MongoDB. ATLASSIAN

The Effective 🚀 Developer

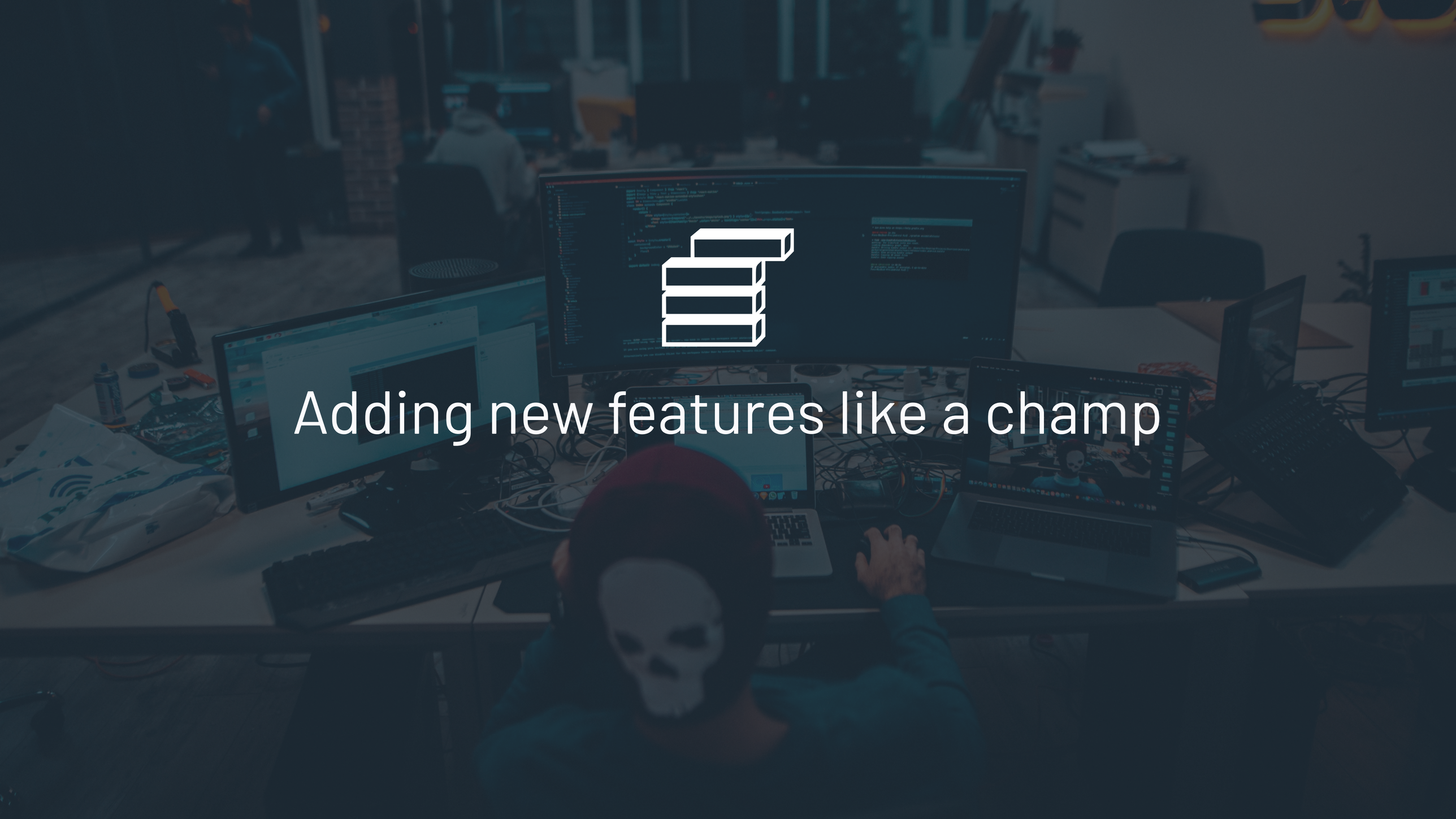
Work Smarter 💡 , Not Harder 💪


Sven Peters

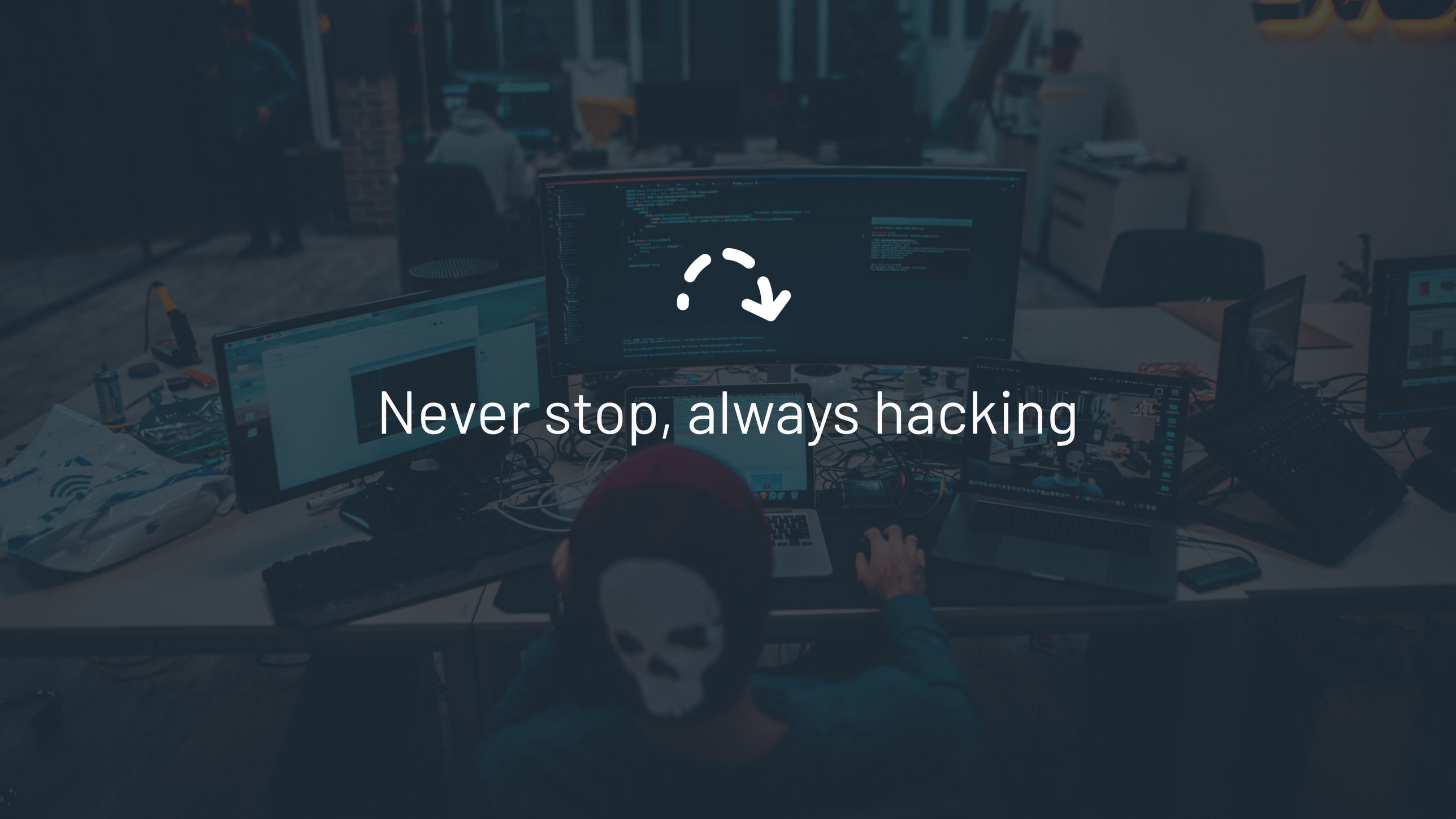

@svenpet


Writes smart code

Fixing bugs immediately



Adding new features like a champ


Never stop, always hacking

Live and breathe code

That was me in 2003

Super effective

Super effective , really?

Smart code

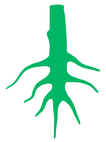
Fixing all the bugs

Adding features

Never stop

Live and breathe code

Don’t touch



Not root problem

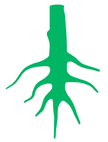
Never used

Context switching

Broader view

Efficient

Don’t touch



Not root problem

Never used

Context switching

Broader view

Efficient

Effective?

