merixstudio

How to get stakeholders on board

A guide for change leaders to gain buy-in for software modernization

www.merixstudio.com

Table of Contents

03 The human factor in software modernization

Part 1:

04 Why are people afraid of changes?

04 Psychological resistance to change

06 People-centered change management methodologies

09 Key change management takeaways

Part 2:

11 Who do you need to convince and how to do it?

11 The board

15 The users

16 In-house tech team

The human factor in software modernization

Software modernization is all about changes that usually go beyond the code or even the software itself. A broad and profound approach to this cross-disciplinary process increases the chances of success, but it also makes modernization a long and demanding journey. The journey within which you can't paddle your own canoe even if you're the whistleblower. Changes will affect many people in the company, their work, behavior, and responsibilities, so if you don't want them to sabotage the process, ***you need to get them on board****.*

Our experience suggests that the factor usually tipping the scales in software modernization is ***the positive attitude and the awareness of changes by a wide range of stakeholders****. Or rather their lack, as the negative consequences of insufficient buy-in across the company most frequently result in postponing, delaying, or even abandoning the system upgrade. That takeaway is reflected in numerous surveys, such as The State of IT Modernization 2020 or 2020 Legacy Modernization Report. Their respondents cite low priority among other initiatives, a shortage of cooperation between business and IT, or an inability to show clear ROI as the main barriers against software modernization.*

Most of these obstacles can be overcome with a deliberate plan based on solid analysis (and we give a tone of advice on how to do it within the Modernize and grow whitepaper), but before you even start preparing for the change, you need to gain allies. If you would like them to stand by your side, it is necessary to understand what is holding them back.

Why are people afraid of changes?

Change is related to two things that activate people's deepest and most primal fears: ***of the unknown and of loss****. Both induce inner defense mechanisms, thus arousing distrust for any ideas that disturb a safe daily routine. Because these anxieties are psychologically determined, it's hard to combat them. On the other hand, we know enough about them to find ways of lessening their impact.*

PSYCHOLOGICAL RESISTANCE TO CHANGE

FEAR OF THE UNKNOWN

We usually associate fear with negative experiences related to a given situation, person, or subject. However, one of the most primal and fundamental fears concerns the opposite: **the things we don't know, so we cannot predict their impact on us**. The unknowns are inherently appraised as aversive, and that reaction has its evolutionary basis. It has helped the human species to survive by remaining vastly vigilant and avoiding situations and objects that might cause unpleasant effects on them.

What exactly induces the fear of the unknown? Two main factors pull the trigger and raise the anxiety level:

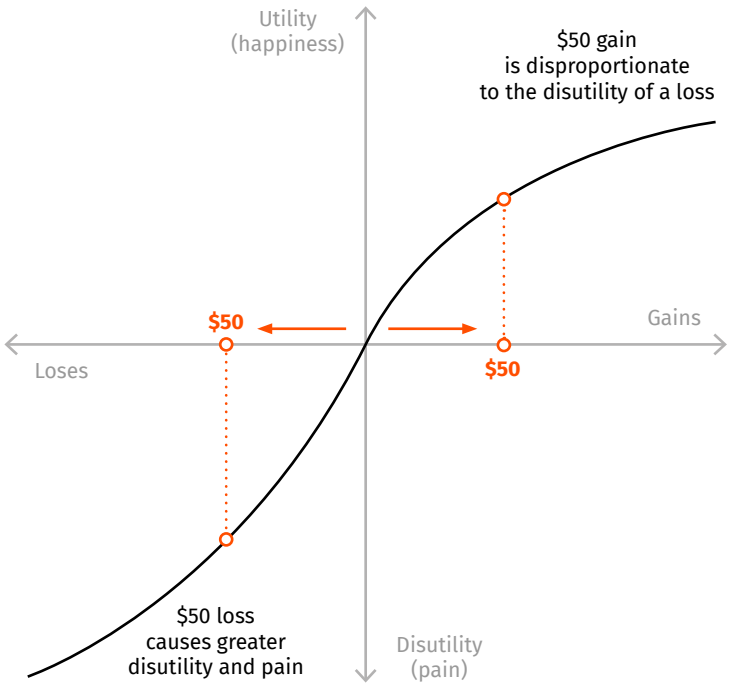
- **lack of information** that enables to predict the future and possible influence of events,
- **loss of control** decreasing people's sense of agency and depriving them of the impact they have on the circumstances within which they operate.

