TOP-DOWN OR BOTTOM-UP

WHICH APPROACH IS BETTER?

Prepared By : Muhammad Tariq Arif Hussain

Date : 27 July 2022

Table of Contents

1. The top-down and bottom-up approaches.

1.1 General Overview

The top-down and bottom-up approaches are both general purpose strategies for processing information and knowledge. They can be developed and implemented to serve various business and research contexts, including fields like software, humanistic and scientific theories (systemics), management and organization. Fundamentally, such broad-scope and generalized strategies are required when solving problems of varying criticality. However, this research aims to explore the impact of either approach on software architectures, specifically microservices. The method identifies its underpinnings from other related fields on a higher-level, whilst working with limited literature and a generous reliance on the inductive research approach. However, in order to entertain a contrasted opinion, the following hypothesis is formulated:

When a feature rich application begins to accommodate an increasing number of users, the higher is the tendency to choose either one approach.

To start off with our discussion, we will first acquire some conceptual clarity. It will include some answers to questions like,

'What is the top-down and the bottom-up approach?'

'In which areas are they predominantly applicable?'

'What are the trade-offs for choosing either one approach?'

1.2 Conceptual Clarity

Top-down and bottom-up approaches are techniques for processing information and knowledge. In the top-down model an overview of the Information System (IS) is formulated without emphasizing each system-detail. And in the bottom-up model, the starting point of analysis is from its individual parts which are linked together to form a whole system. However, in-practice, the terminology is also applied within non-technical contexts like management and leadership by following a similar principle.

1.2.1 The top-down approach of management.

- The top-down approach involves a directive that originates from the leadership's stand-point. It is therefore called the "traditional-approach".
- In the top-down approach to management, a team or project manager makes decisions that trickle down through a hierarchical structure.

1.2.2 The bottom-up approach of management.

- The bottom-up approach for management requires a much more democratic and collaborative effort to decision-making, knowledge work, along with initiating changes and improvements. It involves the organization or the team as a whole.
- Such an approach is quite prevalent in sectors that prioritize disruption and innovation.

1.2.3 The hybrid-approach to management.

- The hybrid-approach combines both top-down and bottom-up approaches to minimize the disadvantages of both approaches. Conventional top-down and bottom-up approaches have inherent weaknesses because of which very few organizations apply a purely top-down approach to management.
- Most teams apply a hybrid approach that falls somewhere along a spectrum between the top-down and bottom-up management styles. When it comes down to it, effective managers know how to balance the efficiency of the top-down approach with the collaborative and creative advantages that come from the entire team [1].

2. Top-down and bottom-up approaches in System Design

Top-Down design requires extensive planning and research to develop a product, whereas bottom-up design leans towards experimentalism to develop a product. Even though the overall goals for a product are outlined beforehand in bottom-up design, the assembly of the product is done on a system-by-system basis. While some insist that one approach is better than the other, those who have invested in design-thinking methodology know that a blend of the two approaches often produces the best results. Characteristic situation where one approach is better suited than the other is illustrated with the following decision-trees:

2.1 Experimental Development

- Will your Product Development Cycle[1] be heavily experimental?
- Are you trying to make something completely new?

⇓ If so, a Bottom-Up iterative approach might be best for your project.

2.2 Tight Budget

- Is your project constrained by a tight budget?

⇓ If so, a top-down approach can help you maximize savings of your Product Development Cycle.

2.3 Large and complex systems

- Are you building a large, complex system?

⇓ Complex systems and machines benefit from a top-down approach because it breaks down a project's goals into smaller problems that are more easily solved.

2.4 Collaborative Effort

- For your project to be successful will you need everyone's voice to be heard?

⇓ If the problem you're trying to solve is going to require a lot of creativity a bottom-up approach can help leverage all of the creativity in your group by letting them experiment and voice their opinions [3].

[1] Product Development Cycle: It is the process of taking a product from an idea through its market release and beyond. This cycle involves product managers, developers, designers, QA testers, and others. Stages can be summarised as: Develop the idea. Validate the idea. Build a prototype. Create the messaging. Build the product. Release the product. Improve the product. [2]

3. Top-down and bottom-up approaches in Computer Science

An algorithm in Computer Science is a specific procedure for solving a well-defined computational problem. The development and analysis of algorithms is fundamental to all aspects of computer science: artificial intelligence, databases, graphics, networking, operating systems, security, and so on. Algorithm development is more than just programming. It requires an understanding of the alternatives available for solving a computational problem, including the hardware, networking, operating systems, security, and so on [4]. Algorithms are designed using two approaches, which are the top-down and bottom-up approaches. In the top-down approach, a complex module is divided into submodules. On the other hand, bottom-up approaches begin with elementary modules and then combine them further. The prior purpose of an algorithm is to operate the data comprised in the data structure. In other words, an algorithm is used to perform the operations on the data inside the data structures.

