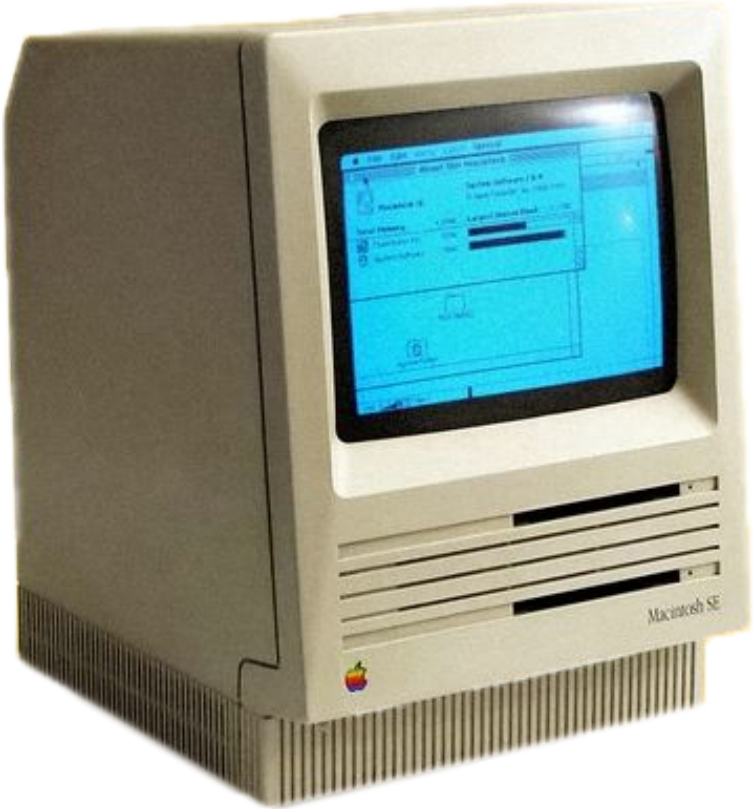
Past, Present, and Future of Software Engineering

Stefan Bukarica, Samson Goodenough, Chen Jin, Jordan Dubrick

190563930, 190723380, 170631720, 201859360

What is Software Engineering?



Introduction

Software Engineering is the application of engineering principles to create and develop software products for computer systems

Almost every person, business, and service relies on quality software in order to conduct many tasks

We want to identify trends in the industry in order to be able to project what the future holds in software engineering

Past of Software Engineering

Advent of computing

Advancements in the computing field ramps up in the mid 20th century.

Cold War is a catalyst for major developments in technology.

Margaret Hamilton

Evidence points to the term first being coined in around 1963-64 by American computer scientist and NASA engineer Margaret Hamilton during her time working on the SAGE project.

SAGE

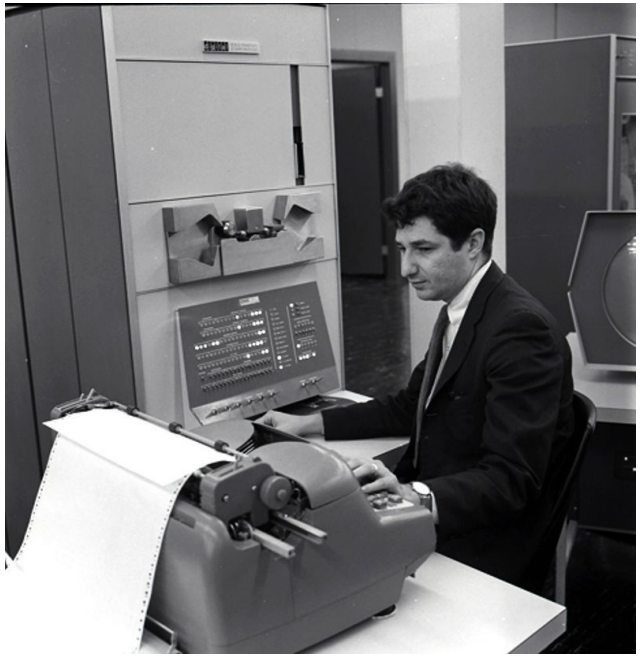
Largest computer ever built.

Produced Images of airspace based on radar data.