The awareness of the roots of such a strong, natural fear is crucial to deal with resistance to change.

THE LOSS AVERSION

"Losses loom larger than gain" – this asymmetric relation discovered and studied by Daniel Kahneman and Amos Tversky immensely determines our choices and decisions. It also became a root problem for their broader concept

of behavioral economics, prospect theory (which, by the way, significantly contributed to awarding a Nobel prize to Kahneman). One of the main conclusions of the survey the authors conducted was that **the pain of losing something is greater than the satisfaction of an equivalent gain**. That dependency is particularly noticeable in an environment of uncertainty. It makes people favor a certain gain over another that's potentially greater but bears a risk of losses. What's more, even if the outcome of two alternative offers is the same, such as receiving $50, people prefer to get the full amount upfront than to be first given $100 and then deprived of $50 – even though they end up with the same amount of money in their pocket! **The negative emotions caused by loss are much stronger than the rational judgment of benefits or the possible additional gain**.



Prospect theory explained. Source: boycewire.com

But how do people identify what's gain and loss for them? The prospect theory says **it depends on a reference point determining how an outcome is perceived**. That might be some benchmark or just a status-quo that helps evaluate if the choice they face will bring them positive or negative consequences.

Although the authors surveyed how people make financial decisions, the mechanisms they discovered can be applied to a whole plethora of fields, e.g., designing user experience, marketing activities, insurance products... **and communication within the software modernization process – assuming one can learn the proper lessons from Kahneman & Tversky explorations**.

FIGHT-OR-FLIGHT RESPONSE

When people act in an environment of uncertainty, they do not always act rationally. The unknowns turn their vision of the future into a massive threat, as the survival instinct tells it's better to assume the worst. **And when a human needs to face the menace, there are two possible reactions: run or fight**.

The mechanism commonly known as the fight-or-flight response transferred to software modernization circumstances explains employees' negative behavior. It's manifested most frequently either in avoiding engagement in the process or opposing it. You might want to nip those attitudes in the bud while executing change.

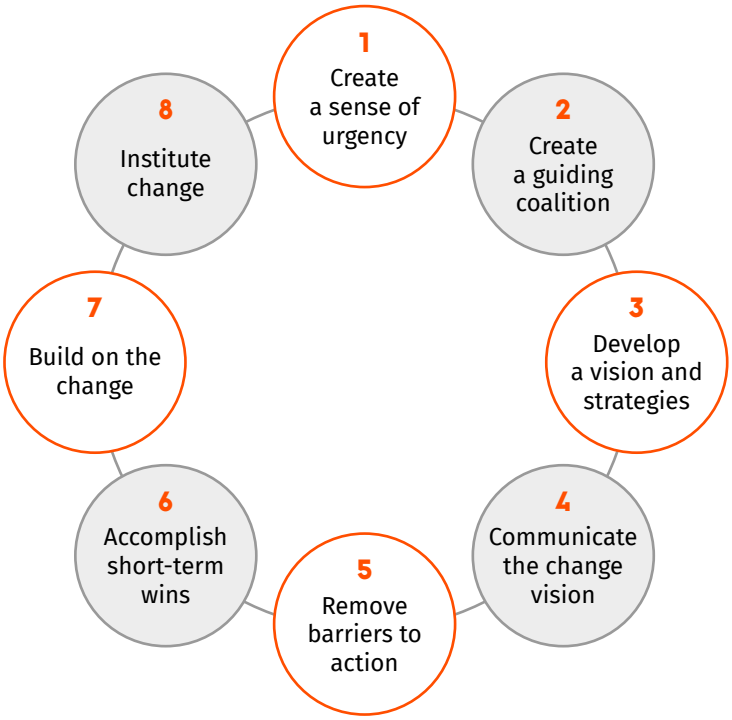
PEOPLE-CENTERED CHANGE MANAGEMENT METHODOLOGIES

The aforementioned psychological theories gave rise to several people-centered change management methodologies. They deliver battle-tested processes that can help to **tame the fears and neutralize the sense of danger**.