Harder

Smarter

The Effective 🚀 Developer

Work Smarter 💡, Not Harder 💪

Team Code

To be more effective

Writing good code

takes time

Writing good code

The machine

doesn’t care ...but humans do

Writing good code

Read code

more often than

write code

```
class clc {
    public static final double RATES_OF_I_TAX=10;
    private static final double RATES_OF_H_INS= 1.5;
  protected static final double RATES_OF_S_INS= 7;
  public static void main(String args[]){
  int grosssalary=1000;
  int net=clc.calc(grosssalary);
  clc.printincomeinformation(grosssalary, net);
  }
  public static int calc (int salary){
    double deductiona= salary*(RATES_OF_I_TAX/ 100);
    double deductionb=salary*(RATES_OF_H_INS/100);

    double deductionc = salary  * (RATES_OF_S_INS/ 100);
    int tmp=salary -(int)deductiona-(int)deductionb-(int)deductionc;
  return tmp;
  }
  public static void printincomeinformation(int allsalary, int homepay)
  {
    System.out.println("Gross income: " + allsalary + " doller\nNet income:   " + homepay + "doller\n");

    if (3000<allsalary)
    {
      System.out.println("You get great salary!\nDo your best at work!\n");;


    };
    if (3000>=allsalary) {
      System.out.println("You're glowing up now!\nDo your best at work!\n");
    }
  }
}
```

```
class clc {
    public static final double RATES_OF_I_TAX=10;
    private static final double RATES_OF_H_INS= 1.5;
  protected static final double RATES_OF_S_INS= 7;
  public static void main(String args[]){
  int grosssalary=1000;
  int net=clc.calc(grosssalary);
  clc.printincomeinformation(grosssalary, net);
  }
  public static int calc (int salary){
    double deductiona= salary*(RATES_OF_I_TAX/ 100);
    double deductionb=salary*(RATES_OF_H_INS/100);

    double deductionc = salary  * (RATES_OF_S_INS/ 100);
    int tmp=salary -(int)deductiona-(int)deductionb-(int)deductionc;
  return tmp;
  }
  public static void printincomeinformation(double allsalary, double homepay)
  {
    System.out.println("Gross income: " + allsalary + " doller\nNet income:   " + homepay + "doller\n");

    if (3000<allsalary)
    {
      System.out.println("You get great salary!\nDo your best at work!\n");


    };
    if (3000>=allsalary) {
      System.out.println("You're glowing up now!\nDo your best at work!\n");
    }
  }
}
```

I don’t understand it 🤯

▶ ohh…. I wrote it 🤦

```
class clc {
    public static final double RATES_OF_I_TAX=10;
    private static final double RATES_OF_H_INS= 1.5;
  protected static final double RATES_OF_S_INS= 7;
  public static void main(String args[]){
  int grosssalary=1000;
  int net=clc.calc(grosssalary);
  clc.printincomeinformation(grosssalary, net);
  }
  public static int calc (int salary){
    double deductiona= salary*(RATES_OF_I_TAX/ 100);
    double deductionb=salary*(RATES_OF_H_INS/100);

    double deductionc = salary  * (RATES_OF_S_INS/ 100);
    int tmp=salary -(int)deductiona-(int)deductionb-(int)deductionc;
  return tmp;
  }
  public static void printincomeinformation(int allsalary, int homepay)
  {
    System.out.println("Gross income: " + allsalary + " doller\nNet income:   " + homepay + "doller\n");

    if (3000<allsalary)
    {
      System.out.println("You get great salary!\nDo your best at work!\n");;


    };
    if (3000>=allsalary) {
      System.out.println("You're glowing up now!\nDo your best at work!\n");
    }
  }
}
```

```
class clc {
    public static final double RATES_OF_I_TAX=10;
    private static final double RATES_OF_H_INS= 1.5;
  protected static final double RATES_OF_S_INS= 7;
  public static void main(String args[]){
  int grosssalary=1000;
  int net=clc.calc(grosssalary);
  clc.printincomeinformation(grosssalary, net);
  }
  public static int calc (int salary){
    double deductiona= salary*(RATES_OF_I_TAX/ 100);
    double deductionb=salary*(RATES_OF_H_INS/100);

    double deductionc = salary  * (RATES_OF_S_INS/ 100);
    int tmp=salary -(int)deductiona-(int)deductionb-(int)deductionc;
  return tmp;
  }
  public static void printincomeinformation(int allsalary, int homepay)
  {
    System.out.println("Gross income: " + allsalary + " doller\nNet income:   " + homepay + "doller\n");

    if (3000<allsalary)
    {
      System.out.println("You get great salary!\nDo your best at work!\n");;


    };
    if (3000>=allsalary) {
      System.out.println("You're glowing up now!\nDo your best at work!\n");
    }
  }
}
```

```
class clc {
    public static final double RATES_OF_I_TAX=10;
    private static final double RATES_OF_H_INS= 1.5;
  protected static final double RATES_OF_S_INS= 7;
  public static void main(String args[]){
  int grosssalary=1000;
  int net=clc.calc(grosssalary);
  clc.printincomeinformation(grosssalary, net);
  }
  public static int calc (int salary){
    double deductiona= salary*(RATES_OF_I_TAX/ 100);
    double deductionb=salary*(RATES_OF_H_INS/100);

    double deductionc = salary  * (RATES_OF_S_INS/ 100);
    int tmp=salary -(int)deductiona-(int)deductionb-(int)deductionc;
  return tmp;
  }
  public static void printincomeinformation(int allsalary, int homepay)
  {
    System.out.println("Gross income: " + allsalary + " doller\nNet income:   " + homepay + "doller\n");

    if (3000<allsalary)
    {
      System.out.println("You get great salary!\nDo your best at work!\n");;


    };
    if (allsalary<=3000) {
      System.out.println("You're glowing up now!\nDo your best at work!\n");
    }
  }
}
```

```
class clc {
  public static final double RATES_OF_I_TAX=10;
  private static final double RATES_OF_H_INS= 1.5;
   protected static final double RATES_OF_S_INS= 7;
   public static void main(String args[]){
int grosssalary=1000;
int net=clc.calc(grosssalary);
   clc.printincomeinformation(grosssalary, net);
   }
   public static int calc (int salary){
     double deductiona= salary*(RATES_OF_I_TAX/ 100);
     double deductionb=salary*(RATES_OF_H_INS/100);

     double deductionc = salary  * (RATES_OF_S_INS/ 100);
     int tmp=salary -(int)deductiona-(int)deductionb-(int)deductionc;
   return tmp;
   }
   public static void printincomeinformation(int allsalary, int homepay)
   {
     System.out.println("Gross income: " + allsalary + " doller\nNet income:   " + homepay + "doller\n");

     if (3000<allsalary)
     {
       System.out.println("You get great salary!\nDo your best at work!\n");;

     };
     if (allsalary<=3000) {
       System.out.println("You're glowing up now!\nDo your best at work!\n");
     }
   }
}
```

```
class clc {
    public static final double RATES_OF_I_TAX=10;
    private static final double RATES_OF_H_INS= 1.5;
  protected static final double RATES_OF_S_INS= 7;
  public static void main(String args[]){
  int grosssalary=1000;
  int net=clc.calc(grosssalary);
  clc.printincomeinformation(grosssalary, net);
  }
  public static int calc (int salary){
    double deductiona= salary*(RATES_OF_I_TAX/ 100);
    double deductionb=salary*(RATES_OF_H_INS/100);

    double deductionc = salary  * (RATES_OF_S_INS/ 100);
    int tmp=salary -(int)deductiona-(int)deductionb-(int)deductionc;
  return tmp;
  }
  public static void printincomeinformation(int allsalary, int homepay)
  {
    System.out.println("Gross income: " + allsalary + " doller\nNet income:   " + homepay + "doller\n");

    if (3000<allsalary)
    {
      System.out.println("You get great salary!\nDo your best at work!\n");;


    };
    if (allsalary<=3000) {
      System.out.println("You're glowing up now!\nDo your best at work!\n");
    }
  }
}
```

```
class clc {
    public static final double RATES_OF_I_TAX=10;
    private static final double RATES_OF_H_INS= 1.5;
  protected static final double RATES_OF_S_INS= 7;
  public static void main(String args[]){
  int grosssalary=1000;
  int net=clc.calc(grosssalary);
  clc.printincomeinformation(grosssalary, net);
  }
  public static int calc (int salary){
    double deductiona= salary*(RATES_OF_I_TAX/ 100);
    double deductionb=salary*(RATES_OF_H_INS/100);

    double deductionc = salary  * (RATES_OF_S_INS/ 100);
    int tmp=salary -(int)deductiona-(int)deductionb-(int)deductionc;
  return tmp;
  }
  public static void printincomeinformation(int allsalary, int homepay)
  {
    System.out.println("Gross income: " + allsalary + " doller\nNet income:   " + homepay + "doller\n");

    if (3000<allsalary)
    {
      System.out.println("You get great salary!\nDo your best at work!\n");;


    };
    if (allsalary<=3000) {
      System.out.println("You're glowing up now!\nDo your best at work!\n");
    }
  }
}
```

```
class clc {
    public static final double RATES_OF_I_TAX=10;
    private static final double RATES_OF_H_INS= 1.5;
  protected static final double RATES_OF_S_INS= 7;
  public static void main(String args[]){
  int grosssalary=1000;
  int net=clc.calc(grosssalary);
  clc.printincomeinformation(grosssalary, net);
  }
  public static int calc (int salary){
    double deductiona= salary*(RATES_OF_I_TAX/ 100);
    double deductionb=salary*(RATES_OF_H_INS/100);

    double deductionc = salary  * (RATES_OF_S_INS/ 100);
    int tmp=salary -(int)deductiona-(int)deductionb-(int)deductionc;
  return tmp;
  }
  public static void printincomeinformation(int allsalary, int homepay)
  {
    System.out.println("Gross income: " + allsalary + " doller\nNet income:   " + homepay + "doller\n");

    if (3000<allsalary)
    {
      System.out.println("You get great salary!\nDo your best at work!\n");;


    };
    if (allsalary<=3000) {
      System.out.println("You're glowing up now!\nDo your best at work!\n");
    }
  }
}
```

```
class Salary {

    private static final double RATES_OF_INCOME_TAX = 10;
    private static final double RATES_OF_HEALTH_INSURANCE = 1.5;
    private static final double RATES_OF_SOCIAL_INSURANCE = 7;

    public static void main(String args[]) {

        int grossSalary = 1000;
        int netIncome = Salary.calc(grossSalary);

        Salary.printIncomeInformation(grossSalary, netIncome);
    }

    private static int calc(int salary) {

        double deductionIncomeTax = salary * (RATES_OF_INCOME_TAX / 100);
        double deductionHeathInsurance = salary * (RATES_OF_HEALTH_INSURANCE / 100);
        double deductionSocialInsurance = salary * (RATES_OF_SOCIAL_INSURANCE / 100);

        int reducedSalery = salary - (int) deductionIncomeTax - (int) deductionHeathInsurance - (int) deductionSocialInsurance;

        return reducedSalery;
    }

    private static void printIncomeInformation(int grossSalary, int netIncome) {

        System.out.println("Gross income: " + grossSalary + " dollar\nNet income:   " + netIncome + "dollar\n");

        if (grossSalary > 3000) {
            System.out.println("You get great salary!\nDo your best at work!\n");
        }
        if (grossSalary <= 3000) {
            System.out.println("You're glowing up now!\nDo your best at work!\n");
        }
    }
}
```

```
class Salary {

    private static final double RATES_OF_INCOME_TAX = 10;
    private static final double RATES_OF_HEALTH_INSURANCE = 1.5;
    private static final double RATES_OF_SOCIAL_INSURANCE = 7;

    public static void main(String args[]) {

        int grossSalary = 1000;
        int netIncome = Salary.calc(grossSalary);

        Salary.printIncomeInformation(grossSalary, netIncome);
    }

    private static int calc(int salary) {

        double deductionIncomeTax = salary * (RATES_OF_INCOME_TAX / 100);
        double deductionHeathInsurance = salary * (RATES_OF_HEALTH_INSURANCE / 100);
        double deductionSocialInsurance = salary * (RATES_OF_SOCIAL_INSURANCE / 100);

        int reducedSalery = salary - (int) deductionIncomeTax - (int) deductionHeathInsurance - (int) deductionSocialInsurance;

        return reducedSalery;
    }

    private static void printIncomeInformation(int grossSalary, int netIncome) {

        System.out.println("Gross income: " + grossSalary + " dollar\nNet income:   " + netIncome + "dollar\n");

        if (grossSalary > 3000) {
            System.out.println("You get great salary!\nDo your best at work!\n");
        }
        if (grossSalary <= 3000) {
            System.out.println("You're glowing up now!\nDo your best at work!\n");
        }
    }
}
```

