

2. Lesen sie "A Brief History of Software Engineering" und fassen sie die wichtigsten Punkte zusammen:

- Seit einer NATO Konferenz 1968 versucht man bessere Methoden zu finden um komplexe Systeme zu verbessern. Das hätte durch automatisierte Programmdokumentation, analytische Verifikation usw. geschehen sollen. Dieser Fall ist nicht eingetreten, testen ist immer noch erforderlich.
- Der rasende Anstieg an Hardwareleistung führt dazu, dass die Programmierqualität leidet, weil der Profit im Vordergrund steht. Unsere Limitationen bestehen inzwischen nicht mehr aus schwacher Hardware sondern aus unseren geistigen Limitationen.
- Viele Entwicklungen die man Heute als neu Verkauft haben bereits in den 1950er 1960er Jahren existiert. Zunächst haben erste Forschungseinrichtungen und Universitäten Zugang zu Computern gehabt, später waren sie auch für Unternehmen unerlässlich. Allerdings gab es noch keine "Interaktivität" zwischen Mensch und Computer.
- Die ersten Computersprachen haben sich entwickelt, zunächst Fortran später auch Cobol.
- Ab 1963 erschienen erste "time-sharing" Systeme die die Interaktivität brachten, die den "Batch-Processing" Systemen fehlte. Allerdings waren diese Systeme so viel schwieriger als jeder glaubte. Diese Probleme riefen die NATO auf den Plan, sie sponorte eine Konferenz um diese Probleme anzugehen. Man erkannte das die damals aktuellen Techniken der Softwareentwicklung nicht ausreichten und das neue Methoden entwickelt werden mussten.
- Dijkstra und Hoare haben das "Strukturierte Programmieren" und "Datenstrukturen" maßgeblich mit beeinflusst und 1966 mit Semaphoren und dem "CSP" Konzepte aufgezeigt um konkurrierende Prozesse zu behandeln
- Eine andere "Bewegung" führte zu der Entwicklung von Unix und C. C sollte zur Entwicklung von Applikationen für Unix dienen, trug aber nicht den Geist des "Strukturierten Programmieren" mit sich.
- C's einziger Vorteil war ein "bischen Syntax", allerdings konnte man die Regeln in C leicht brechen, was viele Programmierer wertschätzten. Dafür sind diese Systeme Fehleranfällig und Teuer in der Wartung.
- Der Durchbruch für Microcomputer im Bezug auf Softwareentwicklung im Jahre 1975, began als ein Texteditor mit "file-system" und "debugger" mit Pascal entwickelt wurden.
- Dijkstra erwähnte oft: "Testen kann die Anwesenheit von Fehlern zeigen, aber niemals deren Abwesenheit."
- Die Entwicklung der Modularisierung half bei der Konstruktion von System die eine große Zahl an Menschen erforderte.
- Die Personal Workstation "Alto" war ein weiterer Meilenstein.
- Als "Heilmittel" gegen die Komplexität wurde das Objekt Orientierte Programmieren" angeboten.
- Die Leistung der Computer steigerte sich immer weiter und erste Open Source Software wurde geschaffen.
- Leider wird die Software langsamer schneller als die Hardware schneller wird.
- Der Autor schlussfolgert das Komplexität überall in der Softwareentwicklung möglichst bekämpft werden muss. Qualität in der Programmierung reduziert den Aufwand in der Wartung enorm. Der Autor kritisiert auch das die akademisch Gebildeten einfach die Methoden und Werkzeuge der Industrie übernommen hat ohne zu hinterfragen oder zu kritisieren.