

## Modern Organizations

Are facing huge challenges responding to:

rapid changes in regulatory and market conditions customer and user demands

ever moving trends powerful shifts in technology capabilities

Organizations can no longer choose between optimizing for stability or optimizing for speed

## Organizations

need to be flexible to deal with constant changes of the business environment

Therefore, the most important thing is not the current org structure

but the ability to shapeshift and readapt

### We need to shape the laws of change of the org, not only the org itself

## **Is Product-led Organization OUT?**



### All the Product mng guidelines and rules still apply,

#### but we must adapt it to the <mark>enterprises' culture</mark> we need a structured approach for continuous delivery instead of copying a few startups' rituals



## SW delivery has been suffering from problems for years. Problems that new tech promises to solve, but rarely (or never) does





## it's no surprise that in <mark>function-based org</mark> there is a strong <mark>focus on automation and tooling</mark> adoption, rather than cultural and org changes

## Agile

## Helped demonstrate the enormous value of smaller, more autonomous teams

### However, traditional orgs have been limited in their ability to fully benefit from Agile due to their org models



## UNKNOWN COSMIC TIME MACHINE



"One of the most important things is how many single points of failure do we have in a project?" <u>Dr. Thomas Zurbuchen</u>



Most of these failures are because of the cultural and org models

Org design for the sake of management convenience or reducing headcount actively destroys the ability of Org to build and operate SW systems effectively



People communicate with whomever they need to get the job done. The speed of SW delivery is strongly affected by how many team dependencies the org preserves





## By the Conway's Law:

# Organizational design systems will always mirror their own communication structure



His original wording was:

"Any organization that designs a system will produce a design whose structure is a copy of the organization's communication structure".

Melvin E. Conway. 1967



Org that is arranged in a functional style is unlikely to produce SW that are well-architected for end-to-end flow



Customers don't care which department of your company services their product

Products	Services	Account Management
Homes	Claims	Wealth Advisor Solutions
Automobiles	Risk Consulting	ePolicy Options
Valuables	Wildfire Defense	Welcome Fireman's Fund Client
Watercraft	Preferred Service Providers	Help
Excess Liability	Signature	Contact Us
Personal Accident & Health	Hurricane Assistance	Auto ID Card Requests
	Identity Management	
	Collectors	
	Collector Car	
	Protection	

Traditional org charts are out of sync with the new reality of frequent reshaping of teams in environments of uncertainty and innovation

Conway's law: org design prevails over SW architecture design

#### Two sides of the same coin



## \*How much awareness does the HR have about SW systems?

it's very ineffective, or even irresponsible, for orgs that build SW systems to decide on the structure, responsibilities, and boundaries of the teams without input from technical leaders

### Road to a Modern, Team-first Organization 2024



## Where to Start?



## How to TEAM?



#### Team Anti-patterns:

Constant change of team members

Too much

cognitive load



Copyright@STKI 2024 Do not remove source or attribution from any slide, graph or portion of graph

Ad hoc

team

- Citizen tasks for gov online services: passport, taxes, etc.
- Banking products: online money mng, bank transaction automation

**STKI.INFO** 

0010 Introstuture **Business Stream** By Stream-aligned team **Business Streams** Security Procurement **Everyone else** supports it

## Teams' structure is complex but they're also the best shot at delivering value continuously



- It's not a universal formula to achieve good SW delivery
- It doesn't mean to dictate to outstanding players how to perform
- Clear patterns for teams and organizations that struggling to adopt modern approaches to SW delivery



### **The 4 Fundamental Team Topologies**



Source: Skelton, Matthew; Pais, Manuel. Team Topologies



### **The Four Fundamental Team Topologies**

Source: Skelton, Matthew; Pais, Manuel. Team Topologies



There are no "Ops" or "support" teams. Stream-aligned teams are responsible for live operation **Copyright@STKI\_2024** Do not remove source or attribution from any slide, graph or portion of graph



#### Stream-aligned team

Team empowered to **build & deliver value** as quickly, safely, and independently as possible. **No hand-offs** to other teams to perform parts of the work



full spectrum of delivery

1<sup>st</sup> line to the customer and able to quickly incorporate **feedback from customers** while monitoring their SW in production

#### **Expected Behaviors**

- steady flow of feature delivery
- quick to course correct based on feedback from the latest changes
- constantly learn and adapt (experimental approach)
- time and space to address code quality
- proactively & regularly working with the supporting teams (complicated subsystem, enabling, and platform)
- autonomy, mastery, and purpose

24 Do not remove source or attribution from any slide, graph or portion of graph



#### Stream-aligned team

#### **Enabling team**

#### "CoE" - Guidance, not execution

The mission of enabling teams is to help stream-aligned teams **acquire missing capabilities** (UX, testing, automation, etc.)

