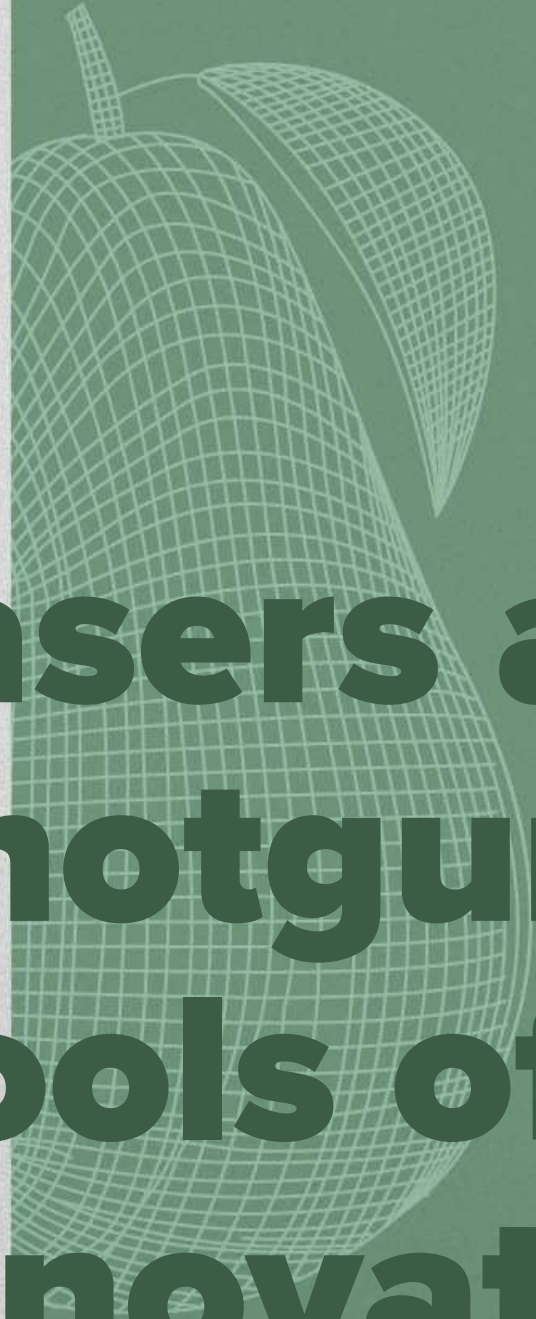


Sep, 7 | Kyiv



Lasers and Shotguns: The Tools of Innovation

 WEEKEND
products



Introduction

Software Innovation Comes
Down To Writing A Product

20+ Years Ago, We Employed
“Top Down Analysis”

In The 2000’s, We Went “Lean”
with the MVP Concept



Introduction

“The minimum viable product is that version of a new product a team uses to collect the maximum amount of validated learning about customers with the least effort.”



Eric Ries

Author of “The Lean Startup”

Introduction

The MVP approach relies on two principles...

“Minimal” can’t exceed “viable”.

Post-MVP iterations must be valuable and can’t exceed available resources.



Introduction

But what if...



What must be (minimally) accomplished is well known, established, and expected by users?

Or the concept is so innovative users don't know what they want or need?



Introduction

My name is Chuck Ros

I'm a Senior Business Analyst at SoftServe, based in Atlanta in the USA.

I've worked most of my 30 year career as a programmer and product leader.

I've built and sold 3 software companies. And I've helped dozens of companies build dozens of products.



<https://linkedin.com/in/chuckros>

BSCS, MBA

IT WEEKEND
products

Introduction

At Emerald Software, my 3rd company, we set out to build a just-in-time order entry system.

Its principle feature was its flexibility and configurability: we could accommodate any company's order process, no matter how complex.

We built a *platform*, not a product.

And we found a huge market in automating employee hiring.

Not once did we sell it as an order entry system.

