PHILIPS

The agile way to add Sec, Dev & Ops into Waterfall

Xander Heemskerk
Philips Product Security P
May 16th 2023, Okura Amsterdam

innovation you

Subjects

- ❑The Philips context
- ❑Digital transformation in Healthcare
- ❑The Philips solution creation
- ❑Regulatory context
- ❑The adversary /Hacker context
- ❑Agile way to add sec to DevOps attempts
- ❑Agile security
- ❑Open Issues

Who is this?
and what does he do?

- Xander Heemskerk
- >30 years in IT “grew up through the ranks”
- >20 years in It security
- Advanced development from architecture evolved into Cyber Security role

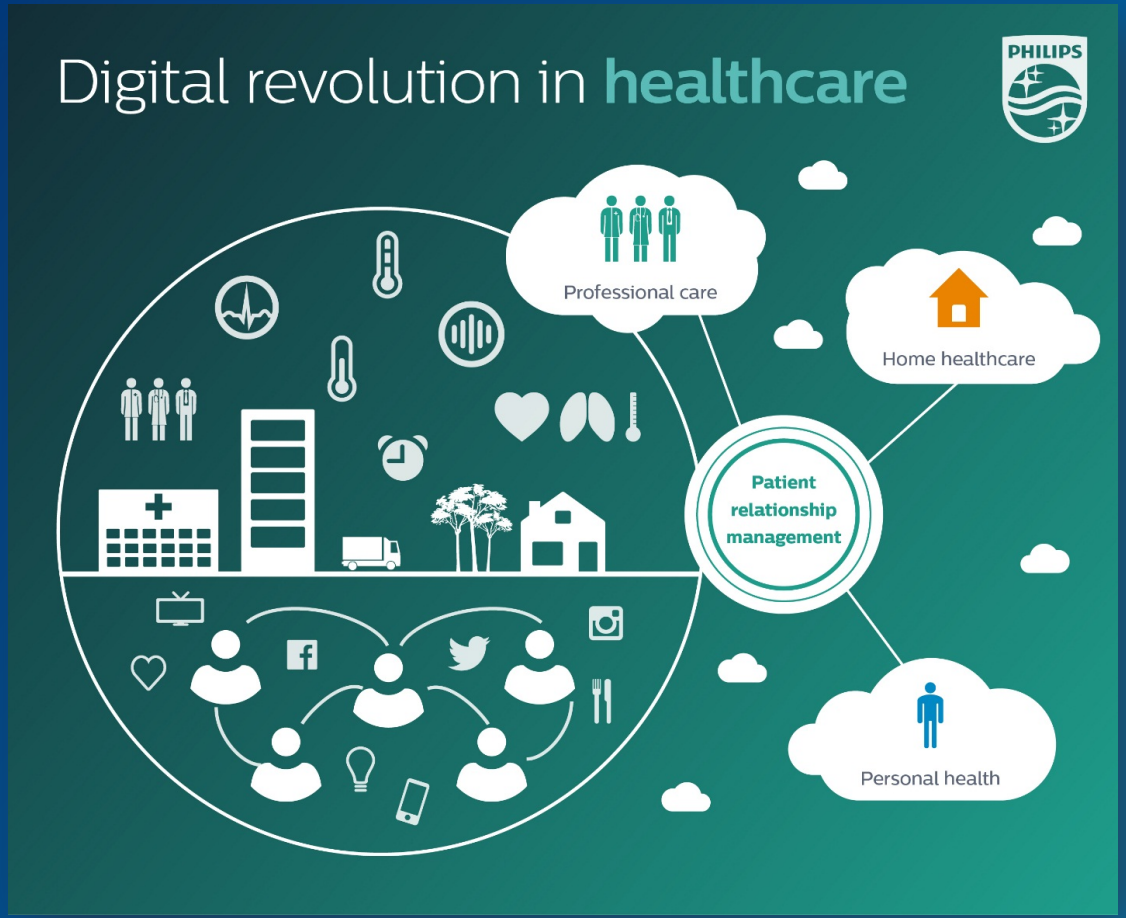
- Role in Philips
  - Product Security
  - Governing security is appropriately addressed in the products and service that are sold (distributed)
  - Not about driving agility