500,000 assembly statements

The SAGE project's sheer scale not only brought legitimacy to Software engineering and demonstrated its usefulness, but also laid the foundations for every major Software endeavor after it.

Software Crisis



The 1960s were a time of innovation in the field of software engineering, but these rapid advancements also brought with them a slew of problems.

Software crisis: 1965 - 1985

Tech advancing was faster than engineers could keep up.

New challenges, high demand, high complexity

Major software failures

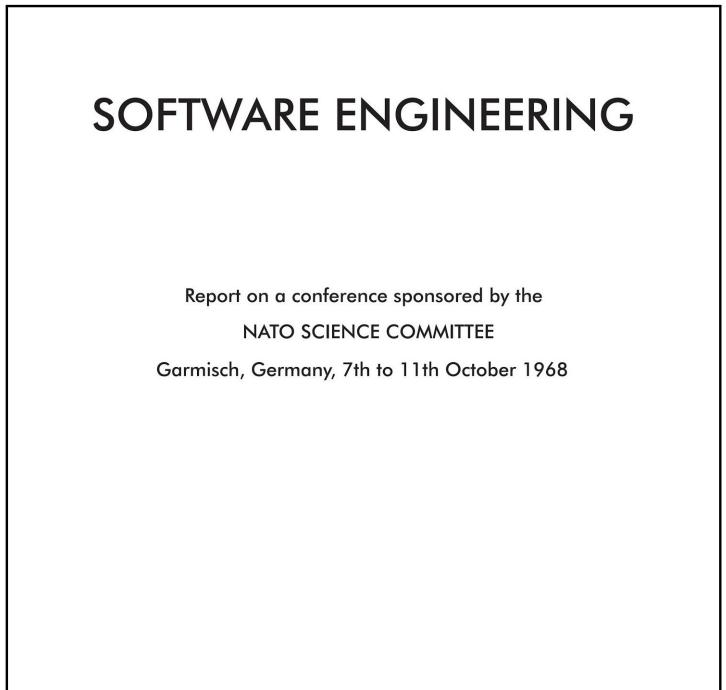
NATO conferences 1968-1969

The NATO conferences looked to address some major issues of the Software Engineering Crisis.

Resulted in the creation of two landmark documents.

“Software Engineering” and “Software Engineering Techniques”.

Design, production, service, specification, software quality, software flexibility.



SOFTWARE ENGINEERING

Report on a conference sponsored by the

NATO SCIENCE COMMITTEE

Garmisch, Germany, 7th to 11th October 1968

Golden Age

1970s Software Engineering grows faster than any occupation in history, leading to the birth of subfields within software engineering that we are familiar with today.

Middleware

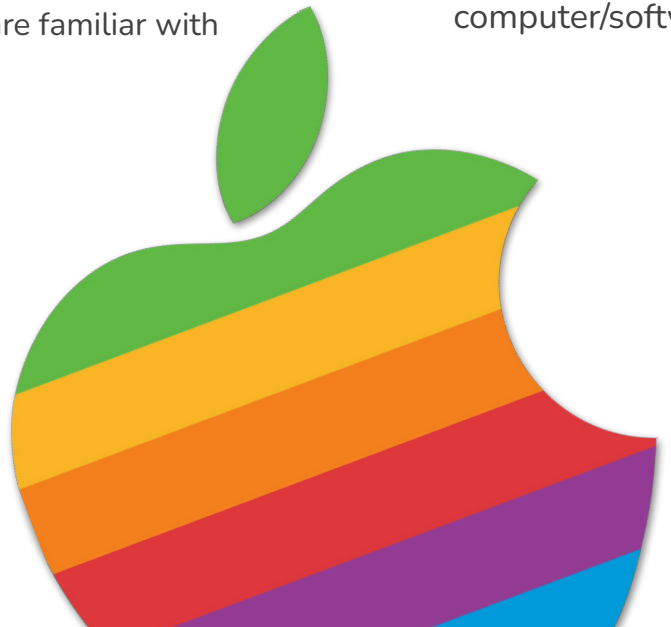
Embedded software

Databases

Security

C Programming Language

Rise of hundreds of commercial computer/software companies



Golden Age Cont

Many software engineering standards, structure, and strategies pioneered in the 1970s

1980s brings with it the rise of personal computers, results in software boom as commercial software and operating systems skyrocket. Software and computers become mainstream.

