

# Estimation Center of Excellence

Software projects are like skyscrapers. They both need solid foundational elements — for one, cement; for the other, code — and teams comprised of members with various skillsets. They require planning that allows for some amount of flexibility, along with access to modern tools that workers can use in construction. No contractor worth his or her salt would approach the creation of a complex structure like an office tower without these items already in-hand.

So why are there still so many companies out there that fail to do a comprehensive job of software planning and estimation? Some may argue — incorrectly — that estimation and planning have no place in a company that practices agile development. Others adhere to the fact that they've never done it before, so why start now?

Much like a contractor breaking ground on an office tower, software project managers should have a plan in place to ensure that their development processes remain on track, on budget, and on pace to deliver a fully functioning project. This requires a method to proactively manage software projects from inception to development.

A software [Estimation Center of Excellence](#) (ECoE) is a great starting point. An ECoE is a separate entity within an enterprise, and exists for the sole purpose of accurately planning and estimating software development projects. It's ideal for organizations that are developing and deploying new tools and applications on an ongoing basis.

An ECoE is not a physical structure, but an ideological construct comprised of people, processes, and tools, all working together to achieve more efficient software development. In an ECoE, teams of **people** work together to ensure that development remains on track, results and data are reported correctly, and information is shared. These teams work inside of an overall **process** that is defined by specific business cases and provides structure around the direction and purpose behind the organization's software estimation and development efforts. **Tools** are used to gather metrics and report data, because hard facts and information should be the girders that house a solid software development structure.

Let's take a look at how each of these components work.

## Process

[Establishing an ECoE](#) begins with creating a process that places a framework around software development projects. All projects should begin with a plan that showcases clearly defined goals, direction, purpose, and an outline to help organizations achieve each of these in turn. Thus, project managers should sketch a blueprint that addresses guiding mandates and incentives, pre-determined milestones and decision points, and success criteria. After all, how can a software project be deemed successful if success isn't predefined?

Before doing this, though, they'll want to do an honest assessment of their current situation. They should start by analyzing and evaluating their organizations' current estimation processes (or lack thereof). Consider what is working, what isn't, and where they want things to be, given the already-established criteria for success. Then, begin charting an appropriate change management strategy.

It's important to acknowledge the old axiom that change is hard. It's even harder in enterprises where managers have become used to doing things in certain ways. Establishment of an ECoE will very likely disrupt those

“certain ways” by introducing new and potentially unfamiliar kinds of processes. There’s likely to be pushback, but project managers can respond with data and defensible facts to support the establishment of the ECoE that indicate that whatever changes are introduced, they will be for the better.

## People

They’re at the heart of the process itself, so it’s very important for companies implementing an ECoE to hire the [right type of staff](#) to run the Center. Indeed, there are certain characteristics and skills that comprise the ideal ECoE staffer.

Technical skills are important. All ECoE team members should have basic software development and analytical knowledge, the latter to better crunch the data and numbers that help steer projects through to completion. It’s also helpful to have people with functional sizing skills, as these can help determine how long a project will take, how many people should be involved in its development, how much it will cost, and so on.

Pure management skills are also vital. It’s important for ECoE employees to be able to articulate information to each other and to executive management. Therefore, communication, presentation and mentoring skills are all valuable commodities. And since the ECoE is based on the concept of planning and execution, team members should plan for data collection, template building, training and deployment, especially in the case of large enterprise projects.

## Tools

Just as a major construction project cannot be completed with manually operated shovels and hammers, software estimation requires the use of automated tools to help keep development on track. The good news is that the ECoE toolbox can consist of solutions that most corporations likely already have at their disposal.

Project portfolio management (PPM) software, which many companies use for financial, risk, and resource management, can work with your software estimation tools and development process. The standard PPM process involves creating a proposal for a project, developing a business case for that project, and getting executive buy-in and approval. Unfortunately in many cases the cost and schedules in these proposals come from the business and represent what they would *like* to have happen. This is an ideal place to insert a feasibility estimate to validate the customer target cost and schedule with a more realistic picture of what is *likely* to happen. This puts a formal process in place that allows managers to clearly articulate and validate project plans and goals and have them officially sanctioned by corporate leaders.

Other enterprise management tools can also be used within the ECoE, including standard business intelligence and reporting, estimation software, time and cost accounting, and pricing solutions. All can help teams better estimate the overall time and costs associated with their development projects, and keep things on track.

Development tools can be used to ensure the project’s continued viability and performance throughout the production cycle. Configuration management and defect tracking software can help managers get a better feel for the overall size and scope of the project and monitor for defects and stability throughout the entire process, right up until deployment.

## The Building Never Ends

Creating an ECoE based on these three core pieces does not happen overnight. The time it takes to build a fully

functioning ECoE is dependent on a number of factors, including size of the organization and amount of employees being impacted. Some deployments can be accomplished in a month or so; others may take up to a year and a half.

Even after deployment, the ECoE is never really completed. It should be considered a living thing that teams continue to build upon based on lessons learned from completed projects and past experiences. Much like a building that needs to be remodeled based on the needs of its tenants, the ECoE should grow and adapt over time to continually meet the evolving needs of the business.

In that sense, the work surrounding an ECoE is never truly done, but that just means the Center will continue to deliver benefits. Development processes will become more efficient and projects will be delivered on time and as expected. As a result, it will not be just software that's being built, but trust in the company and its solutions.