

INGREDIENTS FOR WORLD-CLASS SOFTWARE DEVELOPMENT

Experiences from  Spotify  Microsoft  **RIOT**
GAMES **ERICSSON** 



Are you curious about your craft and want to get better as an engineer / developer / leader building products and solutions using software?

Then please join **Erik Schön** coder | software researcher | system engineer | R&D leader on a journey from **Hell's Kitchen to MasterChefs ...**



WHY: Inspiration and new perspectives on principles and practices of modern, world-class software development.

WHAT: Ingredients for World-Class Software Development - Experiences from Spotify, Riot Games, Microsoft & Ericsson

What are the key capabilities needed for state-of-the-art software development? This talk distills the essence of what the software community has learnt over the past 50 years including lessons from movements like Agile, Lean software development, DevOps and Lean Startup as practiced by companies in different industries like Microsoft (cloud, development tools and office applications), Ericsson (telecom), Spotify (online music) and Riot Games (online gaming). Intended audience: practitioners curious about their craft wanting to learn and get better at their daily work, e.g. engineers/developers, product owners/product managers, ways of working experts and leaders.

HOW: 40 min presentation in English including interactions and Q&A.

WHEN: November 2, 14-18 CET.

WHERE: Lean Product And Process Development Exchange (LPPDE) Virtual Summit November 2023 - online.

WHO: Erik Schön, coder, software researcher and system engineer turned leader executive and strategist, Erik Schön has led organizations of up to 400 people in complex product development. He was a member of Steve Denning's Learning Consortium (Microsoft, Spotify, Riot Games, Ericsson, ...) and one of the leaders of a large Lean/Agile transformation journey involving 2000 people in 10 sites in 3 countries over 7 years. Erik currently heads Erlang Solutions Nordic, evolving the open source Erlang/Elixir/OTP/BEAM ecosystem for scalable, secure and reliable software. His latest books are The Art of Strategy – Steps Towards Business Agility and The Art of Leadership - Purpose and Integrity for Sustainable Success – both published by Yokoso Press.



INGREDIENTS FOR WORLD-CLASS SOFTWARE DEVELOPMENT

Experiences from



Lean Product and Process Development Exchange
LPPDE Virtual Summit
November 2, 2023



Erik Schön
Managing Director Nordic, Erlang Solutions



Why developers like cooking ...



Randall Koutnik

@rkoutnik

Follow

Why programmers like cooking: You peel the carrot, you chop the carrot, you put the carrot in the stew. You don't suddenly find out that your peeler is several versions behind and they dropped support for carrots in 4.3

9:22 PM - 16 Jan 2019

1,845 Retweets 5,689 Likes



68

1.8K

5.7K



What do you need to make a great dinner?

Great tools!

Great ingredients!

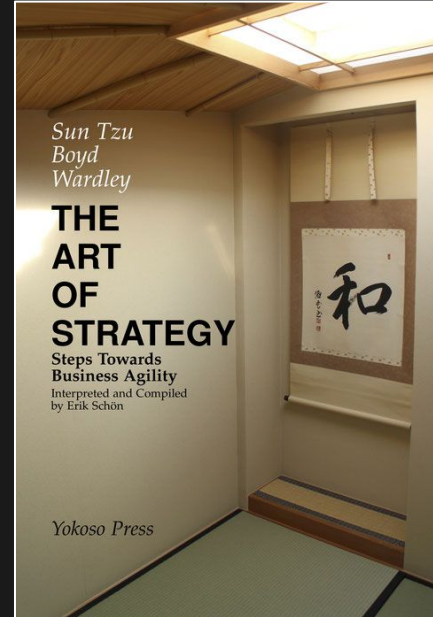
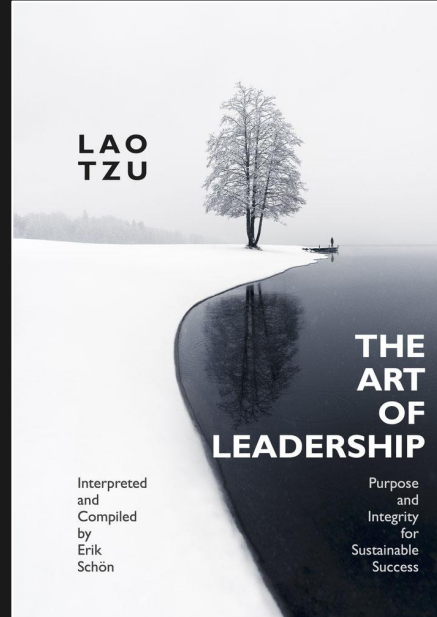
Great recipes!

