Digiteum

20 Tips to Get an Accurate Estimate
for Your Software Project

Cut risks and gain control over your software project

What do Quibi, Fuhu, and Magic Leap have in common aside from catchy names? They all were once promising products that tragically failed due to underestimated complexity and poor budgeting.

This fate hits hundreds and thousands of products. In fact, **45%** of software projects run over budgets, delivering **56%** less value than expected.[1]

NHS digital modernization[2]
Estimated: around £6 billion
Real cost: more than £10 billion

healthcare.gov website[3]
Estimated: $292 million
Real cost: $2.1 billion

An **accurate estimate helps you cut risks** and **paves the way to the success of your software project**.

It gives you confidence and control. You see the scope and know exactly how much money, time, and resources you need to get your product out the door. You can weigh the odds of hitting your delivery and business targets. Most importantly, you secure yourself from the bottomless pit of budgeting fails and business losses.

But getting an accurate estimate requires all hands on deck. It involves communication, research, and active participation. Both from you and your vendor.

Why should you get involved?

Why shouldn't you? After all, it's your project and your budget. Quality estimate guides you through your decisions and determines how effectively you can allocate your resources. Without it, you may compromise your success by draining your budget and wasting your opportunities.

And that's why you should get involved. Show up on discovery calls with your vendor. Bring other stakeholders with you — the more the merrier. Ask questions and give honest answers. Gather and share all the information you can to help your vendor give you a precise estimate. Your effort will pay off.

This is **Tip 0** and perhaps the ultimate advice from the list below.

Tip 1. Bring everything you have to the table

The more materials you provide to your vendor, the more precise estimate you get. It can be anything from user profiles to a signed-off design. Below are some examples of what you could share with the estimation team.

Must have:

- Documented business process/scenarios/use cases
- User descriptions and types
- User journeys and flows
- Mapped out technical and business requirements

Nice to have:

- Project priorities
- Wireframes/mockups
- High-fidelity design
- Design guidelines/brand book
- Links to similar solutions and inspirations
- Compliance and legal requirements

Tip 2. Highlight your core features

Can you define your competitive advantage, your killer features, and your value proposition? Share this information with your vendor. This will help the estimation team understand what lies in the core of your system and assign priorities.

Tip 3. Show the vendor your existing system

If you have an existing system, share a test account with your vendor. Or conduct a detailed demo and show them around. The more your vendor knows about your system, the more clarity they have. And the more accurate estimate you receive.

What if you have no or only some materials to share? No worries! Your vendor can help you **build up the foundation for your project using the following strategies**.

Scoping Workshop

What is it? Scoping Workshop is an **interactive 2- to 4-hour session** with your estimation team. In this short time, you and your vendor collaboratively define the scope and priorities of your project, fill in the gaps, and prepare all you need for cost-effective implementation.

Does it suit me? Scoping Workshop is ideal for you if you have some documented requirements or other materials but lack details and need help setting up the scope, priorities, and plan for your project.

What do I get? Deliverables depend on your input and goals, but at a bare minimum you get:

- Detailed report with documented requirements
- Project priorities and roadmap
- Tech stack and team recommendations
- Precise time and cost estimate

Discovery

What is it? Discovery is a **research phase that takes from 4 weeks to up to several months**. It involves profound business and technical analysis and covers every aspect of your software project. It consists of interviews, workshops, and in-depth research to build a solid foundation for your project.

Does it suit me? If you have nothing but a project idea or a business need, Discovery is the right start. BAs, architects, engineers, and design experts translate your idea into product vision and development strategy aligned with your business objectives.

What do I get? Discovery ends with a rich set of deliverables. Depending on your needs, you receive:

- Project goal statement and value proposition
- Report on competition
- User personas
- UX flow diagram and design concept
- Documented and prioritized requirements
- Project scope and tech stack
- High-level system architecture and integration strategy
- Product roadmap from MVP to final release
- Precise time and cost estimate

Tip 4. Clarify business rules and hidden logic

No one knows the ins and outs of your business process like you do. There are unique business rules, dependencies, workflows, and hidden logic that may have a great impact on your system and its features. Think about how you could transfer this information to vendors. Failing to do so may result in underestimation and costly changes.

