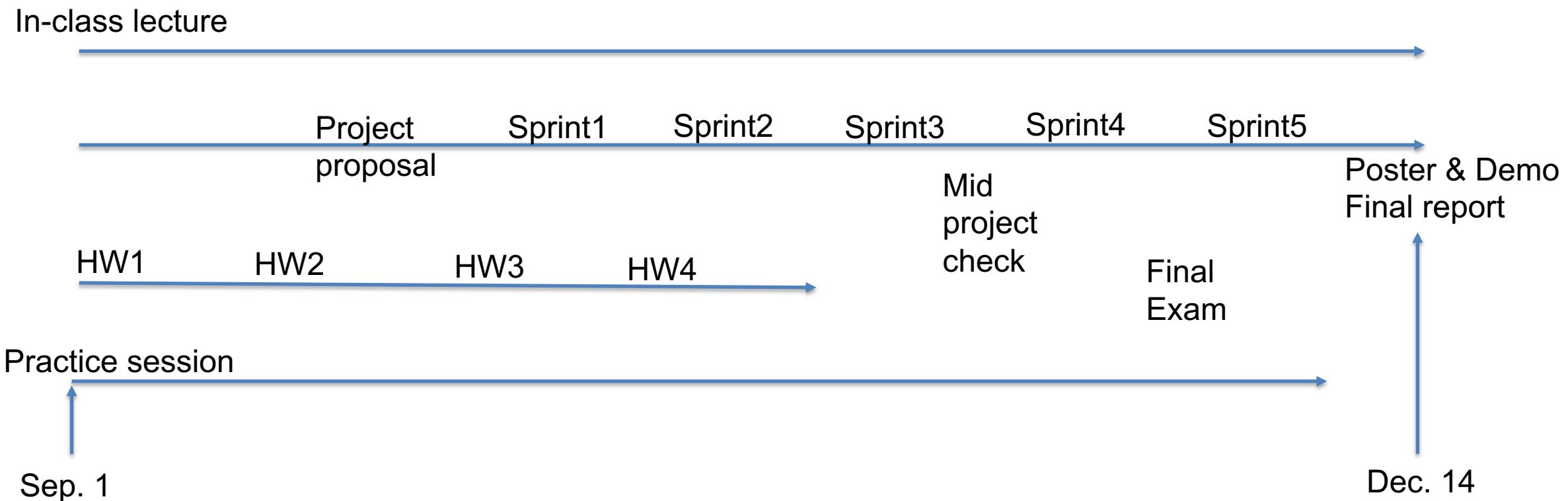
# This Course is About

- The students are expected to **apply the principles** to systems in practice by working on **semester-long team projects on web services**
- Imagine that each team is creating its own startup. A team **proposes** what to build and students apply software engineering principles to **build** their software products.

# Course Changes

- 3 units (~Spring 2018) => 4 units (Fall 2018~)
- 5 sprints => 6 sprints (12 weeks) (Fall 2018~2019) => 5 sprints (10 weeks) (Fall 2020)
- 2 exams => 1 long exam (Fall 2018~)
- Typescript/Angular (~Fall 2018)=> (NG)Javascript/React (Fall 2019~) => **Typescript/React (Fall 2022)**
- Django (No change)
- AWS (~Fall 2018) => MS Azure (Fall 2019) => AWS (Fall 2020) => **AWS + FriendliAI (Fall 2022)**

# Overall Course Structure



# Class Grading Components

[The Percentage Can Change]

Class participation/attendance /surprise quizzes	10%
One in-class coding exam (open notes, ~ 4 hours)	20%
Four individual warm-up programming assignments	20%
<b>Team project</b>	<b>50%</b>

# Course Materials

- **There is no required textbook in this class.**
  - If you want to read more about the topics covered in the class, I recommend to read the following books.
    - "Engineering Software as a Service: An Agile Approach Using Cloud Computing", by Armando Fox and David Patterson
    - "Software Engineering. A Practitioner's Approach (6th ed.) ", by Roger Pressman
    - "Code Complete", by Steve McConnell
    - "Design Patterns: Elements of Reusable Object-Oriented Software", by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
    - "Extreme Software Engineering. A Hands-On Approach", by Daniel H. Steinberg, Daniel W. Palmer
    - "Structure and Interpretation of Computer Programs (SICP) (2<sup>nd</sup> ed.)", by Harold Abelson, Gerald Jay Sussman
- ...