3.1 Defining the top-down approach

The top-down approach basically divides a complex problem or algorithm into multiple smaller parts (modules). These modules are further decomposed until the resulting module is the fundamental program which cannot be decomposed any further. After achieving a certain level of modularity, the decomposition of modules is ceased. The top-down approach is the stepwise process of breaking a large program module into simpler and smaller modules to organize and code programs in an efficient way. This approach is a form of stepwise refinement. Starting from an abstract design to a more concrete level, until we reach a level where no more refinement is needed and the design can be implemented directly. Most design methodologies are based on the top-down approach. **A top-down approach is suitable only if the specifications of the system are clearly known and the system development is from scratch.**

3.2 Defining the bottom-up approach

In the bottom-up approach, the fundamental parts are designed first. These parts are then combined to make the higher-level module. This integration of submodules and modules into the higher-level module is repeatedly performed until the required complete algorithm is obtained. With the bottom-up approach, layers of abstraction are implemented from the bottom. By design, the operations of the previous layers are subjected to higher levels of abstraction to form powerful operations. This process continues until operations desired by the system are reached. **If a system is to be built from an existing system, a bottom-up approach is more suitable.** For a bottom-up approach to be successful, we must have a good notion of the top level. Without a good idea about the operations needed at the higher layers, it is difficult to determine what operations the current layer should support [5].

4. Top-down and bottom-up approaches in Service Oriented Architecture (SOA).

In SOA, software is developed as services which can communicate with different platforms and languages and packaged to form applications. Services are the fundamental building blocks of SOA. These services are self-contained units of software designed to perform a specific task. Consider a situation where an application needs an additional component to serve a specific function, but this add-on is positioned elsewhere. In this situation SOA can be applied to communicate the necessary data and technology services and coordinate the function [6]. A simple analogy would be of having a word-processing tool which only supports the colour green for text, but in order to get other colours, the user much connect the word-processor to a "colour-palette" component over the internet. The communication protocols between the two components, i.e., the word-processing software and the colour-palette are all situated in networks that allow interconnectivity, irrespective of the individual technologies the services are built upon.

A typical example for SOA is that of integration between enterprise applications and legacy systems. It originated as a solution for integration problems between enterprise-wide applications which ran on very disparate systems. Modernization of such enterprise applications usually cannot eradicate legacy systems as a whole and build a new replacement entirely. Usually a "web-use" interface and a "partner-exchange" interface represent the existing business systems and legacy systems. **Enterprise level design approaches borrow from the principles of SOA to significantly utilize disparate services to serve complex functions** [7].

In order to conceptualize SOA as a system with multiple interconnected moving parts can be confusing. However, for the sake conceptual support, certain tools are helpful. Consider SOA as a layered structure with 5 levels, arranged according to a hierarchy which serves human-computer interactions at the upper layers. To illustrate such an architecture, the following relations between the various layers of SOA are presented (bottom to top):

1. The Object Layer: The bottom layer, i.e., the Object Layer sits at the bottom and consists of legacy systems, including custom-built applications and databases that contain business functionalities. These legacy enterprise objects can be leveraged to build composite services – proof that SOA doesn't have to mean rip and replace.
2. The Component Layer: Just above the Object Layer is the Component Layer consisting of enterprise components. These components are responsible for realizing the functionality of services.
3. The Service Layer: The middle layer is the Service Layer, which is where exposed services (both individual and composite services) carrying out business functions reside. **The Service Layer acts as a bridge between the lower-level layers (the Object Layer and Component Layer) and the**

higher-level layers (the Process Layer and Enterprise Layer). Enterprise components can be exposed as services in this layer, making reuse a possibility.

4. The Process Layer: Right below the Enterprise Layer, we arrive at the Process Layer. Here, the services exposed in the Service Layer are combined through service orchestration or service choreography to create a single application that realizes and automates a business process.
5. The Enterprise Layer: The Enterprise Layer is the upper-most layer. This layer is also called the Presentation Layer because it's the end user's point of access to the composite enterprise application, typically over the Internet [8].