Example: When we work on digital transformation projects, we often have to migrate manual internal processes into workflows in a custom digital system. These processes may include unique rules and informal agreements between people.

Translating these nuances is not an easy task in itself. And it gets harder if we can't identify hidden logic and business rules in time. This may lead to underestimation, delays, and expensive reengineering.

Outcome

Outcome
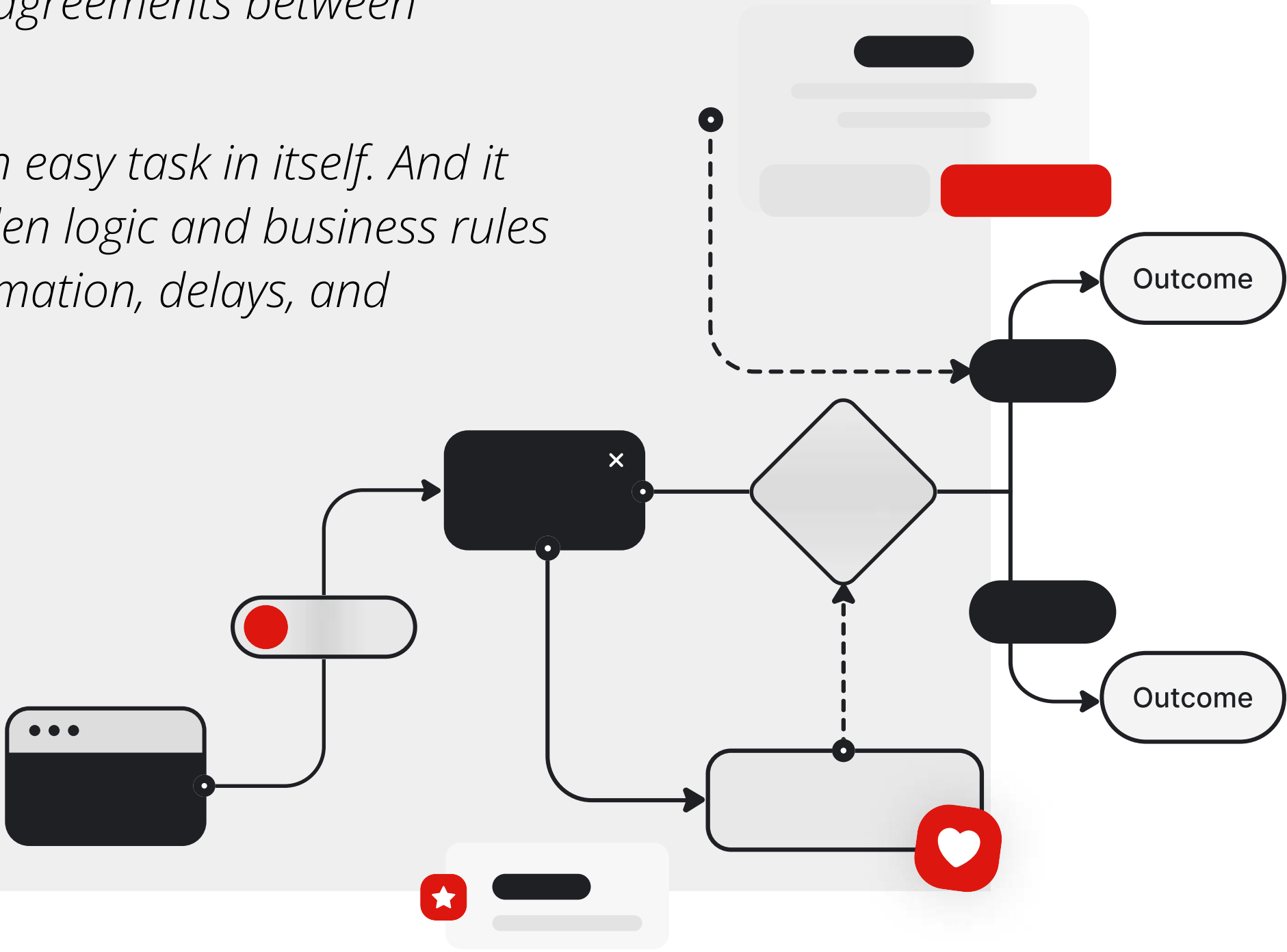
Tip 5. Define success metrics

Clearly define what success looks like for your project. This may refer to both what your system should do (functional requirements) and how it should behave (non-functional requirements).