Great skills!

WE INTERRUPT THIS PROGRAM FOR A

COMMERCIAL BREAK

ERIK SCHÖN



Coder
Software Researcher
System Engineer
Product Manager
Inventor
Manager
Leader
Executive
Speaker
Writer

ERICSSON 

NETENT

framfab 

 Erlang
Solutions

BUSINESS OUTCOMES

ERICSSON 

Red Bull 
RACING

aws 

VOLVO







WhatsApp



The **ERLANG/ELIXIR** open-source ecosystem

2x FASTER development of new services

10x BETTER services, down-time less than 5 minutes/year

10x SAFER services that are very hard to hack or crash

10x MORE users and transactions - within milliseconds

10x LESS costs and energy consumption



RabbitMQ

120+ people
Founded 1999
London
Stockholm
Kraków
Budapest
Americas

Consultants
Services
Products
Training
Support
Conferences

Part of **TRIFORK**
1200+ people in EU & US
Founded 1996

 Erlang
Solutions



What tools?

What ingredients?

What recipes?

What skills?

It depends ...

What tools?

What ingredients?

What recipes?

What skills?

WHAT
IS
THE
JOB
TO
BE
DONE?



WHAT ARE THE RIGHT

ingredients, skills,
recipes, tools

FOR THE JOB?



What did we forget?

Great tools!

Great ingredients!

Great recipes!

Great skills!

Preparations!

Time!

Creativity!

Between Hell's Kitchen and MasterChefs...

Better ingredients!



Better skills!

Better recipes!

Better tools!

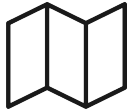
Better preparations!

A pinch of creativity!





Let's talk about ingredients, recipes, skills,
tools, preparation and creativity
in a software development context!



What have we learned about
software development since Ivar Jacobson
invented blocks, modules, sub-systems,
sequence diagrams & use cases

@  in 1967?