4.1 Bottom-up approach in SOA

In the bottom-up approach in SOA, as the name suggests, the initial point of focus is the bottom layer, i.e., the Object Layer. By identifying the required services/components at the Object Layer, and integrating each subsequent layer on top, forms the fundamental design principle of the bottom-up approach. This approach has been followed and proven by many integration projects.

4.2 Top-down approach in SOA

The top-down approach is not exactly the reverse of the Bottom-Up Approach, as one might presume. It involves finding an alignment between IT and Business processes at an enterprise level. Here, alignment refers to the general agreement on the business goals of the enterprise for the technologies to meet before they are built. A lot of interdisciplinary concept work is involved before proceeding towards a fully-engineered enterprise-wide application. This approach promises the maximum benefit from SOA. Thus, identifying the required business process services at the enterprise level will be a starting point for the top-down approach, and its nested business processes are realised only after the enterprise-wide business processes have been aligned. Finally, the bottom layers, or the most basic building blocks are identified by the bottom-up approach. In a way, the top-down approach is a "hybrid" approach as it is not purely technical [7].

5. Top-down and bottom-up approaches to Microservices

As the name suggests, microservices are finely-grained services. It can also be called a distant cousin of SOA. Microservices are designed to work independently and to focus on a single task. A key characteristic of microservices is being “loosely-coupled”. It basically means deploying a particular application by following the microservices architectural approach. Two systems are said to be loosely-coupled, if changes to the design, implementation and behaviour of one does not affect the other. Without relying much on any other application/system for its technical and functional requirements, microservices are confined to the definitions set by their developers. This stand-alone feature enables developers to build scalable applications and use off-the-shelf services more quickly [9].

As organizations recognize the benefits of a microservices architectural approach, they face the question “How do we divide our monolithic applications into more manageable pieces?” Most companies answer this question is one of two ways, the top-down approach or the bottom-up approach. Each approach offers a distinct set of advantages – top-down is more easily understood by non-technical decision makers, and bottom-up makes more efficient use of existing development resources. However, each approach comes with its own set of challenges.

5.1 Top-down approach to Microservices

The top-down approach involves a functional decomposition of the application, beginning at the business level. The architect separates functionalities based either on domain (domain-driven design) or on business capability. A bank, for example, may create separate microservices for lending, wealth management, and online banking.

5.1.1 Challenges of the top-down approach.

Organizations that have adopted a top-down approach to microservices architecture have encountered a series of complications. For example, if the organization creates an independent microservice for every operation or function, the resultant architecture becomes difficult to manage. Furthermore, the top-down approach requires each microservice to arrange its own infrastructure, thereby needing additional servers and IT resources, driving up their infrastructure costs.

Another challenge related to the top-down approach is that the resulting service may not align with the workflows of the organization’s development and support teams. Deploying and supporting a single service – such as online banking – may require coordination among several IT teams, resulting in a less efficient use of technology resources.

5.2 Bottom-up approach to Microservices

The bottom-up approach is purely technical, using the availability of development resources as the basis for defining the required microservices. The architect may look at service functionalities- such as a bank's deposit functions – and bundle them together to create a single microservice. Or divide larger applications by technology, such as Java or Node.js, and create microservices based on those groupings.

5.2.1 Challenges to the bottom-up approach

While relying on a bottom-up strategy offers the potential to utilize technology resources more efficiently, it offers its own share of problems. The resulting services, for example, may be misaligned with business function ownership, since a single microservice – such as deposits – may span several different lines of business.

Problems may also arise around the timing of releases. Some business units may want to have more aggressive release cycles, and if services are not aligned to those cycles, releases could be held up while the development team waits on other departments.

5.3 The "hybrid" approach

What if there was a way to leverage the greatest benefits of each strategy. That would indeed be a recommendation when opting for a top-down or a bottom-up approach. Consider the following approach:

- Begin with a domain model or functional decomposition.
  - This effort will engage all the stakeholders meant to benefit from the microservice.
- Optimize your services inventory using a technical approach, either separating or folding functions together from the operational perspective, to determine how you're going to manage those services.