# Course Lectures and Practice Sessions

- Lecture - **Tuesday, Thursday 11:00-12:15PM**
  - Proposal Evaluation
  - Mid project check
  - Sprint meetings (TAs)
  - Final project presentation
  - Programming assignments
- Practice session – **Wednesday 6:30-8:20PM**
  - Step-by-step practice on software development principles
  - You may need to stay longer to finish up **in-class exercises**
- Don't miss practice sessions: we try to make lectures and practice sessions go hand in hand.

# Syllabus (subject to change)

(<https://github.com/swsnu/swppfall2022/>)

Week	Lecture	Practice Session	Homework / Project / Exam
9/1	Course overview		HW1 out 9/1
9/6, 8	Challenges to make software; Version control	Environment Setup(Docker) + Python + HTML/TypeScript	HW2 out 9/7, HW1 due 9/9 6pm
9/13, 15	SaaS architecture; Building software; ORM	Git	Team formation due 9/13 6pm, HW2 due 9/16 6pm
9/20, 22	Project sprints; Requirements and specification	React	HW3 out 9/21
9/27, 29	Testing	Redux	Project proposal due 9/28 6pm

# Syllabus (subject to change)

10/4, 6	Software development process; Design patterns	Frontend Testing (Jest + Enzyme)	HW3(feature) due 10/6 6pm, Project sprint 1 begin (bi-weekly meetings with TAs at the end of sprint)
10/11, 13	Taming complexity; defensive programming; Design patterns	Django	HW4 out 10/19, HW3(testing) due 10/13 6pm
10/18, 20	Design patterns	Django + Python Testing	Project sprint 2 begin
10/25, 27	Design patterns; code refactoring	Integration + CI + SonarCloud	HW4 due 10/27 6pm

# Syllabus (subject to change)

11/1, 3	Operation	Design Pattern	Project sprint 3 begin
11/8, 10	Operation, Mid-presentation	Project Mid Presentation	
11/15, 17	ML pipeline	Deployment	Project sprint 4 begin
11/22, 24	ML pipeline	Code Refactoring + Optimization	
11/29, 12/1	Operation	Final Exam 11/30	Project sprint 5 begin
12/6, 8	TBD	Testing Session	
12/13, 15	TBD	Poster Session 12/14	
12/21			Project final report (due 12/21 6pm)