Insights from complex software development



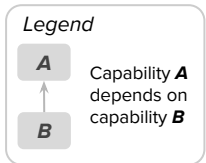
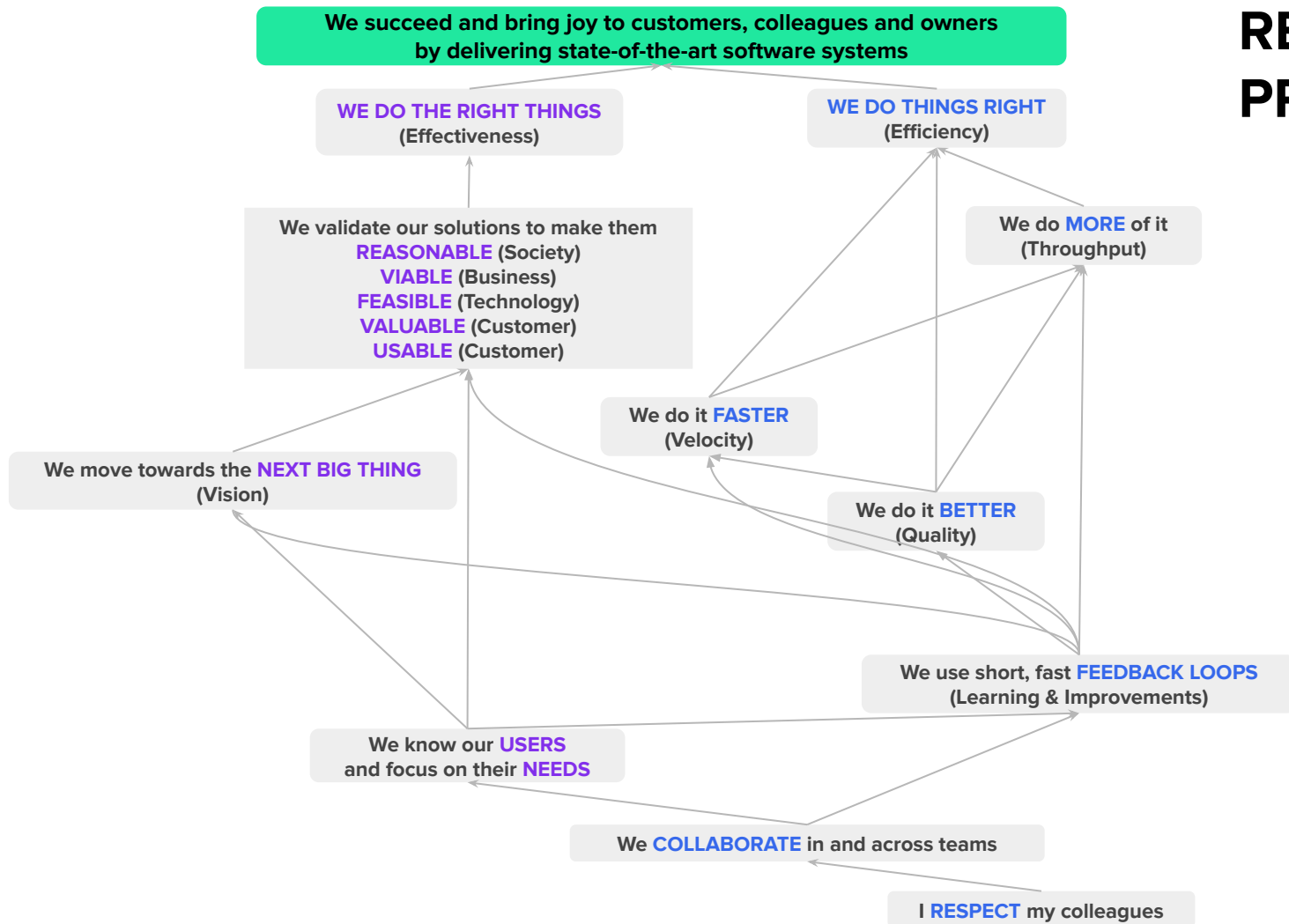


Key question:

**How do you know that you are working
on the most important thing right now?**

Think about this for 1 minute, starting now!

RECIPE, PREPARATION





**How about a few concrete examples
right now?**

INGREDIENTS, SKILLS, TOOLS

We succeed and bring joy to customers, colleagues and owners
by delivering state-of-the-art software systems

WE DO THE RIGHT THINGS
(Effectiveness, Discovery)

Lean Startup
Design Sprint
Impact Mapping
Prioritization
Business Model-
Canvas

We validate our solutions to make them
REASONABLE (Society) **sustainable, resilient**
VIABLE (Business) **sustainable business model**
FEASIBLE (Technology) **architecture, performance**
VALUABLE (Customer) **secure, reliable, scalable**
USABLE (Customer) **ease-of-use, ease-of-deploy, desire to use**

WE DO THINGS RIGHT
(Efficiency, Delivery)

We do **MORE** of it
(Throughput)

We do it **FASTER**
(Velocity)

Focus

Work on few things in parallel
Kanban boards for teams and organizations

Visualise the work
“Elephant Carpaccio” slicing
Find and remove wait times
Find and remove bottlenecks
Find and remove handovers
Value Stream Mapping (VSM)

We do it **BETTER**
(Quality)

Pair/ensemble/mob programming
Refactoring (“Scout Rule”)
Code reviews
Test-Driven Development (TDD)
Continuous Integration (CI)
Tools: SonarCube Coverity, XRef, Typer,
Dialyzer, EUnit, QuickCheck, Common Test, ...