My opinions are my own and do not necessarily reflect the views of any of my current or past employers

Philips

- 132 year old company (yesterday)
- Used to be a (the) real Dutch Conglomerate
- Culturally originates from the Physical product development
  - TV, Lighting, Washing machines, Coffee, Cookers, Chips and Chips machines
  - The basis of many technology companies in The Netherlands

- Now Medical device company
- Big Iron like MR scanners and Image Guide Therapy
- Wellness like pregnancy apps and connected toothbrushes and shavers
- 80K people

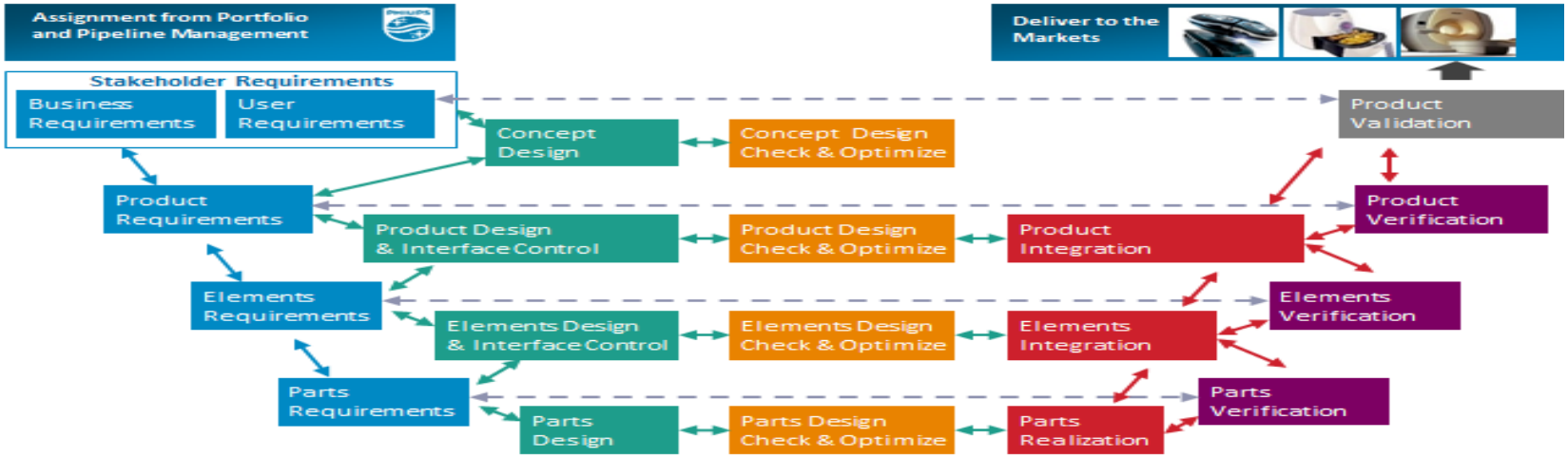


- Digital transformation
- Increasing complexity
- Interconnectivity
- IOT
- Big data
- AI
- Cloud infrastructure
- Meshed Cloud sub-solutions
- Collaborations
  - B2B
  - B2C
  - B2B2C
  - B2G
- Complex Supply Chain
  - Open Source
  - COTS
  - OEM
  - ODM
  - Finished good manufacturer

The Philips solution creation context

- Medical device Companies must have and follow a Quality Management System (QMS)
  - FDA
  - CFDA
  - EU MDR
  - Must be able to verify the right things are done.

