The Complete Guide to Software Process Improvement

Contents

Introduction 1
Basic Principles of Process Improvement 2
  Things to Consider 2
  Set Goals 2
  Get Support 3
  Be Inclusive 3
  Provide Coaching 4
  Keep Going 4
Importance of Process Definition 6
  Things to Consider 6
  Policies 6
  3 Critical Components 7
  Process Interfaces 8
  Roles & Responsibilities 8
  Process Standards 8
  Documentation Options 9
Software Estimation Process Example 12
  Introduction 12
  Process Automation 12
  Process Documentation 13
  Process Overview 14
  Process Detail 14
Summary 16
About the Expert 16

Introduction

The original Software Engineering Institute (SEI) Capability Maturity Model (CMM) was developed in 1987 to provide a framework by which the Department of Defense could promote software process improvement to raise the maturity of contractors' software development practices. The goal was obvious - reduce the large amounts of schedule and budget overruns. "Maturity" relates to the degree of formality and optimization of processes, from ad hoc practices to formally defined steps, to managed result metrics, to active optimization of the processes.

I have worked to promote software process maturity since its origins. The company I worked for was not only motivated to follow CMM guidelines to win more government contracts, but they suffered large financial losses on fixed priced contracts and recognized that poor project planning, oversight, requirements management practices and a lack of development standards were largely to blame. Thankfully, senior executives did a very good job of communicating the benefits of software process improvement and provided the financial support required to increase our maturity levels.

Over the years, both as a private consultant and working for QSM, I have encountered companies that view their improvement efforts as short term. Many adopt the latest development methodology as the ticket to success, but soon find out that there is never a quick fix, and you only get out what you put in. There are other organizations that recognize that their maturity level is low and become so overwhelmed by what they perceive as a monumental task that they just give up.

In this guide, I want to promote two concepts required for successful software process improvement success:

1. **Process improvement must be cultivated.**

   Look at the definition of cultivate (via *dictionary.com*):

   - To promote or improve the growth of (a plant, crop, etc.)
   - To develop or improve by education or training
   - To devote oneself

   The best gardeners know the basic principles: that they need a plan, they will have to adapt to the change of seasons, and that often unseen yet diligent workers are the key to success. They also know that their beautiful garden will improve over time with constant care and attention.

2. **Process improvement is continuous.**

   Inefficient processes are a main reason that businesses and agencies do not realize the success they envision. Software organizations, particularly, are expected to deliver more value with ever-decreasing budgets and schedules. Continuous improvement is a necessary mindset for achieving business agility. The fundamental Agile practice to Inspect and Adapt each iteration and program increment supports continuous improvement. Cultivate mature processes and sustain the momentum in your organization with the ideas and software estimation example in this guide.

Basic Principles of Process Improvement

Things to Consider

Most everyone acknowledges change can be difficult. Designing and maintaining business processes are challenging because of the many stakeholders, practices, and tools being used. As the business changes, processes must change to keep up, usually with a focus on helping workers become more productive. Certainly, now that digital transformation efforts are being undertaken, understanding how to design, implement, and continuously improve processes is a must.

Software process improvement initiatives must be planned and managed like a project. We recommend creating a small group with representatives from each affected area. It may only be project management, product owners, and development teams. The groups involved will be driven by the size of your organization and the way it is structured. Larger organizations tend to have more formal policies and procedures and several groups that must coordinate. Smaller organizations, of course, have fewer resources and often do not need as much structure. Designate a process manager and assign supporting roles as appropriate.

Here are some basic principles to keep in mind:

- **Set Goals:** Determine what processes need to be changed and why.
- **Get Support:** Build the business case and secure moral and financial support.
- **Be Inclusive:** Make sure every area has a voice.
- **Provide Coaching:** Train and coach practitioners to develop skills.
- **Keep Going:** Change must be continuous and in sync with the business.

Set Goals

Setting attainable improvement goals and communicating them well is essential. To define your process improvement goals, you need to understand the current process issues and pain-points.

Perform an informal, yet thorough assessment to take an honest look at the current practices of each contributing group. Identify bottlenecks and inefficiencies, often caused by poor communication and information exchange.

Measurement and Analysis is a CMMI key practice. Gathering actual data for core metrics (size or value, effort, staffing, duration, defects) plus some project characteristics and demographics (application type, development method, complexity, etc.) across the organization is critical to effective process analysis. Creating a repository of completed software project data is a good place to start. Perform regression analysis to benchmark project performance and identify process improvement opportunities. We recommend QSM's SLIM-DataManager application with over 300 defined software project metrics, used with SLIM-Metrics statistical analysis tool to identify and prioritize improvement needs.

Once you have the information you need to identify and prioritize the process changes that will provide the most benefit, set long-range and short-term goals. The more you know about your organization's agility and speed of change, the more realistic your goals will be.

You have no doubt learned about SMART goals, but it helps to be reminded.

- **Specific:** Direct, detailed, and meaningful. *Ex: Reduce software defect density rather than improve software quality.*

- **Measurable:** Quantifiable to track progress or success. *Ex: Ten percent reduction in defect density (# defects / size).*
- **Attainable:** Realistic, supported by tools and resources to attain it. *Ex: Testing processes and tracking systems are mature enough to support the achievement of the goal.*
- **Relevant:** Aligns with functional area skills and company mission. *Ex: Organization seeks to reduce rework.*
- **Time-bound:** Has a deadline. *Ex: 12 months.*

Get Support

Management's enthusiasm for and belief in process improvement will be key to a successful implementation. Change doesn't always start at the top. In fact, it is the folks doing the jobs that need to be done who know what's not working and propose the changes they'd like to see.

