

7+ Painful and Funny Tech Lead's Stories to Level Up Your Productivity

Marek Dominiak



7+ Painful and Funny Tech Lead's Stories to Level Up Your Productivity

Marek Dominiak
Trainitek / Mintra

Platinum Sponsors





Marek Dominiak

I'm run training in EventStorming, Architecture, DDD, and TDD

17 years of professional experience

Co-owner and instructor at



Hands-on Architect
and Team Lead at



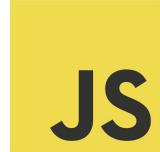
<https://www.linkedin.com/in/marekdominiak>

What does it mean to be a Senior Developer / Tech Lead?



Story no. 1: The first code review

- Year 2008
- 30-people team
- Fresh and new Java 6
- JavaServer Faces
- JavaScript
- Spring 2.x, Hibernate 3.x
- We even had to support ...
- Internet Explorer 6 ...



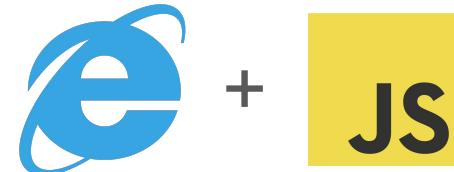
Story no. 1: The Bug

- Bug reported in a JavaScript function on a page
- Found a solution on a similar page
- I applied a very known solution ...
- Copy-Paste Design Pattern
- I was confident in the quick fix
- And was happy not to play with IE6 and JavaScript



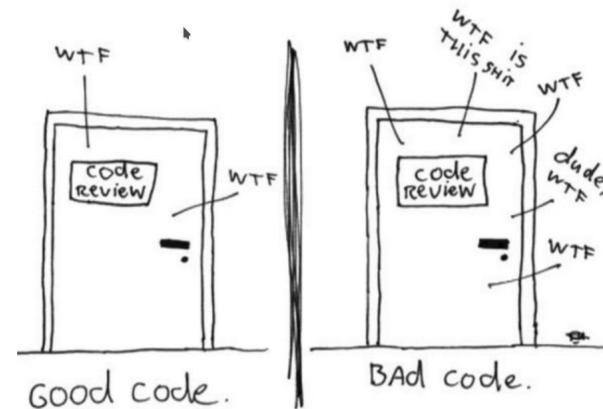
```
function onEdition()
{
    // No right mouse edit + preload
    if( encodeURIComponent(document.location).search(/\?preload\b/) != -1 ) return;
    // /&reload/
    if( !top.pageName.match(/\?Discussion.\?Traduction/) ) return;
    var diff = new Array();
    var status; var pecTraduction; var pecRelecture;
    var avancementTraduction; var avancementRelecture;
    /* ***** Parser ***** */
    params = document.location.search.substring(1, document.location.search.length);
    gtb().split('&');
    var i = 0;
    var tmp; var name;
    while ( i < params.length )
    {
        tmp = params[i].split('=');
        name = tmp[0];
        switch( name )
        {
            case 'status':
                status = tmp[1];
                break;
            case 'pecTraduction':
                pecTraduction = tmp[1];
                break;
            case 'pecRelecture':
                pecRelecture = tmp[1];
                break;
            case 'avancementTraduction':
                avancementTraduction = tmp[1];
                break;
            case 'avancementRelecture':
                avancementRelecture = tmp[1];
                break;
        }
    }
}
```

Ctrl-C -> Ctrl-V



Story no. 1: The review

- Company introduced post-factum code reviews
- Main architect / tech lead reviewed my code
- Solution was okeish, but prone to future errors
- I got defensive and pointed at the original author
- I thought I defended myself quite well
- Review has ended



Story no. 1: The verdict

- After a day I got email with evaluation
- I have got
- I was really shocked
- I was furious
- After I picked up my jaw up from the floor ...
and gathered some courage I approached the Architect

2 / 5



Story no. 1: Lesson Learned

- Architect explained me this:
- Take responsibility for the code you create
- Take responsibility for the code you copied
- Even if you have copied it
- Now it's yours
- It took me some hours to understand it but ...



Story no. 1: Conclusion and Productivity

- I have adopted this mindset and it helped in a various of ways
- Taking responsibility is a key characteristic of a senior developer / tech lead
- This mindset improves work quality and productivity
- Others start to put more trust in you and value your work more
- You are not your CODE - keep emotions away
- Take responsibility for the code you have generated with AI

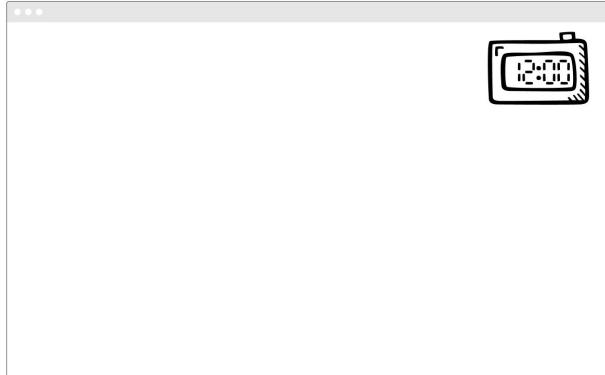


What does it mean to be a
Senior Developer / Tech Lead?

How would you answer this question

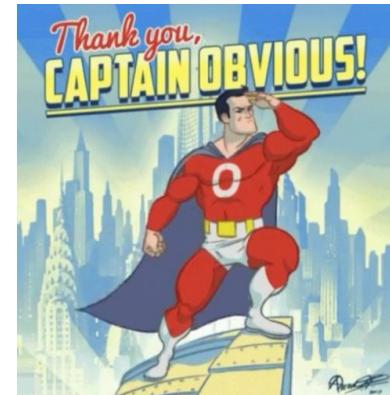
Would you invest six months of two senior developers' time to implement date-time library supporting time zones?