# Finale: Poster & Demo Session

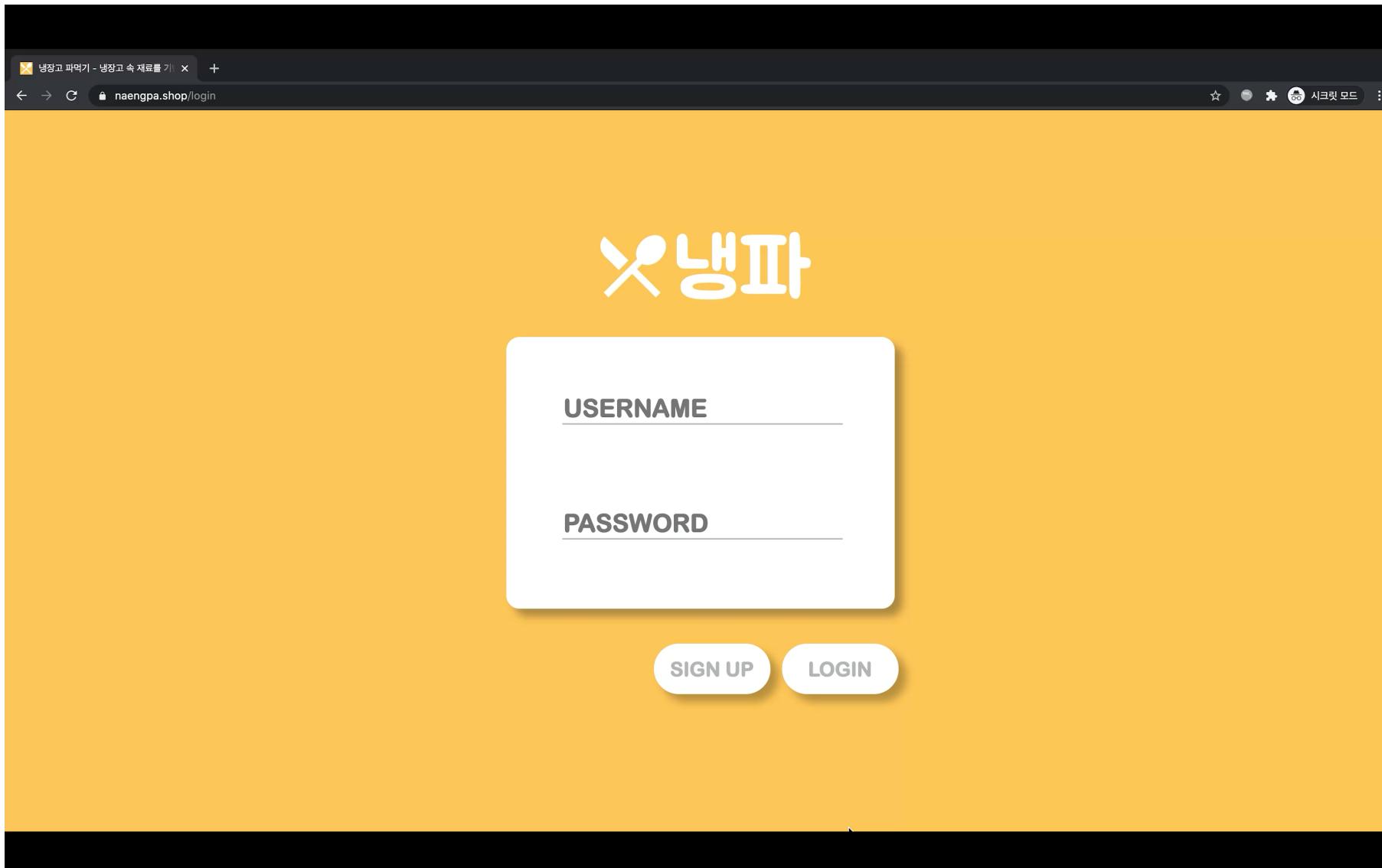
(12/14(Wednesday) 6-9:30pm)



# Main Project

- Theme – Build your own web services
  - Note. Not Mandatory. But many teams use ML features.  
**Good news: GPU resources provided by FriendliAI  
in case you train and deploy your model! ☺**
- Build new services you've dreamed about while learning software development principles and practices
  - Amazon, Google, Facebook, Ebay, Paypal, Uber, Snapchat, Airbnb, Twitter, LinkedIn, ...

# Project Example (2020)



# Project Example (2021)

**MetaBand**

Search Music... 

**Sign In** **Sign Up**

**벚꽃엔딩**  
버스커 버스커  
By (Guitar) (Bass) (Guitar)



**신호등**  
이무진  
By (Guitar) (Drum) (Piano)



벚꽃엔딩 - 버스커 버스커 

0:03 / 4:32

◀ ▶ ⏸

# Project Examples

- <https://github.com/swsnu/swppfall2021> 14 projects
- <https://github.com/swsnu/swppfall2020> 18 projects
- <https://github.com/swsnu/swppfall2019> 21 projects