We move towards the **NEXT BIG THING**
(Vision)

We use short, fast **FEEDBACK LOOPS**
(Learning & Improvements)

Retrospectives

Short, daily stand-ups
Pair/ensemble/mob programming
Code reviews
Test-Driven Development (TDD)
Continuous Integration (CI)
Continuous Delivery (CD)
DevOps

Playing to Win
Wardley Mapping
Objectives & Key Results (OKRs)
4 Disciplines of Execution (4DX)

We know our **USERS**
and focus on their **NEEDS**

Lean Startup
Design Sprint
Design Thinking
Jobs to be done
Impact Mapping

We **COLLABORATE** in and across teams

Cross-functional teams
End-to-end teams
Find, remove/handle dependencies
Feature/System anatomies

I **RESPECT** my colleagues



Where do we start?

WHAT
IS
THE
JOB
TO
BE
DONE?





Where do we start?

Ask your customers:

What's the one thing we do really well?

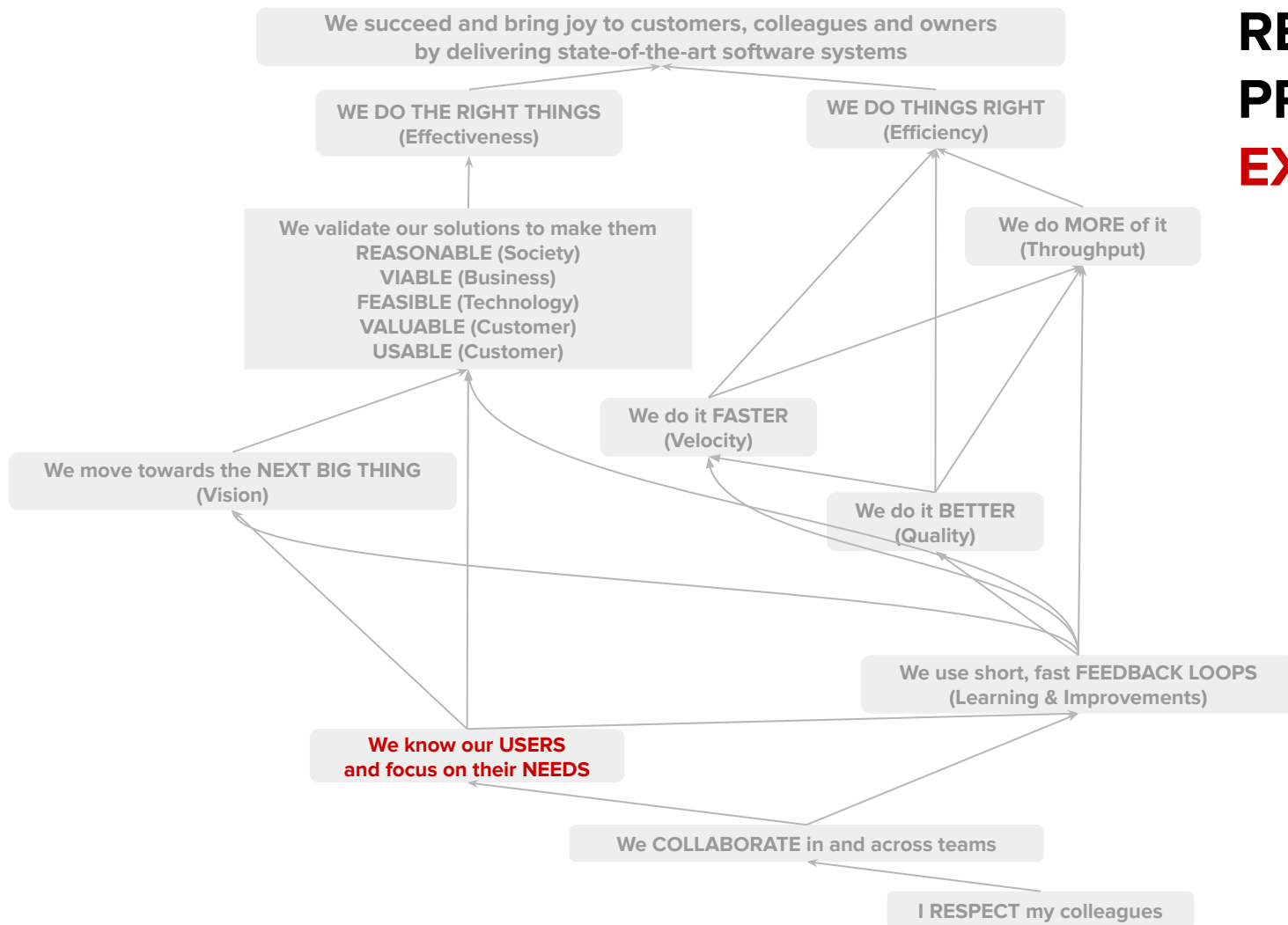
What's the one thing we should improve?