It depends



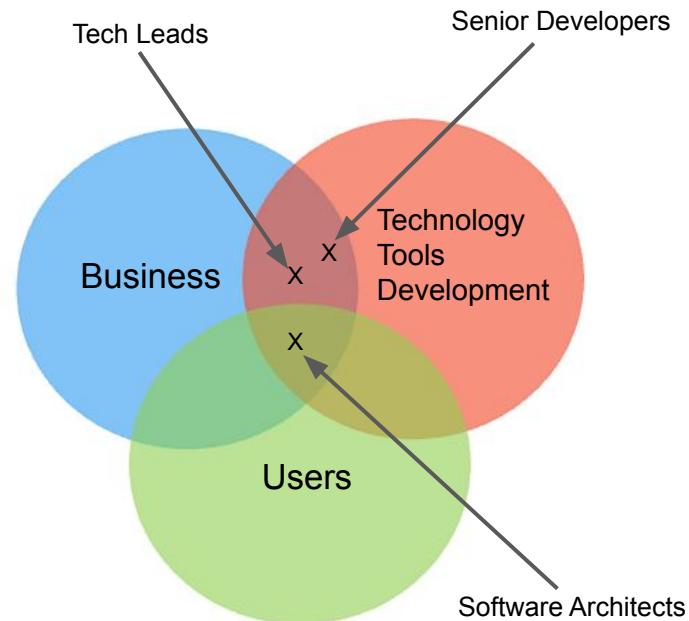
Senior Developer / Tech Lead characteristics

- Often answers to questions with:
- “*It Depends*”
- Provides information on **what** it depends on
- Can explain potential different scenarios
- Sees a problem from different perspectives



What does it mean to be a Senior Developer / Tech Lead / Architect?

- Leadership
- Skill (applying knowledge)
- Being glue between "business" and technology
- Having impact



Business and Communication

The App

Story no. 2: The App

- Year 2014
- Team of 4 developers
- We needed to create an App to serve our content
- Technologies: Cordova / PhoneGap on devices
- Support needed for Android, iPhone, possibly Windows
- OAuth 2.0
- Backend in Java 8, Spring 4.x, Hibernate 4.x



Story no. 2: Challenges

- Encrypted content on devices
- Downloading with resume if dropped
- Needed to be fast on all platforms
- Impl of HTTP server for both Android and iPhone
- Cordova plugins development
- OAuth 2.0 was a bit more complex than Basic Auth



Story no. 2: Results

- Limited use of the App
- Difficulty in maintaining the App
- Unnecessary complexity due to weird requirements
- Project only after 2.5 years was hard to maintain
- Feeling of wasted efforts
- Few attempts to revive the App failed



Story no. 2: New hope

- 2022 a new team appointed
- Based on the insights from the old App developers
- And their older APIs
- Simplifying the requirements
- And removing some of them
- In just a few months the App reimplemented



Story no. 2: Lesson Learned

- Challenge requirements
- By default say “NO” :)
- Don’t be a “YES man”
- Question assumptions
- Ask question “Why do we need this feature?”



Story no. 2: Productivity

- Being sure of what is really needed is IMPORTANT
- Avoid wasted effort
- Saying “NO” is hard thing to do, but can’t be overstated



Reports in the new module

Story no. 3: Reports in the new module

- Implementation of reports in the new module in the system
- Team of 4 developers
- Around 6-7 different reports to implement
- Backend in Java 17, Spring Boot 2.x, Hibernate 5.x
- Quite straightforward

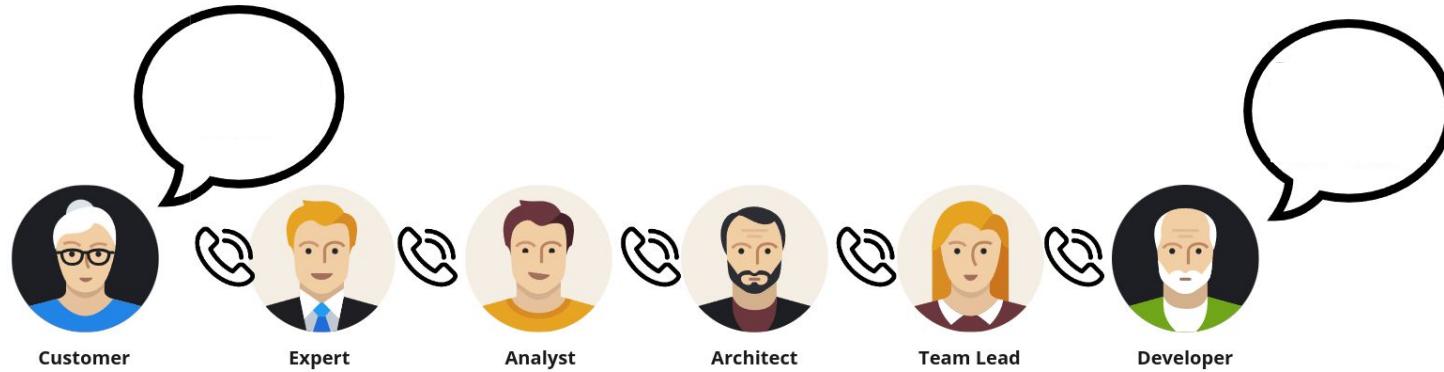


Story no. 3: The challenges

- After initial implementation we got feedback
- “This is not what we want”
- So we improved the initial reports
- We got feedback
- “This is not quite what we want”
- And those iterations continued ...
- Around 6-7 times



Story no. 3: The telephone game



Why are we continuing this madness?