Object Oriented Programming

Structured Programming

Advent of the Internet

The internet brings a vast array of new global markets, further propelling the software field.

Newfound focus on security, privacy, and maintenance. Aging software shifts the field of Software Engineering as many jobs become centered around maintaining software.

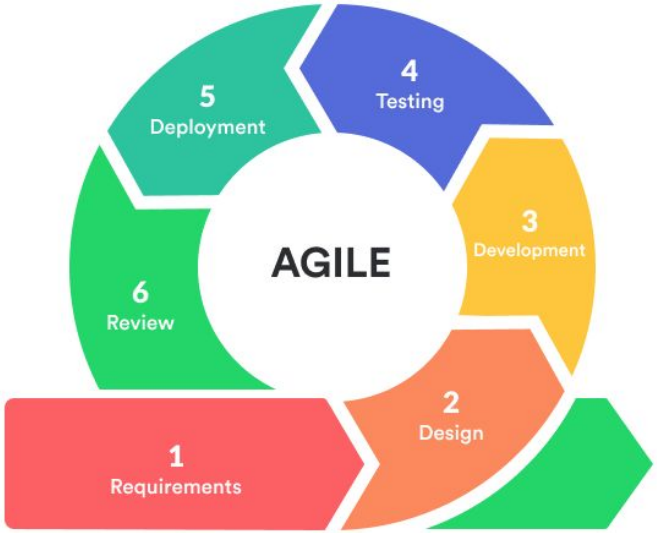
Demand for software engineers skyrocket as entrepreneurs take advantage of this new platform.

Present Day

Agile Software Development

79.4% of Software Engineering Education studies are mentioning this practice in some way.

Noticing increase in educators implementing strategies such as Agile, Kanban, and Scrum into their curriculum to further educate future software engineers.



Informality

As noted by the popular software engineering forum Stack Overflow in their 2020 survey, only 9.7% of respondents noted that they believed higher education was critically important to become a software engineer.

How much of this formal education could be learned by other means?