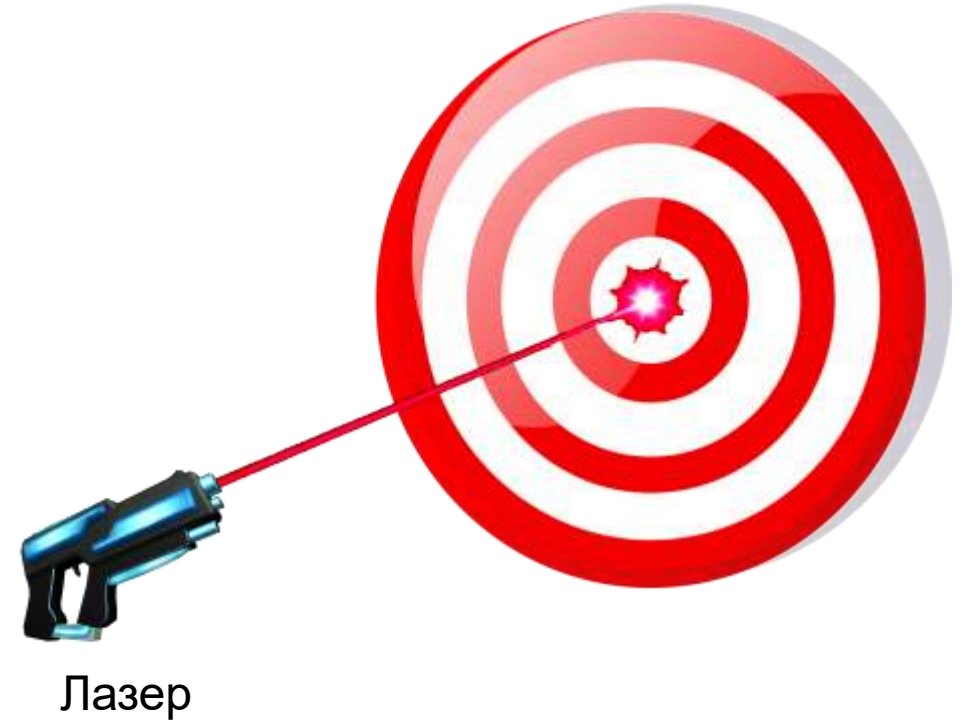


Introduction

The MVP approach requires considerable investment in understanding user expectations and what functionality might be considered.

Building an MVP is like trying to hit a bullseye with a laser.

If you miss the bullseye, the iterations to hit the next time might be expensive.



Introduction

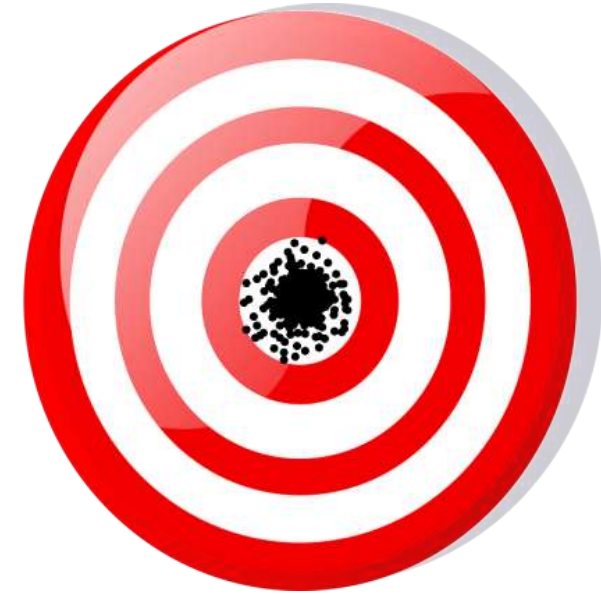
A more flexible approach—building a platform rather than a product—can result in creating a successful product even if you don't hit your bullseye on the first try.

Building a platform is like trying to hit a bullseye with a shotgun.

You're more likely to hit the bullseye, but if you do miss, the iterations to hit the next time will likely be less expensive.