If you have any specific requirements for speed and performance, need to meet certain compliance or security criteria, or want to hit some business goal, communicate this to your vendor. Even if it doesn't impact the scope, it may still change the workload and, therefore, the costs.

Tip 6. Share data for competitor analysis

You know your competitors. You know what they do well, and what you do (or can do) better than them. Talk about this with your vendor. Share a list of competitors, and specify what exactly you like or dislike about them.

Some competitors might be the Apple or Netflix of your industry, and that's why you mention them. Others might be smaller than you but have a killer design, and you want your system to look similar. Or have an excellent set of features or a specific feature you want to replicate.

Tip 7. Get other stakeholders onboard

Analyze your workflow and define the people and departments in your organization who will work with or otherwise be affected by your system. Then bring them into the conversation.

Different stakeholders have different expectations, priorities, and perspectives based on their role in the workflow. Hear them out and share insights with your vendor. This will help the estimation team clarify requirements, create a realistic workload breakdown, and get you a precise estimate.

> **Example:** *If you plan to build an application to optimize supply chain operations, call Darryl from shipping and warehouse management. Customer support portal? Kelly from customer support. Sales automation platform? Jim or Dwight from sales, or better both.*

Tip 8. Invest your time into a pre-estimate discussion

Find time to talk to your vendor and iron out the details before the estimate starts. Even a brief online meeting gives you a chance to get to know the people who will prepare your estimate. You will see how it speeds up the process and information exchange. So you can get a well-informed and accurate estimate.

Tip 9. Talk about your long-term goals

Where do you see yourself in 5 years?

Just kiddin!

If you have plans to scale your system in the future, mention this to your vendor from the beginning.

Talk about your growth vision. Do you expect to onboard 1 million new users within a year, add new modules, scale to new markets, or integrate with large platforms? All this impacts how technical specialists look at the system, what architecture and technologies they offer, and how they break down the workload.

Tip 10. Don't hold back when talking about challenges, limitations, or pain points

This is a tough one. If you want to get a realistic estimate — not an optimistic one — be honest with your vendor. Share about the challenges and problems you have with your current system or process if this is relevant to the project.

Suppose you want to build your system based on internal workflows. But these workflows have gaps or inconsistencies. Explain this to your vendor. Help your vendor understand what doesn't work or works poorly. Failing to disclose this information may lead to severe underestimation and very expensive rework.

__Example:__ You want to replace your current customer support system with a new one. Tell your vendor what exactly is wrong with the current system and why you want to switch. Explain how you use it and what you expect from the new system.

By default, your vendor assumes that you use out-of-the-box customer support as... Well, as it's written on the box. So they will estimate accordingly. If you don't share that the system is missing important features (so you had to build them yourself) or integrations (so you had to tinker with it to make it work), they won't account for it during the estimate.

Tip 11. Pay attention to the questions vendor asks

You can learn a lot about the competence of your vendor based on the questions they ask (or don't ask).

- If the estimation team asks relevant detail-oriented questions, it shows they are invested in your project, trying to understand your business needs.
- Evaluate whether the questions are aligned with your business case. Are they asking about critical features, user roles, or specific business rules? If yes, it means they are considering the practical aspects of your project.
- The quality and depth of the questions reveal vendor expertise and ability to foresee potential challenges and opportunities. This is crucial for a reliable estimate.

Tip 12. Define your priorities, then optimize

You know your business priorities better than anyone else. Even if you don't have a detailed list of requirements, you have at least some understanding of the essential features and functionalities for your MVP (Minimum Viable Product).

Your vendor, on the other hand, can help you prioritize and optimize the scope and create a balanced roadmap taking into account your goals, limitations, and resources. For instance, they can analyze must-have features and suggest cutting unnecessary elements or moving them for further iterations. Or identify features that will maximize value early on and bring them to the front.