But your efforts won't go very far without executive awareness, buy-in and financial support. Process group members must be authorized to devote time to the project. Funds will likely be needed to purchase software tools. All practitioners will require some level of training. When I helped design and implement a software metrics process, we created three training modules to fit various levels of involvement:

- **Leads** – 8 hours
- **Contributors** – 4 hours
- **Executives** – 45 minutes

Make the best business case you can and show how your plan supports organizational objectives and benefits teams. In the early stages of process improvement initiatives, it is challenging to compute ROI. Our clients typically implement pilot projects ranging from three to six months to show the benefit of adopting SLIM-Suite applications to estimate, track and forecast, and analyze software projects. This provides enough time to learn and implement the new process steps and show tangible results – increased estimate accuracy or greater visibility of in-progress project health – to secure executive support. Here are three quick tips to build your business case:

- Pick short projects/initiatives to see results quickly
- Measure key indicators before and after
- Present pilot outcomes, benefits, and lessons learned

Be Inclusive

Each process area, be it project management, estimation, development, or quality assurance, has persons designated as leaders or process owners. Somebody needs to communicate the vision for the improvement initiatives for their area and make decisions. They may have a formal title or simply recognized as the expert and one who gets things done.

Although process affects every team member, it is impractical for everyone to be involved. The key is to ensure those impacted are aware of the improvement goals and that management has authorized and financed the project. Then, periodically update people on status. What has been accomplished so far, and what are the next steps?

The most important thing is to get their input! Every team member brings unique skills, perspective, and experience. Hopefully, this is done when the process assessment is performed. The people closest to the process will gladly let you know what's working and what's not and can help define the goals.

Reality check – not everyone is going to be supportive or want to be involved. People consider certain processes and practices their domain and may become naysayers. Others are simply too busy and won't or can't spare the time.

We will address the importance of change management later in this chapter. My experience is that most people will come around when a few benefits are demonstrated.

Provide Coaching

Software process improvement efforts often fail because organizations try to accomplish too much too soon. Aside from the cultural and organizational obstacles to change, people need time to learn and assimilate new ideas and skills. *"Human memory and comprehension are limited, and it is easy to design processes that are beyond peoples' capacities,"* says Watts Humphrey (*Managing the Software Process, Humphrey, 1989*). This is true in any situation, but I think it is compounded in the software world, because time is always a scarce resource. The pressure is high to complete projects quickly, making it difficult to justify time and money spent on process improvement.

Conduct formal training. Too many workers are asked to learn on their own while continuing to perform their daily tasks. At QSM, we know that adopting an estimation methodology that computes time and effort based on project scope means thinking differently. It also requires learning new terminology and software tool skills. This is why we offer training and free coaching to all customers.

Understanding human learning and skills development challenges (Watts Humphrey) will make your coaching effective. There are four development stages:

- **Installation**: Initial installation of methods and training in their use.
- **Practice**: People learn to perform the methods as instructed.
- **Proficiency**: There is a learning curve as people gradually improve their efficiency.
- **Naturalness**: Finally, the methods are performed without intellectual effort.

To learn more, read our best practices for software process improvement pacing.

Keep Going

There is an old cliché that says, nothing is certain but change. All of us experience this in work and life. Embrace change as the path to excellence.

Tony Robbins' constant and never-ending improvement (CANI) technique is all about committing yourself to a new way of being. The Scaled Agile Framework (SAFe) version 5 defines Continuous Learning Culture as a key Business Agility competency – *"a set of values and practices that encourage individuals—and the enterprise as a whole—to continually increase knowledge, competence, performance, and innovation. This is achieved by becoming a learning organization, committing to relentless improvement, and promoting a culture of innovation."*

Making consistent, small changes to your life is often more effective than trying to change everything at once. How much you change and how quickly is determined by your goals and action plan – set a reasonable pace.

The same is true with software development teams. As practices become well-established and part of business as normal, you will discover little changes that can be made to improve efficiency or provide more insight. When our QSM development and test team started using a bug tracking system, the structure and process for entering bugs and requirement requests was simplistic. As we learned more features like tags, statuses, and how to be creative with the “assigned to” field, information was better organized and easier to find and report. We also eliminated processes initially considered noble but required too much overhead, namely creating formal, detailed test scripts.

Simple ways to establish continuous improvement becomes part of the organization’s culture:

- **Tailor and adapt:** Allow practitioners to tailor processes to fit different project types, such as the frequency of software integration builds for single versus multiple team projects.
- **Test and tweak:** Don’t be afraid to try new methods or make small adjustments. Keep abreast of technology advances that can help.
- **Encompass new disciplines:** Few processes work in isolation. Look for the next related discipline you can adopt or improve, such as collecting data on completed software projects to support estimation.

Importance of Process Definition

Things to Consider

With your process improvement goals and plans in hand, the next step is to define and document the processes.