дробовик

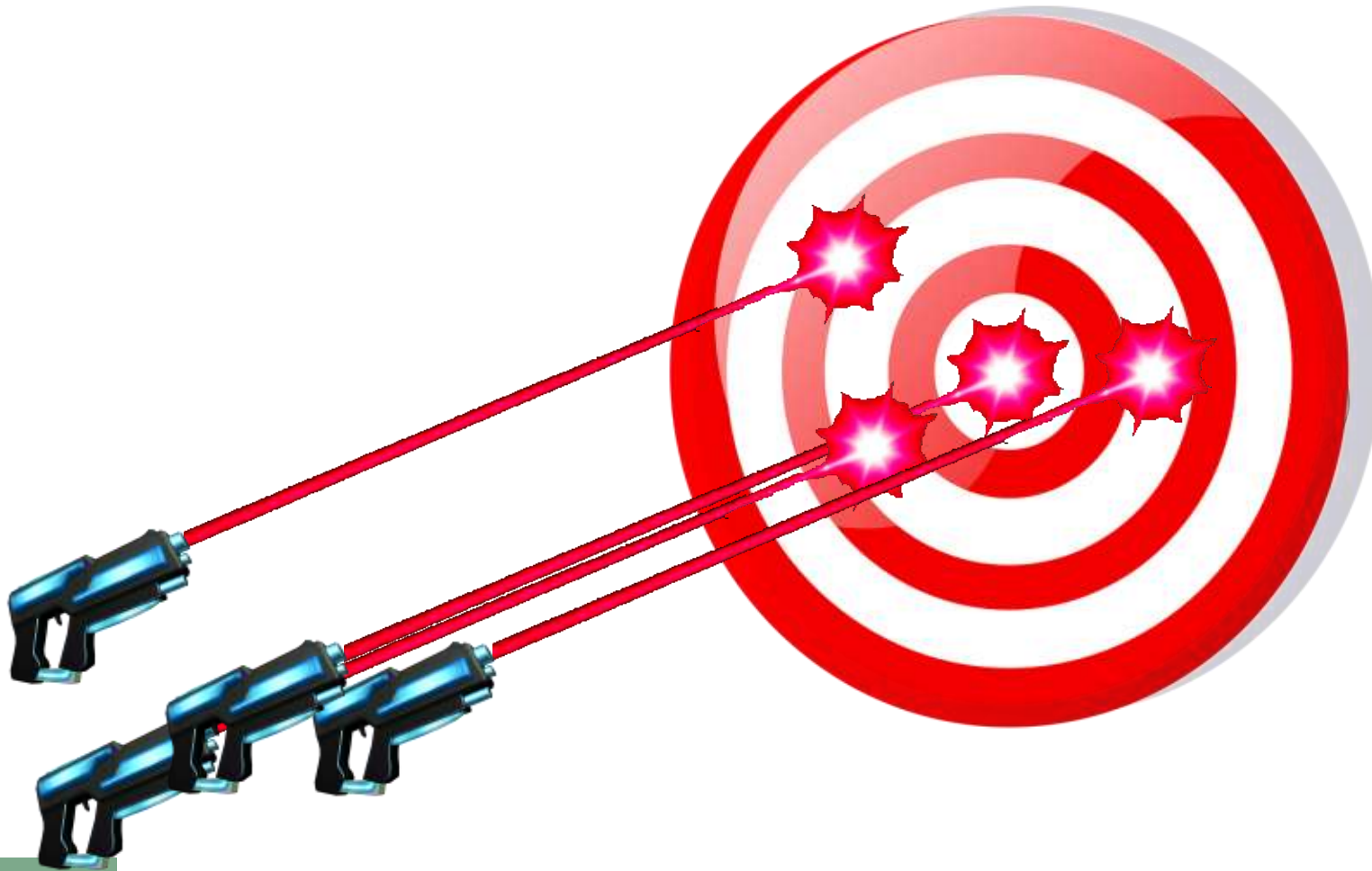


Making MVPs Work

“The minimum viable product is that version of a new product a team uses to collect the maximum amount of validated learning about customers with the least effort.”

- “Least effort” – still requires effort, which is sometimes not trivial.
- Depends on delivering sufficient value to begin gathering validated learnings.
- Relies on subsequent iterations to increase the value delivered.

Making MVPs Work



IT WEEKEND
products

Making MVPs Work

Those iterations—retargeting the laser—are generally done by programmers.

Programmers are among the scarcest, most valuable resources in the company.

Those iterations are therefore expensive.