- V-Model is a standard for medical devices



The Philips solution creation context

- QMS contains detail definitions of processes that must be followed
  - Must have formal specifications of the (secure) requirements
  - Must be validated at the end are implemented according to specs.
  - FDA can come and audit that the QMS is followed
  - Design History File is the basis of the audit
- Product Life Cycle fully described
  - Pre market
  - Post market
- Waterfall cycles can short and incremental for software
- This is not cost effective (possible) for hardware (IOT)

The Philips development context consequences for Agility

- Globally distributed development
  - Everybody in his own timezone
  - Optimized for cost
  - US &| Netherlands designed
  - India Software
  - Hardware IOT China
  - Also Israel, Brazil, etc.
  - Conway's law for agility ?

The Philips development context consequences for Agility

- Self organized teams
  - –complexity in alignment
  - –Self organized is not the same as self approving
  - –For quality and compliance governance from outside the team is still needed

Our team from WC Duck advises to use.....

Increasing regulatory obligations

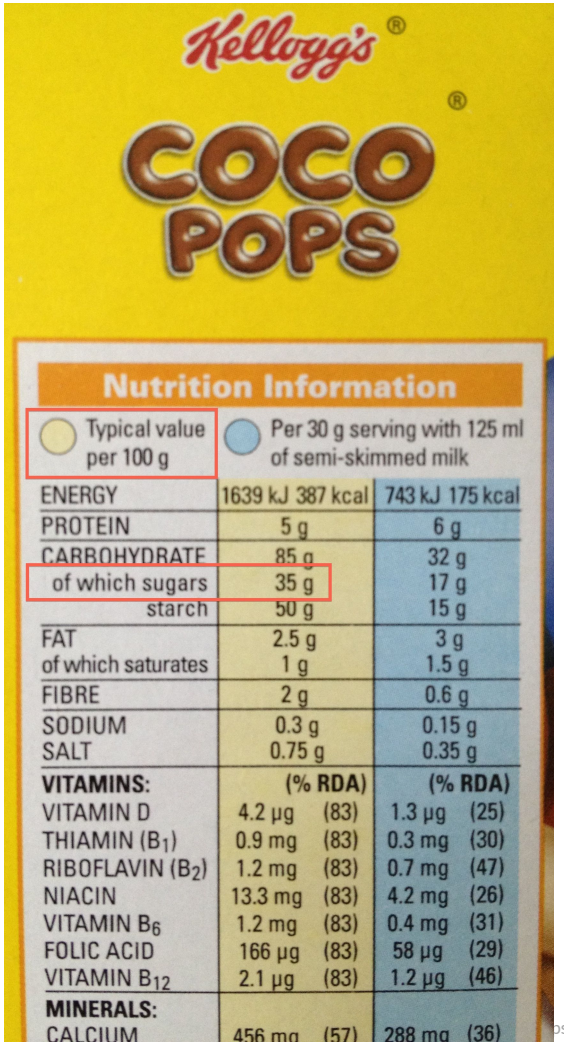
Summary of impact of the ***increasing*** cybersecurity regulations in US and EU:

- Security by design processes need to be demonstrated to customers and regulators
- Stringent conformity assessments (3rd-party certification for critical products and services)
- Ship hold monitoring (not allowed to place products on the market with exploitable vulnerabilities)
- Disclosure of security relevant information such as Software Bill of Materials
- Security monitoring and timely patching required (e.g., CRA = 5 years free of charge)
- Reporting of incidents and vulnerabilities occurring in the enterprise and products (<24 h)
- Denied market access and fines, e.g., 2.5% of the world-wide turnover (EU-NIS2)
- Leadership personally accountable for failing cybersecurity (EU-NIS2)

China, Japan & UK have and are extending regulation too.