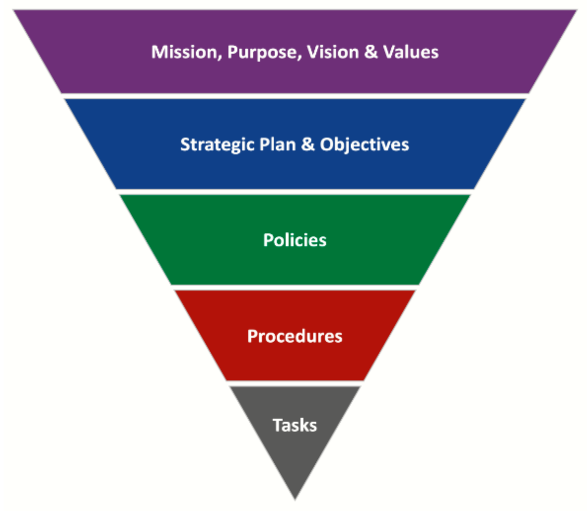
The formality and level of detail of each of these suggestions depends on the size of your organization, company culture, and budget. Start small and expand as appropriate. Here are some basic process components that should be defined:

- Policies
- 3 Critical Components
- Process Interfaces
- Roles and Responsibilities
- Process Standards
- Documentation Options

Policies

It is unfortunate that corporate policies get a bad rap. Howard Walwyn of Prism-Clarity stated, *"in the digital age what really is the point of writing out a few tired phrases purporting to be 'the way things should be done' to sit in a forgotten corner of the web taking up space and interesting no-one?"* He was merely stating common sentiment, because he believes policies are important, saying, *"from good policy we get a clear exposition of what our organization is all about. We get the expectations that our owners or shareholders or managers have about what we are doing and – just as important – why."*

The diagram below, from the University of North Texas, clarifies that the purpose of policies is to translate the organization's mission, vision, and strategic objectives, and provide guidance for developing procedures. Policies are a vehicle for communicating executive commitment and support.


Mission, Purpose, Vision & Values
Strategic Plan & Objectives
Policies
Procedures
Tasks

Organizations must have policies in place to comply with government standards, such as the Cybersecurity Maturity Model. Policies should state the purpose and scope of related procedures and identify exclusions. Keep them as brief as possible, fitting the size and culture of the organization. Policies should:

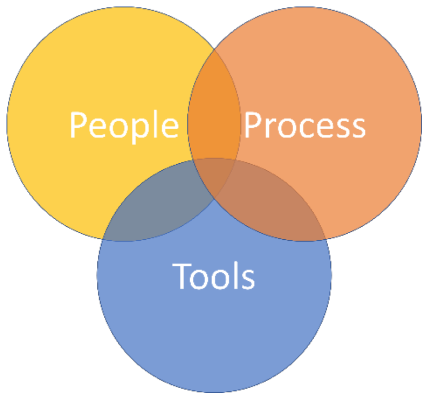
- Communicate the Organization's Vision
- Match Breadth and Formality to Organization Size and Culture
- Limit to Key Practices
- Establish Firm Commitment to Continuous Improvement

3 Critical Components

Poor estimation practices are a major reason why software projects fail. Estimation processes are often ad hoc, not well understood or documented, and performed by specialists. Successful organizations define processes and best practices for estimation as part of the overall planning process.

An important service QSM provides is to help establish an Estimation Center of Excellence – a small group that ensures best practices are being implemented consistently . We provide the roadmap for adopting SLIM methods and designing processes that will have to integrate with existing processes. The roadmap lists the required tasks and artifacts, organized by the three critical components required to make the new process implementation efficient and effective:

- **People:** Who is responsible? Who contributes information or reviews and approves the outputs? What skills and/or training will they need?
- **Process**: What has to be done? What are the tasks? Is there a particular sequence? When is it initiated? What defines "done done?"
- **Tools:** Where are process artifacts (documents, data) stored? What tools are used to perform tasks? Are there support guides and checklists?



Define the level of involvement and interaction between each component (see Documentation Options).

Process Interfaces

Groups within the organization must interact on a daily, weekly, and monthly basis as part of normal business functions. Process interfaces need to be defined to ensure pertinent data flows easily across group boundaries in a straightforward and timely manner.

For example, let's assume the Marketing group is responsible for product development and must define the initial requirements for the system. These initial system requirements will be the starting point and key inputs for both the Hardware Engineering and Software Engineering groups. The process for documenting the system requirements should show that Marketing is responsible for drafting the system requirements document, and that Hardware Engineering and Software Engineering team members participate in the review of that document.

We will address process automation in Chapter 3, but obviously, most project teams use software applications to perform tasks performed by various roles. These applications need to exchange data. Well-defined and documented processes that clearly identify inputs and outputs will support automated data exchange. Identify process interfaces by looking for:

- Related processes, paying attention to process assets and data requirements.
- Stakeholders from related processes and how they participate.
- Software application interfaces that can increase efficiency.

Roles & Responsibilities

A clear division of responsibility for processes across the business, project management and development teams throughout the project life cycle is essential. This is an effective way to ensure that nothing falls through the cracks. Although each procedure may have many participants, one role should be designated as responsible for its proper implementation.