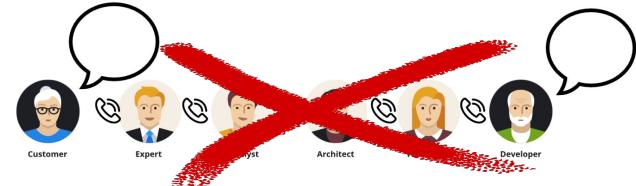
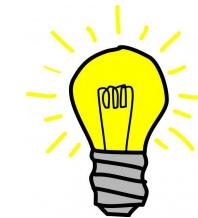
Story no. 3: Two meetings

- Finally we have arranged the meeting altogether
- In sum around 2 hours spent
- We cut off some of the not needed details
- We have finished implementation of the reports
- Everyone is happy
- Devs can work on other more fun stuff
- Business and PM have what they needed



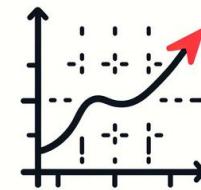
Story no. 3: Lesson Learned

- Bringing people together is crucial
- Access to project experts is essential
- Speak up and fix communication problems
- Possibly use tools like EventStorming
- Just avoid Broken-Telephone game



Story no. 3: Productivity

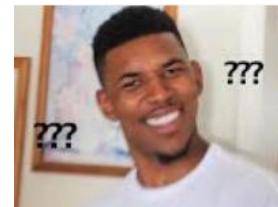
- We get things done quicker
- We avoid miscommunication
- We avoid effort spent on not needed features



The Bug

Story no. 4: The bug

- New bug in the notification system reported by Support
- Angry customers missed important notifications
- Leading to consequences e.g. missed flights
- ...
- I thought, “What The Hell? This is MY code.”
- Clean code, tests coverage close to 100%



Story no. 4: The chase

- I got angry
- I have quickly reviewed the Jira ticket and information
- Looked legit
- ????????
- I tested locally
- It worked



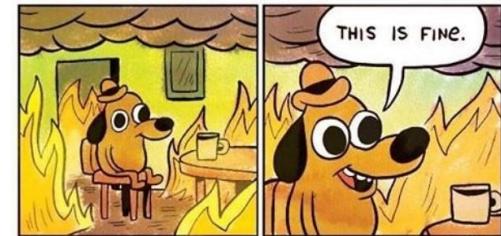
Story no. 4: The deep dive

- I reviewed code and recent changes
- I wrote another more specific integration test case
- Everything works fine
- I have tested the same thing on our staging env.
- Everything works as designed
- I took a break to clear my mind



Story no. 4: Deep dive

- Support is nagging and I have no answers
- I checked SQL queries
- I checked for errors in logs
- I restored the newest DB backup
- Some hours have passed
- I took another break to clear my mind



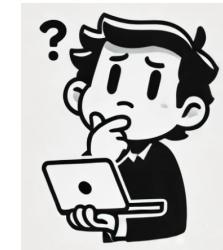
Story no. 4: The discovery

- I restored an older DB backup and found what has happened
- I discovered the client had created notification configuration
- Deleted it, then the Event happened - the notification wasn't sent
- And then the user recreated an identical notification configuration
- The code was correct
- The behavior of the system was correct
- However, I spend a lot of time and effort on it ...



Story no. 4: Lesson Learned

- Doubt all provided information: Zero - trust ;)
- Doubt assurances from others
- Check information in detail
- Imagine that you are on a minefield
- *“15 minutes spent on reading the ticket and verifying information can save you from 2 days of debugging”* - an Old Chinese Proverb
- Improve observability (revisions, audit, better logging, event sourcing)



 Eric Chapman @eschapman · 2d
Replies to @BJSmithArt
Last week I drove two hours to push the power button on a server that three separate people assured me was already on.

2	3	3	💣	💣	1
💣	💣	💣	5	5	3
💣	5	5	💣	💣	💣
2	💣	💣	💣	💣	3
0			4		2
0		1	💣		1

Story no. 4: Productivity

- Be critical thinker - it pays off
- Don't waste time on non-existing issues
- Don't attach emotions to tickets related to your solutions



On S3

Story no. 5: On S3

An innocent question from CTO:

“Marek, could you tell me why in 2018 when moving our services to AWS, we stayed on Elastic File System storage for quite some time instead of moving to S3 storage immediately? It was 10 times more expensive to be on EFS.”

Story no. 5: The fun

- I started archaeological process of gathering information from:

- Long Emails threads



- Confluence documentation



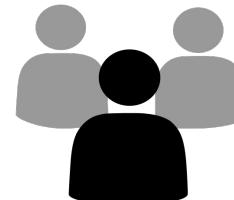
- Commits history across some projects



- Jira tickets and comments

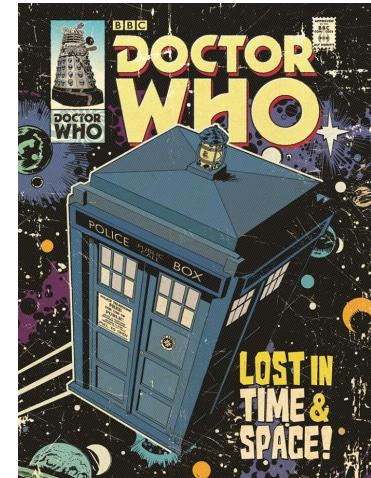


- Coworkers



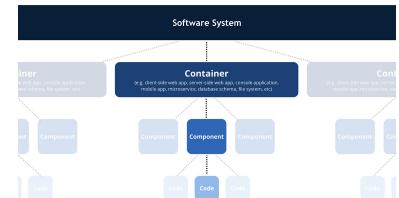
Story no. 5: Dr. Who

- ...and after just **six** hours of
- being lost in time and space
- I successfully recovered context of the decision and found the answer!



