

Contents

BECOMING SOFTWARE

Some boundaries that appear to be fuzzy when we try to define them (appeared wed. // thurs.): - fuzzy line between data and software - fuzzy line between software and hardware - tolerance to noise, software being less tolerant to changes. That was something that could define software from data. - fuzzy line between software and signal (signal vs. noise) - machine learning algorithm and the training data, adaptability - developer/ - forking (when does one software become another software?) - Is a social network a piece of software? - FPGAs and Hardware Definition Languages

(FRIDAY) - labor (is training an ML machine labor? “the reCATCHPA insight”) - process (Agile practices) “using” your users to define the requirements of the product (features in the product become defined by the feedback of the release cycle) - services and agile processes collapsing the divide between production and use - Agile methodologies collapse product, labor and process. - there is also the pain and labor that users bring to software that is rarely documented (e.g., activists using Signal) - comment based on a study of floss chat tools (secure messaging?) - thin / fat client software, where does it live? - poc.gtf pdf is an executable is an image / alchemistowl.org/pocorgtfo

Applying agile: agile refers to a set of practices, a number of which needs to be adopted to qualify as an agile process.

Political Economy of Software Engineering - software engineering is an expensive process - Tom DeMarco: why does software cost so much? It mostly has to do with process. - study of linux kernel: how much did it cost? compensated through economies of scale - web development changed the landscape - more people joined the bandwagon: ex-yugoslavia a place where development gets outsourced to, also Russia and India (all hackers are Russian now, too) - when you establish a relationship of outsourcing

- objective is not to define what is software,
 - describe the how of software becoming
 - discussing “what is software” is productive, worth elaborating writing software does not come with the kind of reflection that this discussion provides
- it seemed obvious what is software, but it became more difficult
 - interest in corner cases: popular encryption algorithms, AES: you can write a program implementing it: software. but now it is embedded in the chip, hardware.
 - code sequences in pdf: nobody would consider pdf to be software
 - postscript can also do this

- proof of concept (gtfo) you can read it and execute it (peter will add a link)
 - burnt out ruby developer: he retired all the software and manuals. he published a file with a printing queue,
 - corner cases: narrow down the whole thing, we cannot come up with a definition of software, we can think about to come up with a feature set -if these features are fulfilled it can be considered software
 - yesterday: fpga: reconfigurable hardware Jean: what is software? he said, i can only tell you what is good software. qualifiers that can help narrow down the path: queer software
- methodology: we can apply extreme programming as a way to write together (XP ==> programming in pairs or doing extensive code review, unit testing of all code) take a topic, ask people and document in twos
 - yesterday we had these machines with different use, with the punch cards looking at how they were used in business use mixed with what martino and anita showed: binaries as sets of instructions could we develop other algorithms with these machines do an addition or hangman on the tabulation machine what could be this sequence with the operator that handles the cards user, developer, operator, tester, sufferer, translator, technical writer, marketer, manager, product owner, devops etc.
 - software engineering practices are spreading -> a managerial culture that is poisoning other practices. . . it has effected start-up culture and how capital gets invested the individual genius: the falsehood around the garage kids the idea with rushing to the market: minimum viable product. pervasive today grasshopper: data flow design tool architects are using it and are actually building software urban systems: generative system, they work mostly in china software with sliders, their entire design process is building software so that you can design cities but moving the sliders the idea of a star architect, discussed in media, there is a much less glamorous process of building hospitals, staidums configurable and modular cities
 - architects see themselves as the original system architects and design BIM: every toilette is set up, the idea that you can have a process of when the toilette was built and fixed and replaced. the small guys are doing this
 - roles and labor relations very interesting
 - a joke: bastardization of the agile method as a self-help method
 - * i want to understand how software is made and services etc.
 - * seeing it also as a vessel for having fun it is expensive, industry, investment maybe there is a way to misuse it make little steps that makes it funny, stupid or real relating also to what silvio was presenting there is a link between the method and the output

that we see as software in which way? i want to figure it out
more... everybody is a start-up: bad metaphors

cargocult: by richard feynman
rituals that people look at
colonialist practice?
in polynesia: pacific islands that were occupied by americans
the planes used to bring cargo, wealth
so they built an airport out of wood and stuff that was laying around hoping
that the planes would come back by replicating the form hoping that things will
materialize people being occupied, with no relationship with them that's maybe
what you are referring to a metaphor in computing: cargocult i don't know if this
actually happened this came after feynman published an article that was called
cargocult science just by following the formal aspects of scientific method, people
expect their results to be scientific i am not sure if it happened due to something
that happened in polynesia or because of feynman's storytelling burning man:
cargocult

- why did you think about cargo cult? somehow as an image: interpreting
agile practices as something different misusing of it taking at face value
getting blind to some of the metaphors

doing it a bit more fun and playful
this could be misused as a methodology to understand agile processes
stripping the agile methodology some of its aspects and putting some of it back
some of those words: if you look at them
scrum i don't know well but it refers to rugby

idea: to look at the origins of terms in agile methods, like scrum
take a scrum metaphor and insert yoga into it what happens?
agile scrum yoga
agile yoga is a thing

- what this guide would be if it is about providing people with guidance on
how to observe data vs. software vs. hardware to be interesting
guidance expand upon making these distinctions

some of the programs we are having in making a distinction between software and even
using information system
if you are using several pieces of software
then the things they do between using different software becomes the most important
one way you can understand this is seeing software as automating practice
and look at some workflows of people using software
if data and software is a thing, isn't it reasonable to map this out as an information
data coming, breaking down the function

- cargocult: mistakes, can you see something where you don't see the infrastructure but just the gesture, people on the radio are talking with someone else saying we don't have any more milk, drop us some i am unhappy about the term: the balance of power the occupied and the occupier the occupier made a mistake cargocult you don;t understand the infrastrucutre and try to make sense it is great mistakes
- unrelated relationship between observation and evaluation from what i gather: evaluation is a core element of the agile method in order to be able to review and adapt to the moving target this observation and evaluation: evaluation is not only observation, it is something plus some criteria to have some criteria: in the sense of the cargocult, what they replicate is what they see, an image

what are these things:

framing device: data vs. software, the roles (a taxonomy device) – relates to identifying obvious and less obvious forms of labor cargocult: method of observation

META THINKING:

asking what is software as a method?

terms relevant to describing software: computing programming developing programming languages algorithms data. . . development processes roles architecture protocols

what are the economic forces acting upon software? how do you find out?

CARGOCULT PROPS: - coconut shells - cardboard laptops - candy wrappers - chewin gum as stickies - code reviews

A qualifier like “good”, “bad”, “spy”, “queer”, “proletarian”, “bourgeoisie” can help narrow down definitions.

What is not software?