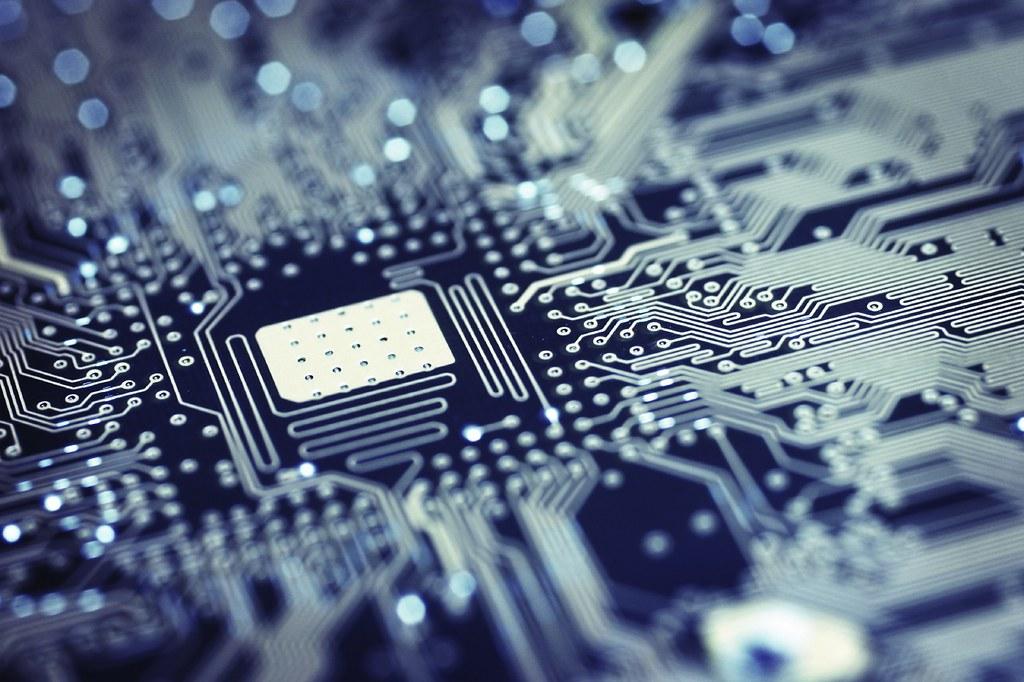
Emerging Trends Research

Jordan Radushev - Semester 6, Class RB05



Domain Driven Design

# Overview

Domain Driven Design (DDD) is a concept in software development regarding the structure, maintainability, and the logical separation of the code. The greatest priority of each DDD project is the business domains and their implementation and respectively the domain forms the core of the logic. It is currently a very popular topic among the software industry. Respectively quite a lot of companies implement this concept within their projects and the developers are expected to be fluent in applying it.

My main goal for this research is to acquire more in-depth knowledge about DDD both theoretical and practical. I want to learn not only about the DDD concept itself, but also the way it fits in different types of projects. From my experience as a developer, I can say I have been in occasional situations where I use a particular technology/concept in a wrong way while thinking it was a “silver bullet” in software.

# Research

**Main research question:**

*“How can the domain driven design concept be applied efficiently in software projects?”*

**Additional questions:**

* What is Domain Driven Design?
* What are the advantages and the disadvantages of this concept?
* How can DDD be applied in a real-life software project?

## Methods

A nice way to acquire theoretical knowledge is **library research**. This way I can get into the basics of DDD. Searching for different trustworthy sources of information and investigating the **good and bad practices** is also a key part of the topic, so I can get knowledge on how to use DDD properly and not misuse it eventually. A nice could-have is also an **expert interview** which can give me a clear perspective from someone with first-hand experience with DDD.

After collecting enough information, I may try to implement what I have already learned in a real-life project (**prototype**), where I can apply the DDD principles and achieve a practical justification to my research.

## Deliverables

At the end of the research possible deliverables could be:

* Research document, including detailed information about DDD.
* Small project to show what I have learned from theoretical study.

# Report

The aim of this report is to show my research about the emerging trends topic I chose for this semester.