KOTTER'S 8 STEP MODEL OF CHANGE

That's one of the most popular models implemented in managing different organizational change processes. It was developed by **John Kotter**, a Harvard Business School professor and an experienced management consultant who introduced the concept in his book Leading Change.

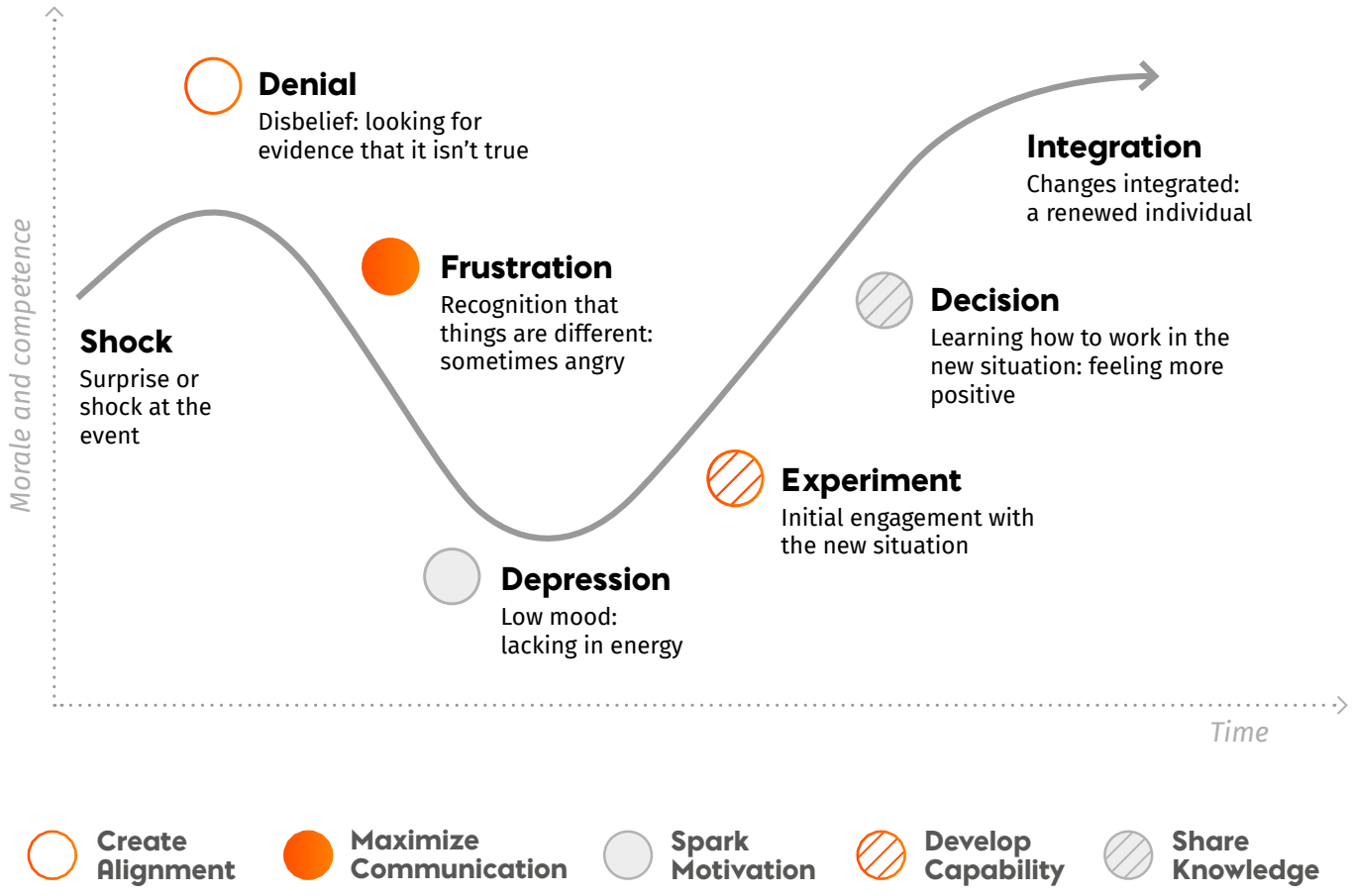
The model emphasizes the significance of **preparing the employees for the change** and helps to understand their reactions and anxieties. It equips transformation managers with the 8-step process and useful tools to overcome people's resistance. The main focus in this approach is put on **gaining allies**, a real army of volunteers supporting the idea, and a strong guiding coalition of effective leaders who drive and coordinate the process. Kotter suggests that the change won't be accomplished unless we ensure the buy-in from at least 75% of a company's management. He also gives prominence to **post-change activities, which can institute the transformation and consolidate its results**.