Story no. 5: Lesson Learned

- Spend some effort on having effective documentation
- Good documentation can save you a lot of time
- Architectural Decision Records are nice to start with
- C4 model is useful and doesn't have to be heavy
- Start small and then introduce additional tools



Story no. 5: Productivity

- Effective documentation helps with onboarding
- DRY - Saves you time on answering the same questions
- Documentation can be less painful to write using support of AI tools
- It can save you a lot of time in the future



Technology, Tools and Impact

Story no. 6: The car

- I bought a second-hand car 1.5 years ago
- It looked like it had a rear camera, but it was dead
- I was really busy with work, life, trainings...
- I really missed the rear camera
- I thought about selling the car
- 1 year has passed, and I needed an annual checkup



Story no. 6: The workshop

- During the checkup I asked if the guy could check the rear camera
- The guy found a hidden handle
- It activated the video; it was always on ...
- I have got a rear camera working
- Together with 5 other cameras! (360 camera)
- **6** in total ... No comment ...
- I thought about not telling this to anyone



Know your tools!

Story no. 6: Back to IT

- Postgres
 - ❑ Window Functions
 - ❑ Columns with JSON content
 - ❑ Have you ever run EXPLAIN for your SQL queries?
 - ❑ Do you know about SELECT FOR UPDATE?
 - ❑ Do you know about SELECT FOR UPDATE SKIP LOCKED?



Story no. 6: Back to IT

- JIRA
 - Automation Rules in JIRA
 - Shortcuts like "." for quick actions and Ctrl-K for searching
- IntelliJ Idea
 - Column selection mode
 - Structural search
 - Async Profiler
 - Debugger: Conditional breakpoints

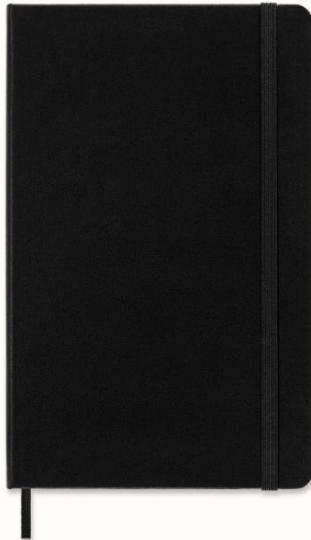


Story no. 6: Back to IT

- Hibernate / JPA
 - ❑ 1st Level cache - Entity Manager
 - ❑ Batch sizes
 - ❑ Stateless Session
 - ❑ 2nd Level Cache and it's implementations



Story no. 6: Find a difference



19 Euro

VS



1.9 Euro

Story no. 6: Productivity

- There is no Golden Hammer, Free lunch, Silver Bullet
- Know the tools even you haven't chosen them
- Keep emotions away, those are only some tools
- Learn about many various tools and what they can do
- Choose the right tool for the job
- Don't spend too much time going too deep
- Learn to use AI tools effectively



Time, Tech Debt, Upgrades and Money

Story no. 7: Time, Tech Debt, Upgrades and Money

- How do you approach your technical debt?
- Project on Java 11, Spring 4.x, Hibernate 4.x
- Elasticsearch 5.7, MySQL 5.7 and a few dozens other libs
- It was fine for some years, so upgrades were postponed
- It got more challenging and harder to keep it secure
- We started to get hit with interest on our credit card debt
- We finally had to start upgrading the stack



Max CVSS	9.8
EPSS Score	49.26%
Published	2018-01-04
Updated	2022-04-07

Max CVSS	9.8
EPSS Score	0.74%
Published	2023-04-20
Updated	2023-08-28

Story no. 7: Tech Debt, Upgrades and Money

- Spaghetti upgrades: Java, Spring, Hibernate, MySQL
- Infinispan, ElasticSearch, ActiveMQ
- Javax -> Jakarta
- Few dozens other libraries
- The whole stack has to be changed at the same time
- That is complicated, scary, and risky
- An outdated version of MySQL on AWS led to increased costs

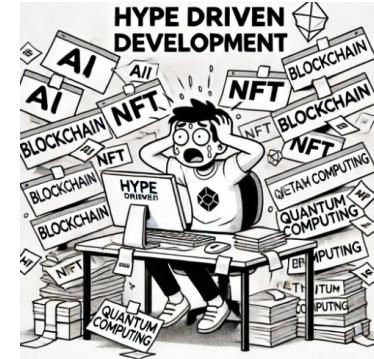


elasticsearch



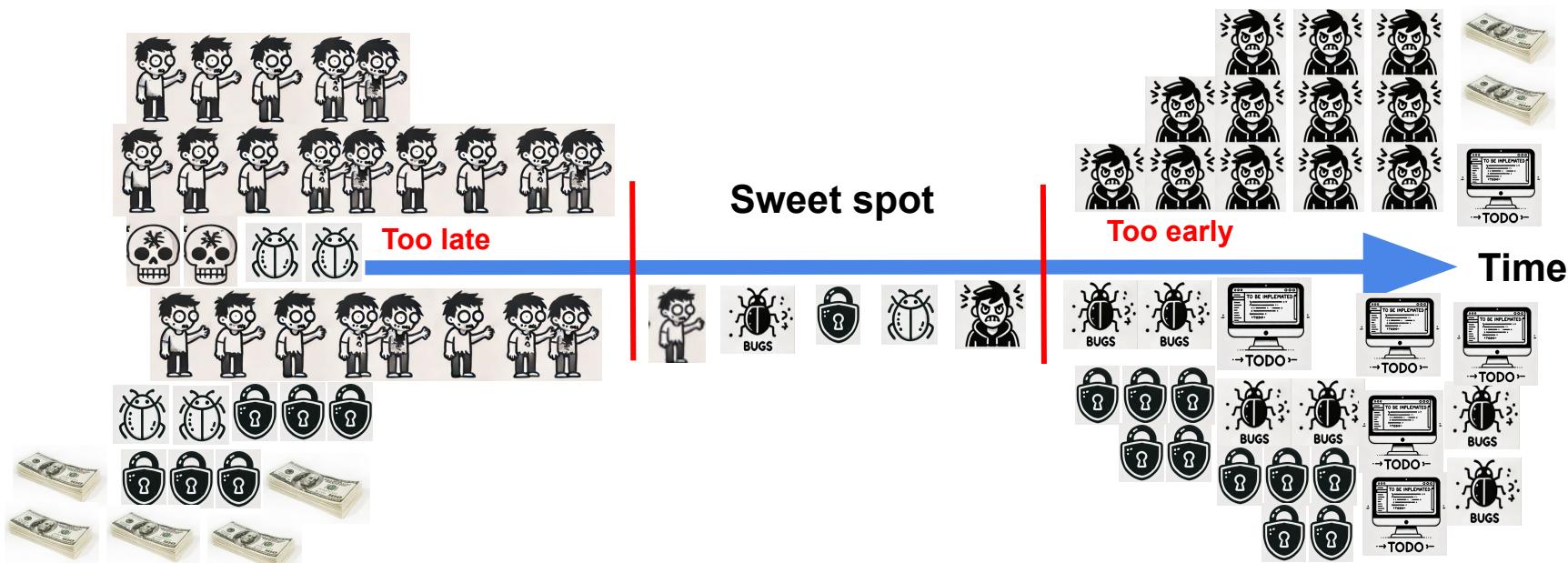
Story no. 7: Tech Debt, Upgrades and Money