It is very helpful to list all software procedures by role so that each person knows their project responsibilities in this area and the PM can make sure the project schedule will accommodate these activities -- especially from a resource loading perspective. Small project teams that have one or more persons performing multiple roles will need this information to guide tailoring. For example, I am assigned the role of tester in QSM's software testing process. I assist with test planning, but I am not responsible for it, and I don't manage testing procedures. Three key steps are recommended:

- Define the minimum set of roles for software projects by process area
- Define responsibilities for each role
- Perform skills needs assessment to plan training and coaching needs
    - Expert
    - Skilled
    - Familiar
    - Unskilled

Process Standards

Most organization functions, like accounting, project management, and quality control, follow standards to comply with regulations and promote consistent quality work. Government and industry standards are not always applicable or necessary so consider creating your own. Home-grown standards are often the most beneficial. It can be difficult for team members to agree on which standard to adopt. A familiar example is agreement on story point counts for user stories – Agile teams are often free to make their own determination in the spirit of promoting autonomous teams. However, inconsistent measures hinder the utility and application of development performance metrics.

A clear motivation for moving from a project to an organizational focus is to achieve consistency of development practices. It also makes the resulting software products more maintainable. Guidelines are more relaxed than standards and can provide the information needed with less formality. Leave room for teams to work the best way they see fit. A simple checklist can suffice for some process artifacts, such as small project requirements documents and project plans.

Recall the earlier suggestion to enlist your key experts to develop process assets. It pays off when you capture their experience of what works and what doesn't from a technical engineering perspective. Nuances understood about applying different tools, how some applications need to be tested and other things are perfect for guidelines, checklists, and formal standards**.** Standards help you:

- Reduce the need for training and tool support
- Promote consistency and efficiency
- Basis for process and quality measurements

Documentation Options

The options for documenting process are almost unlimited. Process flow diagrams are common and are helpful in communicating the big picture. The outputs of early processes are the inputs and triggers for later process.

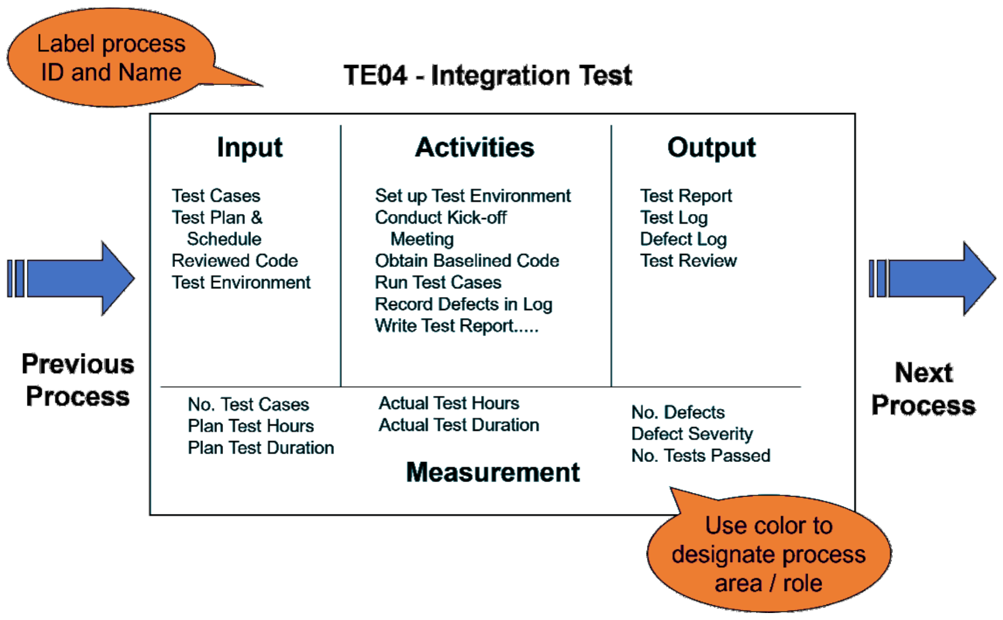
The detailed description of each process should clearly define the following elements:

1. **Inputs** – data and artifacts required to begin and the information source
2. **Activities** – specific steps to be performed in sequence with responsible role(s)
3. **Outputs** – data and artifacts created and where they are stored
4. **Metrics** – data to validate completion and support continuous process improvement

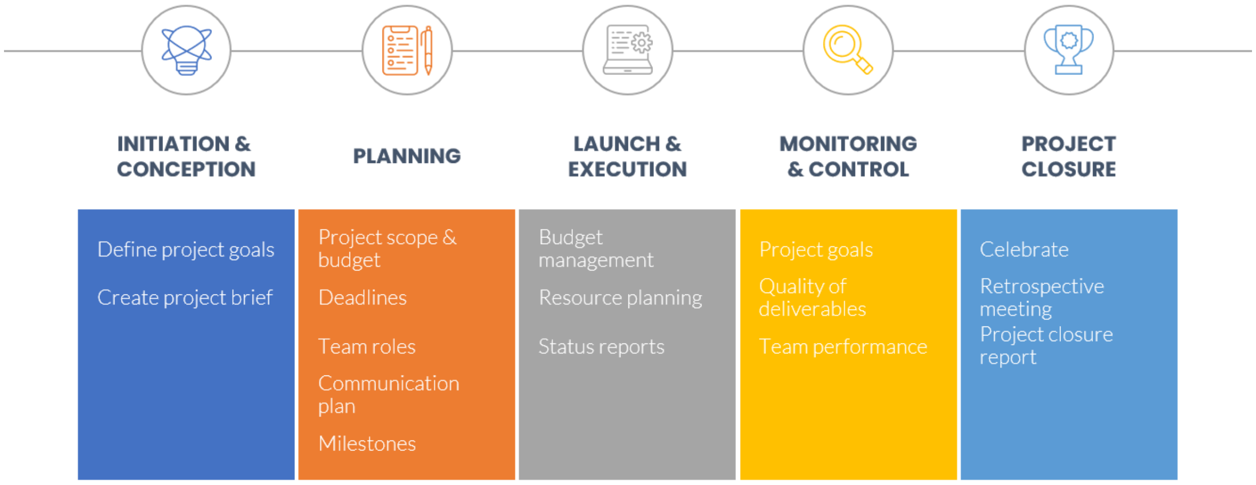
Supporting Information to Capture:

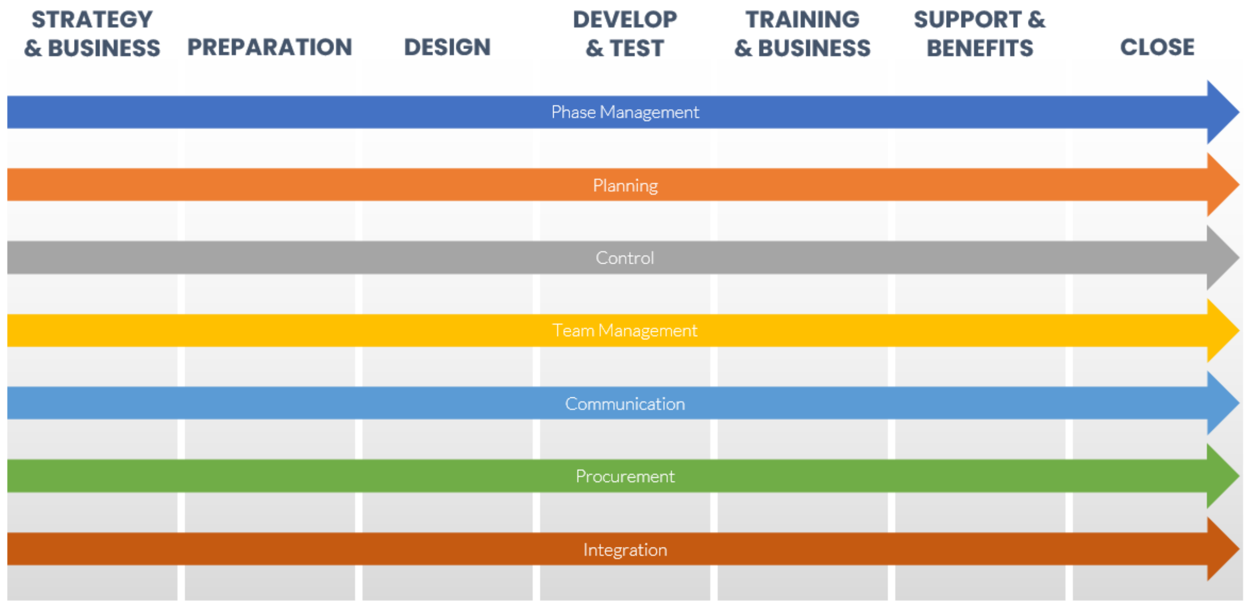
- Outputs that serve as Verifiable Objective Evidence for formal process assessments
- Estimated time required to complete

A simple table like the one below for an integration test process is my preference.



Process flow diagrams that list sequential and related business functions are great providing the "big picture."


INITIATION & CONCEPTION
PLANNING
LAUNCH & EXECUTION
MONITORING & CONTROL
PROJECT CLOSURE
Define project goals
Create project brief
Project scope & budget
Deadlines
Team roles
Communication plan
Milestones
Budget management
Resource planning
Status reports
Project goals
Quality of deliverables
Team performance
Celebrate
Retrospective meeting
Project closure report


STRATEGY & BUSINESS
PREPARATION
DESIGN
DEVELOP & TEST
TRAINING & BUSINESS
SUPPORT & BENEFITS
CLOSE
Phase Management
Planning
Control
Team Management
Communication
Procurement
Integration

Software Estimation Process Example

Introduction

Many ideas and "things to consider" for cultivating software process maturity are described in the previous chapters. This chapter uses the example of defining a top-down, scope-based estimation process, only one component in a larger software estimation and lifecycle management process. High-level estimation methods are quicker and easier than bottom-up methods because they can be performed very early in the planning process when detailed information is not available.

The process overview and detail formats presented are manual – PDF documents easily made available online. The introduction to the larger process document (not shown) serves as the policy statement, linking the process to corporate strategies and goals, and describes the benefits that will be realized.

A Rough Order of Magnitude (ROM) Estimate Quick Ref Guide was created to show specific software tool screens and related data used in each step. An estimation process management checklist was also created to get feedback from practitioners and assess the thoroughness of activities performed.

The following examples are presented:

- Process Automation
- Process Definition
- Process Overview
- Process Detail

Process Automation

Process automation is not a luxury these days – it is a necessity.

It is important to define processes manually and tie them to supporting software applications and data sources. It is an iterative process. No tool is going to implement a process exactly. It may only fulfill part of your needs. There may be methods and features the tool offers you have not considered that will influence the process design.

In Chapter 2 we mentioned the need to understand process interfaces. The outputs of one or more processes provide the inputs to others. Most interfaces and associated information and data will be digital. Identify whether a tool integration is required, or if a simple data export will do.

