Measuring the ROI of Your Software Projects

Crema

Evaluating technology investments for high-end professional services companies and consultancies is a valuable practice.

Amidst the wealth of information and opinions on these topics, it can be difficult for even the most sophisticated organizations to gain a shared understanding of how to adequately measure the ROI of their software projects.

If you're a digital product owner or in the C-suite at your organization, you're likely evaluating how technology investments will impact your bottom line. Building custom technology or licensing expensive software solutions are difficult, yet necessary investments to see major gains in your business or organization.

Measuring the ROI of those investments can be an even more complex equation, making it that much more difficult to explain their financial impact.

→

This guide will give you a clearer understanding of how to evaluate your technology decisions and measure their impact on your organization.

Note: *We are not bankers, accountants, or private equity professionals. We're technology practitioners, leaning on our 11-year history of designing and building software products for hundreds of companies. This advice is not one-size-fits-all, but we're confident that you will benefit from the knowledge and exercises shared below.*

In this guide, we'll explore the following topics:

1

Where is software impacting business? **Everywhere.**

It can't be overstated how much software is impacting our world. Even in "non-tech" industries, businesses rely on flexible digital tools and software to meet their internal needs and the expectations of customers.

What's more, in light of COVID-19 (with social distancing and the shift to remote work for many) accessing goods & services digitally is simply our new reality. Plus, users expect a seamless experience 24/7. Operating well as a team in a semi-remote or fully-remote fashion creates new challenges and opportunities to improve as an organization.

Even if your organization doesn't view itself as a "technology company," tech enables organizations to operate well and support their customers. It's a key part of new revenue generation and stimulates innovation within your organization.

Every business uses some technology, but not every business strategically calculates the potential returns these tech investments can create for the organization.

Hopefully, today can be a turning point for your organization by embracing new software and moving forward with confidence in your investments.

Areas that tech can improve:

- Sales enablement
- Internal communication
- Customer support
- Education & professional development
- Inventory management
- Electronic records
- Security
- Marketing tech
- Recruitment
- Human resources
- Facilities operations
- Managing vendors
- Tracking freight
- Internal support
- 3D printing
- Compliance

2

Setting your baseline

The first step in analyzing a potential software investment is calculating how much your organization spends on operating to achieve the same results. In other words, what are you spending to maintain the status quo? Further, how is the status quo holding your team back?

Teams are usually shocked by the cost of their current processes after adding it all up. Between salaries, software licenses you may or may not fully utilize, outside vendors, redundancies, training, etc., your current costs could be higher than expected.

If this is the case, your organization might be spending tons of money supporting a system that is out of date, insufficient, and unable to meet the growing needs of the business, which in turn hinders the ability to grow and innovate.

Outdated systems become so expensive due to the intangible costs of job dissatisfaction. Old and insufficient systems lead to laborious and unfulfilling work for employees, raising levels of absenteeism, and creating a disengaged workforce.

In other words, **the cost of supporting the status quo can be both quantitative** (hard costs) **& qualitative** (soft costs). Usually, it's a compound matter.

→

Coming into Q4 of 2020, we consulted a company where if they did nothing, they were going to spend close to $30M over the next 5 years on their existing processes.

What kind of software can you build and benefit from between $0-$30M? Something incredibly robust – and for a fraction of the cost.

If the quality of the software you're using, or the current processes you have are suboptimal, then it's worth looking into how software can change your organization for the better. You could potentially revitalize your processes, stimulate innovation, add a valuable asset to the balance sheet, transform your organization, and save money.

How would your business be different if you optimized the work with software tailored specifically for your organization?

Calculate what you're already spending

Multiply $X by 3-5 years

- Headcount/salaries _____

- Licenses (Salesforce, NetSuite, etc.) _____

- Third-party services _____

- Training _____

- Marketing _____

- Reskilling _____

- IT Customer service _____

- Security _____

- Compliance _____

- Maintenance _____

- Outside vendors _____

- Legal _____

Total _____

x 3-5 years _____

Action steps:

1. To set a baseline for the project, calculate what you're already spending if you change nothing in your current process.

2. Multiply this number by 5 years to calculate what you could reasonably spend with your current processes.

3. Pick your jaw up off the floor 😲 You didn't know it was that much, did you?

3

Analyzing the potential returns of software investments

Rethinking processes and the business as a whole leads to micro-innovations and new ways of thinking for the organization. This can lead to new lines of revenue, optimizations, cultural renewal within the company, and so much more.

Now that you understand your baseline spend if you were to make zero changes, there are three areas where you should focus your attention when calculating the potential ROI of your software project.