> *Example:* *If you build a dashboard with data tables, you probably want to add filters. But with minimum data at the start, this feature may have little to no value. So you can move it down the list of priorities and develop it later.*

Tip 13. Share the deadlines

Knowing your delivery goals will help your vendor optimize the roadmap and adjust the workload. Or better yet suggest a development strategy to help you meet your deadline.

> *Example:* *Let's say you are building an application and want to showcase it at an expo in 3 months. In this scenario, you don't even need a fully functional product — a prototype with killer features could be enough to generate interest and onboard investors. Your vendor can help you hit this goal by optimizing the roadmap.*

Your proposal is ready. But your job is not done yet. Whether you received a one-list Excel Sheet or a 20-page proposal, checking the Grand Total number is not enough. Go through the estimate to make sure you have clarity and fully understand where the numbers come from.

The following tips will help you read and comprehend your estimate effectively:

Tip 14. Schedule a call to talk through the estimate

The majority of companies skip this part. Don't make this mistake. Get on a call with your vendor and listen to the estimate presentation. It won't take long. Rebuilding the system because of poor scoping or misunderstanding takes much more.

Your vendor will walk you through the estimate, clarify all uncertainties, and answer your questions. You will better understand the reasons behind each number, the grounds for the proposed range, and the difference between the optimistic and pessimistic estimates.

Moreover, during the presentation, your vendor may reveal the complexities you haven't thought of which could have led to underestimation. It will help you make informed decisions about your project and better judgment when choosing a vendor.

Why do estimates come in ranges?

The realistic estimate is 33% input, 33% experience, 33% teamwork, and 1% hunch. It's true, even the best of us can't predict all the ins and outs and always leave a margin for surprise. That's why honest estimates often come as a range.

The more information you can provide at the start, the smaller the gap between an **optimistic** and **pessimistic** number you get. Like in the examples below:

Scenario 1

Request: *I want to build an app like Pawpton for cat sitters. Can you give me a ballpark number?*

Result: *Ballpark estimate $50k to $150k*

Scenario 2

Request: *We need a booking application for pet owners and sitters. Below you will find the list of main features. Kevin from Product already sketched some wireframes (attached) and Lora from Marketing has a customer profile to start with. Let's jump on a call to discuss.*

Result: *Workshop report with estimate $70k-$80k*

Scenario 3

Request: *We want to hire a development team to build a P2P booking application. In the RFP, you will find the detailed specifications, signed-off design, architectural diagram, and preferred stack. We have exactly 6 months to build the first MVP, and another 3 months to polish the edges. I cc'd our architect Mark to schedule a discovery call and discuss the next steps.*

Result: *Proposal with estimate $73k-$75k*

Tip 15. Look for logic and reason in the workload breakdown

You can't get into the minds of your estimation team. But you should be able to get the logic behind workload breakdown and see how it affects the cost of your project.

- Make sure the features are broken down reasonably into the components/ business functions of the system.
- Different phases, stages, and major components should be singled out in the estimate (e.g., Discovery, UX/UI, development, testing, debugging).
- The breakdown should be detailed enough to provide a clear view of the system and be easily readable.
- It should include everything you discussed with the vendor. The development part must be particularly detailed.

Example: *Workload breakdown for building a website to order pizza online.*

POOR WORKLOAD BREAKDOWN   **VS**   REASONABLE WORKLOAD BREAKDOWN

Task	Days	Comments
Architecture	X	
UX/UI design	X	
Data layer	X	
Navigation	X	
Website frontend	X	
Logic for pizza ordering	X	
Checkout	X	
CMS	X	
QA/debugging	X	
Project Management	X	
TOTAL	XX	

Task	Days, min	Days, max	Comments
UX/UI			
UX prototype	X	Y	Wireframes, clickable prototype
UI kit	X	Y	Responsive UI
Design system	X	Y	Set of reusable components
General			
Architecture	X	Y	
Data model	X	Y	
Development			
Homepage/pizza list	X	Y	Display all available pizzas
Pizza search	X	Y	Filters, search pizza by ingredients
Pizza selection	X	Y	Select pizza by size, customize ingredients
Shopping cart	X	Y	Add, remove from cart
Checkout	X	Y	Add billing, shipping, apply coupons if there are any
Payment	X	Y	
Confirmation email	X	Y	Send confirmation email to user
Post pizza order to order tracking system	X	Y	Send orders from the new website to restaurant order management software
CMS: info pages	X	Y	Integrate with CMS to create info pages (About us, Contact us, etc.)
QA	X	Y	
Debugging	X	Y	
Project Management	X	Y	
TOTAL	XX	YY	

Tip 16. Check hierarchy and dependencies

Dependencies between different components in the system impact priorities and project planning. A reasonable project estimate should show a clear hierarchy between system components. If you see it, great! It means your vendor understands dependencies and considers them when estimating your project.