One advantage of this approach is that it involves both IT and the business at the highest level. Both teams work together from the very beginning to designate microservices that make sense from the business perspective, but are also efficient for IT to manage on a day-to-day basis [10].

Conclusion and Future Works

The top-down and bottom-up approaches encompass a multitude of applications in information processing and knowledge management. Its range can vary from humanistic fields like management and psychology to technical fields like the intersection of IT and business. At the beginning of this read, a hypothesis was formulated, to check which of the two approaches would turn out better and more favourable than the other, universally. A blend of both approaches, the so-called “hybrid” approach seems to qualify as a clear winner.

The debate between the two strategies engages parties in both sides, while illustrating valid criticism from each side. While the top-down party criticizes the excessive updates and rework involved with the bottom-up approach, the bottom-up party defends their ground by criticizing the extensive planning and strict policy enforcement with the top-down approach. The decision is never so black and white, but rather situated in a grey area. Here contextual demand overrides technical and non-technical prowess.

In this attempt to compile marks in the vast territory conquered by the two approaches, a bigger question was stumbled upon. The question that begins to surround us is of even more fundamental nature. It is the question of ever-increasing digitalization which now impacts every sphere of human interaction. In this digital-age, we generate so much information and access such rich forms of media, dare I say that the grand scheme of things can be indeed be sufficiently illustrated and futures forecasted.

The topic of digitalization cannot be indulged without mentioning the influence of big-tech. We have all felt it. It is now part of human-nature. Looking at it from a layman’s perspective, big-tech encompasses social-media, search engines, operating systems, video-streaming sites etc. But this is not a composition about big-tech. A more tech-savvy individual will recognize where this is stemming from – cloud-computing.

Future works of research is set in the direction of cloud-computing. Specifically, topics of interest include:

- Underrated technologies in the cloud-computing space, like OpenStack and how is it different from the conventionally used options like AWS and Azure.
- How to circumvent vendor-lockdowns? Can cloud technologies be improvised to mitigate such situations?
- Other nascent technologies, industries and markets that have yet to enjoy the spotlight of mainstream digitalization.

So where will this journey lead us? We shall find out more!

References

1. Asana, T., 2021. “*Top-down approach vs. bottom-up approach: What’s the difference?”*. [online] Asana.com. Available at: <https://asana.com/resources/top-down-approach> [Accessed 11 July 2022].
2. Fitoussi, V., 2022. “*Product Development Cycle*.” [online] Productplan.com. Available at: <https://www.productplan.com/glossary/product-development-cycle/> [Accessed 27 July 2022].
3. Maxey, K., 2012. *Top-Down and Bottom-Up Design*. [online] Engineering.com. Available at: <https://www.engineering.com/story/top-down-and-bottom-up-design> [Accessed 11 July 2022].
4. Encyclopedia Britannica. n.d. “*computer science - Algorithms and complexity”*. [online] Available at: <https://www.britannica.com/science/computer-science/Algorithms-and-complexity> [Accessed 11 July 2022].
5. Differences, T., 2017. “*Difference between Top-down and Bottom-up Approach (with Comparison Chart) - Tech Differences*.” [online] Tech Differences. Available at: <https://techdifferences.com/difference-between-top-down-and-bottom-up-approach.html> [Accessed 11 July 2022].
6. Community, Software Development, 2022. “*What Is Service-Oriented Architecture?”*. [online] Medium. Available at: <https://medium.com/@SoftwareDevelopmentCommunity/what-is-service-oriented-architecture-fa894d11a7ec> [Accessed 15 July 2022].
7. user, p., 2009. “*SOA Top-down & Bottom-up approach Vs Integration, modernization and SOBA*.” [online] The Spiceworks Community. Available at: <https://community.spiceworks.com/topic/2454971-soa-top-down-bottom-up-approach-vs-integration-modernization-and-soba> [Accessed 26 July 2022].
8. MuleSoft. 2022. “*Services in SOA | MuleSoft*.” [online] Available at: <https://www.mulesoft.com/resources/esb/services-in-soa> [Accessed 26 July 2022].
9. Richardson, C., 2022. “*What are microservices?”* [online] microservices.io. Available at: <https://microservices.io/> [Accessed 26 July 2022].
10. Buslovich, D., 2022. “*Microservices architecture: Choosing your approach*.” [online] Logic2020.com. Available at: <https://www.logic2020.com/insight/microservices-architecture-choosing-your-approach> [Accessed 27 July 2022].