# Main Project

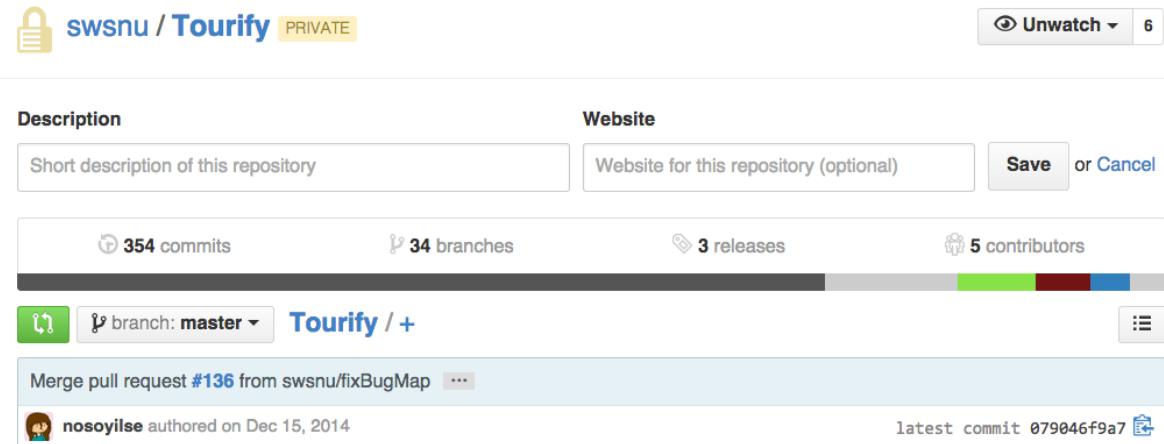
- Team formation & Project proposal
  - Sprint 1 – Spec (What do you want to build precisely?)
  - Sprint 2 – Development
  - Sprint 3 – Development
  - Sprint 4 – Development
  - Sprint 5 – Development
  - Poster/demo session (project report)
- \* Bi-weekly meetings (15-20 minutes per meeting) with TAs
- \* E.g., 18 teams (60 students) – team1-team6 TA1, team7-team12 TA2, team13-team18 TA3 (we don't know yet how many teams we have)

# Main Project

- Group: a team of **4** students
- Start forming teams early!  
We finalize project proposals on September 28. The earlier you submit, the better.  
We start team projects from October 3.
- Development environment
  - Backend: Python/Django
  - Frontend: Typescript/React.
  - IDE: vscode (other options - PyCharm, IntelliJ)
  - Of course, you can make use of the tools and libraries you need to make code better. We will learn new tools throughout the course.

# Main Project

- Agile software development process
- Git for version control
- Github for project management
  - Milestones
  - Issues
  - Pull requests
  - Code review