Tip 17. Look into a suggested roadmap

If your vendor knows your priorities, goals, and limitations, they can suggest a project roadmap along with the estimate. This means breaking down the workload into phases and grouping tasks into logical, manageable iterations. So you can ship your project gradually, bringing value early on and reducing the risks of running over budget.

Tip 18. Analyze cost management ideas

Does your vendor offer strategies for effective cost management? This may include optimizing project scope, re-prioritizing features, or simplifying the design to help you hit your budget and timeline goals. If your vendor suggests opportunities for managing costs effectively, it's a good sign. They understand project requirements and are invested in your success.

Tip 19. Go through assumptions

Your estimate may include a series of assumptions — factors and conditions that your estimation team relies on when calculating the time, resources, and costs. Assumptions usually refer to design, technologies, scope, change requests, resources, or any other aspect that may change during the project.

> **Example:** *We assume that the application is compatible with iPhone 13-15 only. If you want to scale the application to iPad or support older or newer versions of iPhone, the provided estimate is not valid and needs to be reevaluated.*

Tip 20. Accept the fact that you may have to re-estimate

You may come across a note saying that you have to get a re-estimate after a certain phase or stage of the project. More often than not, it relates to UX/UI design.

> **Example:** *Phase 1 of the project will take xxx-yyy hours, assuming the current design is complete, approved, and won't change during the implementation. Changes in design will require re-estimate.*

Why? Because changes in the design may (and will!) significantly impact the development workload, and therefore, costs. Only after having a complete signed-off design can you have a precise estimate for the development, testing, and debugging.

More about this in FAQ!

FAQ

What are QA and testing?

QA and testing ensure the quality, reliability, and performance of your software. These tasks include **finding and reporting errors and defects and validating fixes after the issues are resolved**.

You may see either testing or QA in the estimate. Know that these are the two sides of the same protective shield safeguarding the quality of your software.

What are stabilization, debugging, and bug fixing in the estimate?

Don't let the names scare you. Stabilization is the essential phase in software development. Debugging and bug fixing are just other ways of calling it, but essentially, they refer to the same tasks. Your estimation team allocates time to these tasks to **analyze and resolve issues and make sure your software works and performs as it should**.

Some vendors may not include this phase in the estimate. But it doesn't mean it's not there. It only means they added XX% of time for stabilization (or debugging, or bug fixing — choose one that gives you less creeps) to each feature or task.

What is the normal percentage for project management, testing, and debugging in the estimate?

Testing, debugging, and project management (PM) usually account for **15%-30% of the estimate**, depending on:

- Supported platforms. If you are building a cross-platform application, you have one codebase but still need to test your software on all supported platforms.
- Available documentation. The better the documentation is, the less PM time is accounted for.
- Size of the team. The smaller the team, the less PM time you will need.
- Number of stakeholders. The more stakeholders are involved, the more PM time will be allocated for communication.

Testing and debugging should be added to each major component in the estimate. If the project is divided into phases, testing, debugging, and PM should be added to each phase separately.

What are the infrastructure (environment) setup, architecture, data model, and documentation in the estimate?

As we mentioned in Tip 15, reasonable workload breakdown revolves around business functions and system components. However, you may still come across technical tasks like infrastructure setup, architecture, documentation, or data model in your estimate. These are the **critical activities aimed at building the foundation for the software and software development process**.

Some vendors don't show them in the estimate but add the time for technical tasks into each component. So if you don't see them listed in the breakdown, it doesn't mean it's not there (same as debugging). Your vendor simply doesn't single them out as separate tasks so that not to confuse you.

How much does design impact the estimate?