- On the other end too optimistic upgrades can be **painful**
- So-called Hype-Driven Development
- Too fast IntelliJ IDEA Upgrades
- E.g. Using AngularJS for the whole project
- We tried too early to upgrade to Spring Boot 3
 - ❑ Some parts were not ready - we were Alpha Testers
 - ❑ Missing Libs: Old SAML deleted, new one didn't exist yet



Story no. 7: The Sweet spot for the Upgrade window

- Find your sweet spot between lagging behind and upgrading too early
- Martin Fowler's advice: “*If it's painful, do it more often*”



Story no. 7: Productivity

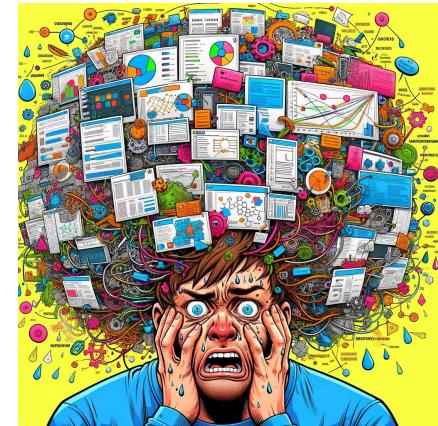
- Find your good spot for upgrades
- Both lagging behind and being too fast can be costly
- Setup notifications for easier upgrades
- Do upgrades more often, so its impact will be less painful



On Big Models and Perfectionism

Story no. 8: On Big Models and Perfectionism

- 2008, an attempt from Analyst to create a big model
- Around 100 entities and counting
- Almost everything has Many-To-Many relationships
- Hard to keep it all in head
- Clunky, slow and not fun to work with
- Obviously attempt failed

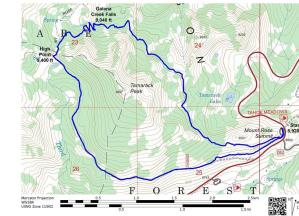
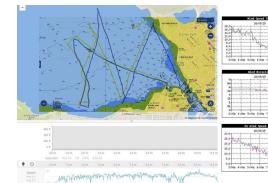


Story no. 8: The Model?

- **Domain:** Cartography, Geospatial Data, Transportation Networks
- **Problem:** Route Planning and Navigation: Assist users in efficiently moving from one place to another (A -> B)
- **Possible Models:**
 - Google Maps, OpenStreetMap (in cities or between cities)
 - Marine Navigation Systems (e.g., Navionics, iNavX)
 - Mountain Navigation Systems (e.g., Gaia GPS, AllTrails)
 - Air Navigation Systems (e.g., ForeFlight, SkyDemon)
 - Classic physical Maps