Kotter's 8 Step Model of Change. Source: creately.com

KÜBLER-ROSS FIVE-STAGE CHANGE CURVE

The model is also known as the **five stages of grief**, and this somber name is not accidental. Its author, a psychiatrist Elisabeth Kübler-Ross, came up with it as a result of her work with terminally ill patients. The curve at the model's heart presents the emotions and their level experienced by the people who will soon approach death. After introducing the theory within the book *Death and Dying* in 1969, the concept was applied across industries and business activities far away from the medical environment. The possibility of its wide implementation in different areas grows from the people's common reaction scheme to even much less severe physical burdens connected to the feeling of loss.

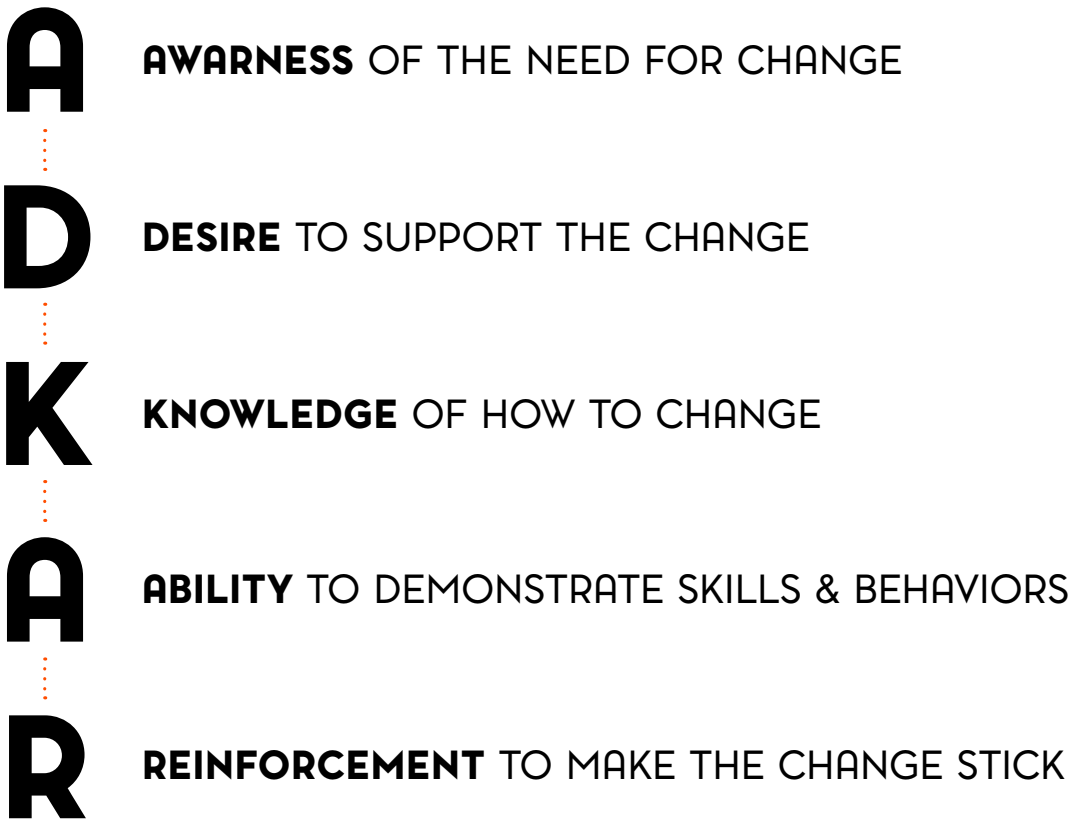


Kübler-Ross Five Stage Change Curve. Source: management30.com

In the organizational change conditions, the model helps managers understand employees' emotions and empathize with them deeply. It also hints at how to react to behavior and communicate with a person at a particular stage of the grief curve.