If the MVP is too far off the bullseye, the sum total of all iterations may be greater than the resources available.

Which means you will run out of money before the product is successful.



Making MVPs Work

MVP's still rely on assumptions, even if they are very well researched and validated.

And those assumptions might change as the MVP is being built.

Product teams should always assume they are trying to hit a moving target.



Making MVPs Work

For the MVP approach to work, we have to balance delivering a product close enough to the mark...

such that the cost of iterations doesn't suck the company dry.



Making MVPs Work



IT WEEKEND
products



The Platform Approach

We probably still have to make assumptions about what our product should be.

And unless we're *perfect* in figuring out what our users want and need...

We'll still need to tweak our software after we go to market.



The Platform Approach

So what can we do to make our iterations faster and cheaper?

Faster – minutes and hours, not days and weeks.

Cheaper – services and support resources, not our most expensive technical resources.

Better yet – let our users perform their own iterations.



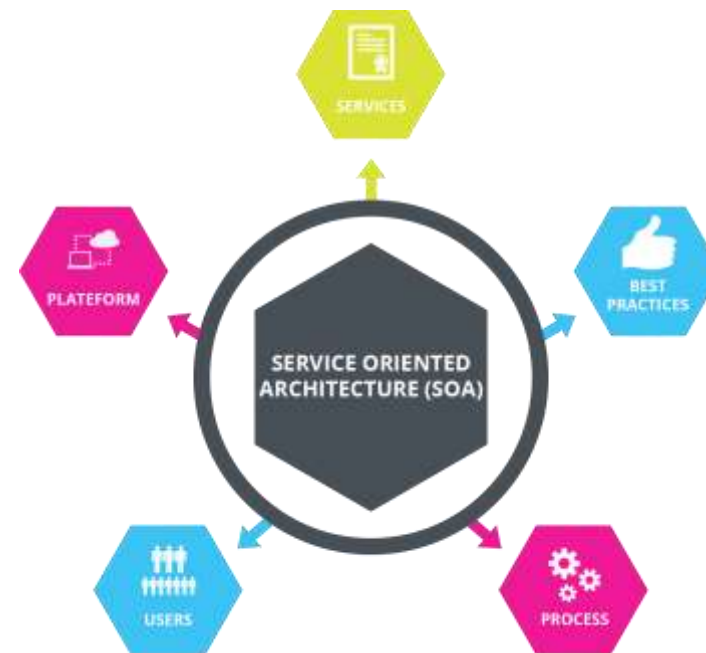
The Platform Approach

Build in flexibility and configurability into our products.

Sufficient to overcome missed or change assumptions or bad data.

Perhaps even sufficient to appeal to users with needs you never even anticipated.

Build a *platform* more than a product.



The Platform Approach

“A platform is a group of technologies that are used as a base upon which other applications, processes, or technologies are developed.”



Wikipedia
The Source of All Knowledge

The Platform Approach

Why a shotgun approach? Why build platforms now? What's changed since the MVP concept emerged?

The MVP concept pre-dates cloud.

Programmers are more valuable than ever, and iterations more expensive than ever.

Customers expect more flexibility and agility than ever.



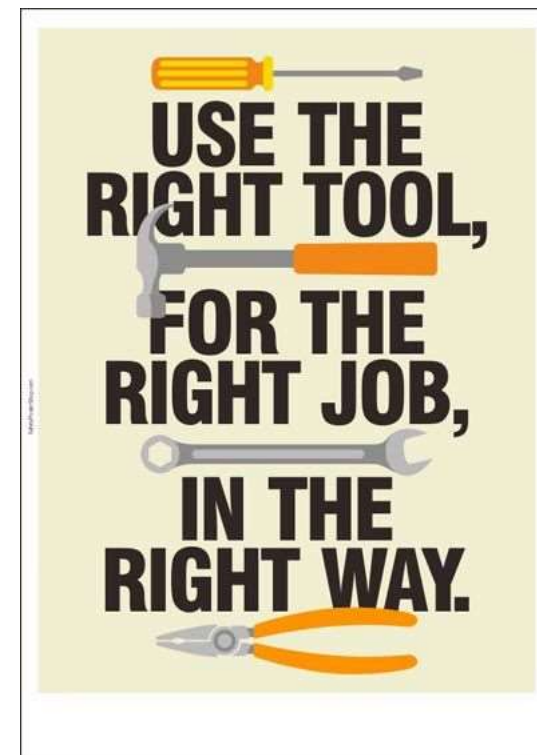
A Typical Programmer

The Platform Approach

Is MVP dead? Are shotguns *better* than lasers?