To automate a software estimation process using SLIM applications, you need to decide the types of estimates needed, whether you plan to gather completed project data, and whether desktop or web-based tools are best.

There are three SLIM-Suite desktop tools that support the estimation process:

- SLIM-Estimate – Estimates a single project / release
- SLIM-MasterPlan – Estimates multiple projects or tasks / program or portfolio
- SLIM-DataManager – Historical project repository / basis of estimation / validation

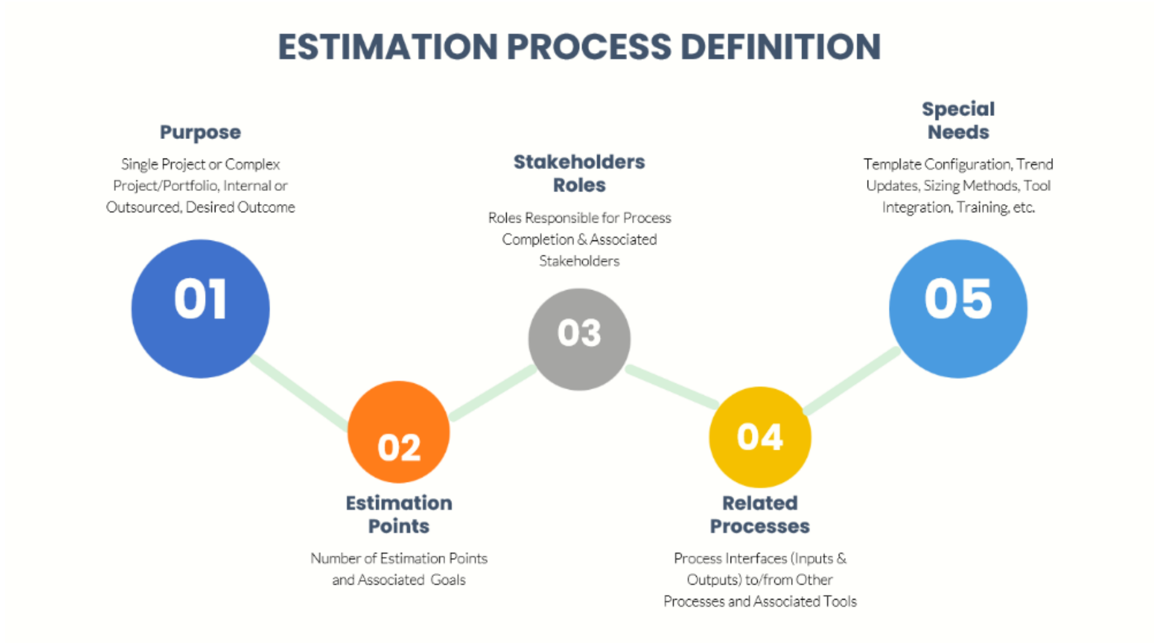
SLIM-Collaborate, QSM's online application, performs the functions of SLIM-Estimate and SLIM-DataManager and provides a portfolio view different than SLIM-MasterPlan. It facilitates collaboration among multiple stakeholders with built-in process workflow.

Process Documentation

In Chapter 2, we described several things to consider when designing your processes. Below is an example of a simple diagram you can create to guide designers and decision makers responsible for creating and modifying processes in their areas.

For our SLIM top-down estimation example, we have labeled five process components that require decisions or specifics.

- **Purpose:** What is the purpose of the estimate? Is it a single release and team, or a complex program that includes multiple platforms and includes non-software costs? The answers will determine which SLIM-Suite tool(s) is best for each situation.
- **Estimation Points**: When and at what frequency do estimates need to be calculated? Many organizations start with a ROM estimate, because so little is known early on, then update estimates as scope is decomposed and better understood.
- **Stakeholder Roles**: Which role is responsible for the process or the process owner, and what other roles participate, either to supply data, contribute to the work, or approve results?
- **Related Processes**: Estimating software projects will likely be a process related to larger program and project management and/or development processes. What are the interfaces and what systems supply inputs or receive outputs from the estimation process?
- **Special Needs:** All processes should be tailorable to accommodate a project's special needs. SLIM-Estimate or SLIM-Collaborate template settings or sizing methods may need to be modified. What special stakeholder, technical, or managerial needs must be accommodated?


ESTIMATION PROCESS DEFINITION
Purpose
Single Project or Complex Project/Portfolio, Internal or Outsourced, Desired Outcome
01
Estimation Points
Number of Estimation Points and Associated Goals
02
Stakeholders Roles
Roles Responsible for Process Completion & Associated Stakeholders
03
Related Processes
Process Interfaces (Inputs & Outputs) to/from Other Processes and Associated Tools
04
Special Needs
Template Configuration, Trend Updates, Sizing Methods, Tool Integration, Training, etc.
05

Process Overview

The outcome of the Process Definition step, either creating new a new process or modifying existing ones, may result in multiple processes that are similar, but contain different inputs, activities, outputs, roles, and supporting software.