ADKAR MODEL

ADKAR model is a structured framework created by Jeff Hiatt, the founder of Prosci® – the world's largest body of change management knowledge. It's based on the assumption that **organizational change should be achieved through personal evolution**. The model combines reaching transformation business goals with providing support for the employees to go through the process more easily. Its name is an acronym for the five stages required to be covered: Awareness, Desire, Knowledge, Ability, and Reinforcement.



Prosci ADKAR® Model. Source: www.change2value.com

No matter which model you'll implement, there are basic rules applicable to all of them which need to be taken into account while gaining buy-in from stakeholders.

KEY CHANGE MANAGEMENT TAKEAWAYS

1

Identify target groups

There's no winning formula for effective communication with all stakeholders. It should be tailored to specific needs, expectations, and concerns of every target group.

2

Discover the reference points

According to the prospect theory, there is a baseline from which people evaluate the situation in terms of gains and losses resulting from the change. This point of reference is usually just their status-quo, but it might also be a benchmark for comparison, like a competitor, family member, friend, or neighbor.

3

Pinpoint particular fears and concerns

The cited fundamental fears manifest themselves in a number of concerns specific to different change-affected audiences. Uncover and counteract them, yet stay honest – some of the pessimistic predictions might be reasonable and likely to happen. In that case, it's better to get accustomed to the existence of the problem and talk the possible solutions over.

4

Show opportunities and create a need

Even if people across all departments find the outdated system the ball and chain of their everyday work, it's still not easy to convince them to transform it. They have to feel a strong need for change. Kotter calls it the sense of urgency achieved "through a bold, aspirational opportunity statement that communicates the importance of acting immediately". Shaping this demand can also help to avoid moving the software modernization tasks down the list of priorities.

5

Emphasize the gains and establish short-term wins

As software modernization tends to be a long-term and intricate process, its positive outcomes might not be apparent and profitable for all stakeholders in a short period of time. Try to specify the gains and show mostly the ones that concern a particular group. Don't focus only on the benefits obtained after finishing the whole project but also on quick wins that can bring smaller yet noticeable improvements. This way you can show the short-term benefits to the board as well as motivate and engage the team in reference to the Kaizen Blitz approach.

6

Provide information coverage all the way

Knowing that people fear to act in an environment of uncertainty, the challenge is to keep them informed through the whole process of change. On the other hand, remember about attention span shortening and information overload. Try to be as concrete as you can by keeping your messages short and customized.

7

Invite to the discussion, listen, and engage

Effective communication always depends on effective listening. People can't engage when you only serve them monologues and introduce visions. They need to feel they have an impact on the process and that their opinion is taken into account. Create space for a broader discussion and be available for people who want to share their remarks.

8

Find change agents

Securing a broad endorsement of software modernization is just halfway to success. Some people need to be more convinced than others to drive change, not only accept it. These are change agents, the leaders who can lift a bit of a burden from your shoulders but need to be empowered and included in the decision-making procedures.

9

Prepare and present a plan

If you lead the change, people have to trust that you can control it and have a plan. Considering the possibly protracted and often unpredictable nature of software modernization, you can't make a highly detailed timetable for the whole process. However, a solid strategy with a consistent roadmap is a must.

10

Anchor the change in the organization

Software modernization doesn't end with implementing the last changes in the system. To make the change last, proper modifications have to reach other areas of the organization as well as its culture. Also, the support towards the change needs to be continuously extended and deepened by employees and leaders.

Who do you need to convince and how to do it?

How to put all that knowledge into practice in specific terms of software modernization management? You need to know the stakeholder groups inside out and adjust your actions to their specific needs. Although there might be a dozen of them, it is safe to distinguish three main groups of the modernization process-affected parties: the board, users, and the tech team.

THE BOARD

THE GOAL

To secure a budget and ensure a place for software modernization on top of the organization's priorities.