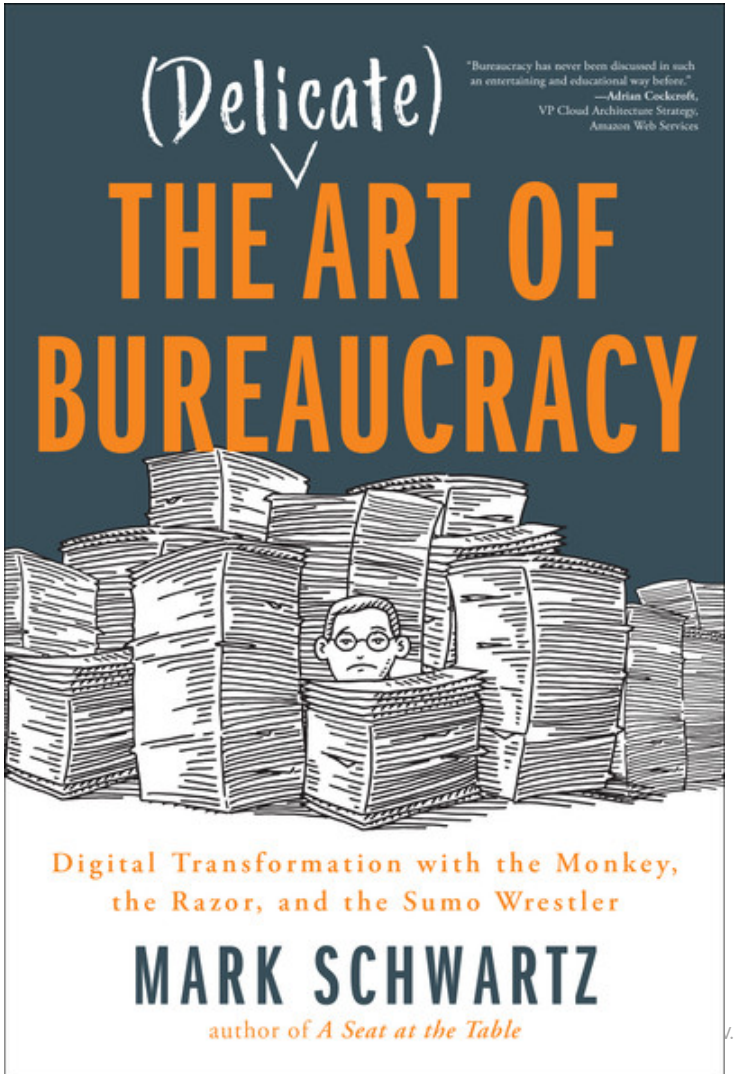
The Regulatory context consequences for Agility

- Joe Biden's Software Bill of Material (SBOM) executive order
  - –Solution needs to know and administer and present "what is in the product"
  - –Download an opensource library that solves an issue has strings attached….
  - –Applies to the complete supply chain
    - •Suppliers using open source have the same strings attached
    - •The string they need to present to us their customer.



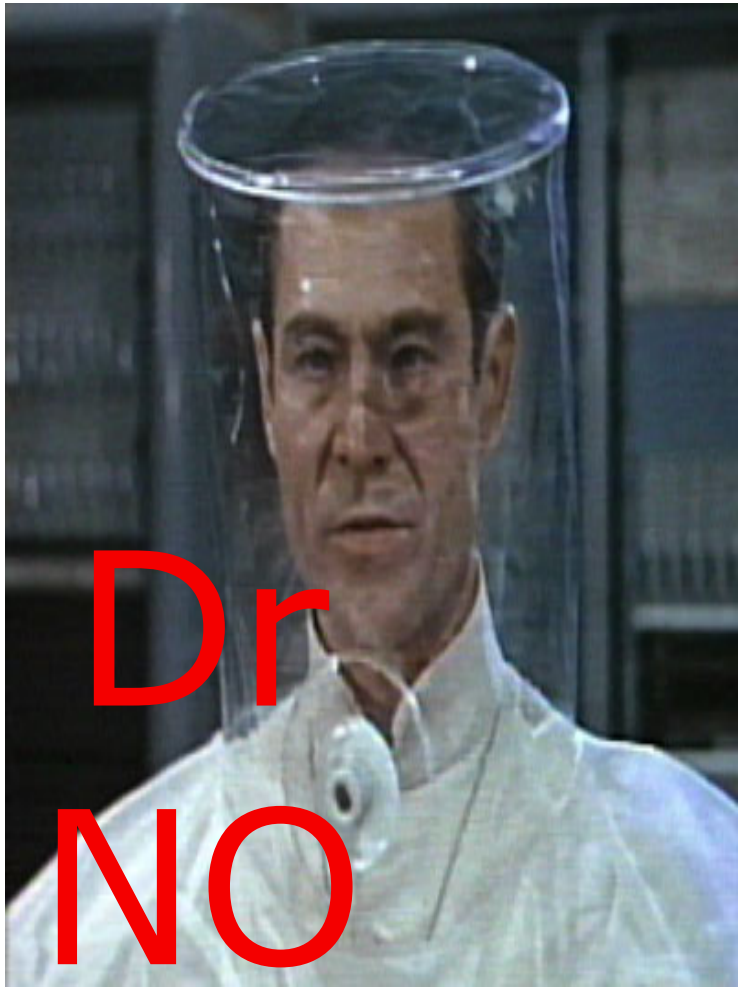
The Regulatory context consequences for Agility

