

# Change Management Toolkit

Last Modified on 07/24/2025 5:31 pm EDT

Change management is hard. We've developed a range of tools to help with the people-side of implementing DevResults.

- **Principles and best practices**
- **Responsibilities matrix**
- **Permissions matrix**
- **Roadmap for data management**
- **Stakeholder buy-in**

## Principles and best practices

---

### 1. **Communicate, communicate, communicate:**

- Tell people over and over what's happening, why it matters, how it benefits them, and what's expected of them in return.
- "Sell it" and *keep selling* it internally, especially when organizational or leadership attention is a scarce resource.
- Don't assume *everyone* has read or listened to *everything* you've written or said previously. Most big email blasts are not fully digested.
- Find different ways, audiences, and venues to say the same thing. Don't over-repeat yourself, but make sure everyone has heard the same message in at least 2-3 different spaces.
- Don't assume *anyone* has drawn the logical conclusions or personalized the implications of your public statements (e.g. "M&E teams should do X..." does not always translate to "I need to do X..." in users' minds).

### 2. **Think politically**

- This follows from the above, but be strategic in how you communicate and who you communicate with.
- Don't explain too much too early. Give the long-term vision, but fill in the details piecemeal. This goes double for training (more on that in #3).
- Don't assign someone to be the face of or lead for a particular initiative just because their job title suggests that it should be them (if you can).
- Do find a way to involve the well-liked, gregarious, tech-savvy person (even if junior) to a significant role with as much authority as possible/warranted.
- Identify the other initiatives that will soak up everyone's attention and take an active approach to mitigating those risks (e.g. coordinate with those implementation leads, take them out to working lunches, trade favors, do each others' communication/internal-selling work).

### 3. **Don't skimp on time/resources for change management**

- Companies in well-endowed sectors invest *heavily* in change management.
- This *can* mean training and documentation, but it can also mean coaching, troubleshooting, consulting, listening and other forms of social/emotional labor.
- Training should be delivered when the platform and people are all ready to start using the system in the real world. This not only creates a sense of urgency, but aids in learning retention.

### 4. **Don't do anything for someone they can do themselves**

- It's tempting to bear the burden of menial tasks early on. You might tell yourself that you're "paving the way towards widespread adoption." But beware: some people ask for assistance as a ploy to avoid work or put off change. If you sense that you're *offloading* rather than *instructing* someone, reverse course immediately.
- Challenge people not to dismiss themselves or denigrate their considerable talent/intellect by saying "I'm not a techy person." This has become an acceptable dodge in our professional culture, and it should be totally unacceptable to say about one's own self! Don't let people give themselves an easy way out.

### 5. **Focus on quick wins (aka "the snowball strategy")**

- Sometimes you aren't at liberty to decide what comes first, second, third, but if you do have a choice, choose to focus on the workflow/tool/process that causes the most pain AND/OR is the easiest to adapt.
- This is similar to "the snowball strategy" to paying off debt: Should you start by paying off your highest interest loan first? Absolutely. Is it more psychologically rewarding to pay off your smallest loan first,

regardless of interest rate? Yes. What should you do? In the cases where behavior change is required, the thing that produces endorphins most likely.

- You could play around with games (**gamification**), parties (**data parties**), and other such elements of "fun" to incentivize people, but this is highly dependent on the culture of your organization, how competitive (and friendly) they are, etc. Use at your own discretion.

## Responsibilities Matrix

---

Getting the most out of DevResults requires keeping information current and complete. We recommend establishing clear internal guidelines for who should be responsible for updating and managing various parts of DevResults.

**Download the Responsibilities Matrix** to think through:

- Real-world stages in your project cycle
- The corresponding updates in DevResults to keep the site up-to-date
- Who should execute them.

You can customize these stages and actions as needed and identify the individuals who will be responsible.

## Permissions Matrix

---

To help decide which groups of users should have which permissions, we've assembled an Excel file that shows the default permissions for each standard group. You can use this file to think through how to modify permissions for user groups, develop custom user groups, and assign users to the appropriate group.

- **Default Groups tab:** shows our default groups and permissions settings. (Locked from editing.)
- **Custom Groups tab:** use this worksheet to think through modifying groups or developing your own.
- **Users tab:** allows you to list your users and to assign them to permissions groups.

**Download the Permissions Matrix** or see **Permissions Overview** for more information.

## Roadmap for Data Management

---

The most challenging element of our work is helping organizations implement plans for using our software. Rolling out data systems across an organization requires aggressive change management processes and a big investment in making them work. We collaborated with MERL/Tech workshop participants to create a start-to-finish roadmap for choosing and implementing a data management platform.

**Download the Roadmap for Data Management.**

This roadmap covers a range of steps, questions, potential pitfalls, and suggestions for an organization seeking to refine results management, specifically around choosing and implementing tools for doing so. To use this roadmap, we recommend you:

1. Customize the list for your organization
2. Classify the priority level of different steps
3. Seek buy-in at your organization
4. Assign responsibilities
5. And most importantly, grant authority to make decisions and act.

## Stakeholder Buy-in

---

This guide covers essential components of cultivating stakeholder buy-in, including a worksheet for crafting your messages to different stakeholders and a checklist summary.

- [Download the Stakeholder Expectations Worksheet](#).
- [Download the Stakeholder Buy-In Checklist](#).

## Cultivating stakeholder buy-in

In order for your organization to successfully adopt DevResults, the people involved will need to accept and embrace the changes that come with new software.