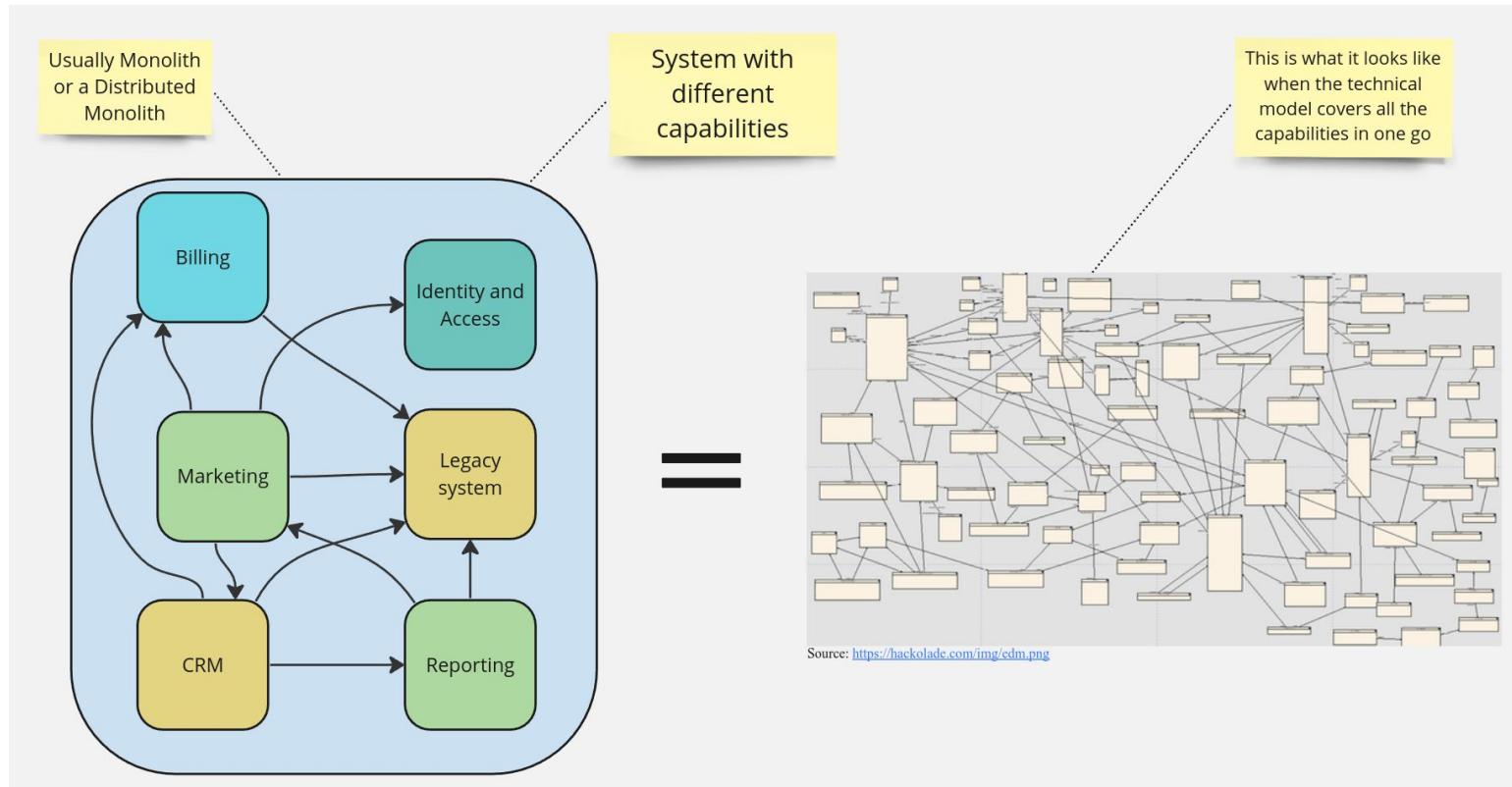
Google Maps



Story no. 8: On Models

- A model is a **simplified** representation of a domain, focusing on essential elements to **address specific problems**
- **All models are "wrong"** as they cannot capture every aspect of reality, yet some are useful for problem-solving
 - “*The Map is not the Territory*”, *Alfred Korzybski*
- **There can be Models live in the solution space**, representing potential approaches to tackle challenges within a problem context