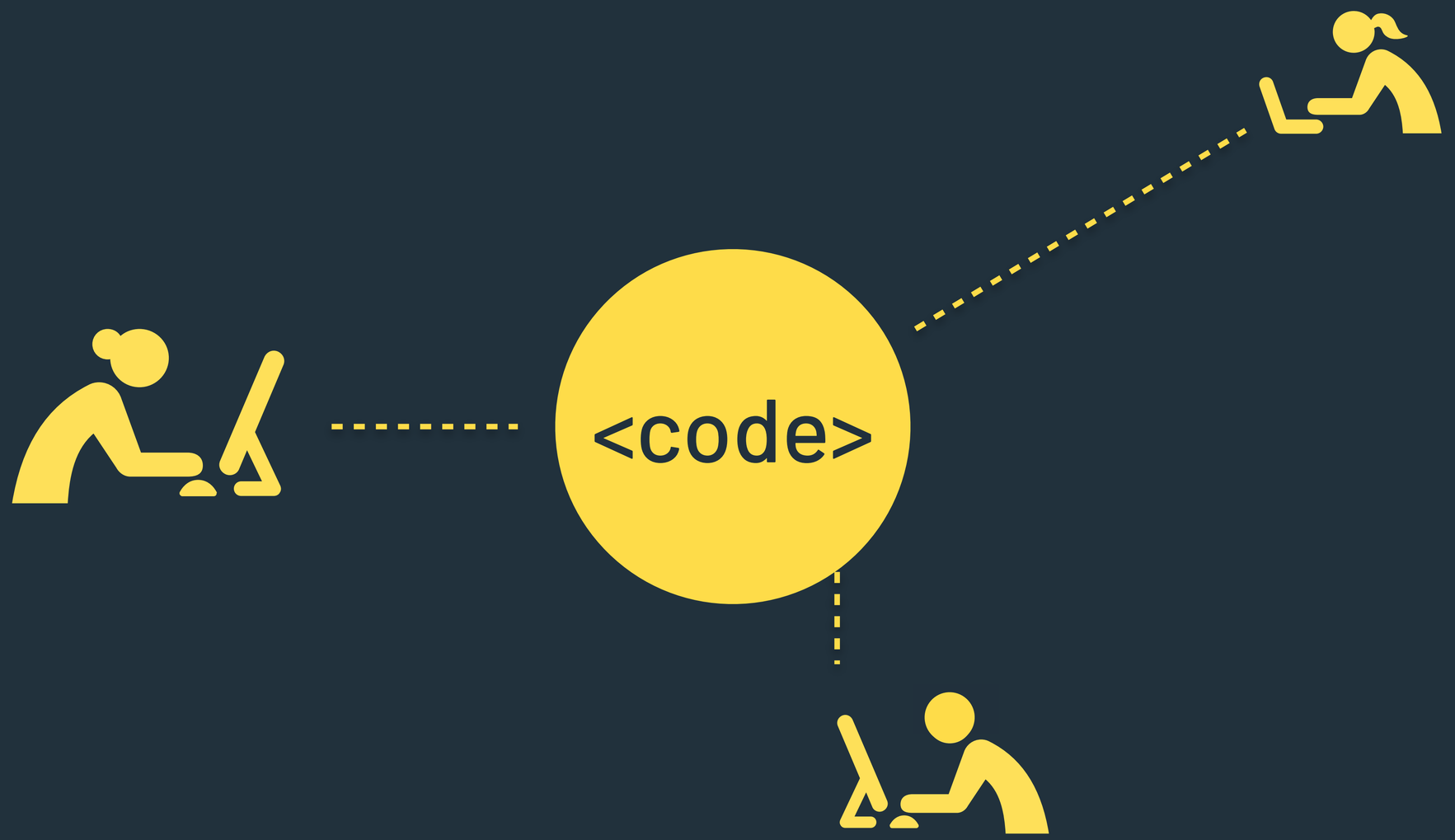
“

The huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.

Clever code

Show off

Simplicity over Cleverness


<code>

Is your code simple enough?

