ished in Schuberg Ph

e **2** free member-only stories left this month.
for Medium and get an extra one

enk van der Schuur
ep 7, 2020 · 10 min read · Member-only · Listen

evOps: Bridging the Dominant Divide een the Business and IT

DevOps Success Needs a Successor

Photo by Jonas Verstuyft on Unsplash

practices have had a huge positive impact on teamwork. However, Ops paradigm mainly focuses on the more technical aspects of ng value as a team. A question posed after DevOpsDays dam 2019 illustrates the issue: we've come so far, but where is the s? In this article, I will share how we are trying to go beyond the of DevOps to tackle the divide between agile DevOps teams and

Henk van der Schuur
49 Followers

Customer Director at Schuberg Philis

Follow

More from Medium

Mark Schaefer
20 Entertaining Uses of ChatGPT You Never Knew Were Possible

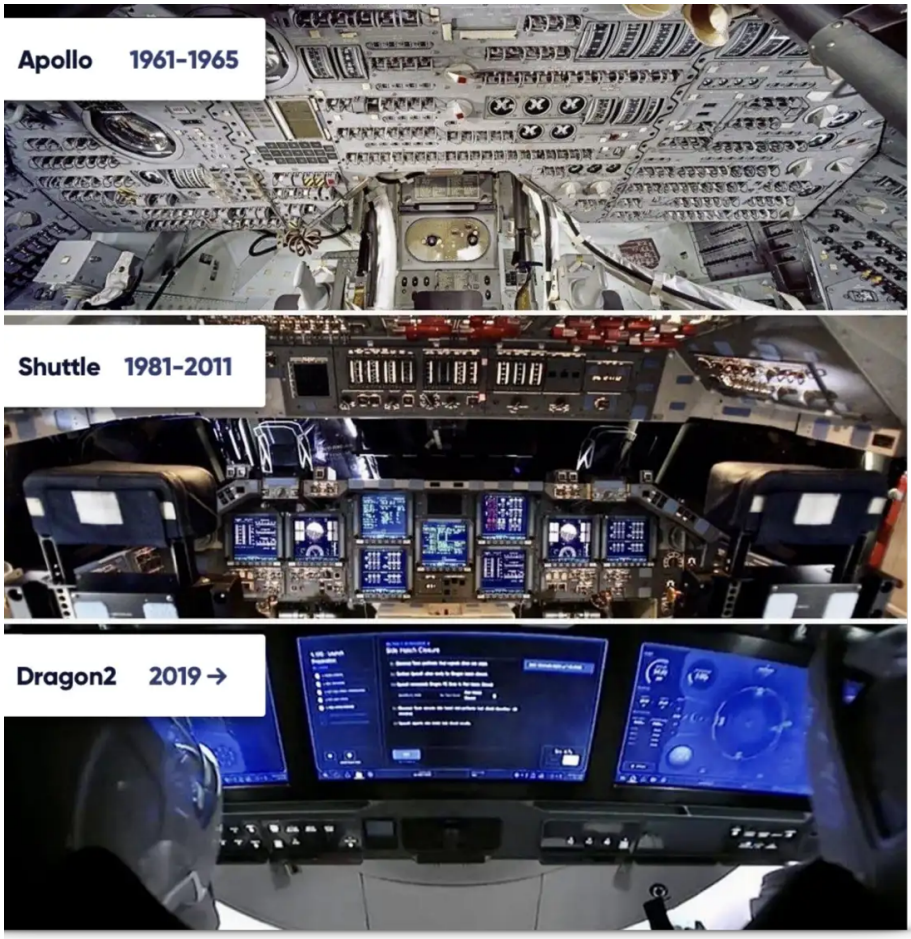
Kairsten Fay in CodeX
Today's Software Developers Will Stop Coding Soon

Jason
How a Simple Script Helped Make Me over $1000/month

Mehek Kap... in Stories From H...
This woman vlogged about her life in a polygamous relationship, and now she ha...

ness. We have found our approach makes it possible to do true tsourcing while still avoiding much of the typical friction between s owners and developers, and I hope to inspire you to try our h to BizDevOps yourself.