Many enterprise clients are focused on "net new revenue" for obvious reasons. They want to grow their company revenues through scalable solutions and be more competitive in their industry. But there are other areas that are equally as important: operational improvements & intangible values.

Let's break down some key questions to ask in each area.

Revenue gains

Competitive advantage: What is the value if you gain a competitive edge in your market?

Net new revenue: How much net new revenue could you generate with the same amount of staff by operating more efficiently? Will this create an entirely new line of revenue?

Revenue gains

Winning more customer accounts: Does this technology allow you to win new customer accounts you previously couldn't serve? Either through the software itself or by reallocating employee time and accountabilities from cost center activities to revenue generating activities?

Winning consumer preference & evangelism: What would it mean for you to win consumer preference so much that people rave about their experience to other potential buyers?

Increased customer lifetime value: What would it be worth to grow your customer lifetime value?

Operational improvements

Increased understanding of customer needs: What would it be worth to have more & better customer data at your fingertips?

Checks and balances for team workflow: What if you could ensure that your organization is working well and giving people the correct permission levels based on their roles?

Standardize project workflow & deliverables: What would it be worth if you could standardize your projects, team workflow, and deliverables to clients?

Decrease number of engineering/consulting hours per project: What would it be worth to take on more projects with the existing team that you have?

Delivering projects on time and on budget: How could you meet or exceed client expectations, all while streamlining your planning and operations?

Operational improvements

Rapid adaptation in minutes: What would it be worth to be able to adapt in minutes instead of days or weeks and respond to project changes quickly?

Real-time analytics to improve project outcomes: What if you could understand what's working best and where improvements need to be made within your projects, all in real-time?

Opportunistic access & coordinated analysis: What would it be worth to understand data across all your customers and products that allowed you to create more intelligent applications?

Intangible values

Brand value & prestige: What is the value of boosting your customers perception of your brand?

Minimize recurring issues: What if your teams' overall quality improved?

Peace of mind: What would it be worth if all your leaders slept better at night?

Confidence in strategy: What if you gained the confidence needed to move forward?

Awards & industry recognition: What would it be worth to impress the industry and your stakeholders?

ROI
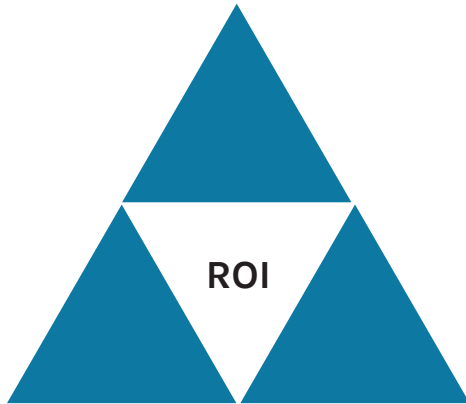
This line of questioning is to open up the potential opportunities in which your organization stands to gain across the board - tangibly and intangibly.

Some organizations will care about impacting many aspects of the business, while others will only focus on a few. You know your organization best, but know that the more areas the software can positively impact, the greater the possible returns will be.

Therefore, we've observed a positive correlation between the potential high value brought by the software, and the funding and internal support around software investments.

Action steps:

1. Identify all of the possible areas of your organization that could benefit from an investment in software, and write them in a goal format.

   Example: By the end of 2021, earn $1M in net new projects without hiring new staff.

2. Assign a dollar value to each item on a per annual basis. Use your best judgement.

   Example: If an average customer is $250K per year, and you think the software will help you take on 2 new projects, then mark $500K for net new revenue.

3. Finally, add up all the potential revenue gains, operational efficiencies, and intangible benefits in dollar format, and multiply these by 4 years.

   Note: We're using 4 years because here as it's assuming you wouldn't see the full benefits of the software until it's built and implemented across the organization, which takes time.

4

What will it take to implement?

Now that we've covered some of the potential compound benefits of doing this work, let's identify what it will take to bring your software to fruition.

It's not as straightforward in most organizations as simply deploying capital and building software to replace what you have. This work takes people, technology and culture to enable the changes needed to transform engrained processes and tools. Change is challenging and messy, and people want to feel safe & secure amidst changing roles, processes, and culture.