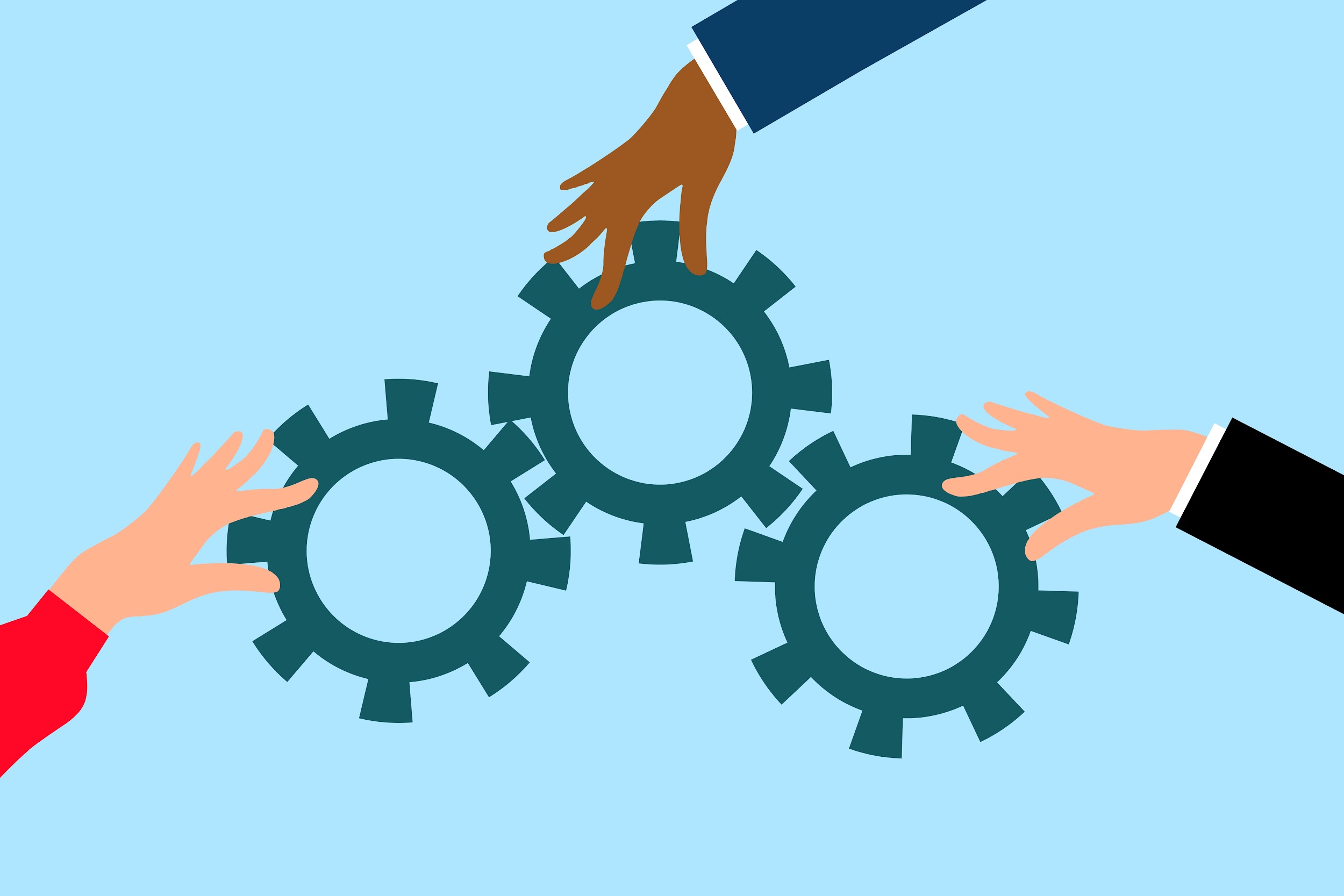
In the following section you will read about my research, regarding the main question of the report along with the sub-questions (refer to page 1).

## Theoretical research

### What is Domain Driven Design?

Domain Driven Design is a software development concept, concentrated mainly on the domain part of the software project, as the name says. This means that the core of the project is the domain specific business logic. This way, the domain expert within the project is directly involved in the development process.

DDD is a widely used concept in software development nowadays. In simple terms, it is a type of code structure which allows the development team(s) and the domain expert to speak the same language. Many times a project requires a domain specific expertise, the developers are often confused and need additional knowledge, so they can make the software work in compliance with the domain expert’s expectations. Trial and error in the prototyping phase of a given project often costs a lot of time which eventually can cost a lot of money.