Not at all. Both are tools that should be applied when appropriate.

It's the job of product leaders and innovators to figure out when it's best to use a laser and when it's best to use a shotgun.



When To Use Shotguns and Lasers



When To Use Shotguns and Lasers

Scope

- Well-known, well-defined, fixed
- Not well-known, undefined, variable

Resources

- Funded, resources/personnel available
- Limited funding & resources

Time

- Short time to market, long iteration curve
- Longer time to market, shorter iteration curve

Laser

Investment in

Risk of missing the bullseye is high

Cost to

The inevitable

Usually not

Without plenty of

Building

Short time to market AND short iteration curve?
Laser/MVP – but you better not miss!

Options are more expensive & time-wise)

More time to invest in flexibility, iterations are faster

IT WEEKEND

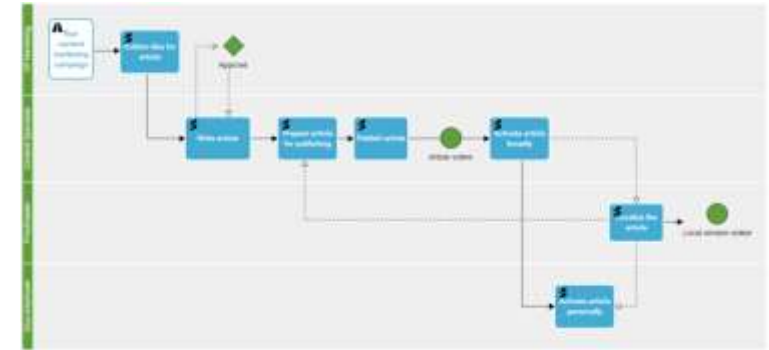
products

Implementing a Shotgun Approach

1. Generalize underlying processes.

Thoroughly understand what the system needs to do, not just what features users want to see. Features are desired by specific users, processes are useful to many.

Example: an employee timekeeping system is made up of a check-in/check-out process, document and records management processes, and review and approval workflows.



Implementing a Shotgun Approach

2. Anticipate where flexibility is most required.

What functionality is most at risk for missing the target?
What is most important to make flexible? What
functionality would require the most expensive
iterations after the initial release?

Example: a document-centric system where the
documents are subject to change, like tax forms.



Implementing a Shotgun Approach

3. Utilize more off-the-shelf services to inherit flexibility.

Off-the-shelf systems and services—particularly public cloud services—come pre-made with configurability and flexibility.

Example: Incorporating an open source BI platform like SuperSet as the dashboarding and data visualization module of a business system.

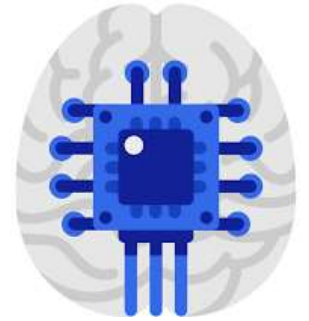


Implementing a Shotgun Approach

4. Apply the architecture of “engines” and “services”

Concentrating functionality into an independent service provider forces you to generalize processes and provides opportunity for non-app-specific configurations.

Example: use a business rules engine instead of hard-coding decision trees that are subject to change or client variability.