Junior Devs

Make the canary test

Invest time in

code reviews

to inspire jr. devs

ensuring consistent code quality

code reviews

done right

Seek to understand

Seek to understand

Criticise ideas, not people

Seek to understand

Don’t rant, make suggestions

Criticise ideas, not people

Seek to understand

Don’t rant, make suggestions

Don’t spoon feed

Criticise ideas, not people

Seek to understand

Don’t rant, make suggestions

Comment on positive things

Don’t spoon feed

Criticise ideas, not people

Seek to understand

Don’t rant, make suggestions

Comment on positive things

Don’t spoon feed

Criticise ideas, not people

Learn

To be more effective

Wow

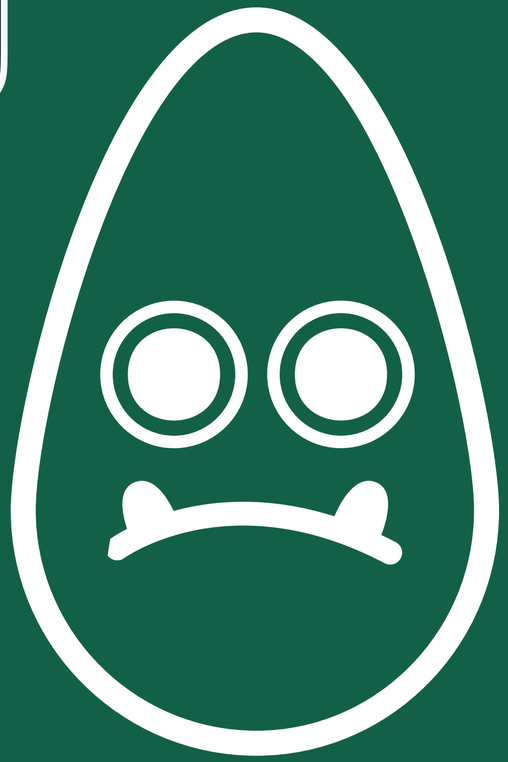
Cool

Great idea

New technology

I' ll also **survive** microservices, and serverless hype.

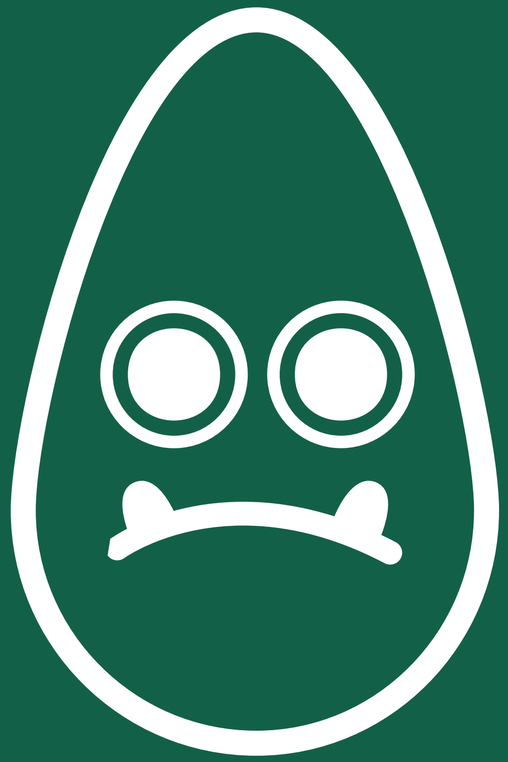
Really, a **new** front end framework, again?



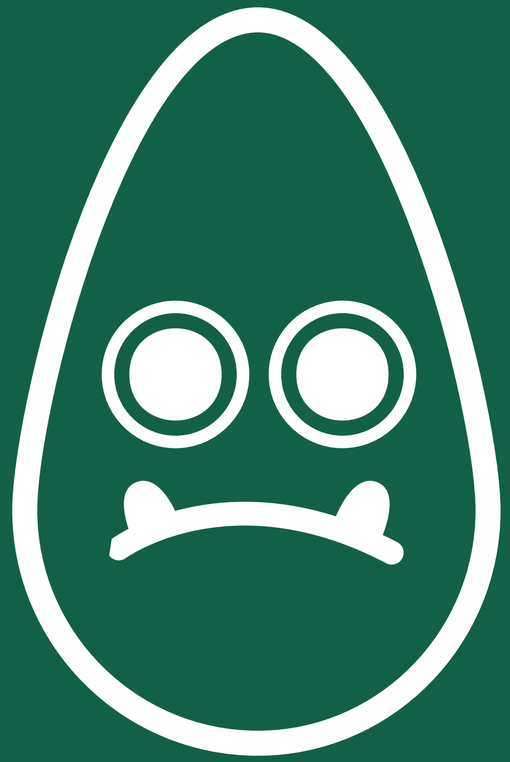
Code Reviews? Not for me

Scrum boards? I already know what's **important**.

Grumpy Developer



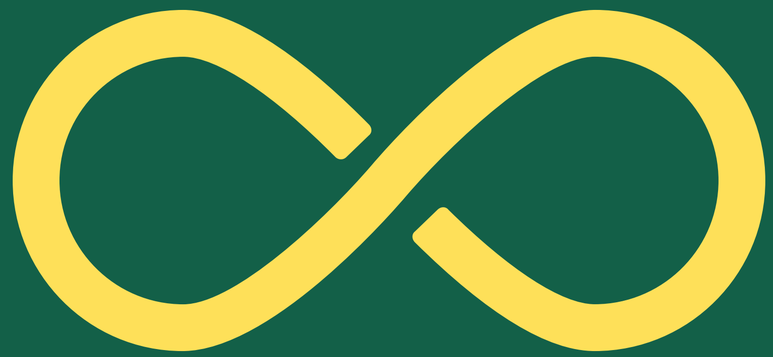
Grumpy Developer



I know how to
develop software ...now

Do you think
software development
is changing ?

Continuous **learning**




Podcasts
Audiobooks


IMPOSTER SYNDROME IS THE
INABILITY TO ACCEPT ONES
OWN ACCOMPLISHMENT AND
THE CONSTANT FEAR OF BEING
EXPOSED AS A FRAUD.

Meetups


Book club


Video
Fridays

But why learn?