Applying DDD in software projects, especially in the bigger ones, may save a lot of time, money and effort from developers’ side. Prototyping becomes much easier and also by applying DDD in a project, it is like an investment for writing better and easily understandable code. In today's world, when it comes to software development, the complexity of the project grows enormously and there are plenty of projects out there which contain much more than 3 classes and 5 files. Most of the projects on the enterprise level contain hundreds/thousands of classes and files within the project. Without a proper code structure, there is a high chance for the code to transform into a complete mess.

### Pros and cons of DDD

This concept is often taken as an universal recipe for writing high quality code. The truth is that DDD has many advantages in specific use-cases, however it could also make the development process much slower and harder for the developers if it has been incorrectly used for the specific project.

The main advantages of DDD are:

* **Better communication** - DDD acts as a bridge between the domain experts (often non-technical people) and the developers
* **Readability** of the code, since it should be easily presentable to the domain experts.
* **Flexibility** when it comes to adding more functional requirements to the system. When applied properly, DDD is easily configurable and new requirements are often implemented more easily.

It may sound that DDD could be implemented in any project context, it has some major drawbacks such as:

* **Larger codebase**. DDD requires more code to be written which may be a problem when the project becomes too large.
* **Overburden** if DDD is misused. Generally, if the project does not require domain specific knowledge or there is not a distinguished domain expert who really navigates the project between some kind of domain boundaries.

To apply DDD or not is rather the last question by taking the decision whether to apply DDD or not. Before that the team should answer questions such as:

* Does the software development process involve a specific domain and eventually a domain expert?
* Is the software product meant to contain many user stories? (At least 20)
* Will the application complexity grow a lot more than in the initial plan?

If the answer of the above-mentioned questions is rather no, so is also the answer to the question *“Should I introduce DDD in the project?”*

#### Research methods

The subquestion above summarized the initial part of the research. Respectively, I needed to get more in-depth knowledge about Domain Driven Design. This is why I started with more detailed **library research**. I have read a few articles, regarding the topic, and also, I managed to read part of the most famous DDD book (listed in the literature list). Part of the initial research were also the **good practices** for applying DDD successfully within a given project.

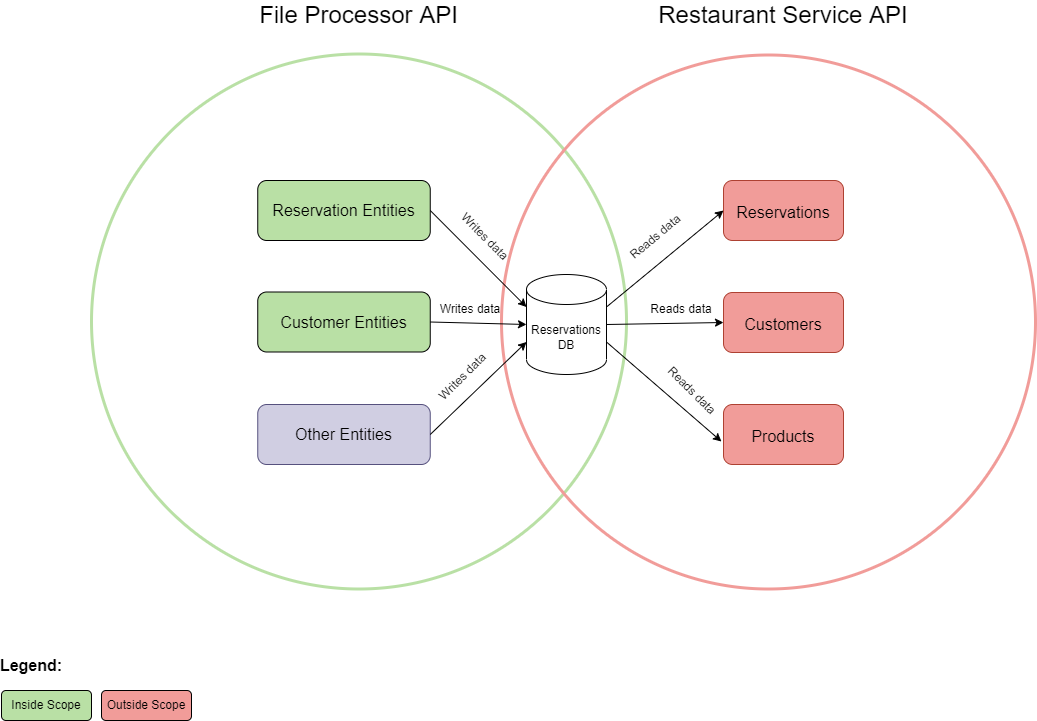
## DDD in practice

In this section you will find the practical outcome of my emerging trends research.

