

# How a Center of Excellence Can Help Teams Develop Excellent Software



The ways that enterprises handle software development have changed immensely over the past couple of years. Today, many organizations are upending traditional business cultures as they strive for greater collaboration through things like DevOps and other agile development methods.

But even as things change, some core principles must remain the same. Business stakeholder requirements need to be delivered within a reasonable timeframe and budget, with a good user experience and solid return on investment.

Software development estimation can check off all of these “must have” boxes. Estimation helps development teams and organizations ensure that their projects remain on track, even (or perhaps especially) in highly agile environments.

The problem is that not every organization does software estimation correctly—or at all. However, enterprises that are already undergoing transformation are primed to integrate software estimation practices that offer long-term benefits.

Most of these organizations would benefit from creating an [“Estimation Center of Excellence,”](#) or ECoE. That may sound aspirational (and it is), but the concept is based on a proven methodology focused on a combination of people, process and tools.

An ECoE is a separate entity within an enterprise. It’s comprised of employees whose primary functions are to ensure the development and implementation of sound software estimation processes. An ECoE can help

businesses avoid potential mistakes and software failures. It can also provide them with a way to deliver better solutions more efficiently and productively, saving time and money.

Like DevOps, establishment of an ECoE is really more of a shift in culture, rather than technology. Thus, it's important for managers to first figure out if such an entity will be accepted, and whom it will comprise. Some organizations may need to hire estimation experts, while others might be able to pull from existing internal resources. Others may need to overcome resistance to change, which is natural when anything new comes along that could upset the status quo. And everyone will need to determine their anticipated ROI, which will change based on each organization's unique goals.

Once those baselines have been established and it's been determined that software estimation and an ECoE is the correct way to go, enterprises can begin implementing their estimation solutions. The best way to do this is to follow a four-stage model that takes organizations from creating an initial ECoE blueprint to "flipping the lights on" and keeping things running.

### **Defining requirements**

The first stage involves defining the requirements for the ECoE. During this stage, managers identify their organizations' specific pain points and goals, and begin building a business case for addressing each of them. Management should assess the current state of software estimation within their organizations, and tailor the ECoE as applicable. They should also begin identifying the major stakeholders who will become part of the ECoE and advertise their role in a successful implementation.

At this point, it's also important for management to set forth ground rules for ECoE stakeholders to follow. Everyone should be on the same page in regards to the estimation process the organization has chosen to adopt. Disagreement or lack of buy-in poses the risk that projects will be derailed, thereby undermining the ECoE's purpose.

### **Design and development**

The second phase may be the most intense part of the process. A lot should happen here: Staff should be hired and trained, business processes defined, and service level agreements reached.

There's a lot of additional detail that should be attended to, particularly related to establishing business ROI. First, historical data repositories should be established to support estimation and business-case ROI analyses. Then, several pilot estimates should be performed to help the staff gain experience and fine-tune the estimation methods so they may be able to deliver the expected ROI.

This is also another chance for management to gain further support behind the ECoE effort. They should be prepared to communicate with key individuals throughout the organization to gather support and build the value of the ECoE. It's an opportunity to unify the organization as a whole behind the program. That's important as, like DevOps, software estimation requires buy-in from everyone to be truly successful.

### **Launch and pilot operation**

Finally, the moment everyone's been preparing for: the ECoE becomes operational. During this stage, staff should be mentored and trained as they gain experience. New users should be brought on board as necessary, complementing the individuals who are already in place. These teams should develop case studies of successes achieved showing the efficacy and value of the ECoE's operations so far.

The people working in an operational ECoE have now become part of something more than just their company: They are members of a community of stakeholders practicing successful software estimation techniques. Thus, during the third stage, some organizations may choose to implement sharing circles that allow these

stakeholders and their colleagues to share knowledge and practical experience with the rest of the estimation community. Ideally, this will help them and their peers enhance their software development efforts.

### **Full and ongoing operation**

This final stage never really ends. In fact, over time, ECoE teams should strive to improve the processes they employ based on lessons learned from completed projects and their overall experiences. Hiring and training continues as needed. Skillsets will change and evolve. In other words, much like in agile development, the ECoE will adapt over time to continue to adequately address the business's needs.

It's true that developing an ECoE is not a quick fix. It can take many months for a completely operational ECoE to go online, and even then, it becomes an iterative process that managers will continue to refine over time.

But that's OK, because an ECoE offers long-term benefits and will pay for itself by reducing risk, maximizing business stakeholder demand, and optimizing existing resource capacity. Organizations will gain greater productivity and better insight into the entire software development process and ensure better outcomes.

It's well worth the time and effort.

*This article originally appeared in [SD Times](#).*