All **answers** are on

1st

Try and solve the problem

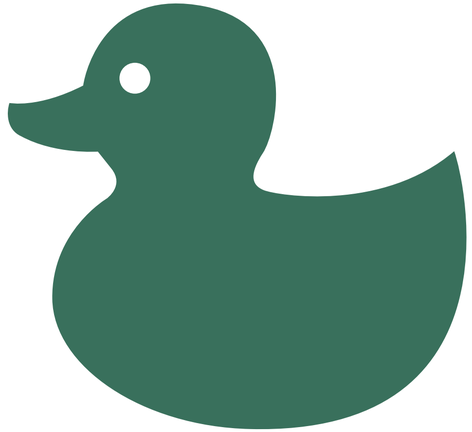
2nd

Ask for help

2nd

yourself

Ask for help



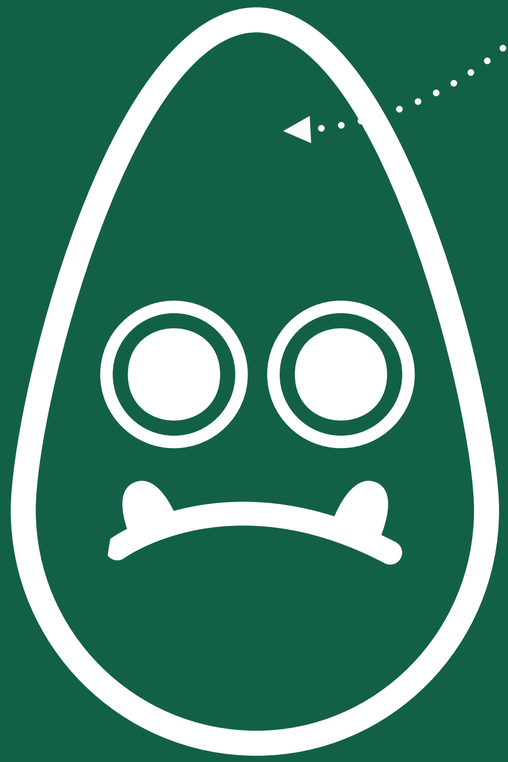
3rd

Ask for help

I don’t know

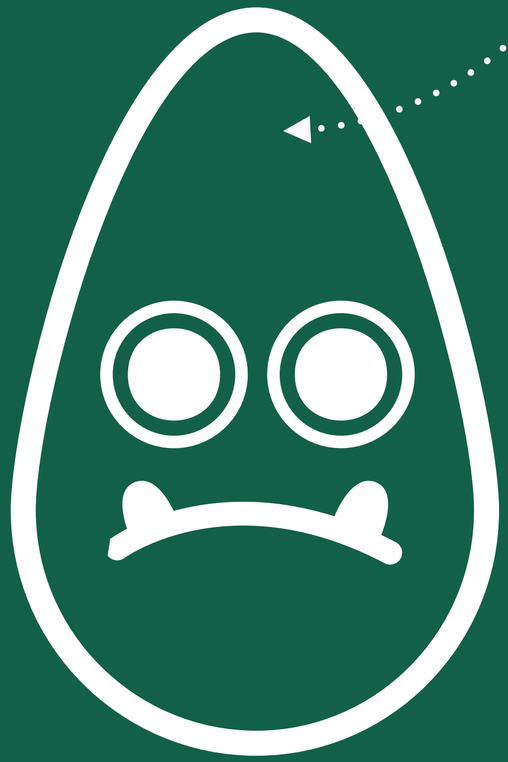
make new connections

Learn



get rid of some connections

Learn



to unlearn


Experiment
Learn
Unlearn

Experiment

To be more effective


BACKLOG
TEMMUZ
AĞUSTOS
EYLÜL
EKİM
KASIM
ARALIK
OCAK
ŞUBAT
MART
NİSAN
MAYIS
HAZİRAN
This might not be the best idea
It’s all on the board*
*The rest is in Jira

Known unknowns

Known unknowns

Better solutions?

Known unknowns

Better solutions?

New technology?

Known unknowns

Don’t be shy – just try

Better solutions?

New technology?

Missing opportunities?

Coding sessions

1.5 hour /week

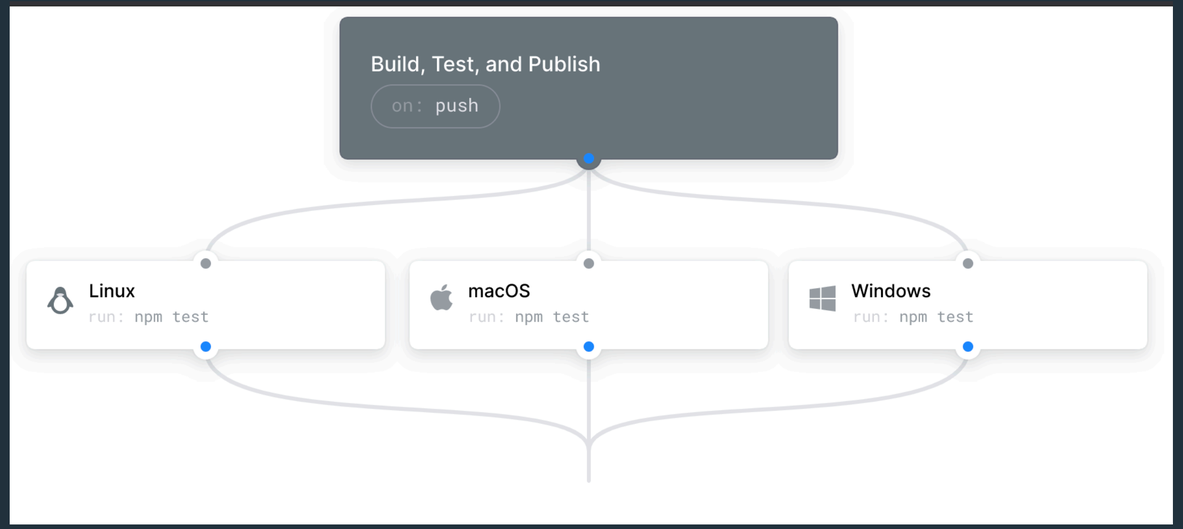
1.5 h Coding

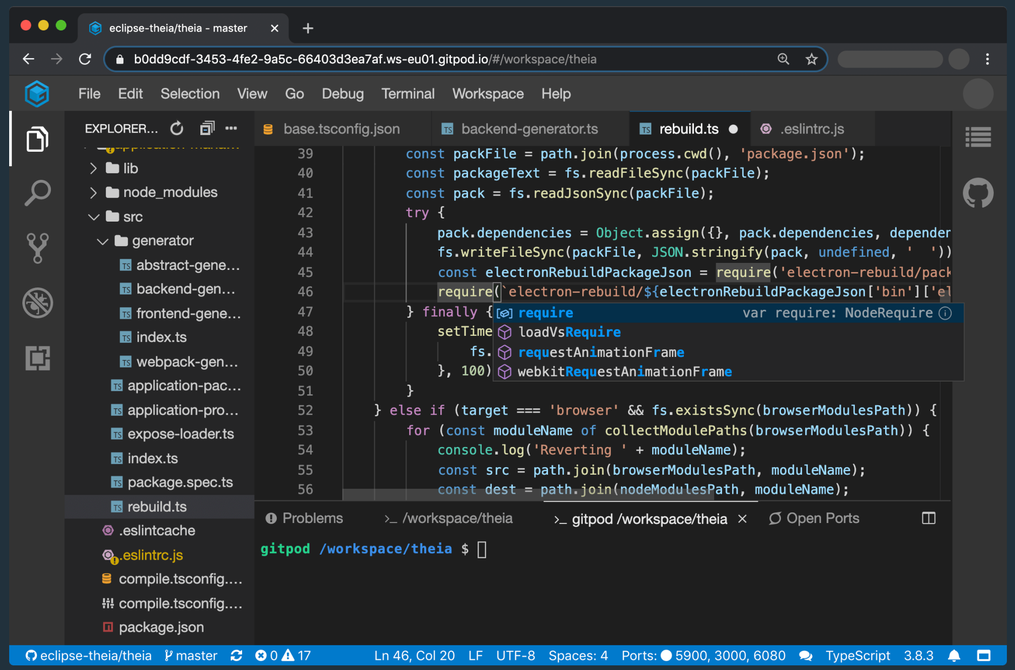
1.5 h Improving

Tools training

GitHub Actions

Gitpod





1.5 h Coding
0.5 h Training

2 h Improving

Tools training

Innovation

The problem is not the lack of **ideas**

It’s the lack of **time** to try them out

The problem is not the lack of **ideas**

It’s the lack of **time** to try them out

How much time do you need?

How much time do you need?


- BEFORE -
Planning
24h
Hackathon
- DURING -
Event
- AFTER -
Follow-up

- BEFORE -

Planning

Plan dates

Announce
and tell stories

Find your teams
aka “Pitch it”

- DURING -

Event

Remember,
time is ticking

Have fun

Get energized—
eat and drink!

- AFTER -

Follow-up

Presentations—give it all you got

Voting, celebrations, and winners announced

Get onto a product roadmap—and plan for next Hackathon

Every quarter

1.5 h Coding
0.5h Training
0.7h Hackathon

2.7 h Improving

Known knowns...

Great ideas

Paper Cuts

Innovation week

Every 8 weeks

1.5 h Coding
0.5h Training
0.7h Hackathon
5.0 h Inno Week

7.7 h Improving


Invest in
discovery time

Work on what matters

To be more effective

Why solve it?

Validation data?

What's the **problem**?

Solutions?

Project Poster

Team: *Name of project owner*
Names of team

Status: ACTIVE

Problem space

Why are we doing this?

Problem statement

What problem/need are you trying to solve/fulfil?

Impact of the problem

What's the impact of this problem on our customers and to our business?

How do we judge success?

What are the goals of the project and the success criteria by which they will be measured?

Possible solutions

High level ideas of possible solutions. (Can be filled out later)

Validation

Validation

What do we already know?

What data or insights do you have to validate this?

What do we need to answer?

What assumptions are we making that need to be validated/refuted?

What are the gaps in our understanding?

Ready to make it

What are we doing?

Succinctly describe what the project will deliver. i.e Elevator pitch.

Why will a customer want this?

How would you sell or market this to your customer?

Visualise the solution

User experience, designs, mockups or prototypes.

Scale and scope

T-shirt size estimate of the effort to make this.

As a user

I want a new statistic button

so that I can see the data sets

Jasmine

As ~~a user~~

to know the number of data sets

I want ~~a new statistic button~~

I can plan for growth

so that ~~I can see the data sets~~

Think like your users

Jasmin
Developer In Training

Lucas
Full-Stack Developer

Tina
Data Scientist

Personas

JASMIN

DEVELOPER IN TRAINING

The Tech Learner

> "Learning to program will put me on the path to a better career. MongoDB seems like the easiest data layer to learn on and it's free!"