- Regulator wants **evidence** SDLC processes are followed to prove Secure design
  - –Documents are the means of communication.
  - –Threat modelling
  - –Static code analysis
  - –Security testing
  - –Risk Management



The Regulatory context consequences for Agility

- Ship hold monitoring zero exploitable vulnerabilities
  - –Risk management negotiating power in getting rules applies and still release products
  - –Zero exploitables vulnerabilities reduces the possibility of risk management
  - –Agility reduction



The adversary /Hacker context

- Hackers are extremely agile
- Just need to find one hole to attack
- Can stop an attack angle and start a new approach

- As opposed to defending the complete inside/out surface
  - Process
  - People
  - Technology
  - Along the complete supply chain of a solution
- To defend the mindset of the hacker is needed

- Risks come not just from the bad guys
  - Security investigators help to find issue
  - But disclosures could still lead to reputation damage.
  - Coordinated Vulnerability Disclosures (CVD) should resolve that

Agile way to add sec to DevOps attempts

- Adding automated security checks CI/CD
  - Automated static code analysis code scanning
  - Automated SBOM library scanning
- DevSecOps security testing
  - Inline DevSecOps staff that does penetration testing
    - Part of the solution creation team
  - Standard scripts with the human touch.
- Still need independent testing from regulatory approval
  - For the independent view for the “hacker spirit”
  - Minimizing the test scope by having standard test automated

- Self organizing teams are not self approving teams.
  - Governance is still needed

Agile way to add sec to DevOps attempts

- Risk Based approach
  - –Regulatory rules have strict description
  - –Almost always with a risk-based element
  - –Risk context is the ‘agility lubricant’
  - –Risk acceptance by business leaders’ way to close arguments
  - –Safety consequences for medical devices

  - –Except Cyber Risk Management not helped by standard risk quantification:
    - •adversary work 100% on 1% probability of exploit

Security requires Agility

- The adversary/hacker is agile therefore defense must have agility too
- Solution teams following a QMS don’t like requirements to change
- But due to vulnerabilities “popping up” out of the blue the security requirements are not stable (agile?)
- This requires DevOps teams that know the solution and must be able to instantly switch to vulnerability analysis and patch management
  - Expectation management must be done with business management that have delivery date expectations
  - Specially when consumer IOT hardware is involved business management is not used to that agility.
  - Alternative is having people on the bench to resolve vulnerabilities but that would drive up BOM cost too much.

- Agility is not a one-way street

Agile security

- **Individuals and interactions** over processes and tools makes everything negotiable
- Security officer in the trenches of negotiations with stakeholder like
  - –Solution creation teams,
  - –Business development & marketing
  - –Legal & Privacy
  - –Regulatory
  - –Business' leaders
- Tactics used:
  - –Choice architecture
  - –Risk management
  - –Social engineering (influencing)
  - –Hardball governance

Open issues

- Agile Quality Management System (QMS)?
- Agile change management
- e.g. change management for large populations using different speeds for different groups but one message
- Agile risk appetite /risk tolerance?


PHILIPS