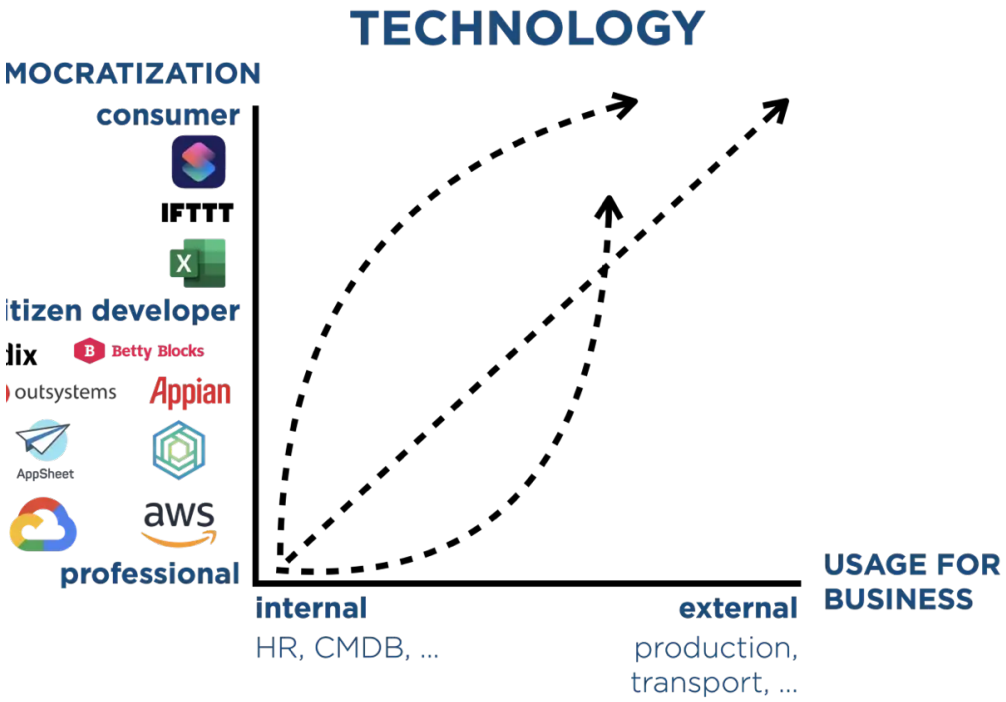
too often, business stakeholders are unexpectedly involved in the velopment process, and business requirements tend to get lost in on. This is why the solution provided by the development team often addresses the needs of the business, and little value is delivered. we best bridge that dominant divide between the Business and IT?



plex technology is further abstracted away over time (source: Michal Malewicz, SpaceX: Simple, beautiful interfaces are the future)

ently I stumbled on this picture. It struck me that, if anything, our acecraft have become *more* technically complex over time. Yet, *w*, we have managed to remove this complexity from sight over the least it *looks* as if everyone could control a (SpaceX) spaceship rs.

chnology is hidden, it tends to become accessible for a broader his is known as democratization). Where public cloud technology ML, Deep Learning, blockchain—feel free to complete your own d bingo card) from Microsoft, Google and Amazon is still targeting onals, their respective low-code platforms—PowerApps, AppSheet eyCode—are designed for citizen developers: those who 'create new applications for consumption by others using [...] shared services, eneration language (4GL)-style development platforms and cloud ng services' (source). The same applies to the low-code platforms ndix, Betty Blocks, and others. Moreover, programming tools such , Apple's Shortcuts, and good old Excel can easily be used by the consumer.



Tech is increasingly impacting products and services

this technology becoming available for a wider public, products and

are increasingly impacted. It's not only internal or secondary
processes that are impacted: the customer-facing, primary
s of businesses are increasingly changing because of the driving
technology and innovation. Just look at the premier provider of
ıd parcel services in the Netherlands, PostNL, which has developed
alternative for postage stamps. Regardless of how this trend will
precisely — it's happening. Under these circumstances the business
more from teams than 'just' doing their 'DevOps trick': they expect
understand that it's all about creating value. If they don't do that,
ces are that 'the business' will swipe their credit card and starts
themselves. That last point is an important difference compared to
ıess-IT gap in the 1980s.

ized that sending letters was going to be impacted by digital technologies and developed a digital replacement for a postage stamp (this was created 'way back' in 2013)

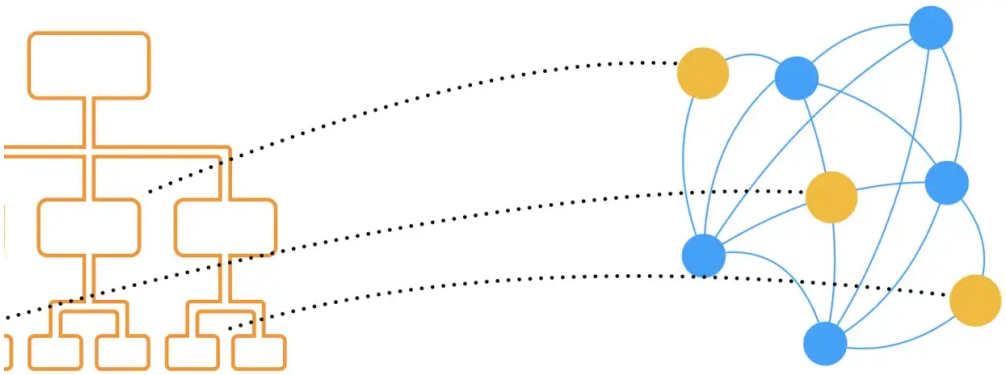
the thing is, today, IT is mostly *still* totally separated from the
. It's separated within organisations (different departments, islands,
well as between organisations (when IT is being outsourced to a
ty). Neither side understands the other, yet both expect the other to
to understand *them*. There's a lot of finger pointing and, as a result,
resses are built. A *corporate immune system* is delaying innovation
ys smart people finding an answer to the question how are the
y's products and services going to be impacted/changed/improved by
new technologies?