THE MAIN CHALLENGE

In 85% of companies that undertake modernization, it takes from six months to a year to kick off the work from the moment of the initial budget proposal. Why so long? The first reason is, of course, the high cost and the long duration of the project. But the key thing about this argument is the low awareness of how much organizations really pay for maintaining the legacy system and, in a broader sense, the lack of understanding of the technology's impact on business.

Despite their strong and still growing position (Gartner predicts that by 2030 a CIO will become the second most important business strategist and operational leader, after the CEO), CIOs still struggle to convince stakeholders that IT should be the core driver of business value. That might be especially hard in terms of the software modernization process, which is more about transformation than actually producing new elements. The clue to gain the board's buy-in is to show **the value of improvement in a business context**.

FREQUENT CONCERNS & HOW TO ADDRESS THEM

"We can't stop the software's development to modernize it"

That's one of the most common concerns we hear from our clients, which actually inspired us to create the Modernize and grow whitepaper. An organization can't just suspend its activities or the app's further development for months only to upgrade the software. **The thing is, it doesn't have to**. To be more specific, the outcomes of modernization should accelerate the company's growth and the development of its system. To convince the board these are possible to achieve, you will need to come up with concrete solutions ensuring the project will not disturb business performance, like:

- **an extensive preparation process,**
- **well-conducted communication with stakeholders,**
- **solid strategy with a thoughtful roadmap,**
- **accurate prioritization of goals and task,**
- **gradual transition to the new technologies,**
- **and many others we present across the chapters of *Modernize and grow*** (don't hesitate to use them all!)**.**

To refute the argument mentioned above, try to synchronize the modernization roadmap with a software development plan. This way, you can, among others, show how the upgrade will speed up the deployment of already scheduled features even before the entire modernization process is completed.

"We've got more urgent business needs than code improvement"

This argument is flawed in two ways. First of all, this reasoning underestimates the size of tech debt and the scale of its negative impact on the business. Year after year, it will only grow, and so will the modernization's scope. One day, the debt can reach a breaking point and bring about critical errors that will completely paralyze the business. Then, the costs of software fire-fighting, rebuilding users' trust, and even paying penalties if sensitive data had been disclosed can overrun an estimated budget for software modernization. So, remember to start your presentation with **a comparison of the costs of performing a system upgrade and postponing it** (and then finish with a catchy quote attributed to Karen Lamb: "A year from now, you will wish you had started today").

The second misunderstanding concerns what software modernization is truly about. With too much focus on code refactoring, **the software upgrade is perceived as an IT-internal project or at least a tech-oriented case**. To clear it up, show the whole picture of the process, all the areas the outdated system touches, and the business benefits of modernization.

"The outcomes are too distant and not viable"

Evaluating software modernization ROI might be like reading the tea leaves: the timescale is long, the risk of complications and running over the initial budget or scope is high, and a big part of outcomes translate indirectly to the business benefits. It is no wonder that 1 in 4 companies can't estimate even an approximate payback period of this technology investment (and almost a half predict it will reach over four years).

Nonetheless, a lack of the project's financial analysis and clearly set profits can make you fall at the first hurdle of gaining the board's buy-in. To build the modernization's value upon economic advantages, you need to use proper indicators. Unfortunately, typically used IT metrics tend to focus on efficiencies, while they should speak to revenue and customer experience. If you want to harness them to either legitimize the necessity of modernization or show its predicted ROI, translate them into business value.

SOFTWARE MODERNIZATION TECH OUTCOMES TRANSLATED INTO BUSINESS BENEFITS

Performance improvement

What's the real benefit of making the app load 30% faster? Combining this figure with the information that a 1-second delay in page response time can result in a 7% reduction in conversions might switch the focus to improving user satisfaction. Place the performance metric in this context together with the UX improvements software modernization includes, and it will resonate better. Then, you can refer to metrics like customer churn or customer retention to illustrate more business-related profits.

Adapting the system to integrate with API's and new technologies

In this case, the goal is to spotlight the value of enhancing the competitiveness gained with software upgrades. Staying ahead of competitors is one of the key business drivers of software modernization. That's because the board members' reference points are usually benchmarks they compare the company to. Profound market research should deliver accurate arguments to show how innovation will help catch up with the competition or even outperform them.