Here are some tips to ensure your staff understands what's going on and what is being asked of them:
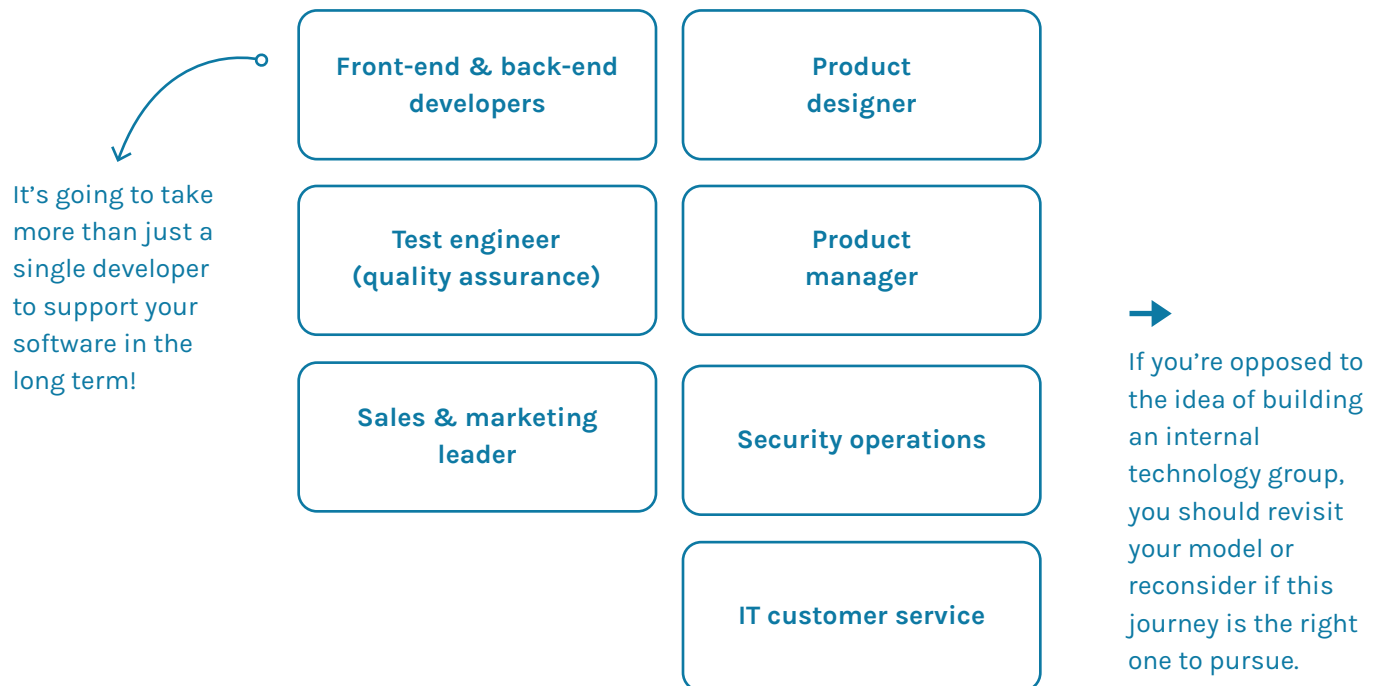
- Articulate a rollout plan that doesn't become too burdensome on your staff or communicate rushed change.

- Communicate early and often about the changes you're hoping to make.

- Articulate a roll out plan that doesn't become too burdensome on your staff, or communicates rushed and overnight change.

- Get your staff's feedback early and frequently to garner buy-in, build excitement, and ensure you are solving their headaches in the right ways.

- Celebrate your wins!

As much as there is room for savings in these software investments, some costs will need to be reallocated from one type of role to another.

For example, you may need fewer operational roles on your team to operate the new model (a savings of $X/year), but you may also need additional roles like developers to support the software over the longterm (a cost of $Y/year).

If your organization does not have any of the following roles, you will need to think about hiring them at some point in your model to support the software and embrace becoming a modern technology organization.

Roles you will likely need on your team longterm:

It's going to take more than just a single developer to support your software in the long term!

Front-end & back-end developers	**Product designer**
Test engineer (quality assurance)	**Product manager**
Sales & marketing leader	**Security operations**
	IT customer service

→ If you're opposed to the idea of building an internal technology group, you should revisit your model or reconsider if this journey is the right one to pursue.

The optimal route would be to partner with a digital product agency for 1-3 years, and work with them to hire and find the appropriate staff to support the product longterm.

Choosing to build software with an outside partner is a good idea for groups that don't have internal development capabilities. This allows you to move quickly, without having either the ongoing overhead, or having to let certain team members go once your product reaches maturity.

Expenses to plan on:

- Hiring staff or a digital product agency for 1-3 years

- Re-skilling & re-training staff

- Software tools, such as third party services, hosting, etc.

- Security operations

- A temporary period of lost productivity (you're never 100% productive when you make a change. Your team will need time to adjust and get ramped up.)