IT is often separated from the business (bonus points if you recognize the movie shot)

A typical business-IT conversation

archical structure of many organizations doesn't help. It creates a zone that discourages transparency and vulnerability. As an attempt natic cooperation, external consultants, project managers, agile and/or so-called 'thought leaders' are hired to function as rs between the two sides. Clearly this only addresses the effects of le, and fails to tackle its cause. Worse: it just introduces more d.

ıow *do* we bridge the gap between those who write code and those o write emails? Maybe I can share a bit about how we work together erg Philis, and what I've learned during my time there.



Making space for team members from the customer

the collection of blue dots on the right is a team. Developers, rs, and Sales are roles that are typically represented in the dedicated r teams of Schuberg Philis. When we're about to embark on a with a customer, one of the first things we do is think about the roles those in the business to play (the yellow circles). What bilities do we expect? And which roles seem to fit those bilities? Do these roles generate the required mandate? Who do we contact to get this approved and arranged on the customer side?

Whole system in a (single) room

the roles are allocated, before the start of the project, we go and sit
in a room. We close the door. Call it a cross-functional team, self-
team, X-team or even co-creation—the name doesn't really matter.
t putting the people with the right knowledge, expertise, vision, and
together.

his "w*hole system in a room".* In this setting, there's not much
y. It's all about getting unnecessary management out of the way and
experts in the lead.

See you on the other side, kiddo

. is to then cross over in each other's space and commit to mutual uilding—to share the responsibility for building a bridge to the other ally, IT has someone from the business who talks their language, .tes them, and even can be an advocate for them. The business then ›re ownership of tech decisions and understands better what's Since both sides are involved in a single team, there's no finger .

ıallenge is to find people from IT and the business who can operate ırple shaded area and won't meet resistance from the business and respectively. This is not a walk in the park. It really *is* more work n be very demanding to go over to 'the other side'. We see from our es, too, that they find it difficult to cross the space sometimes and ;pect to either business or IT people. Each side has its own language :tices. Screaming '#noestimates!' isn't really helpful if business is cope with milestones, budgets and release deadlines. Yelling t!' is just as unconstructive when engineers are trying to understand

’ before sharing their advice, let alone committing to a deadline. I
at, at the end of the day, bridging the gap boils down to things like
ility, honesty and transparency. Getting out of your comfort zone.
ly trying to understand the other person. When such an atmosphere
ished, there’s no finger pointing—and the other party praises you for
nership you’ve created together (this then replaces the more usual
tional customer/supplier relationship).

ting into the same room with the right roles and disciplines can be
ry valuable for sharing project updates (think of the Sprint Review,
nce). In addition, we’ve found it to be a powerful way to kickstart
Getting into the same room from Day One creates an atmosphere of
l transparency, which helps us realise short time-to-value together.
ntroduce the Proof of Value Pressure Cooker™. This consists of five
hat are typically executed in 1–2 weeks. These phases are detailed

re

un is half the work. This phase typically starts *before* the pressure
tarts and is performed by the more solution- and/or technically
team members. It involves getting to the heart of the matter:

nat is this really about?

(why, why) are we doing this? Are we doing this?

) can we be successful?

ll questions have a satisfactory answer, can the team continue to the
se.

Formulating a core question at the start of a Pressure Cooker™ helps the team to focus

se is where the business takes the stage, and shares their ge, experience, frustrations, wishes, ideas. IT is listening, in an c way, trying to ask smart questions. In addition, the technical team s should ask themselves:

e understand?

are we missing?

we start? When do we start?

a primary business process helps establish an initial understanding of the scope and complexity of the problem at hand

h

e problem domain has been laid out by the business, it's time for IT t and share how they understood the explanation made by the s. Visualizing this interpretation helps mutual understanding. er, sketching initial screens helps the team keep the final goal in hile answering the following questions:

s what you mean?

t should be changed?

s going to bring you value? How? Why?