- **Testing** – unit tests/functional tests/integration tests

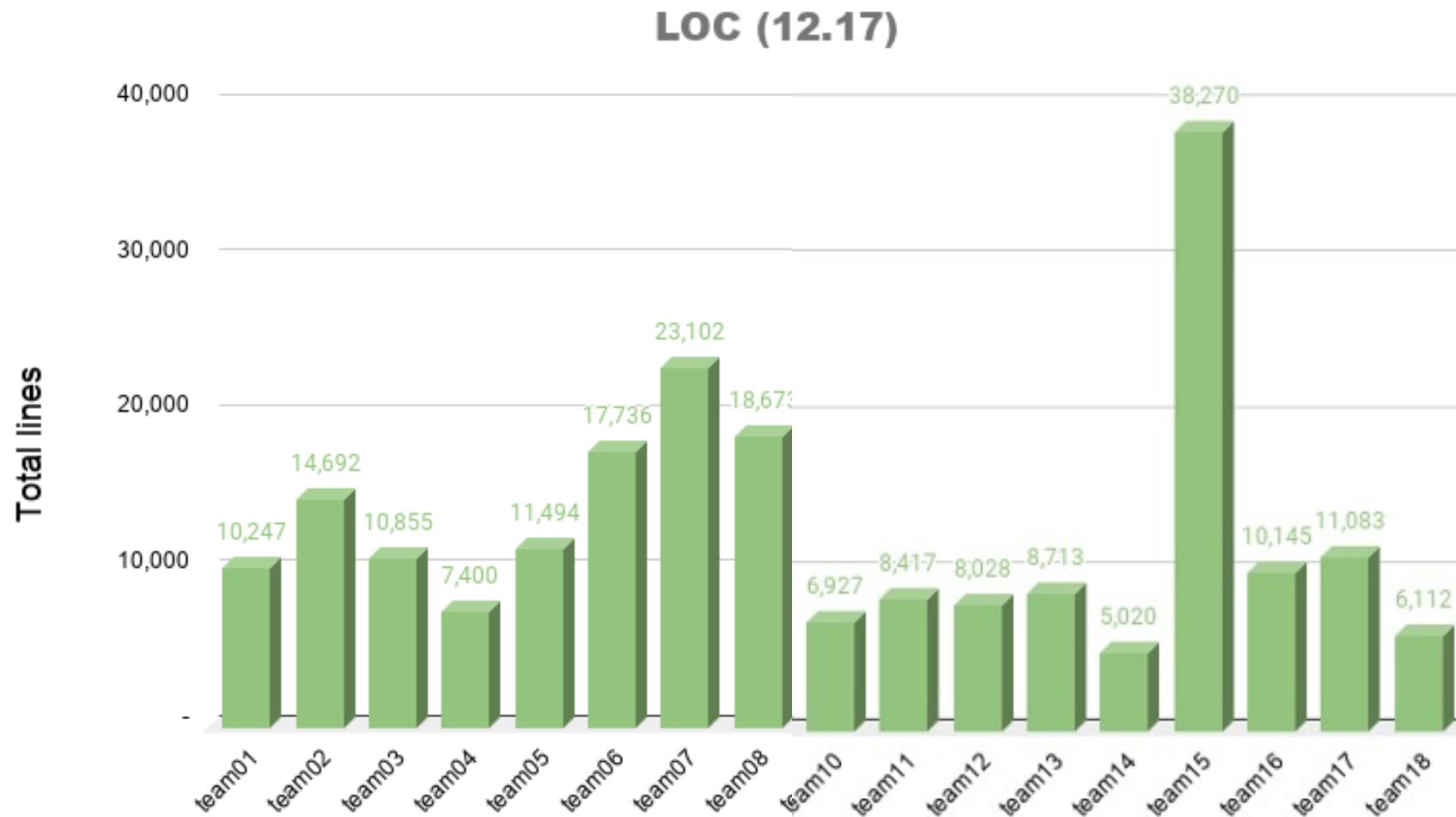
# Deployment

- The back-end of your project will run on a server in AWS EC2
- You will use web frontend/web backend/database (e.g., MySQL).
- You will test if your service runs reliably.
- You will optimize performance (e.g., caching and scaling) in the wild

# Project Evaluation Criteria

- Idea / functionalities
  - The service should depend on Machine Learning features.
- Correctness of functionalities
- Code quality
  - Clean, modular, ...
  - Testing
  - Exception/failure handling
- Load testing: performance, scaling
- Project process (e.g., team collaboration, sprints)
- Peer review ~~(free rider)~~ – you really have to design and write significant amount of code