Action steps:

1. Begin entertaining bids for the software project. Usually, an agency will present multiple options and multiple phases with tradeoffs for each approach.

2. Begin the planning process for the recruitment of new hires based on the list above.

3. Calculate what a transition period of 4 months of slightly less than normal productivity could cost the organization.

4. Calculate what training & reskilling your current team looks like.

5. Budget for marketing & education costs, not only for your internal team, but also for clients and partners.

5

A strategic mindset for organizations

Building software is a capital intensive process. By this point, you might have numbers ranging in the high six figures to low seven-figure sums. Similar to if your company was buying buildings or equipment, software development expenses for internal operational software are typically recorded as capitalized expenditures (expenses that become assets).

Internal software that has a useful life of at least 2 years, that won't be intended for sale in normal operations, and that has been created with the intention of being available for use by the entity CAN be classified as capital expenditures for a maximum of 5 years.

Did you know?

The benefit is that the asset is depreciated over time, allowing your company to take the tax deduction on your income statement in more than one year. This can be a benefit, especially as you continue to add value to the existing asset by way of new features, modules, etc. Those expenses can also be depreciated on the balance sheet.

Action steps:

1. Connect with your CFO and discuss how they would plan to layout the cash for the project and plan to depreciate your software project over the course of 2-5 years.

2. Ensure you have a shared vision for the project and what it can bring to the organization.

6

How to calculate your potential ROI

If you've made it this far, you can start seeing what the financial picture will look like for your software investment and if it will be worthwhile. You've put in the work to understand the required inputs and the target outputs. Now let's put the pieces together using the numbers you've calculated along the way to get your potential ROI.

Follow the processes articulated below. You will want to multiply the top by 4 years and the bottom by 5 years in this scenario. This accounts for the first year where full results won't be hit while the software is being built and internal teams and customers onboarded.

Plug in your numbers and see the potential return you could generate

$$\frac{\left(\substack{\textbf{Projected}\\\textbf{revenue}\\(\$2M/year)} + \substack{\textbf{Operational}\\\textbf{improvements}\\(\$2M/year)} + \substack{\textbf{Cost}\\\textbf{reductions}\\(\$1M/year)} + \substack{\textbf{Intangibles}\\(\$x/year)} \right) \times \textbf{4 years}}{\left(\substack{\textbf{Staffing}\\(\$x/year)} + \substack{\textbf{Technology}\\(\$x/year)} + \substack{\textbf{Marketing /}\\\textbf{education}\\(\$x/year)} \right) \times \textbf{5 years}} = \$X$$

*Additional questions to review:*

- Is it 3x or 20x the original investment? (Even some of the best venture capital investors in the world don't see 20x returns. If you can get to a double digit ROI, congratulations! 🎉)

- Does this sound reasonable or do you need to go back and adjust your numbers accordingly?

- Is there still a case to be made if you remove a section like "projected revenue"? Play around with the numbers and error on the side of being conservative with your estimates.

- Is the case still strong to move forward with your project?

7

Results vs. returns

Many clients we work with have to be coached to understand that although results may be experienced quickly in software projects (like improving a small process), the financial returns of this small result may take years to see in terms of a financial return.