Trying to sketch initial screens helps everyone keep the final goal in mind

type

here the magic happens. Based on all the notes, drawings, sketches, r input from the previous phases, an initial prototype is built. As a e makes the results of the pressure-cooker process much more and tangible, it implicitly asks these questions:

are the most important requirements?

lear enough who are our audience and who are the end-users?

are we going to deliver? And, equally important, what are we not to deliver?

Coding away in Schuberg Philis' Lab 271

e

e prototype phase—and *lots* of coffee—the prototype is presented to ness. It's essential to ensure that the business understands what's no'd and delivered (as well as the effort that went into it :-)). This about asking and answering these questions:

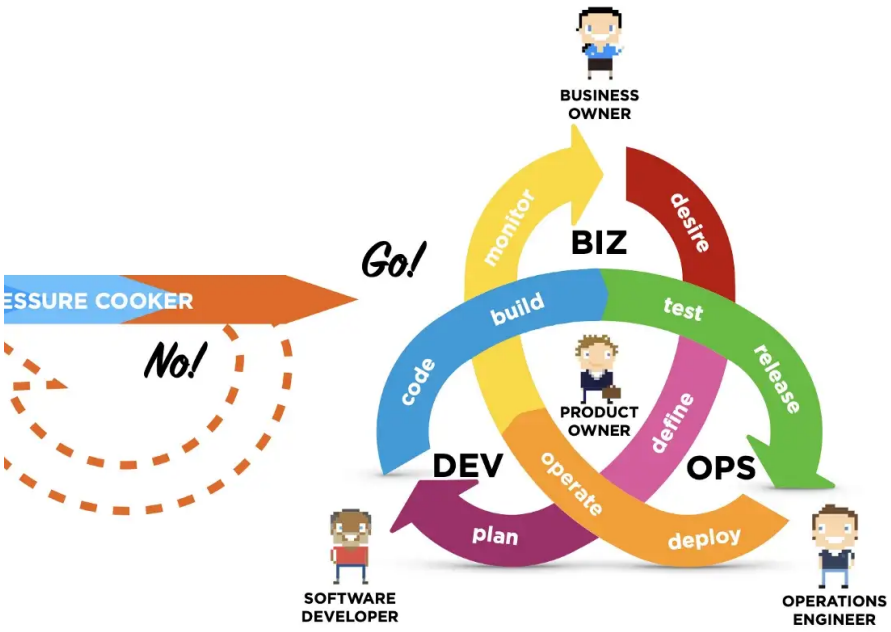
nich extent was value delivered?

what does it mean to go live?

: what needs to be improved?

A Pressure Cooker™ Demo

be clear: my intention is not so much to introduce a new way to kickstart a new project. There are many more ways to do this. The if you *start* a project with *The whole system in a room*, you get an early f what you can expect during the actual project in terms of ation, communication, creativity and critical thinking. If it hurts, do *arly*, right?



Introducing BizDevOps!

re cooker gives you early insights. Either you know the whole thing ing to work like this—e.g. the problem was not clearly defined, the was not sufficiently present, feature creep is already occurring, and ıg is needed, *or* you have a quick, transparent way to add actual hich could serve as a kickstart for the way of working during the rest oject: BizDevOps.

DevOps, it all starts with a business need. Within the team, the defines that need in the form of requirements, which should be and refined enough for the technical members of the team to plan d them. After testing (it's best if this is done by the business, too), ınd deployment, monitoring data from both the technical and al operations create primary input for the team — and the business ular—to form new functional and non-functional wishes and ıents.

ps not only means getting together during the start or design of a it also means getting together during the run phase. *Sit* behind the nd users. *Feel* what they are experiencing when they have to wait nds during each and every login.

duct owner (or component owner, value stream owner, …) basically e glue between the three disciplines depicted above and he/she be quite crucial for the success of this way of working. The product virtually a mini-CEO who should be *fully* mandated to make s on behalf of the business. What's more, this role should be d by all other disciplines.

m up:

Needs a Sequel ♻️

to repeat the DevOps success of bringing Development and ns together, with the Business. We need to double down on the approach of bridging gaps, and now work to bridge the gap between ness and IT. Different roles, beings, and struggles are involved. But gles are real.

ıy = Comfort 🛋️

your way of working, governance, and reporting. Get in the same th at least three different roles: tech, business, project nent/facilitation/delivery. Don't even start if these 3–4 roles are not ıtly established.

ogy is Everyone's Business 🚀

ıst the DevOps trick' alone isn't going to cut it anymore. Business be involved. BizDevOps is about organizing a short time-to-value, tually reduces risk as it allows things to fail early, and to fail fast— with the business. After all, technology is now everyone's business.

y is based on my keynote talk at DevOpsDays Amsterdam 2020. In case any questions, remarks or suggestions: please feel free to comment below ne a line!

s Business Value Software Development Agile Dev Ops

21