SUMMARY

Jasmin is making a big career transition, from customer service to tech. She's learning programming on her own time while working her day job, and developing her own app as a learning tool/portfolio piece. She's especially conscious of price and ease of use (in that order) since she's paying for everything herself.

EXPERIENCE

Current Role
2 Years

Technology Roles
6 Months

MongoDB Experience
Beginner

Primary MongoDB Learning Tools
MongoDB University Certificate in progress
Documentation

BUYING POWER
Decider

COMPANY TYPE
Freelance (Individual)

WHAT I DO
- Juggle my day job with learning to code in my free time
- Install, develop, and troubleshoot the tech I choose
- Manage all aspects of my project

GOALS
- Ultimately transition to a career as a Full-Stack Developer
- Troubleshoot my own IT issues at an advanced leve
- Build an app to manage call center schedules as a portfolio piece

WHY I LIKE MONGODB
- Atlas, Compass, and MongoDB University are free
- Atlas is easy to set up and get started with on my own
- Plentiful third-party tutorials available

PAIN POINTS
- Learning on my own without a good support network
- MongoDB University courses require prior programming and database knowledge
- Understanding what to do next after completing each MongoDB University course

WHAT I USE
- **Atlas** *Beginner*
- **Compass** *Beginner*
- **Stitch** *Beginner*

I MIGHT ALSO USE

LUCAS

FULL-STACK DEVELOPER

The Advocate

> "I love building elegant software. I always try to keep business logic in the code instead of third-party tools so they're easy to change, even if it creates more work."

SUMMARY

Lucas is a MongoDB superfan. He brought MongoDB into his company, follows Eliot on Twitter and attends MongoDB World annually. He's very interested in the philosophy behind software tools, and prefers those that are open-source and responsive to customers of all sizes.

EXPERIENCE

Current Role
1 Year

Technology Roles
8 Years

MongoDB Experience
Advanced

Primary MongoDB Learning Tools
MongoDB University Certificate

BUYING POWER
Influencer

COMPANY TYPE
Small Business (1-50 employees)

WHAT I DO
- Design, develop, and implement custom applications
- Write clean code using a wide range of technology and programming languages
- Fix bugs and rewrite old software

GOALS
- Deliver 5-star apps that elevate the company's reputation
- Create bulletproof data security with excellent UI and solid technical foundation
- Co-found my own startup in a few years

WHY I LIKE MONGODB
- Elegant data platform
- Stitch (serverless platform) for app development
- Great performance even with high query volume
- Easy, convenient, and gets out of my way

PAIN POINTS
- Data integrity concerns when handing off data to someone else
- Finding documentation for new features
- Getting stuck with legacy technology or infrastructure

WHAT I USE
- **Atlas** *Advanced*
- **Compass** *Advanced*
- **Stitch** *Intermediate/Advanced*

I MIGHT ALSO USE

TINA

DATA SCIENTIST

The Statistician

> "We would like to do machine learning. We're currently working on that, but we're learning, too. So the main thing for us is data integration and data visualization."

SUMMARY

Tina analyzes data from several databases, then builds clean data sets when needed, usually in SQL. She can write scripts to query MongoDB but she learned how to work on relational databases, so she's working on shifting her mindset to a document model. Visualizations are important to Tina as she works with data from various sources to provide insights.

EXPERIENCE

Current Role
4 Years

Relevant Experience
8 Years

MongoDB Experience
Beginner / Intermediate

Primary MongoDB Learning Tools
Documentation

BUYING POWER
Consulted

COMPANY TYPE
Corporate (50-1000 employees)

WHAT I DO
- Use scripting to combine data from all sources into a single pipeline
- Identify opportunities for increased efficiency and improved operations
- Analyze the data using statistical methods
- Create visualizations using Charts, Excel, or Tableau
- Provide key performance indicator (KPI) data to key stakeholders

GOALS
- Create a shared understanding of how the company is meeting its goals from a data-driven perspective
- Gain actionable insights from the data

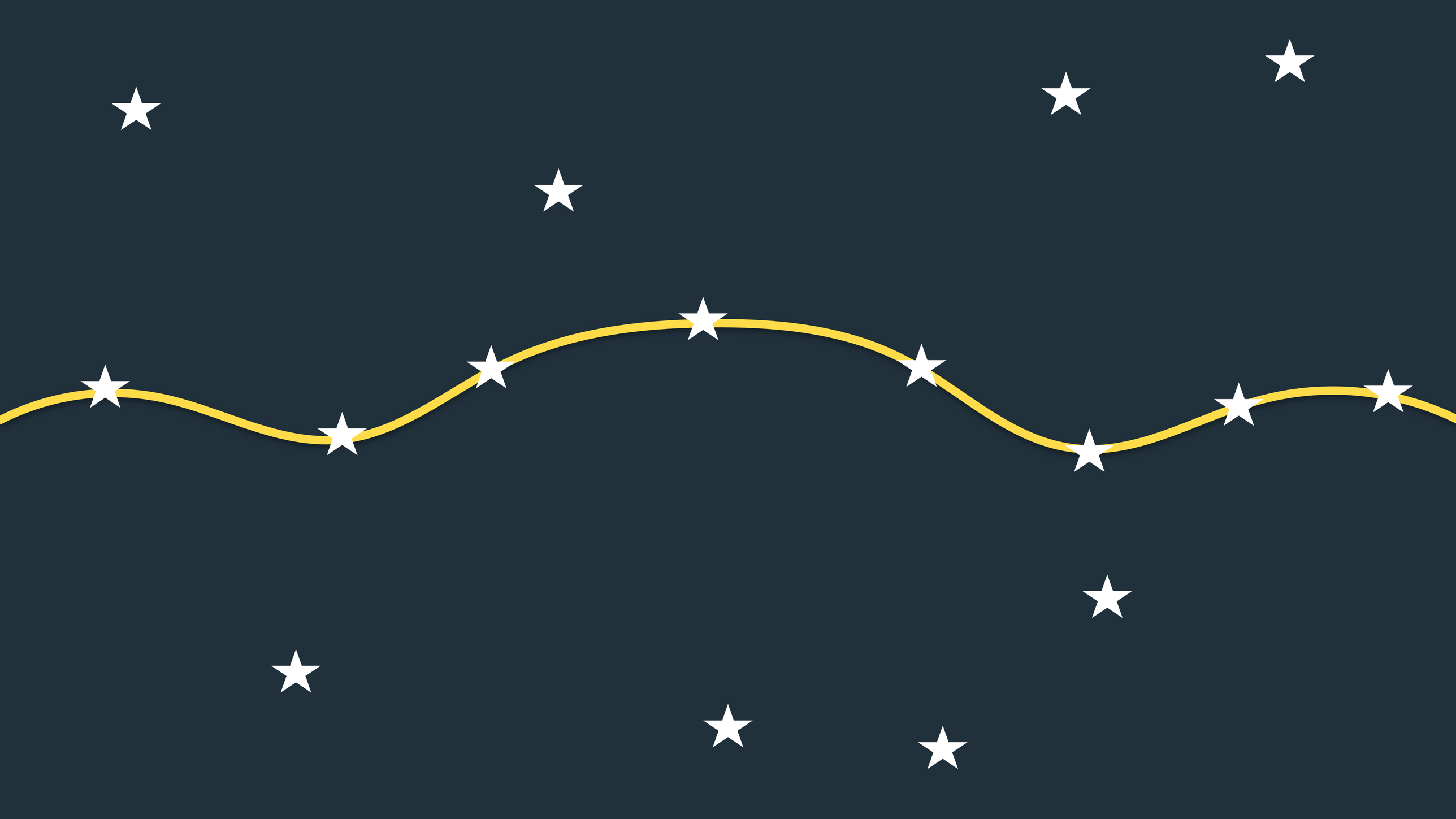
WHY I LIKE MONGODB
- Compatible with Python, R, and other scripting languages
- Works well with Jupyter Notebook for sandbox-type exploration
- Easy to integrate different types of data from multiple sources

PAIN POINTS
- Large and cross-collection queries are time- and resource- consuming
- Data is in multiple places and formats
- Gap in visualization tools that predict based on data trends