Faster deployment

Achieved mainly through introducing or refining the DevOps processes and moving to the cloud, a more time-efficient deployment notably shortens time-to-market of the new features and rolling out software updates. Need examples?

- *Etsy switched from four-hour full-site deployments twice weekly to more than 50 deployments a day;*
- *Adobe has been able to meet 60% more app development demand from the business;*
- *Fidelity Worldwide Investment achieved $2.3 million worth of cost avoidance per year due to automating the software release.*

By the way, that was one of SafeEx's motivations to work with us on the upgrade of their system.

Higher code quality and tech transition

The most beautiful and clean code will be considered "art for art's sake" if you don't present its influence on streamlining the development process. Moreover, it would help if you highlighted that restructuring the codebase, transferring it to more popular technologies, and updating frameworks and libraries make the code work much more pleasant. That might boost the tech team's members' job satisfaction and enlarge the developers' talent pool, thus reducing recruitment costs. Utilize Stack Overflow annual surveys to compare the software engineers' population depending on technology and the solutions they dread to use.

Improved security

It seems as though this benefit shouldn't require an explanation in terms of its business value, yet you might need to specify how the modernization will impact security. Refer to particular GDPR or industry regulations like HIPAA and point out how assumed improvements will make the software more compliant with them. To fire up the board's imagination, you can also recall the impressive amount of $3.86 million, which is the average total cost of a data breach, according to the IBM report.

THE USERS

THE GOAL

To change the users' behavior and shape new habits.

THE MAIN CHALLENGE

If people, in general, don't like changes, digital products' users just hate them – and it doesn't matter if we're talking about users of the organization's internal software or the app the company produces. They all tend to succumb to the baby duck syndrome, which in IT means the tendency to adhere to the first experience when using digital products, e.g., browsing the site or using mobile applications. Once the users learn how the program works, they recognize it as the best one and avoid the necessity of relearning. Believe it or not, it's not only about their mental laziness but also about something more profound: breaking their habits and forming new ones.

FREQUENT CONCERNS & HOW TO ADDRESS THEM

"I like the current system"

Hard to argue with that, isn't it? Well, you need to know that behind this simple statement stands a range of "buts" that need to be uncovered:

- **it takes ages to create a new account,**
- **it looks like the early 2000s,**
- **I need to do half of the operations in a spreadsheet anyway,**
- **new employees need three months to tackle it using basic features** (actually shortening the learning curve and simplifying the onboarding process became the drivers for modernizing the software for one of our clients).

So now, the trick is to **pick a couple of pains and show how the upgraded system will relieve them**. You can use the UX audit and interviews to discover these objections, although they might have already been expressed less directly. Talking with users and observing their behavior is also a gold mine of features worth implementing in a new system. This method will help the affected parties adapt to the new software environment more pleasantly.

Don't be surprised, however, when the users' first reaction to the idea of changing the software will be negative, despite pointing to the impregnable improvements. Why is that? See the following concerns.

"I will break it"

Let's go back to the prospect theory. For most users, especially the in-house ones, the reference point is the current state and already-shaped ways of doing their job. So, the gain defined as completing the task 30% faster should be a strong motivation to back up the change, shouldn't it? Not necessarily.

The fear of losing the sense of safety provided by a flawed yet well-known software or a simple doubt concerning the ability to learn new one might be felt more intensively than a rational, definite profit. To achieve buy-in from the users, you need to assure both tangible, apt advantages and an easy, attractive onboarding to the changed elements of the system.

"It's not my business"

The misconception that technology runs its own course separated from other fields of the business is widespread through most of the non-tech departments, not only the board. Still, the knowledge about essential parts of the system that require modernization is usually distributed all over the organization.

To gather it, **you need to ensure the users the time for consultation/interview and the confidence that their voice will matter**. Remember to highlight their contribution to the success of the software modernization project and inform them about changes in areas they primarily care about.

IN-HOUSE TECH TEAM

THE GOAL

To make the team support modernization.

THE MAIN CHALLENGE

That's probably the most challenging group to convince to modernize the system and keep their high level of engagement through the process. Developers and other IT employees have to **proactively participate in the project and produce software changes**.