Some systems have more intricate designs than others (eCommerce application vs. B2B system with custom workflows). **Unique complex designs have more impact on estimate and costs**. That's why it's very important to understand the proportion and role of design in your project.

For systems and components with complex business logic, focus on getting a more accurate estimate for development. It has more risks to be underestimated and go over budget. The more precisely you can estimate the risky part, the better cost control you will have. (You will thank us later!)

Example: *Coupon/discount system has a very simple design with easy implementation on the frontend but a very complex logic and multiple dependencies on the backend.*

Should infrastructure cost be included in the estimate?

Normally, **infrastructure costs are not included in the estimate**. However, if you need this information to get a full picture of project costs, discuss with your vendor to have it calculated and included in the estimate.

What factors other than scope can seriously influence the estimate?

Design complexity, non-functional requirements (performance, security, reliability, scalability), and **integration with third-party systems** have a major impact on the estimate.

For instance, if you change one third-party system to the other one, it can change the workload and cost. The choice of infrastructure also has a big influence because it determines the skills, time, and effort needed to set it up and running.

How to manage costs on a software project?

Cost management on a software project **goes hand in hand with requirements management and prioritization**. To help you manage costs, your vendor may suggest to optimize the scope, put less critical features on hold, or simplify requirements (e.g., instead of implementing both search and filter on lists, choose the more important feature and postpone the other).

How to optimize costs on a software project?

An accurate project estimate is the first step to cost-effective software development. Use the following **best practices to control costs** further on:

- Prioritize requirements and features based on your business goals, taking into account your limitations, deadlines, and available resources.
- Balance custom and off-the-shelf: create custom business-critical components and use third-party integrations for secondary features where possible.
- Leverage reusable components, libraries, and tools.
- Embrace agile and set up effective software development practices.
- Consider outsourcing part or all development to benefit from lower rates for hiring top-tier tech specialists with your vendor.

1. Delivering large-scale IT projects on time, on budget, and on value. McKinsey & Company.
2. Abandoned NHS IT system has cost £10bn so far. Guardian News & Media.
3. Obamacare Website Costs Exceed $2 Billion, Study Finds. Bloomberg L.P.

Secure the success of your software project with Digiteum

We have delivered hundreds of estimates, detailed proposals, and discovery projects during over a decade in custom software development. The tips we share are the result of immense practical experience backed by the expertise of senior architects and engineers, business analysts, product owners, project managers, and designers.

If you are looking for an accurate estimate for your software project that will:

- Give you control over your project costs
- Put your business goals and priorities first
- Help you bring value early on
- Provide clarity regarding resources and timelines
- Lay out the foundation for the success of your project

CONTACT OUR ESTIMATION TEAM

> "
> Digiteum provides thorough cost estimates, attentive project managers, and well-constructed products. Fast, precise and with great communication and project management skills.
>
> **Mike Burgess**,
> CPO,
> WGSN

> "
> They'll ask if we've thought about doing something else or we could do it faster if we drop these features to get it to market so you could test it. They will tell me if they think I'm going in the wrong direction. I really appreciate that.
>
> **David Zack**,
> Head of Software Development,
> T&F Technology

OXFORD UNIVERSITY PRESS   Diaceutics   PRINTIQUE AN ADORAMA COMPANY   KANTAR

OriginDIGITAL   ORACLE   Diflexmo   T#INKDONATE

native   ROLLOMATIC   LODGIFY   MSD

About authors

Katherine Lazarevich
Co-Founder, Digiteum

katherine@digiteum.com

Katherine is a co-founder of Digiteum with two decades of practical experience in growing successful digital products and building high-performing software development teams. Together with Digiteum, she helps profitable companies reach their business and innovation goals. Katherine is a tech enthusiast and a strong advocate for elegant user-centered design, bringing a wealth of expertise in UX/UI and digital product strategy.

Viktor Lazarevich
CTO, Digiteum

v.lazarevich@digiteum.com

Viktor is a CTO and a founding member of Digiteum. He works closely with Digiteum clients and hand-picks digital technologies, methodologies, and infrastructure solutions to deliver scalable digital systems while keeping deadlines and costs in check. Viktor champions effective engineering practices and fosters a culture of tech excellence across all projects and teams.