Implementing a Shotgun Approach

5. Apply the “Shotgun Rule” to every product requirement decision.

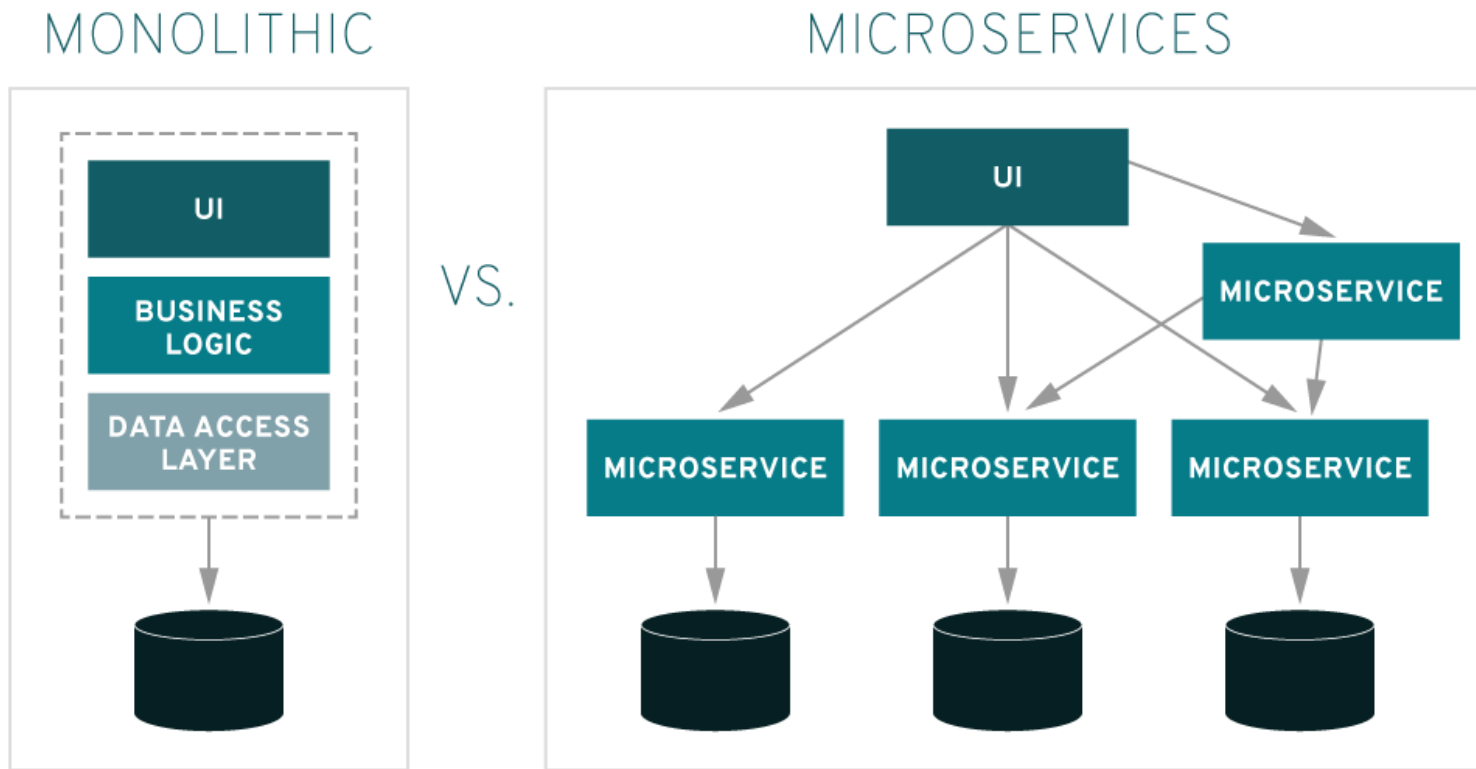
For all design decisions: if a requirement can vary in any conceivable utilization of a feature, make it configurable.

Example: user John insists a sales amount be entered first, then the sales code—but Jane insists the sales code be entered first, then the amount. Solution: allow users to set preference on order of entry.



Implementing a Shotgun Approach

Platform and services-based architectures should be naturally flexible...



Summary

MVP Laser approach: try to get version 1 “just close enough”.

Optimize value through iterations – as few as possible (assuming your MVP was close to bullseye).

Platform Shotgun approach: build in fundamental flexibility and configurability in version 1.

Optimize value through iterations – and utilize cheaper, more abundant resources to accomplish.

And pursue alternative targets with the platform’s agility.



Lasers and Shotguns: The Tools of Innovation

Globalization
Digitalization
Predictive analytics
Self service

Go create something of exceptional
value to your customers!

IT WEEKEND
products