Your first impression might be misleading if you'd assumed this group should be the most interested in system upgrading and thus ready and steady to support it. Even if it seems logical as working every day with legacy code is usually a nightmare, once again – the fear of loss might gain the upper hand. What are developers afraid to

lose? Of course, their jobs. It sounds dramatic, but unfortunately, it's not impossible to happen within software modernization projects. The bigger the technological switch is, the more difficult it might be for the in-house tech team to adapt.

FREQUENT CONCERNS & HOW TO ADDRESS THEM

"I don't have enough skills to modernize the system and use it"

That's the biggest concern of the tech team – **that they might not know the solutions the company wants to implement, not to mention they may not have experience in conducting technological transitions**. Even for highly qualified developers, modernization is usually a great challenge, especially when it entails modifying the architecture (and it usually does even if that wasn't the plan – see the LeoTrippi case study). That brings the risk of changes made in one unit causing damages in another. It often happens when we **don't have a complete picture of all dependencies in the system**. Hardly ever is this knowledge full, even in the case of veteran employees. Over the years, a dozen people might have been working with the code, and if they hadn't created proper documentation, refactoring could feel like field demining.

Internal IT team members will have to face not only these demanding tasks but also the possibility of uncovering their own mistakes and shortcomings. That triggers the fear of undermining their authority and dealing with the overdue consequences of their neglect. Remember that after a while, you can't objectively and fairly judge the situations in which the errors had been made, so it's better to **start the modernization project with a clean slate**. This way, you create a safe environment and a climate of trust. The other thing is to ensure the right tools are in place together with workshops for mastering new solutions. Consider providing soft skills training for tech team leaders struggling with satisfaction swings and other problems related to acting in changing conditions.

"I don't have time for it"

When the organization modernizes and develops software at once, time shortage within the internal tech team is bound to happen. That's why **outsourcing at least a part of the process** is a common way of coping with the problem. It has indisputable advantages for the whole organization, like filling the talent gap quickly, adjusting team composition flexibly to the changing needs, or avoiding the hidden costs of hiring an in-house software engineer.

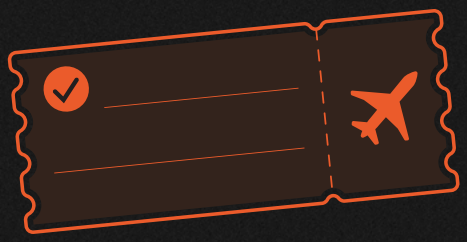
For the tech team, external help brings relief and support in dealing with problems related to the system upgrade. **The challenge is to build the space for knowledge sharing and cooperation between external and in-house teams**. It requires advanced project management and leadership skills from both sides, so while choosing the IT partner for your software modernization project, pay special attention to the PMs' experience and competencies.

“Modernization stops my self-development”

Modernizing legacy code is neither the favorite nor the most attractive part of a software engineer’s job. Developing the system and creating new features is far more engaging, as it delivers tangible, visible outcomes and proof of successful work. It also provides more diversified duties with a clear goal and specified time horizon. Software modernization is seen as quite the opposite with the arduous code fixing, distant effects, and all the risks mentioned above. It creates favorable conditions for the appearance of the **effort-reward imbalance**, which assumes that employees feel their “high efforts spent at work translate to low rewards received, which in turn elicits strong negative emotions and stress reactions”. Additionally, the awareness of the long duration of the software modernization process might be a nail in the coffin of their motivation, encouraging them to look for a new place to work.

The key to overcoming this dilemma is presenting a short-term yet possibly detailed plan, including the quick wins and celebrating small successes. Even if the work is not exciting, try to show the knowledge and experience the tech team can gain and the actual effects it provides for the whole organization.

Anxious to go on the modernization journey alone?



Don't be.

We can help you prepare a thorough plan and pick the most relevant tech solution to implement it!

Merixstudio drives digital innovation to businesses of tomorrow through end-to-end software solutions. Over the last twenty years, we've helped more than 250 clients bring their vision into reality. If you're looking for a trusted modernization partner, why not get in touch with us?

contact@merixstudio.com

+48 570 001 928

ul. Małachowskiego 10, 61-129 Poznań

See our work Behance dribbble merixstudio

Check what our clients think about us Clutch

Follow our stories

Access more content on modernizing software