# LoC (SWPP 2020)



# Pitfall

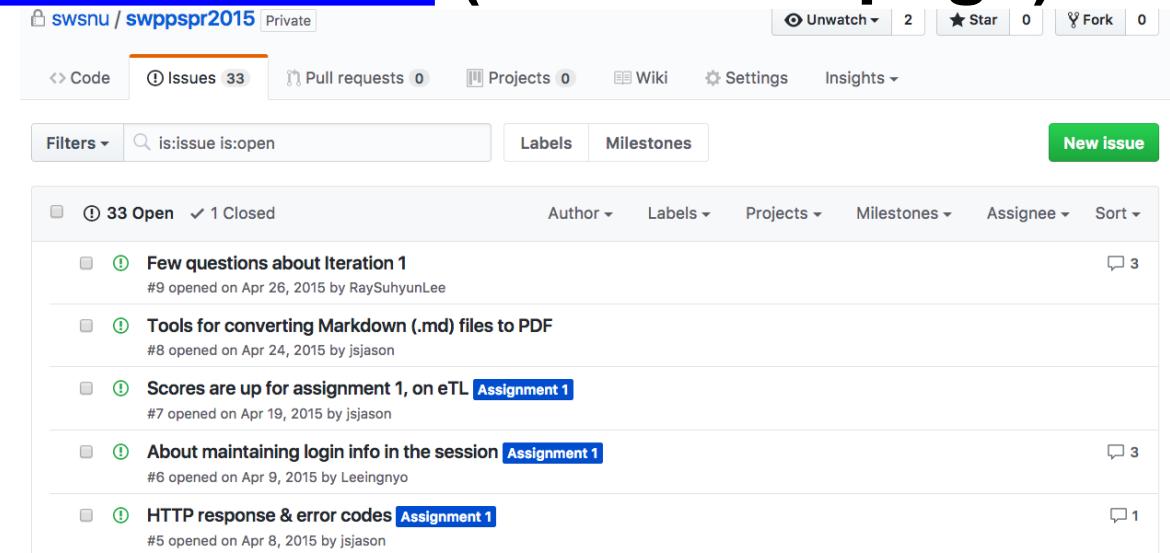
- Pitfall: Dividing work based on the software stack rather than on features -> Don't do this!
  - E.g., Front-end specialist, back-end specialist, customer-liaison, ...
- Better for each team member to deliver all aspects of a chosen story
- Better for app, better for your career growth
- If you divide work by features (i.e., each member works both on frontend and backend), you will get more credits!

# Warm-up Individual Homework

- HW1 – Python basics
- HW2 - Typescript basics
- HW3 – A front-end for a blogging service using React
- HW4 – A back-end service for a blogging service using Django

# Getting Help: ETL+Github+Email

- Class **ETL** for **slides** (**will be posted before classes; however, the slides may not have answers on some questions**)
- Course github site: schedule, homework, project
  - <https://github.com/swsnu/swppfall2022> (**Watch the page**)
- **Github issues for discussion**
- Office hours



# Coding Exams

- In-class coding/theory exam
  - ~ 4 hours
  - Open lecture notes
  - Tentative schedule: Nov. 30 practice session

# Homework/Project Timeliness

- Hard deadlines
- Catastrophic events
  - Major illness, death in family, ...
  - Consult your academic advisor to come up with a plan to get back on track
  - Consult with me about this class

# Cheating

- What is cheating?
  - Sharing code: by copying, retyping, looking at, or supplying a file
  - Coaching: helping your friend to write a programming assignment, line by line
  - Copying code from previous course or from elsewhere in the Internet
  - Especially, be careful about copying code since we open your project code! Be alert about code licenses.
- Penalty for cheating
  - 학사위원회
  - Penalty to the students who provide information as well
  - Level1. Grade penalty (e.g., F)  
Level2. Suspension  
Level3. Expel from the school
- **We run software plagiarism detection tools. => There were unfortunate cases happened two years ago.**

# Attendance Check

- E-Attendance check
- By the rule of the school, if you miss classes more than a specific number, you get F. There is nothing I can help.