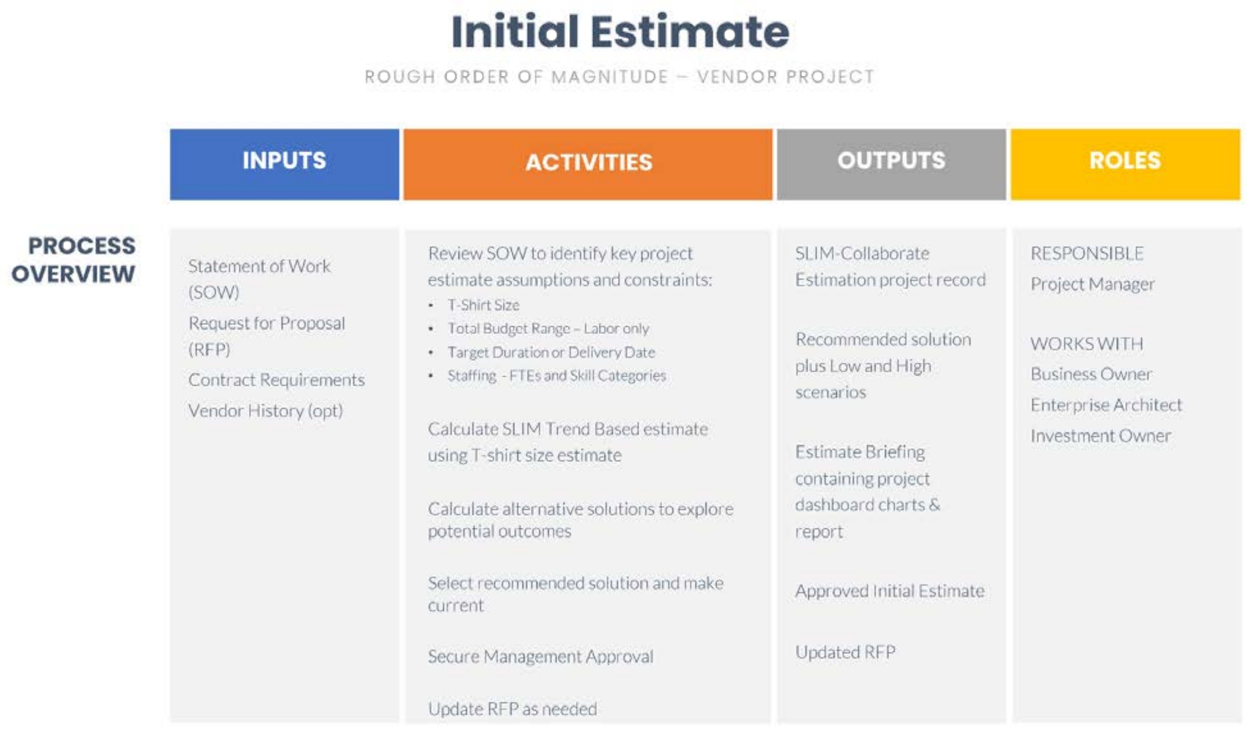
This **Initial Estimate** process example is for a large enterprise that works with vendors to accomplish some of their strategic work. Note the subtitle: **Rough Order of Magnitude – Vendor Project**. SLIM-Collaborate is the estimation tool, but this process can easily be modified to include SLIM-Estimate and/or SLIM-MasterPlan.

A similar process is defined for a detailed estimate to be performed once the project scope is better understood (see Process Detail). There may be variations of the Initial Estimate processes for in-house development, or small projects versus large programs.

Specific estimation artifacts are listed as either inputs or outputs. Inputs trigger the execution of the process. It should not be performed until all the required inputs are available. You can indicate if inputs are optional, as with Vendor History.

The Project Manager is responsible for making sure the initial estimate is performed. Stakeholders that also participate are identified.

A clear statement of process activities to be performed should be listed in order as appropriate. The first activity is to review the SOW to identify key project estimate assumptions and constraints. Since this is a process overview, the list of activities is high-level.



Initial Estimate

ROUGH ORDER OF MAGNITUDE – VENDOR PROJECT

	INPUTS	ACTIVITIES	OUTPUTS	ROLES
PROCESS OVERVIEW	Statement of Work (SOW) Request for Proposal (RFP) Contract Requirements Vendor History (opt)	Review SOW to identify key project estimate assumptions and constraints: • T-Shirt Size • Total Budget Range – Labor only • Target Duration or Delivery Date • Staffing - FTEs and Skill Categories Calculate SLIM Trend Based estimate using T-shirt size estimate Calculate alternative solutions to explore potential outcomes Select recommended solution and make current Secure Management Approval Update RFP as needed	SLIM-Collaborate Estimation project record Recommended solution plus Low and High scenarios Estimate Briefing containing project dashboard charts & report Approved Initial Estimate Updated RFP	RESPONSIBLE Project Manager WORKS WITH Business Owner Enterprise Architect Investment Owner