In parallel, ask yourselves:

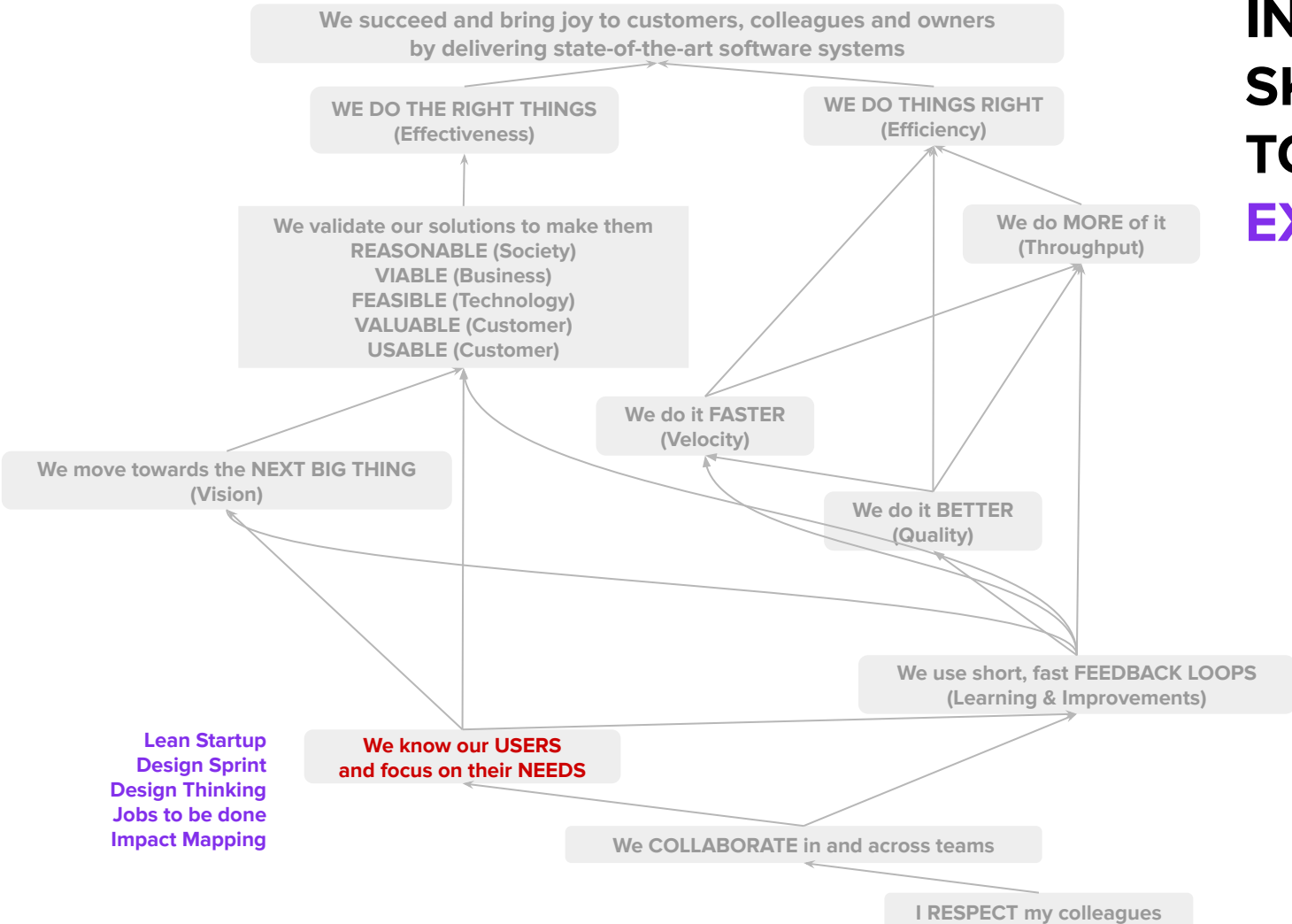
What's the one thing we do really well?