- research
- try out options
- make informed suggestions on adequate tooling, practices, frameworks, and any ecosystem choices around the application stack.

#### Metrics for enabling team:

- Time taken per successful deployment
- Absolute number of successful deployments per day
- Time taken to fix a failing deployment
- Time from code commit to deployment (cycle time)



#### **Build & maintain**

a part of the system that depends heavily on **specialist knowledge** 

Stream-aligned team **Enabling team** 

Complicated subsystem team

Most team members must be specialists in that area to understand and make changes to the subsystem



Stream-aligned team **Enabling team** 

Complicated subsystem team

**Platform team** 

4



Pini Cohen

## **Platform Products:**



INFO

## **Platform Team**

Build and maintain internal platforms that are used by multiple other teams across the organization. Design, development, and operation of the platform, including ensuring its scalability, reliability, and security.

The platform products details are **invisible** to the experience developers

### The basic concept:

## Self Service to the developers

• Self Service is achieved by Automation



## Common Services Provided by Platform Teams

- Identity and Access Management (IAM)
  - API Management
- 禹

- Data services that handle data warehousing, data pipelines, data lakes BI tools that are part of the application
  - Infrastructure as Code (IaC) automate infrastructure provisioning, configuration, and management
    - FinOps Tags
  - Continuous Integration and Continuous Delivery (CI/CD) pipelines automate the software development lifecycle (SDLC)
- Monitoring and Alerting
- Security and Compliance
- Developer Experience (DX) documentation, and training resources
- Document Management

## IDP (Internal Development Platform) for building Platform Products







## Infra-Ops organization evolution





### Not Everyone Needs to Communicate with Everybody



#### **Over communication**

With open offices and instant communication tools, **anyone can communicate** with everyone else

Sometimes org can accidentally fall into a pattern where "everyone needs to communicate" in order to get work done



#### Org design problem

If the org expects that "everyone needs to be present in meetings" to approve decisions, then we have an org design problem

This will drive unintended consequences for the SW systems - lack of modularity between subsystems



#### Less is More

**Conway's law**: many-to-many communication will produce monolithic, tangled, highly coupled, interdependent systems that do not support fast flow

More communication is not necessarily a good thing

The better the Teams' Boundaries and Responsibilities are defined, the better the team's ownership, engagement, and rate of delivery will be



"When code doesn't work . . . the problem starts in how teams are organized and how people interact."

Eric Evans, Domain-Driven Design

## **3 Core Team Interaction Modes:**



Working closely together with another team



- Avoids costly hand-offs between teams
- Long-term collaboration = cognitive load A need for ongoing collaboration suggests an incorrect team's structure

X-as-a-Service



Teams need to use a code, API, or platform that "just works" without much effort



- responsibility boundaries
- Reduced cognitive load
- Slower innovation
- Danger of reduced flow



- Unblocking of stream aligned teams to increase flow
- Enhance the productivity and effectiveness
- Requires experienced staff to not work on building or running things

Neither mode Is better or worse than the other; they simply are useful for different types of work Facilitating

One team helps another team to learn and adapt new tech





### **Stream-aligned Team responsible for its Software**





#### SW delivery as a one-way process (AKA Waterfall) is not efficient, because of the speed of changes and complexity of modern SW



36





### Team Structure alone will not Solve SW Development Problems



Healthy Org Culture

Encourages empowerment, mentoring, listening, and improvement



Clarity of Business Vision

So people can be better aligned to goals and contribute better



SW Development Best Practices

Continuous Delivery, Test Driven Delivery



Healthy Funding and Budget mng

Avoid predefined Capex/ Opex split. Orgs need a clear plan to move in 3 years to OPEX-only org. Avoid making big project budget decisions in advance



Talent Mng and Reskilling

We will need the best people we can get. We must encourage continuous learning, retraining, and investment in human TALENT



## **Organization Model 2024**

#### **Design the Design Rules**





## Galit Fein EVP & Senior Analyst @ STKI