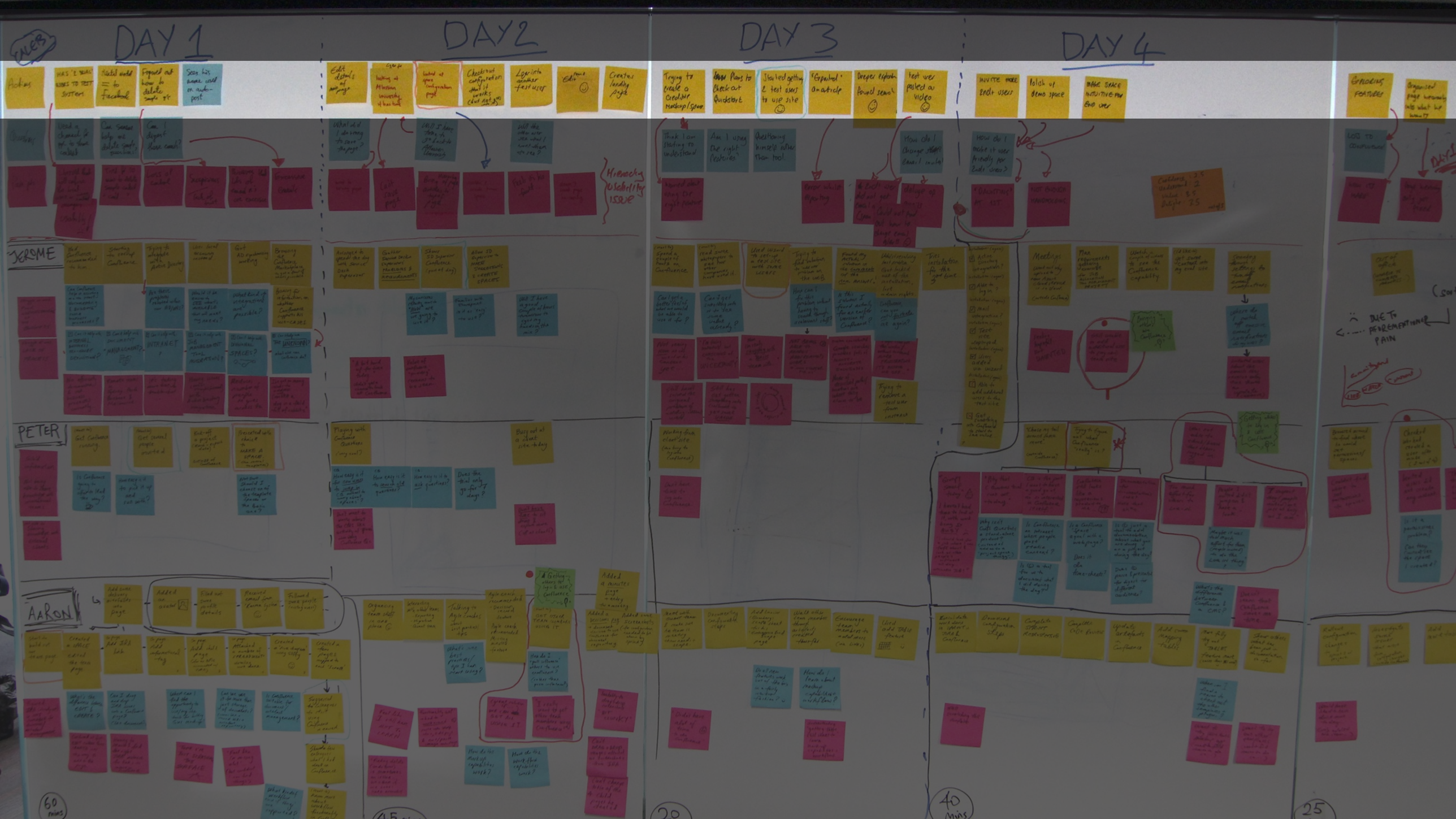
WHAT I USE
- **BI Connector** *Intermediate*
- **Compass** *Beginner*
- **MongoDB on GCP** *Beginner*
- **Charts** *Beginner*

I MIGHT ALSO USE

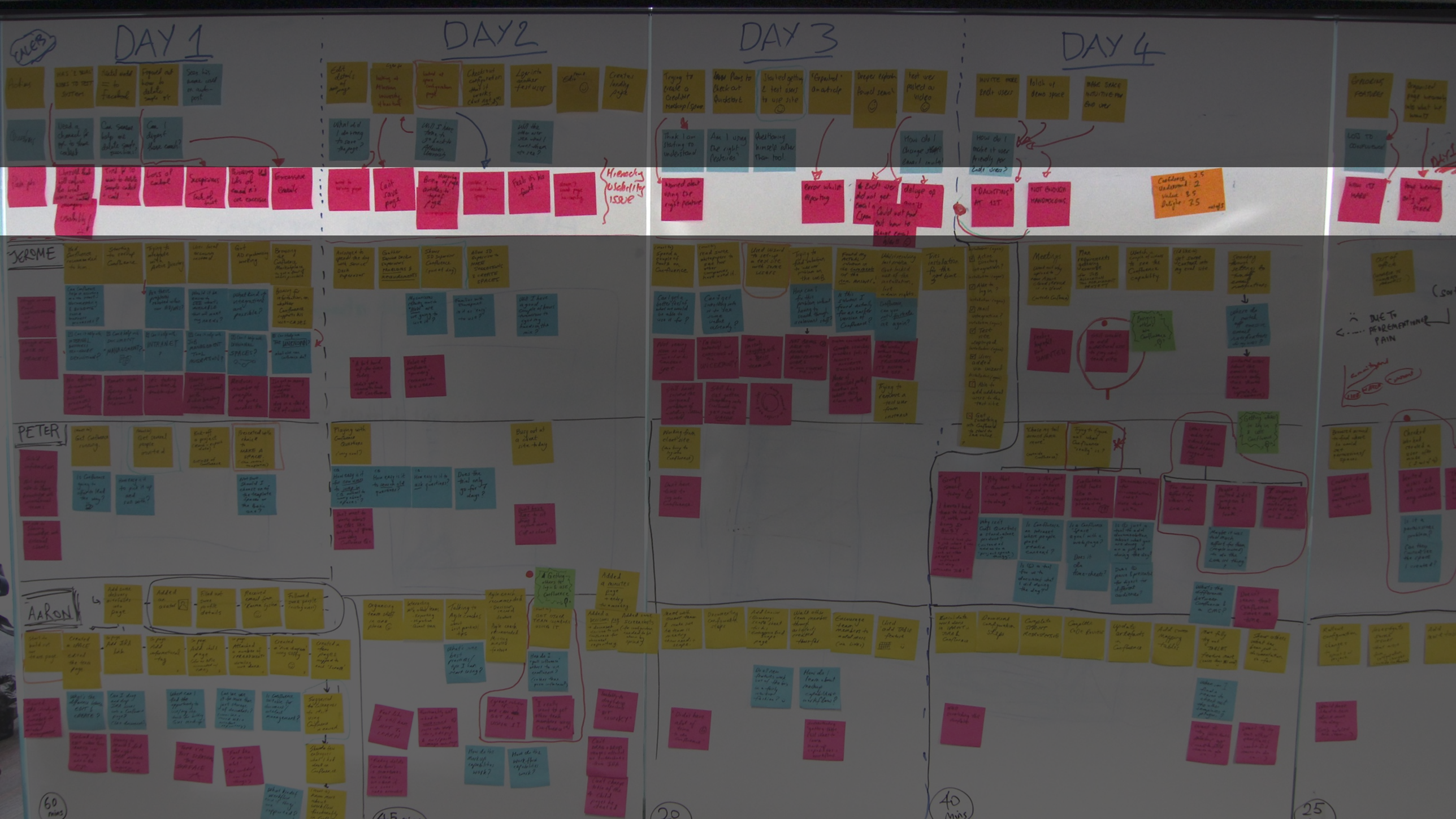

Project Poster
Team:
Status:
ACTIVE
Why are we doing this?
How do we judge success?