Story no. 8: Big models aren't that fun to work with

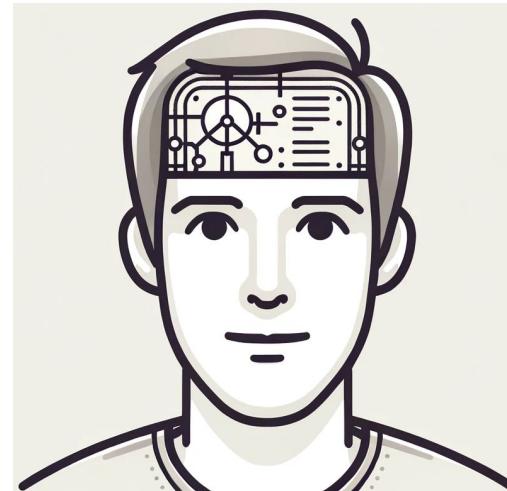


Story no. 8: Big models aren't that fun to work with

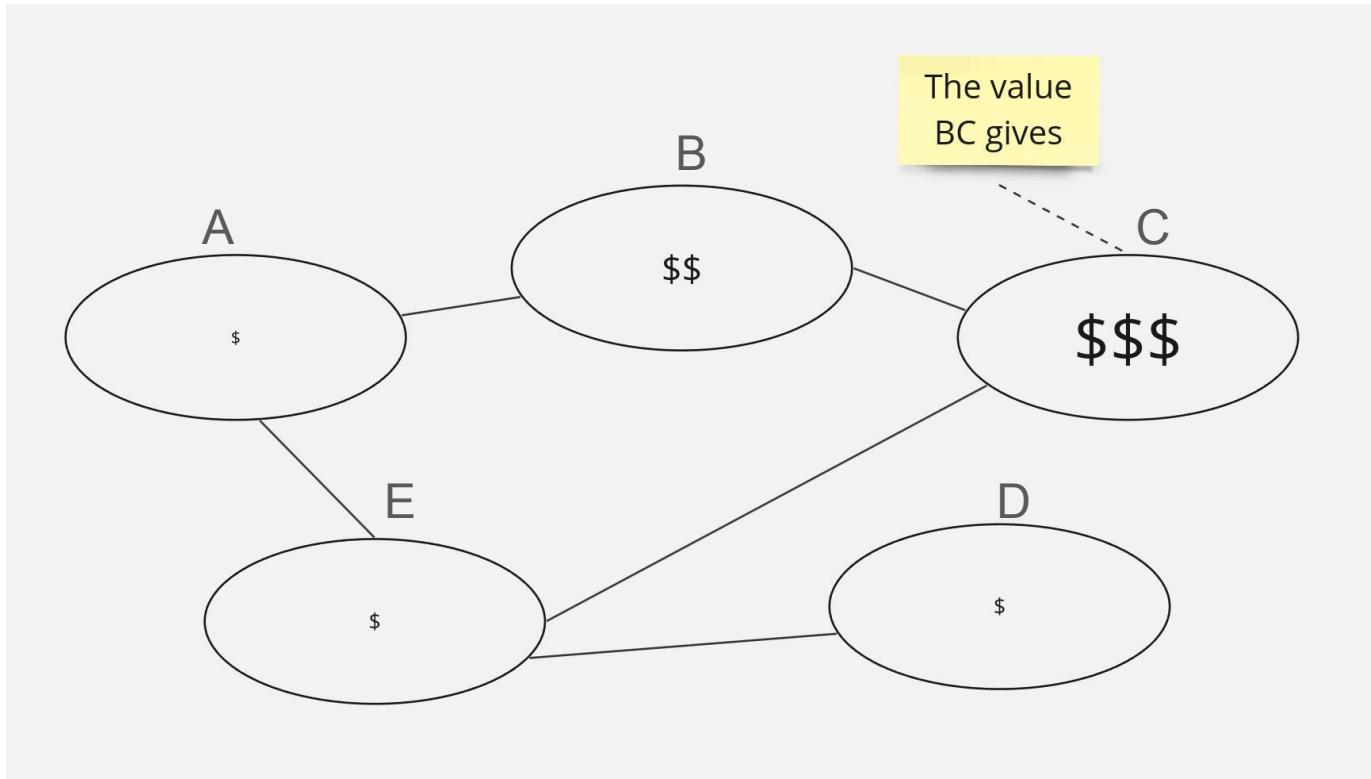


Story no. 8: Many smaller and focused models works better

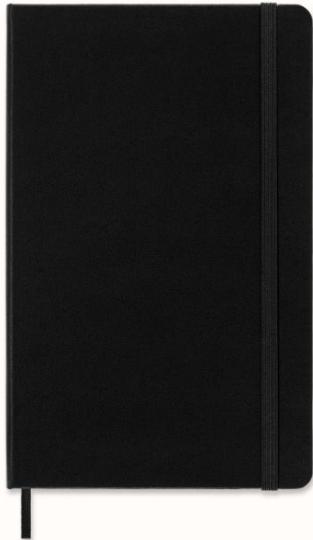
- Smaller cognitive load
- It's easier to maintain smaller models
 - Less bugs
 - Better quality
- It's more fun to work with



Story no. 8: No need to be perfect everywhere



Story no. 8: Find a difference



19 Euro

VS



1.9 Euro

Story no. 8: Throw away the inner perfectionist

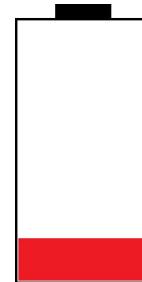
- Don't try to model the real world
- Imperfect models **can** be useful
- Know methodologies like DDD, BDD and TDD and where you can apply them
- Be pragmatic, don't lose hours (or weeks) on finding a way to make the Domain Model perfect!
- Perfectionism costs a lot



On Human Hardware

No 8: Do you face those issues?

- Headaches after work
- Feeling burned out
- Lack of energy
- Feeling distant from family
- Stress affecting personal life
- Struggling with personal errands
- Thinking about retiring?



The solution?



No 8: Take a good care of yourself at Work

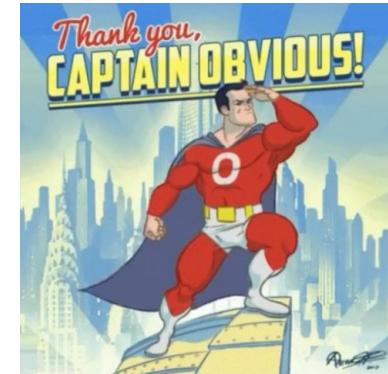
- Don't overwork
- Do ONE thing at a time (multitasking is tiring)
- Take short breaks often
- Take days off work

1



No 8: Take a good care of yourself - Long term

- If you sit more than 6 hours a day, do sports for a healthier back and spine
- Get at least 8 hours of sleep
- Eat healthy food
- Do ONE thing at a time (multitasking is tiring)
- Those are very simple yet hard to implement for most of the people
- **Simple != Easy**
- Only You are responsible for your own health!



No 8: What works for me?

- The work tips (all of them)
- 15-min meditation before work
- 3 times a week - Strength training or BJJ
- Fun with family and friends
- Getting 8 hours of sleep
- Eating healthy - more veggies, and following 16-8 intermittent fasting

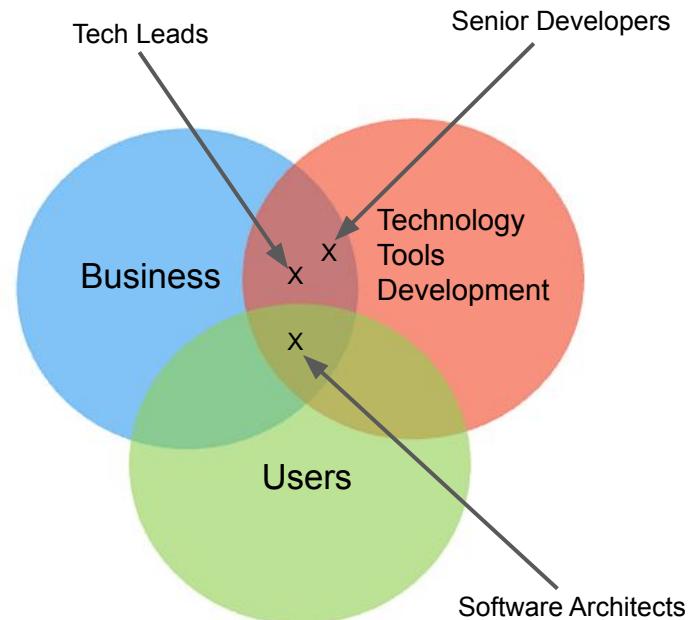


No 8: Productivity

Happy and Well Rested == Productive

What does it mean to be a Senior Developer / Tech Lead / Architect?