### Overview

The sample DDD project I am about to show here is part of our group project for Semester 6 - Dynamic Dash. This project consists of several microservices, but I want to describe the relationship between 2 of them. We have a **Restaurant service** and a **File Processor service**. Both services have their bounded contexts and the intersection between them is the **restaurant-db** (database).

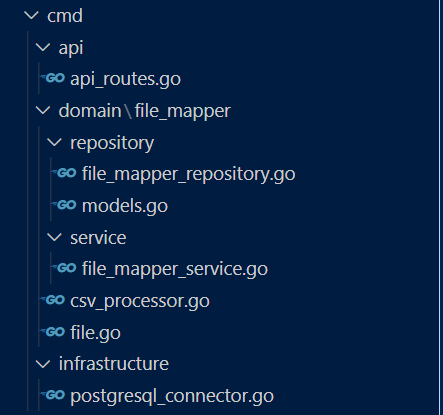
**Restaurant service API** reads from this database and sends responses to the client, and the **File Processor API** writes data to the same database.



**\*The applied DDD principles are in the scope of the File Processor API**

In DDD bounded context means a collection of models within a given domain. Here the reservations are part of 2 bounded contexts. In the **Restaurant API** they are a very important model within the domain and they contain a lot of useful information. However in the bounded context of **File Processor API**, they are nothing more than just a row in a CSV file.

### Folder structure



To structure this project efficiently, I decided to apply the repository pattern within a layered architecture

**Explanation of the diagram:**

* **api/** folder - contains the web layer of the API (controller)
* **domain/** folder - contains the core logic of the application (domain logic)
* **infrastructure/** folder - contains infrastructure-related code such as database connections

I want to pay more attention to the domain folder where the domain logic lives. Here I introduced a **service layer** which contains only business logic and connection (repository) to the database. Within the folder we have a **csv\_processor.go** file where I placed the CSV-related logic. In the future we can add an **excel\_processor.go** and a **gsheet\_processor.go** if we want to add more domain-related functionality to the application. In **file.go** you can find the programming “translation” of the CSV rows.

#### Research methods

For this part of the research I managed to create a small **prototype.** Also, I had a conversation with a tech lead developer from the company I currently work in and I received a few remarks on DDD from an **expert.**

### Conclusion

By the end of this research, I would say that DDD is an investment at the beginning of a given project, so the team can be constantly encouraged to write structured, reliable and flexible code. It could be an overkill, however, if you try to implement it without knowing how to apply it properly. I would definitely say that the domain specific user stories of the project really benefits from applying DDD within the project, especially in the long-term.

On the other hand, I noticed that I wrote more code than I would without applying DDD and why was that so? Mainly because I separated the logic from the domain perspective first which led me to structure the code in more files in my project. These files though were much simpler and smaller. Keeping the logic clean and separate in smaller files may be a huge advantage as the complexity of the project grows.

From a practical point of view, I would not use DDD if there is no domain expert to talk to, especially if the domain is a specific one, not general software. Also, if the team does not deeply understand the fundamentals of DDD, it may unlock even more potential problems in the future. So, from this research I can conclude that DDD is worth trying and applying only if the correct research (DDD specific) is already and there is a domain expert, part of the project.

# Literature list

1. Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software* (1st ed.) [E-book]. Addison-Wesley Professional
2. Ibrahim, A. (2020, August 26). *Do we need DDD in our project?* LinkedIn Article. https://www.linkedin.com/pulse/do-we-need-ddd-our-project-ahmed-ibrahim
3. K. (2020, November 16). *What is Domain-Driven Design (DDD) | Pros & Cons*. Codez Up. <https://codezup.com/what-is-domain-driven-design-ddd-pros-cons/>
4. Khan, A., Oberoi, O., & Core, B. (n.d.). *Domain-Driven Design - DZone Refcardz*. Dzone.Com. https://dzone.com/refcardz/getting-started-domain-driven
5. Smith, J. A. P. (n.d.). *My experience of using modular monolith and DDD architectures – The Reformed Programmer*. The Reformed Programmer. https://www.thereformedprogrammer.net/my-experience-of-using-modular-monolith-and-ddd-architectures/