What's the one thing we should improve?

RECIPE, PREPARATION EXAMPLE



INGREDIENTS, SKILLS, TOOLS EXAMPLES



Lean Startup
Design Sprint
Design Thinking
Jobs to be done
Impact Mapping

The characteristics of good cooking

Optimize limited ingredients creatively for a never-ending menu

Preserve and extend ingredients and combine them in different sequence, color, texture, flavor, fragrance, ...

Use a little bit of the more pricey ingredients to flavor a lot of the more common ones

Take advantage of the seasons

Prepare the food to provide maximum surface area and cook it as to sear in the nutrients while minimizing the use of fuel

Getting the most out of your ingredients means keeping the garbage can empty

Cook and eat to harmonize the ingredients to let the natural flavors speak for themselves



Lao Tzu, Chuang Tzu, China, 500BC

YOU are the Master Chefs!

Prepare
the ingredients
using suitable
recipes, skills, tools –
creatively!





And, remember ...

WHAT
IS
THE
JOB
TO
BE
DONE?



Q&A

We succeed and bring joy to customers, colleagues and owners
by delivering state-of-the-art software systems

WE DO THE RIGHT THINGS

(Effectiveness, Discovery)

[Lean Startup](#)
[Design Sprint](#)
[Impact Mapping](#)
[Prioritization](#)
[Business Model-Canvas](#)

We validate our solutions to make them

REASONABLE (Society): sustainable, resilient

VIABLE (Business): sustainable business model

FEASIBLE (Technology): [architecture](#), performance

VALUABLE (Customer): secure, reliable, scalable

USABLE (Customer): ease-of-use, ease-of-deploy, desire to use

[Playing to Win](#)
[Wardley Mapping](#)

[Objectives & Key Results \(OKRs\)](#)
[4 Disciplines of Execution \(4DX\)](#)

We move towards the **NEXT BIG THING**

(Vision)

[Lean Startup](#)
[Design Sprint](#)
[Design Thinking](#)
[Jobs to be done](#)
[Impact Mapping](#)

We know our **USERS**
and focus on their **NEEDS**

We **COLLABORATE** in and across teams

I **RESPECT** my colleagues

WE DO THINGS RIGHT

(Efficiency, Delivery)

We do **MORE** of it

(Throughput)

We do it **FASTER**

(Velocity)

We do it **BETTER**

(Quality)

We use short, fast **FEEDBACK LOOPS**

(Learning & Improvements)

[Cross-functional teams](#)

[End-to-end teams](#)

[Find, remove, handle dependencies](#)

[Feature/System anatomies](#)

References

[Anatomy of Excellent Development \(AED\)](#)
[The Principles of Product Development Flow](#)
[Inspired](#)
[Extreme Programming Explained](#)
[Scaling Lean & Agile Development](#)
[The Lean Mindset](#)
[Product Aikido](#)

Focus

[Work on few things in parallel](#)

[Kanban boards for teams and organizations](#)

[Visualise the work](#)

[“Elephant Carpaccio” slicing](#)

[Find and remove wait times](#)

[Find and remove bottlenecks](#)

[Find and remove handovers](#)

[Value Stream Mapping \(VSM\)](#)

[Pair/ensemble/mob programming](#)

[Refactoring \(“Scout Rule”\)](#)

[Code reviews](#)

[Test-Driven Development \(TDD\)](#)

[Continuous Integration \(CI\)](#)

[Static code/program analysis](#)

[Code/test coverage](#)

[Retrospectives](#)

[Short, daily stand-ups](#)

[Pair/ensemble/mob programming/software teaming](#)

[Code reviews](#)

[Test-Driven Development \(TDD\)](#)

[Continuous Integration \(CI\)](#)

[Continuous Delivery \(CD\)](#)

[DevOps](#)

Thank You!



www.erlang-solutions.com