This is why it's important to do the work upfront and understand that it's a marathon – not a sprint. Managing stakeholders will be important as you explain the lifecycle of product development and the difference between results & returns.
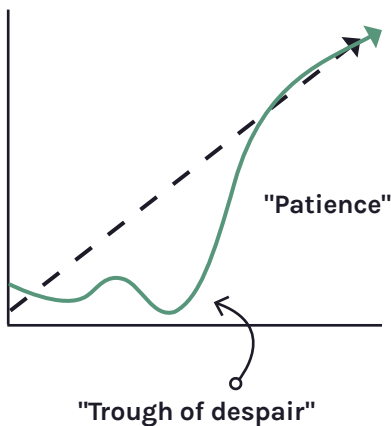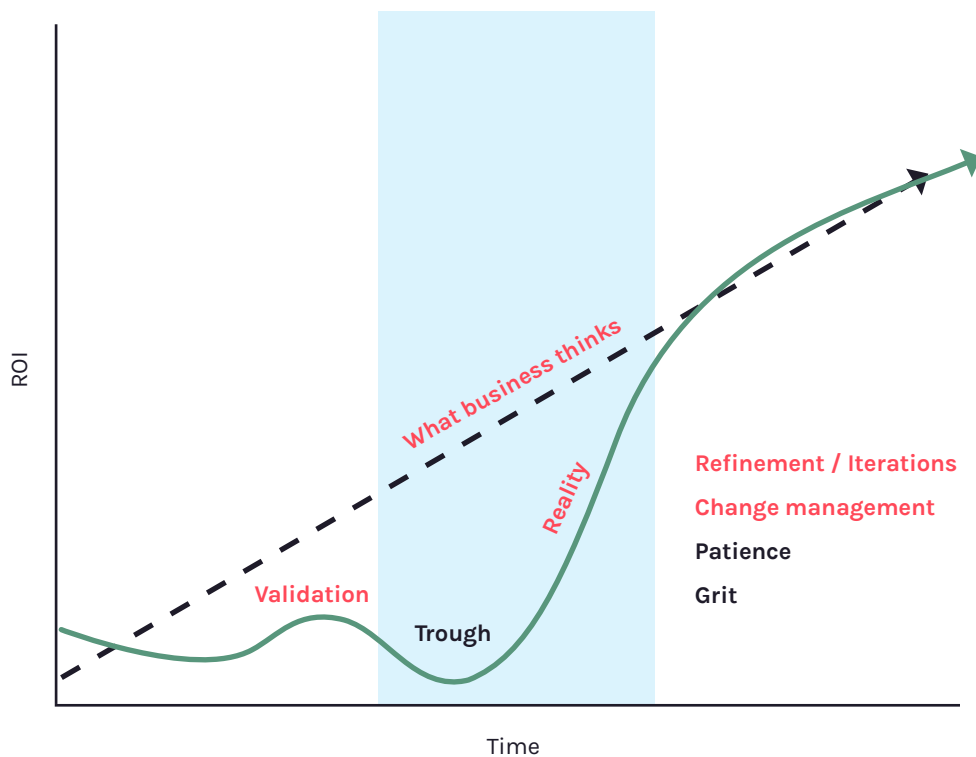
As leaders, we tend to think of progress as a linear progression over time, and that's marked here by our dotted line. In any large endeavor, we're often faced with a "trough of despair," which is how these projects usually play out.

Massive amounts of energy go into laying the groundwork for a successful software project. Small changes and work being implemented don't seem to move the needle enough in the early weeks and months of a software project, and we wonder if we're doing the right thing.

Sometimes, it doesn't feel like the project is going anywhere, but we assure you it's a compounding process. The results are sure to come if you stick to your plan and work closely with your partners. Stay the course.

"Patience"

"Trough of despair"

ROI

What business thinks

Reality

Refinement / Iterations
Change management
Patience
Grit

Validation

Trough

Time

What will you do?

Although, you may see results quickly, understand the financial returns will come exponentially over time. In reality, these projects take time to realize their full potential. That's why you have to have a multi-year mindset.

The reason why many groups defund software projects or pull the plug entirely is because leadership doesn't understand the value of software and the process it will take to get to the end returns. Many projects are cut after their first year. **Remember, the ROI has to be over a multi-year period.** If making huge returns using software was inexpensive and easy, we would all do it!

You likely understand by now why that's a tough pill to swallow. The project had probably just started to pick up speed and start delivering results, but NOT financial returns. Educate your leaders about this process, and show them this book. Hopefully, it will help create alignment and foster good conversations about what results demonstrate progress for the company while holding out for large future financial returns.

8

How to get started

Building software for your organization is an incredibly rewarding experience. You get to break apart day-to-day challenges that your colleagues & clients face and solve them with people, technology, and culture. When you see the software working and that your teams & clients are happy, it will feel like you summited a large mountain (or at minimum a huge career accomplishment).

Today, you're sitting at the base of the mountain looking up - wondering if this is the expedition to embark on. It looks like an enticing journey, but a long & treacherous one to navigate alone.

Move forward from this point in confidence knowing you've already taken the steps to understand what the journey realistically looks like.

If you're ready to move forward, here are some ways to get started:

- Align your initiative(s) toward quarterly or annual business goals

- Gain executive and team buy-in

- Complete due diligence on your solution

- Build a prototype to gain feedback and alignment

- Iterate in design, NOT in code

- Utilize high-leverage solutions like 3rd-party services

- Choose exceptional technology partners (if we can help, get in touch!)

- Set clear metrics and don't move the goalposts

- Measure ROI on a 3-5 year timeline

- Invest in internal/external education and training

- Emphasize learning over perfection

We want to help!

Thanks for checking out our ebook. If you'd like to talk more about how Crema can help your team evaluate the potential of your software project or bring your digital product to life with the help of our software experts, you can reach me directly at nate@crema.us

Learn more about Crema