2023-01-17

Cross-functional Software Development

Working together toward effective, efficient, and user-friendly software processes and products

We live in a world that is constantly changing.

It looks like the digital world and industry move faster than ever, with the continuous stream of new and exciting languages, frameworks, and tools. This might all be an illusion because the way we write software has not changed considerably in the last twenty years. The complexity of software development limits digital transformation. Traditional corporate development strategies and IT solutions are too slow to keep up with growing new demands and constantly changing conditions. According to Moore's Law, the speed and capability of our computers increase exponentially, yet, the 'output' of the majority of software projects does not exhibit a similar trend.

What are we doing wrong?

My extensive experience in the field of software development has led me to conclude that there is a serious lack of quality communication, constructive criticism, and genuine collaboration between technical and non-technical stakeholders at every stage of the creation of software products. This causes misconceptions and errors, which cost money and effort to fix or, worse, result in software solutions that are not relevant. This has been going on for too long and it is time to explore a new mindset that will shift the paradigm of how we write software.

Change is primarily about people.

There is a great need for innovative tools that help organizations visualize and digitize their operations quickly and easily. We need to bridge the gap between technical and non-technical teams and change the way we think about technology together. To succeed, the companies themselves must drive their change, not by individual efforts, rather as an integral part of their operations and cultures. These working places put great demands on communication and organization and therefore our focus should be collaboration, transparency, simplicity, and speed.

Identifying real bottlenecks

Beneficial to understanding our inability to translate the real world into if-statement and while-loops, we first need to look at the real bottlenecks that limit the speed of digital transformation.

- The difficulty for developers to get domain knowledge and communicate with non-technical people.
- The amount of repetitive coding and plumbing needed to create an everyday application.
- The complexity of modern architecture, and the lack of skilled programmers.
- Technical communities cheering technical solutions, instead of promoting real-world value.
- The status quo isn't being challenged, especially because IT specialists are compensated hourly.

Changing the perspective

Concerning overcoming the gap between technical and non-technical teams and changing the way we think about technology, I believe we can add more value to the process and the end product and boost collaboration and innovation in organizations if:

- Technical and non-technical teams work with shared artifacts and models
- Use of visual languages that focus on business needs and hide technical complexity
- Change platforms in real time without downtime
- Adding a cross-functional mindset to agile methodologies

What is Cross-functional programming (XFP)

A programming paradigm where programs are constructed in languages that can be used by both technical and non-technical people.

The principles of XFP are:

- Visual code over text code
- Domain focus over technical focus
- Fault tolerance over compilation errors
- Runtime compilation over static compilation
- Shared business responsibility over technical teams

With XFP, technical and non-technical teams are working together uniformly.

What is Cross-functional design (XFD)

A software design paradigm where solutions are architected in a way that supports XFP.

The principles of XFD are:

- Dynamic schemas over static schemas
- Dynamic databases over static databases
- Dynamic workflow management over static workflow management
- Dynamic code execution over static code execution
- Dynamic UI over static UI

With XFD, we try to build solutions that do not change when the business requirements change.

The perfect storm

The need for XFP has always existed but we have not yet seen the right conditions for a full paradigm shift. Fortunately, we are moving closer to the perfect storm that will unlock the full potential of XFP.

- Cloud computing is now so mature that it was possible to operate and scale infrastructure with minimal knowledge and effort.
- Computers are now so fast that we can generate and execute code in real-time.
- Dynamic languages can take full advantage of the hardware and operate almost as fast as static equivalents.
- With machine learning, we are no longer limited to computer languages. Instead, we can use drawings, natural languages, or video recordings.
- New database technologies allow us to massively scale and manage dynamic data.

Many hard technical challenges are now solved by well-known open-source solutions.

What about low code?

Low-code and no-code platforms have emerged as viable and convenient alternatives to the traditional development process. This is a step in the right direction, but we find that most solutions break the principles of XFP and XFD.

- Built by technical people for technical people
- Requires understanding of coding and coding languages, which can be a hurdle for most non-technical employees
- Vendor lock-in
- Application configuration, not cross-functional programming
- Used by inexperienced software developers. No challenge for experienced developers.

The future for XF?

The world is modernizing and digitizing at an unprecedented rate, but there is a massive imbalance between the supply and demand of talented professionals. The rapid speed of application development anticipated to occur over the next few years is being hampered by the lack of qualified developer expertise. According to IDC, more than 500 million apps will be developed by 2023. Gartner predicts that the market for app development services will expand at least five times more quickly than the IT infrastructure's ability to meet it. As a result, IT projects could be shelved for a lengthy time.

So why waste anyone’s time?

For all stakeholders, cross-functional software development can result in time and cost savings. People with and without technical backgrounds can collaborate. Effective, efficient, and user-friendly software processes and products result from the proper team using the suitable tools, together with the best possible communication. Furthermore, I can honestly assert that cross-functional software development is effective because I have firsthand experience with it. With the help of the aforementioned cross-functional software development approach, I was able to complete a number of projects faster and cheaper while also teaching my team the value of cooperating throughout the whole software development process. Everyone knew their place in the team, as well as understanding the role of other members, and saw the finished result as a whole rather than just from the perspective of their own specialty.

2023-01-17

Any4m - Backend as a service

The Any4m - Backend as a service is our first step towards fullstack Cross-functional Development. It is a fully managed backend infrastructure that helps you build and scale out applications and enterprise solutions. The service includes all backend services that your front-end team needs to successfully deliver your projects on time.

- Access to world class support
- API Management as a Service (authentication included)
- Data Management as a Service (backup/import/export)
- Domain model as a Service (database included)
- Identity as a Service
- Index as a Service
- Scheduling as a Service
- Search as a Service
- Web-sockets as a Service
- Workflows as a Service

Rickard Nilsson
Principal Architect, Founder