- Leadership
- Skill (applying knowledge)
- Being glue between "business" and technology
- Having impact



DONT read afterwards

**The hardest work is to find proper boundaries
of our Services / Bounded Contexts**

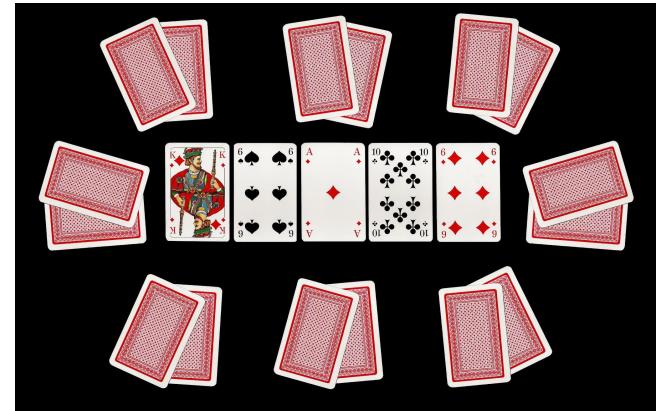
-

Soft skills: “I Message”

- Those are not Soft-Skills, those are Hardware skills, the where the hardware are bio-robots

Strategic Design

- Decide which domain is our **core** domain.
- The core domain should be the one we focus on - best people, most effort, etc.
- Not all the Domains are of the same importance! We all have limited budgets and time ...



Context is king: Best practices are context-specific.

Story about applying conference tools/solutions/architecture styles.

- Even on the newest Spring IO, there is a shift again on importance on modular monoliths, saying that Microservices aren't always that fun, but 10 years ago the same team was claiming that this is the way to go. Be cautious and have NO gods/heroes in any field etc. Use your critical minds.

Taking too much responsibility



Story no 8: On delegation, bus factor and vacations

Story 5 about realizing the need to put the team before ego. 2000 lines of code, it didn't matter anymore.

****Setting expectations:**** Give more freedom and responsibility to others.

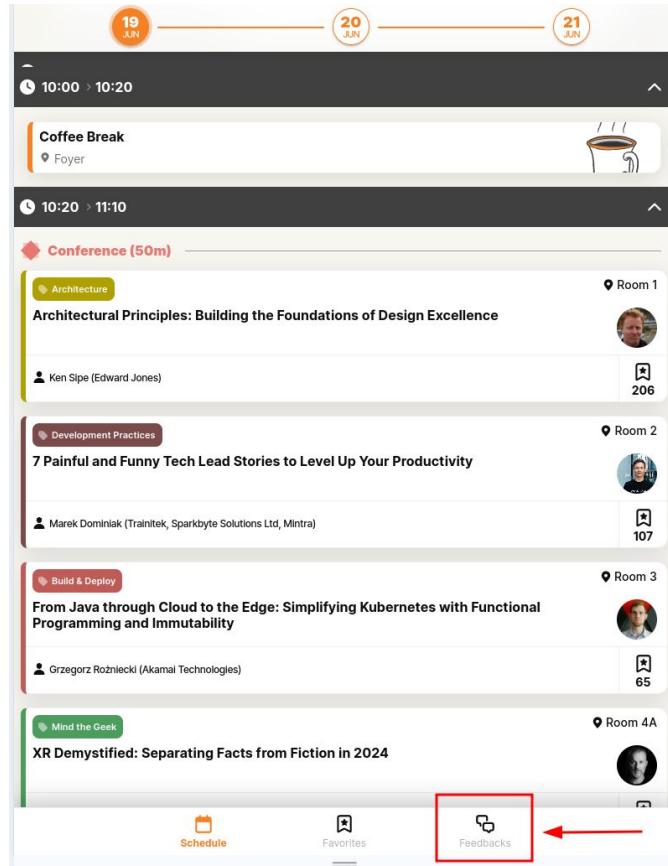
- ****Learn how to speak language with Business****
 - Talk with them avoiding technical terms
 - Talk with devs understanding technical terms
 - Talk with customers, seeing their point of view

References

- [Documenting Architecture Decisions](#), Michael Nygard
- ADR: <https://adr.github.io/> - Lots of information
- The C4 model for visualising software architecture, Simon Brown
- Design IT, Michael Keeling
- The Software Architect Elevator: Redefining the Architect's Role in the Digital Enterprise, Gregor Hohpe
- How to Stop Worrying and Start Living, Dale Carnegie
- Parent Effectiveness Training, Thomas Gordon



Please rate my talk



The screenshot shows a mobile application interface for a conference agenda. At the top, a timeline indicates the dates: 19 JUN, 20 JUN, and 21 JUN. Below the timeline, the schedule for June 20th is displayed from 10:00 to 10:20, showing a "Coffee Break" in the Foyer. The main content area lists three sessions:

- Architecture**: "Architectural Principles: Building the Foundations of Design Excellence" by Ken Sipe (Edward Jones) in Room 1. It has 206 attendees.
- Development Practices**: "7 Painful and Funny Tech Lead Stories to Level Up Your Productivity" by Marek Dominiak (Trainitek, Sparkbyte Solutions Ltd, Mintra) in Room 2. It has 107 attendees.
- Build & Deploy**: "From Java through Cloud to the Edge: Simplifying Kubernetes with Functional Programming and Immutability" by Grzegorz Roźniecki (Akamai Technologies) in Room 3. It has 65 attendees.

At the bottom of the screen, there is a footer bar with three icons: "Schedule" (orange), "Favorites", and "Feedbacks". The "Feedbacks" icon is highlighted with a red box and a red arrow pointing to it from the bottom right.

Contact / Slides



<https://www.linkedin.com/in/marekdominiak>

Slides: <https://github.com/trainitek/presentations/blob/main/DevoxxPL2024/7painful-stories.pdf>

www: <https://trainitek.com>



Slides:

