Business Analytics

UNIVERSITAT DE BARCELONA

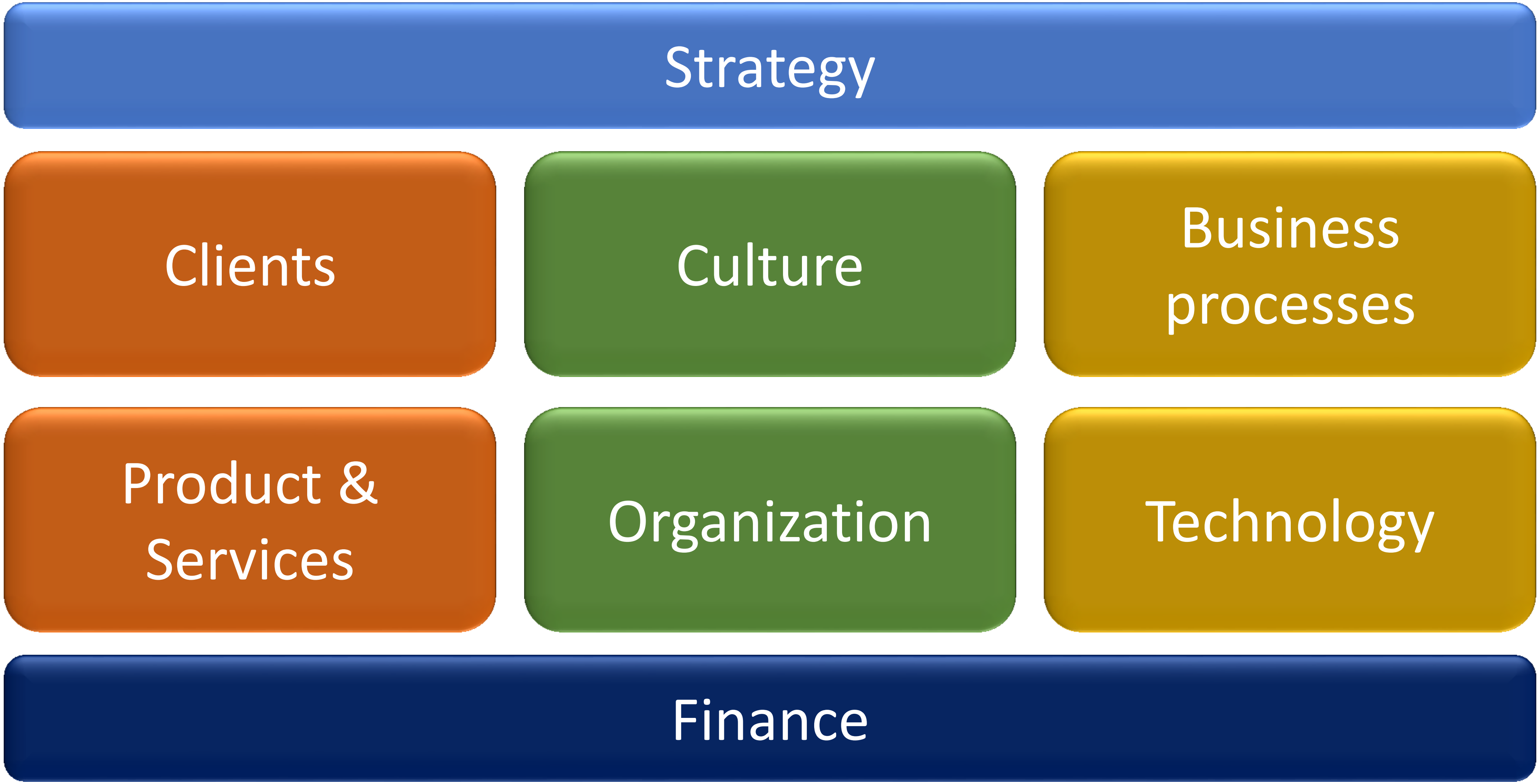
Facultat d'Economia
i Empresa

What are we going to do?

- Presentation.
- What is Business Analytics?
- Impact of Business Analytics in a Company. General approach.
    - Objectives: We will dedicate this sessions to analyze what is the impact of gathering, managing and analyzing data in the different areas of a Company
        - Strategy.
        - Technology.

General Approach

General Approach



Strategy

Clients

Culture

Business processes

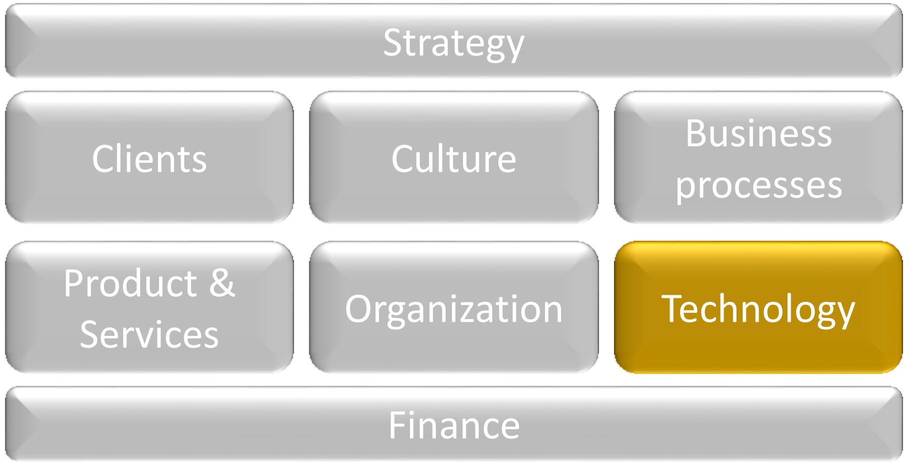
Product & Services

Organization

Technology

Finance

Technology



Strategy

Clients

Culture

Business processes

Product & Services

Organization

Technology

Finance

Understanding Technology.

- Technology is complex, and it can be difficult to evaluate objectively.
- Whether you're a business leader or an investor, you must ensure you’re equipped with the right tools to ask informed questions, evaluate current processes, and make decisions to effectively drive business
- In this session, we’ll discuss the following key points to focus on when evaluating your technology in order to make informed business decisions:
  - The Cost of Technology
  - Technology growth
  - Architecture

The Cost
# of Technology

The Cost of Technology.

- Unlike areas with clear, fixed budgets, technology requirements can change quite quickly and often lead to unexpected costs. To understand technology spending and its influencing factors, we will consider:
    - Investment in New Features
    - Maintenance Costs
    - Technical Debt Costs
    - Infrastructure Costs

Investment in New Features

- Developing a new feature is one of the most cost-intensive areas of technology. Creating new features takes time; It requires identifying the problem, solving it, and transcribing an idea into the code itself.
- Various factors influence the cost of a new feature.
    - You must consider the development environment’s infrastructure
    - Your team
    - The way your projects are managed
    - And integration possibilities.

Investment in New Features

Infrastructure

Infrastructure can have a significant impact on the cost of developing new features.

Cloud infrastructure is in high demand at the moment, so it’s worth considering whether your new features should be developed for on-site or cloud architectures. While both come with their own set of costs, by selecting the infrastructure type before developing a new feature, you can avoid costly reworks in the future.

Project Management

It’s not only the members of your team that can influence the development of your features but how they approach each project. Poorly managed projects can cause unforeseen development roadblocks, which can lead to higher costs.

Flexibility is necessary to react to changing circumstances such as market factors, user needs, technology shifts, etc.

Team

While it may be more cost-effective in the short term, hiring junior over senior developers can prove to be more costly in the long term. Inexperienced technology staff may have greater difficulties or need more. Business leaders should ask whether their development teams are adequately positioned to efficiently manage high-quality technology.

Integration

Ensuring your new features are equipped to handle third-party integrations is yet another factor to consider when evaluating costs. Third-party business application integrations can be particularly challenging because compatible code must be developed that also fits your project's pre-existing architecture. The cost of integration combined with additional licensing fees can significantly drive up development costs.

Maintenance Costs.

- Maintaining your technology is an essential aspect of technology sustainability, but organizations often find themselves spending too much or too little on maintenance-related resources. Creating security updates, fixing bugs, resolving bad code, and data migration can contribute to additional technology costs. Understanding where and when to spend money on maintenance efforts is crucial to technology sustainability. Without maintenance, technology can become susceptible to security threats, bad code, and poor user feedback.
- In general, technology maintenance efforts is divided into four distinct categories:
  - Adaptive
  - Corrective
  - Perfective
  - Preventative

Maintenance Costs.

Adaptive

Adaptive maintenance refers to modifications to the technology so that it can perform despite changes to its environment.

Corrective

Corrective maintenance is the maintenance required to diagnose and fix errors.

Corrective maintenance reflects only 17.4% of total maintenance budgets in general. But some studies indicates between 36.0%-39.0% for corrective efforts as time evolves. This indicates that maintenance-related costs are now trending towards correcting existing errors over perfecting working features

Perfective

Perfective maintenance makes modifications to enhance existing features and functionalities according to new user requirements.

Perfective maintenance contributed to 60.4% of total maintenance budgets in average. 20 years later, a study found that perfective maintenance efforts have lowered to 53.4-56.7% of total budgets.

Preventive

Lastly, preventative maintenance addresses issues or errors that have not yet manifested to improve technology reliability

Maintenance Costs.

- How much should you expect to spend on maintenance costs? Of course, the final number depends on several factors, many of which are unique to the product and industry at hand. A good question to always ask yourself before initiating any maintenance work is the following: “What are the implications/consequences of not adequately maintaining this area of my technology?”
  - Declining client satisfaction. A product that is not well-maintained will suffer from critical user feedback.
  - Your long term product sustainability is also put at risk. A product that does not receive sufficient maintenance updates will have difficulty remaining on the market long-term.
  - That said, it is possible to spend too much on maintenance. If maintenance impacts the ability to innovate and develop new products updates, it may be worthwhile to shift your maintenance priorities. If fixing a problem will cost more than the problem’s relative effect on the product, you might want to bookmark that issue for future product enhancements

Technical Debt Costs.

- Technical debt refers to issues in the technology that lead to future costs due to the complexity of making changes or adding to the technology.
- Think of technical debt as “interest” applied to a technology. The longer the “interest” goes unaddressed, the more you’ll have to pay overtime on the “principal”—or in this case, the original bad code. The longer you ignore technical debt, the more costs (financial, time, and labour) will accrue over time. By addressing technical debt in due time, business leaders can mitigate errors that can lead to higher costs down the line
- Although technical debt can be quite costly, not all technical debt is created equal. Believe it or not, there is actually a degree of technical debt worth having. For example, if the issue costs more time, money, and resources than the problem it creates, it may be worthwhile to leave the issue unaddressed. At the same time, if an issue will continue to worsen over time, then fixing the problem should be a priority
- There’s certainly a balance to strike regarding the amount of technical debt that is appropriate for your organization. Focusing efforts on avoiding technical debt in the first place and correcting it when need be can help you avoid unnecessary costs in the future

Infrastructure Costs.

- Infrastructure refers to the components necessary to operate and maintain an IT environment, including hardware, network structure, operating systems (OS), data storage, software, and more.
- Historically, infrastructure is managed and maintained by on-site IT departments to ensure that organizations’ enterprise software solutions integrate with existing systems and maintain data residency. On-site infrastructure provides in-house control of system management, enables faster backup and restore times, and often does not require an internet connection to access.

Infrastructure Costs.

- Unfortunately, maintaining on-site infrastructures can be very expensive. Your infrastructure’s base costs depend on several factors.
  - Data servers, for example, can heavily impact infrastructure costs. They’re expensive to operate and require climate-controlled environments and sufficient physical storage space.
  - Other factors like the amount of data storage, the amount of information uploaded, downloaded, etc., also impact on-site IT infrastructure costs.
  - Additionally, on-site IT infrastructures require preparation for disaster recovery scenarios. For example, suppose the server loses power or becomes damaged due to external events like natural disasters or a cybersecurity threat. In that case, disaster recovery policies may require additional data servers, backup generators, or threat remediation strategies.

Infrastructure Costs.

- In contrast, cloud infrastructures are quickly becoming a highly sought-after alternative to on-site infrastructures.
  - By hosting their infrastructures on secure web servers, organizations can reduce on-site costs while generating on-demand access to network servers from any location.
  - In addition, cloud infrastructures can create cost savings by slashing IT budgets associated with storing and operating data servers, reducing full-time IT personnel, and creating more resilient disaster recovery plans that are less susceptible to outages
- It’s important to note that cloud infrastructure is not a one-size-fits-all solution. Although the industry is trending towards cloud computing, it might not be right for every organization to implement it immediately.
  - Cloud migration, for example, can cost your organization a significant amount of time and money.
  - Security breaches, for better or for worse, are outside of your control.
  - Lastly, there may be some change management concerns for those that are resistant to the idea of needing an internet connection to access servers

Key Questions.

- Here are a few questions you should ask yourself when evaluating technology costs:
  - Do I have the right talent in place to efficiently manage high-quality technology?
  - How much technical debt do I have?
  - How much am I spending on the different maintenance categories (adaptive, corrective, perfective, and preventative)?
  - Is my product cloud-first? Is my product cloud-ready? Will it need to be in the future?
  - How much am I saving by eliminating on-site data servers?
  - Do I have a clear idea of the actual split between new features, maintenance costs, technical debt costs, and infrastructure costs?
  - Has my technology infrastructure been designed to reduce rework and eliminate redundancies?

Technology Growth

Technology Growth

- Your products will naturally go through several stages of growth throughout their development. From start of concept to product launch, your technology can evolve dramatically. The introduction of new features, regulatory requirements, and market needs can have profound effects on your technology strategy.
- By correctly identifying the stage of growth your software finds itself in, you can shift your teams’ human resources appropriately and keep an eye out for potential issues that can arise farther down the line

Technology Growth

Perfectionist

The perfectionist growth stage refers to times when your efforts and resources are concentrated on improving or perfecting existing technology. The perfectionist growth stage is often less efficient than other stages due to the amount of time and financial resources spent on "perfecting" technology.

High Yield

Compared to other stages of tech growth, the high-yield stage invests time and resources to create new tech features. Unlike the perfectionist and maintenance stages, which focus on existing tech features, the high-yield sstage develops new features for future releases or next-generation products.

Maintenance

Unlike the perfectionist stage, the maintenance stage works to resolve flaws in the technology. Maintenance may require cleaning up poorly written code, responding to user feedback, or making significant adjustments to software architecture. Compared to the perfectionist growth stage, the maintenance stage is considered a more efficient stage because it creates more functional improvements and takes less time than the perfectionist stage.

Exploratory

The exploratory stage refers to a strategic approach to product development. The exploratory stage includes assessing where user problems exist, what the market needs are for a product, and the technical innovations required to create that product. From a resource perspective, the exploratory phase can be one of the most financially intensive aspects, often resulting in high costs and low yield.

Key Questions.

- A product will likely exist in one or several stages of tech growth at any given time. By being aware of what stage a tech product is in or will be in the future, you can shift the focus of financial and talent resources to the most appropriate places. Consider the following questions when evaluating the growth stage of your tech:
  - What stage of growth is my tech in?
  - Am I creating an excessive amount of code compared to similar industry products?
  - How does the talent of my team relate to quality?
  - By focusing on one area of growth, am I lagging behind in the creation of new features or maintenance efforts?

Architecture

Architecture

- IT architecture of a computing system is the structure or structures of the system, which comprise of software elements, the externally visible properties of those elements, and the relationships among them.
- Tech architecture is the organization system between different components, such as an interface or an encapsulated area of software and their relationship to one another. Without architecture, tech would struggle with compatibility between different components, features may not integrate, functions may not perform correctly, and errors may prevent the software from fulfilling programmed objectives

Architecture Best Practices.

Structure

Establishing an efficient tech structure that is consistent across software areas can result in shorter development times. Rather than "reinventing the wheel" each time a new feature or function is introduced, applying the exact structural blueprint can reduce development time, minimize bugs, and improve stability.

Relationship with Costs

Evaluate how the tech architecture plays into operating expenses. Although certain fixed costs like hardware, servers, and general utility costs are unavoidable budget necessities, it’s important to determine whether architecture optimization can reduce additional third-party subscription costs

Scalability and Elasticity

Scalability and elasticity are essential, particularly for client-focused SaaS applications. Not every software client may be the same size or require the same features. By creating scalable, elastic tech solutions, business leaders can attract clients of all sizes and expand their market audience. By creating flexible tech architecture capable of evolving to shifting needs, you can plan for future technological advances.

Reliability

When it comes to disaster recovery scenarios, it's not if it will happen, but when. Is data information backed up on an external system in case of a power outage or system crash? Does the software architecture support fixing bugs from system updates or coding errors? Maintaining a stable and reliable environment is required for long-term software sustainability

Improving Security.

- Tech security doesn't have to be reactive. Requires regular review and hard-wired vigilance to prevent cyber threats. Rather than wait for a security incident to occur, here are a few ways that you can play an active role in improving code security:
  - Designate a Security Officer Whether this means selecting a current team member or hiring a specialist from outside the organization, a security officer should be aware of common vulnerabilities and trending threats
  - Provide Security Information. Security information may include system requirements for developing the code. This should be easily accessible and frequently updated.
  - Train Employees on Security Practices By implementing security training before a security incident occurs, software organizations can turn their team members into their most vigilant protectors.
  - Hold Team Members Responsible.By holding team members accountable for the security of the information in their possession, you can help prevent unwitting accomplices and vindictive insiders

Key Questions.

- Well-developed tech architecture can help you make the most out of your existing work, create standardized processes, and help strengthen security. By understanding and addressing issues with your current architecture, you can help your teams work more efficiently and securely. Evaluate your system’s architecture and security by asking yourself the following:
  - Is my structure scalable to grow with future software offerings?
  - How can my current tech architecture be maximized to minimize additional work?
  - Can my IT be connected with other applications? Is it cloud-ready?
  - How secure is my technology?
  - How can I improve my security?

Thank You

UNIVERSITAT DE BARCELONA

Facultat d'Economia i Empresa