# Other Rules of the Lecture Hall

- Laptops: closed (for offline lectures)
  - Exceptions: when we actually try things together during the class
- Electronic communications: forbidden
  - No email, SMS, instant messaging (Kakao, etc.), calls, etc.
- Presence in lectures
- **No recordings of ANY KIND:** no video, audio recording allowed
- **No auditing allowed**
- **Participate (brownie points :)**

# Announcement

- If you do not plan to take this course, withdraw ASAP.  
We have team projects!
- Yay, HW!
- **HW 1** out today! HW1 due 9/9(Fri) 6pm.
- **HW 2** out on 9/7. Due 9/16(Fri) 6pm.
- Team formation due 9/13 (T) 6pm
  - send your team information to [swpp.22.staff@spl.snu.ac.kr](mailto:swpp.22.staff@spl.snu.ac.kr)
  - Name, student id, email address, github id

# HW1 – Python Basics

- Individual assignment
- Due: 9/9(Fri) 6:00pm

<https://github.com/swsnu/swppfall2022/hw1>

# 실습 관련 공지

- 수요일(9/7) 소개원실 실습에 앞서 원활한 수업 진행을 위해 다음 사항들에 대해 미리 준비를 하고 수업에 참여해주시길 부탁드립니다.
- Docker  
실습과 과제는 조교들이 공지하는 도커 컨테이너 환경을 기준으로 진행됩니다. Ubuntu 18.04.6 LTS 혹은 MacOS를 사용하셔도 괜찮지만 도커 환경을 갖추고 오시길 강력하게 권장드립니다. 설치는 아래 링크 또는 인터넷 검색을 통해 참고하시길 바랍니다.
  - For Mac: <https://docs.docker.com/desktop/mac/install/>
  - For Windows: <https://docs.docker.com/desktop/windows/install/>
- Visual Studio Code  
각자 편한 IDE를 활용하셔도 무방하지만 VSCode 사용을 권장드립니다.
  - for download VSCode: <https://code.visualstudio.com/download>
- 앞으로 대부분 공지사항들은 Github issue board에 announcement label이 붙어서 안내될 예정이므로 [swppfall2022 repo](#) 상단의 “Watch” 버튼을 클릭하여 알림을 받으시길 바랍니다.

# 질의응답 관련 공지

- Github issue board 활용 관련 공지입니다.
- Issue Board를 통해서 질문이나, discussion, 팀원 모집 등의 issue를 남기시면 됩니다. 관련해서 몇가지 유의사항을 안내드립니다:
- Issue board에 질문을 남기고 나서 해결 되었을 때 issue를 close하지 말고 다른 학생들도 해당 issue를 참조 할 수 있도록 해주시기 바랍니다
- Issue에 대해서 답글을 남겨주거나 다양한 tip들을 남기는 학생분들께는 extra point가 주어질 예정이니 적극적으로 issue 논의에 참여하시길 바랍니다.
- 팀원을 모집하는 issue는 팀원 모집이 완료되면 close 해주시길 바랍니다.

# About this Course: Previous Course Takers Said...

회사에 갈법한 개발 경험을 해볼 수 있어서 참 좋았습니다. 수업 내용도 좋았던 것 같습니다.

교수님과 조교님들이 열심히 준비했다는 것이 흠뻑 묻어나는 수업이었습니다.

실제 개발 체험, 확실히 배우는 것이 많았습니다.

교수님이 정말 열심히 준비를 하신다는 게 느껴지고, 피드백도 빠르고 좋았습니다. 코딩도 굉장히 많이 하게되어, 이번 학기동안 제 스스로의 프로그래밍 실력이 step function으로 상승한 것 같습니다.

매 순간 직면하는 신체적/정신적 한계로 인해 ..... 점을 빼면, 정말 배운 것이 많은 수업이었고, 즐거웠습니다.

I got offers from major companies thanks to this class. Thank you!  
This class is worthy of 10 units, not 4 units!  
Enjoy the class. If you cannot, avoid.

# Last Reminder

- This course is very, very challenging! I warned you. ☺
  - **Brace yourself. If you are not ready, it's probably a good idea not to take this class.**
  - **Dropping this class is allowed only in September before you start your team project.**
- This semester the class is more challenging to the staff and students since we have more students enrolled.
- Participate! Be active! If you don't, the challenging class will become more difficult.

Welcome!  
We will have lots of fun!