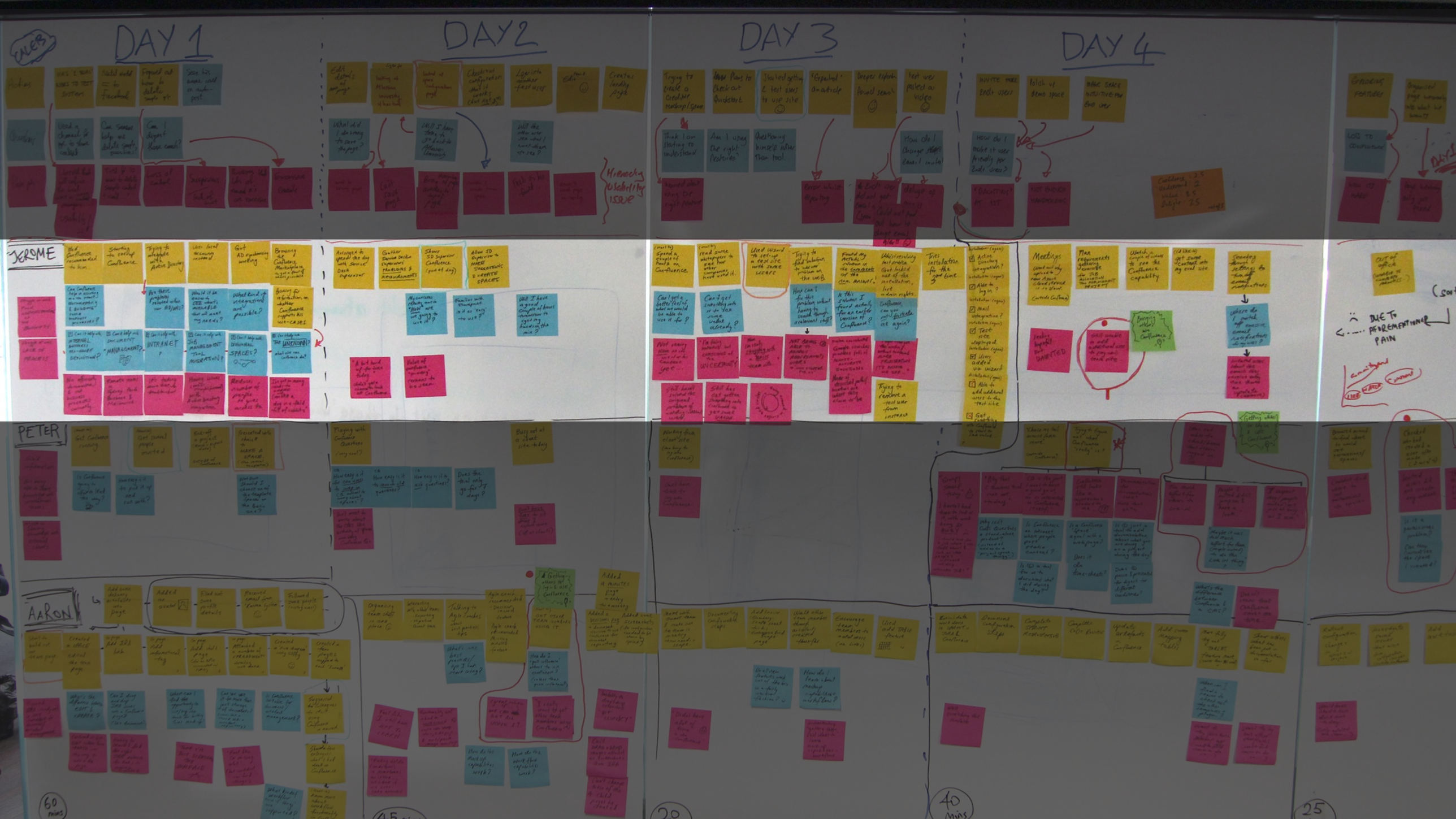



DAY 1
DAY 2
DAY 3
DAY 4
JEROME
PETER
AARON
Hierarchy Usability issue
60 mins
40 mins
25


DAY 1
DAY 2
DAY 3
DAY 4
JEROME
PETER
AARON
Hierarchy usability issue
DIET FERMENTATION PAIN
60 mins
40 mins
25


DAY 1
DAY 2
DAY 3
DAY 4
JEROME
PETER
AARON
Think I am starting to understand
Am I using the right features?
Questioning himself rather than tool.


DAY 1
DAY 2
DAY 3
DAY 4
JEROME
PETER
AARON
Hierarchy Usability ISSUE
60 mins
40 mins
25


DAY 1
DAY 2
DAY 3
DAY 4
JEROME
PETER
AARON
Hierarchy usability issue
DUE TO PERFORMANCE PAIN
60 mins
40 mins
25

Code

Product

Code

I wish engineers spent less time understanding the problem and more time coding.

---

said nobody, ever

Product

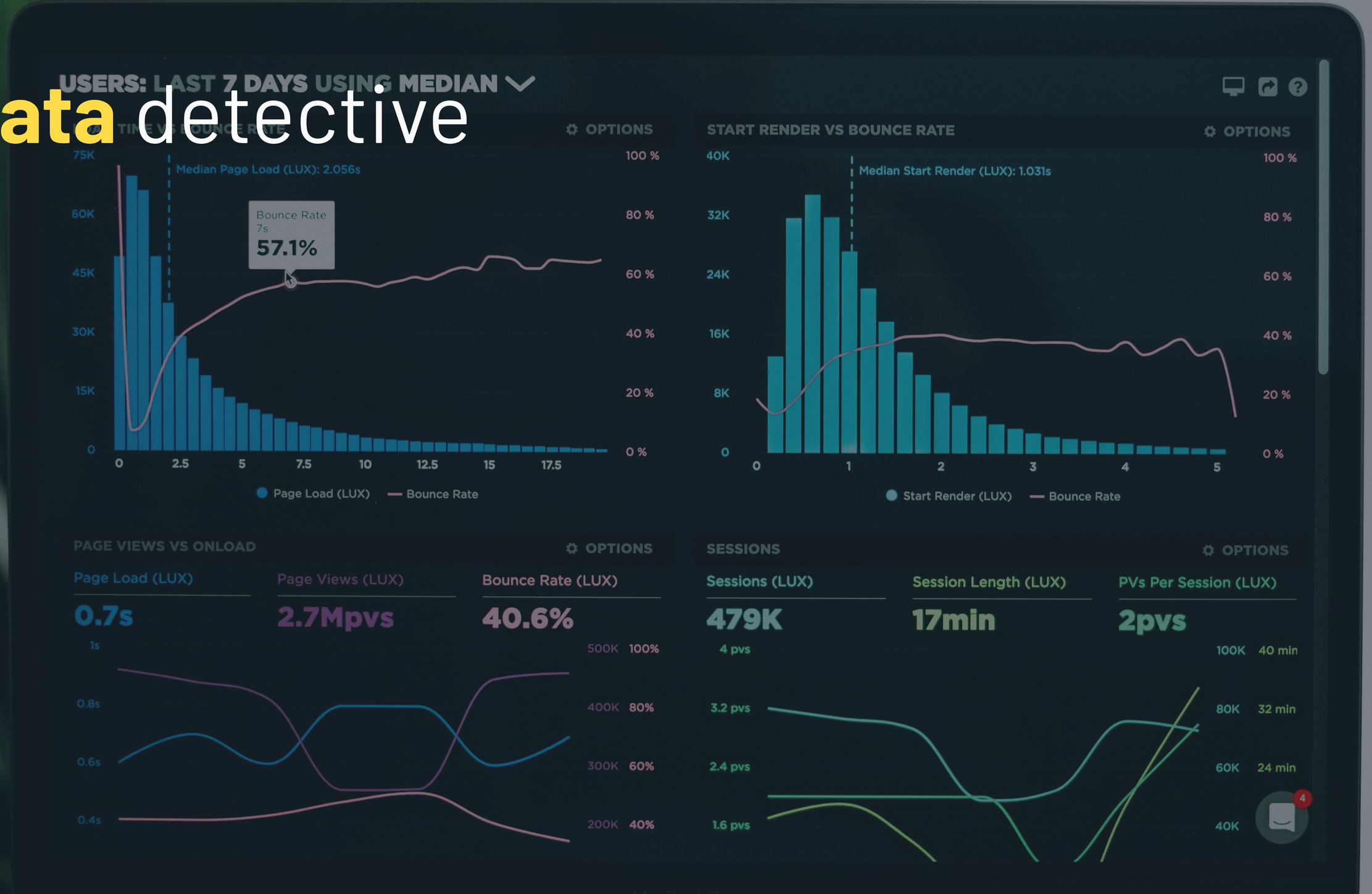
Code

Product **Engineer**

Hallway testing

Developer on support

Data detective



- Team code
- Learn
- Experiment
- What matters

The Effective 🚀 Developer

Team code

Learn

Experiment

What matters

Effective

Team code

Learn

Experiment

What matters

Efficient

Automation

Tools

Processes

Efficient

Efficient

Get more stuff done

Effective

Get the right stuff done

Efficient

Get more stuff done

Effective

Get the right stuff done

Output

Efficient

Outcome

Effective

Thank you

Think **outcome** first, be **effective**, and get **s%*t** done

svenpet.com/talks

@svenpet