The first step is to identify your stakeholders. A majority of these will be the people using the software. **Make a list of everyone who will use DevResults at your organization and at any partner organizations.** (The [Permissions Matrix](#) includes a tab for the user list.) Also include other colleagues who may be affected by the software even though they are not directly involved with it. For example, maybe the communications department should know about the available metrics and visualizations.

## Setting expectations

Anxiety and resistance can stem from uncertainty. If your stakeholders do not know why you are changing their workflows or what they should be doing differently, then they will be more likely to cling to established habits. Setting expectations and enumerating the benefits of an upcoming change can dispel uncertainty and smooth the path to adopting DevResults.

There are four key points to communicate to your stakeholders. These four points will vary based on a stakeholder's role with DevResults, so tailor messages to each group by answering these questions:

**Context:** Why the change is occurring

- Why did your organization choose DevResults?
- What problems will the software solve?
- What problems will remain?
- Why is this change happening now?

**Impact:** What effect the change will have on them

- What specific pain points do these stakeholders currently have?
- How will DevResults improve these specific pain points?
- Will DevResults create any additional pain points?
- How will DevResults change the day-to-day responsibilities of these stakeholders?

**Vision:** What outcome they are working toward

- What will workflows look like when the implementation is done?
- What benefits will become routine for these stakeholders?

**Resources:** Where they can go for help

- Who should these stakeholders ask for help?
- What materials can these stakeholders reference?

Next, think about how to deliver your messages. What channels of communication are available to you? Keep in mind that real-time communication (in-person or even over the phone) tends to be more effective, especially if it is on a small scale; you can put off reading an email indefinitely, but it is virtually impossible to avoid thinking about the content of a one-on-one conversation or small meetings. Sometimes joining existing, standing meetings — even for a different team that you don't usually interact with — can be a good strategy to "meet people where they are."

Cultivate enthusiasm by emphasizing cleaner workflows, streamlined data entry, standardized systems, and the possibilities of data-driven management decisions. Users should see the value-add for their own role, not just the

organization's strategic objectives. That said, don't forget to communicate "up" as well to make sure senior leaders are bought-into and fully aware of the implications, benefits, and risks of a change process.

Finally, it is better to overdo it than to underdo it. Sometimes people need to hear the same thing several times or in several different ways before understanding it. Never assume that just because you sent an email or held a meeting that everyone in the thread or on the call actively listened to, understood, personalized, or committed to do what you said.

**Download the Stakeholder Expectations Worksheet** to craft the message you will use to set expectations for each stakeholder group. Feel free to rename, add, or subtract the groups as you see fit.

## Addressing concerns

Unaddressed concerns can reduce buy-in and undermine the change implementation, so stakeholders should feel comfortable voicing their concerns. Some organizational dynamics can get in the way of such candid communication. For example, a stakeholder might refrain from expressing doubts at a meeting because a supervisor is present. Give stakeholders a way to express concerns directly to you, and make it clear that you welcome their input. Conversely, some people will be more comfortable speaking with their supervisors than with you, so make sure that the leaders at your organization know that you want their input.

In the absence of organization-wide systems, individuals develop their own way of doing things, which often work very well on a small scale. If a stakeholder feels that they'd be losing something by giving up their own tools, see if there's anything you can offer in exchange for a sacrifice on their part that will further organizational objectives.

On the other hand, be sensitive to where a concern or request for support is coming from. Often, people who are stressed, overtaxed, or otherwise resistant to change will raise concerns and questions simply to stymie that change, consciously or unconsciously. If you suspect that someone is offloading their work or behavior change onto you, find a way to push back gently. "I'm not a techie person" is not a valid or acceptable excuse, and someone who is obviously capable enough to work on your team should not insult their own intelligence, nor be given a way out of facing a new challenge.

## Tracking progress

It can take several reporting periods to showcase the full benefits of DevResults. In the meantime, how will you show your stakeholders that the software implementation is working?

**Brainstorm a list of indicators for your DevResults implementation.** These indicators should measure progress toward your organization's adoption of the software, and should be influenced by stakeholder contributions. For example:

- # reporting periods with data
- # tasks completed in the system
- # projects reporting data
- # users trained

Once you have come up with your ideas, select the most relevant indicators for each stakeholder group. **Set your targets** and **decide how you will report your progress.** Will you make announcements at meetings? Send out email updates? Keep a scoreboard in the office? Anything goes as long as your stakeholders can track their progress – although visuals tend to be the most powerful motivators.

As you hit milestones on your indicators, make sure to advertise and celebrate your success. **Think about ways to incentivize progress** on your indicators. Here are some examples.

- Create a "user of the month" award for stakeholders who are making significant individual contributions.
- Showcase the progress that individual projects are making at staff meetings.
- Include DevResults use on performance reports.
- Host a launch party for new projects joining DevResults.
- Turn your projects into teams and develop competitions around using the software. Create a checklist – the first team to complete it gets a prize.

There is no “right way” to track and incentivize progress. The goal is to create a series of short-term wins that will create the momentum you need to carry you to the end of the implementation. Find these quick wins that alleviate the most pain, and only once you can celebrate that victory, move on to slightly higher hanging fruit.

### Hang in there!

Change management is hard, and it often requires you to multitask as a champion, salesperson, troubleshooter, project manager, and ally. While the complexity of the task and the potential for using up valuable time and resources can be discouraging, ***endeavor to persevere***. Companies in other sectors spend much larger percentages of their resources on change management, even for minor software upgrades; how much more should we, as behavior change professionals, dedicate ourselves to implementing thorough and sustainable change?

As you develop your implementations plans, feel free to reach out to us about any change management question.

Didn't answer your question? Please email us at [help@devresults.com](mailto:help@devresults.com).

---

### Related Articles

---