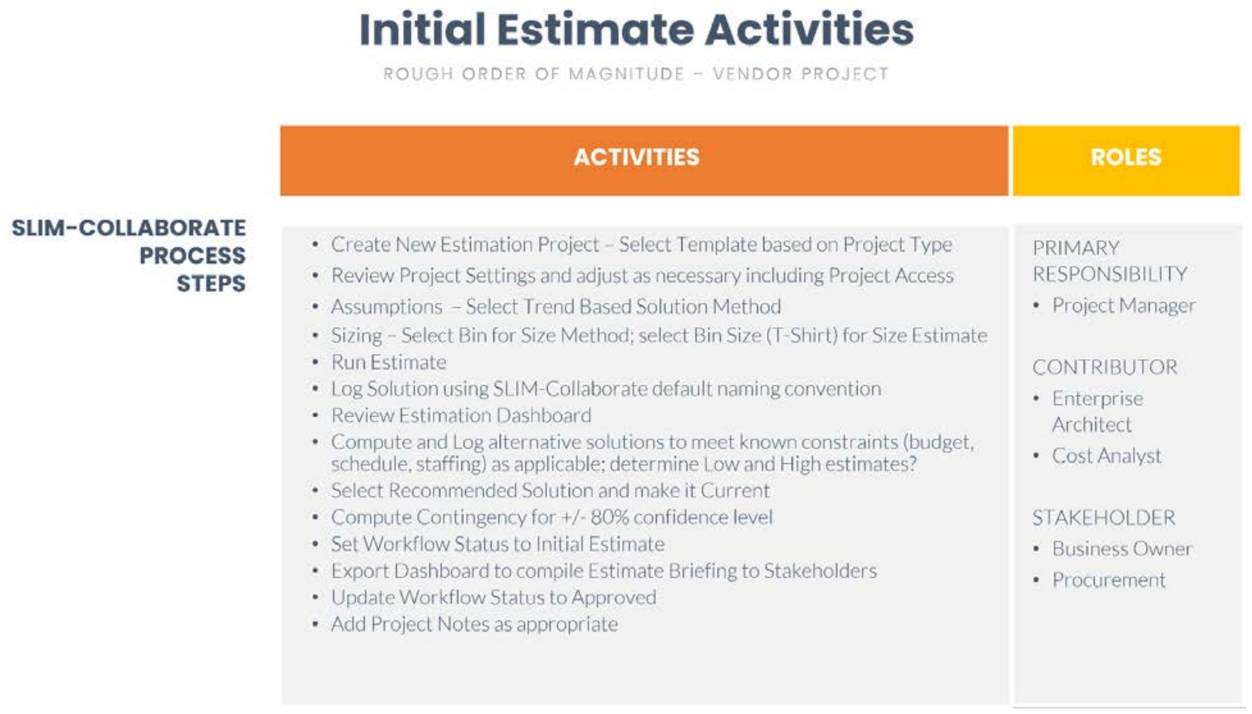
Process Detail

Depending on your process documentation format, you may want to create a detailed process view that lists all activities with instructions.

In this example, a key assumption is the solution method that is to be run to compute a ROM estimate – *"Select Trend Based Solution."* Likewise, another key assumption for a scope-based estimate is an estimate of software size. Because it is early in the lifecycle scope is typically not well known, so the user is instructed to *"Select Bin for Size Method; select Bin Size (T-Shirt) for the Size Estimate."*

Since SLIM-Collaborate is designed to promote collaboration among stakeholders, the detailed list of roles helps the Project Manager configure project access settings - grant permissions for those who can edit project data or simply view it.

Good processes promote consistency, repeatability, and increase performance. Another key benefit is that they provide visibility. A main benefit of SLIM tools is the ability to quickly compute what-if solutions to explore schedule, cost, and staffing alternatives. Setting the *Solution Workflow Status (Initial Estimate or Approved)* clearly communicates that the estimate has been completed. *Solution Log* names and descriptions, along with *Project Notes*, provide places where estimators can elaborate on the assumptions and document constraints, risks, and reasons for decisions made.



Initial Estimate Activities

ROUGH ORDER OF MAGNITUDE – VENDOR PROJECT

	ACTIVITIES	ROLES
SLIM-COLLABORATE PROCESS STEPS	• Create New Estimation Project – Select Template based on Project Type • Review Project Settings and adjust as necessary including Project Access • Assumptions – Select Trend Based Solution Method • Sizing – Select Bin for Size Method; select Bin Size (T-Shirt) for Size Estimate • Run Estimate • Log Solution using SLIM-Collaborate default naming convention • Review Estimation Dashboard • Compute and Log alternative solutions to meet known constraints (budget, schedule, staffing) as applicable; determine Low and High estimates? • Select Recommended Solution and make it Current • Compute Contingency for +/- 80% confidence level • Set Workflow Status to Initial Estimate • Export Dashboard to compile Estimate Briefing to Stakeholders • Update Workflow Status to Approved • Add Project Notes as appropriate	PRIMARY RESPONSIBILITY • Project Manager CONTRIBUTOR • Enterprise Architect • Cost Analyst STAKEHOLDER • Business Owner • Procurement

Summary

Defining, automating, and implementing efficient and effective software project management processes is key to staying competitive, today and in the future. Several business publications, including the Wall Street Journal, Forbes, and Fast Company describe *why every company is a technology company*. Software development is a major technology component. I hope you can use the tips in this guide to cultivate mature processes that lead to successful software projects that meet your strategic goals. Pick the most important areas to improve first and move at a pace you can sustain. Create a software improvement plan to avoid taking on too much at once. As my piano teacher says, if you want to learn a piece quickly, practice slowly.

About the Expert

Laura Zuber, Quantitative Software Management, Inc.

Laura Zuber has 28 years of experience in software development consulting, training, and support. She has conducted training and coaching sessions for all QSM SLIM-Suite tools and helped customers implement SLIM across a wide variety of processes and platforms. Laura has managed software development projects, served as a senior software process improvement specialist, performed process assessments, designed and implemented best practices, and authored numerous training programs. She is a Certified Scrum Master and SAFe Agilist.