Google

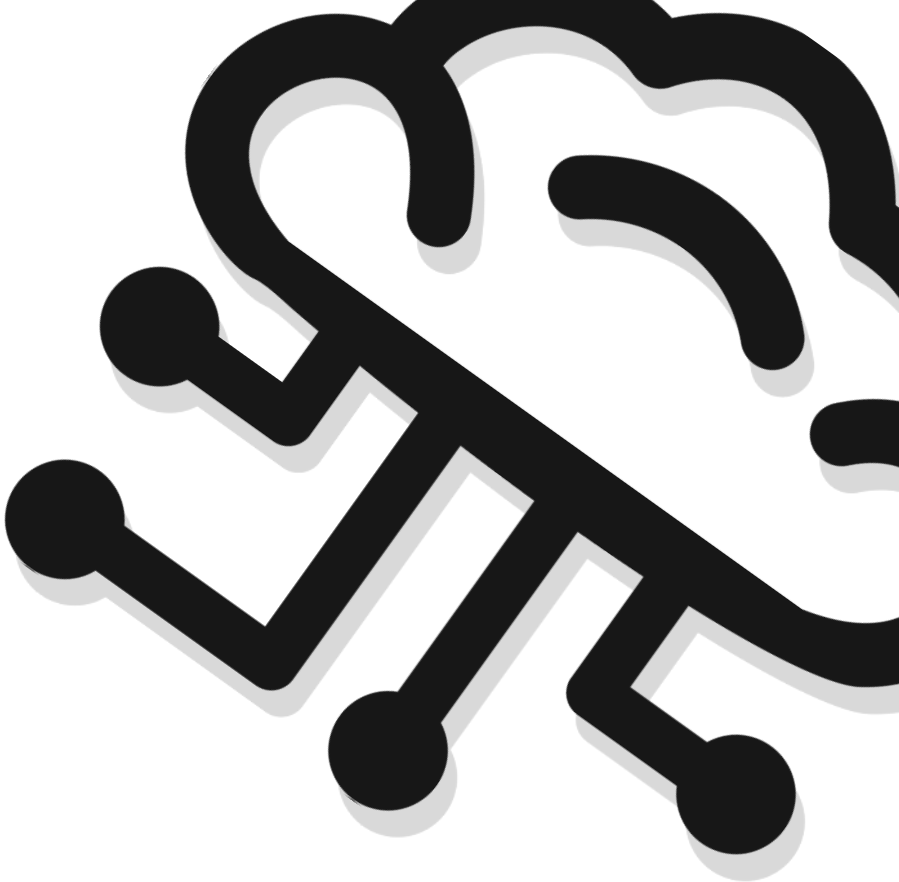
Apple

☒ Formal education

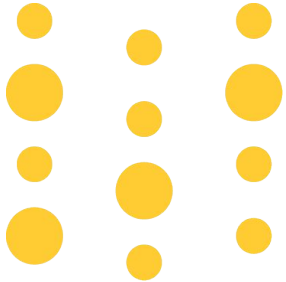
Artificial Intelligence

Current climate of software engineering revolves around a data-centric approach to many tasks

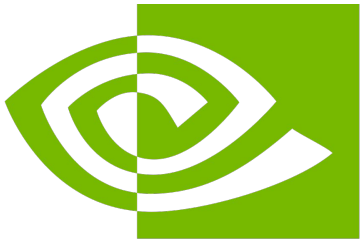
Sourcing mass amounts of data in any field has allowed AI to advance and continue to do so within the past 5 years, and foreseeable future



OpenAI



WandB



NVIDIA

GitHub Copilot

Auto completing code just by context or by a description, GPT generates code that has never been written before.

Revolutionizing coding with AI, allowing software engineers to increase the amount of work done daily.

Available to *anybody*.

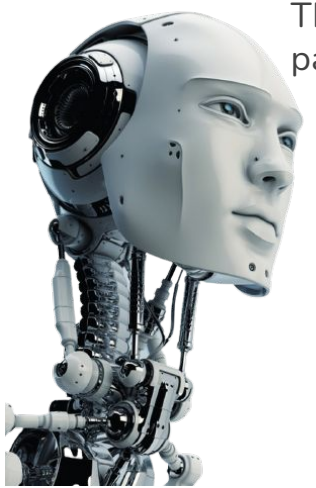
Future of Software Engineering

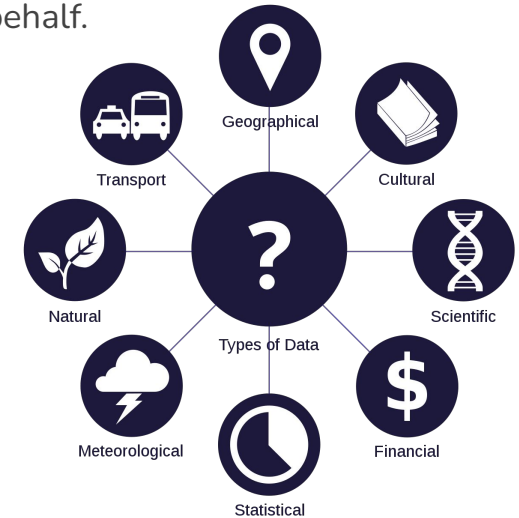
Near Future

With the continuation of data-centric development and the advancement of AI, we expect this to be the main focus of software engineers for the near future.

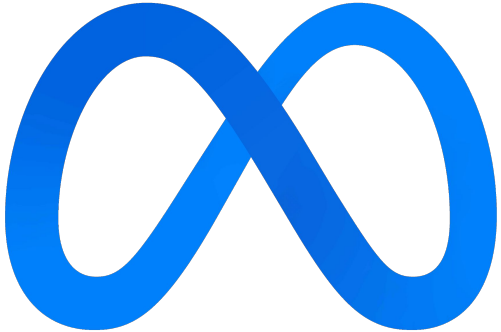
Since most software products require vast systems in order to operate, the expansion of complexity will continue with engineers working across many platforms.

The improvements of AI will result in engineers being able to ease some of the low-level parts of the life cycle such as teaching AI how to program on their behalf.





Further Ahead



Looking ahead, there are multiple emerging technological expansions that will change the life of software engineers

These include:

- Quantum Computing
- Augmented Reality
- Virtual Reality

We expect these things to improve turnaround time of projects and make large projects more feasible, software engineers work will be more abstract and less time consuming

Thank you

Any Questions?