4-Requirements & Process Control

In previous chapters, we developed a measure of success (MOS™) and a network of high-level achievements (HLA™) for both the users and IT staff that are necessary to reach the MOS™. In this chapter, we will further detail the plan, completing our:

- Feasibility of the business and technical achievements
- Detailed functional and technical requirements
- Software development process and quality controls.

These elements will give us our foundation for estimating work, cost and duration and the building blocks for making assignments to the project team. The intent here is to step through the requirements process with techniques that allow us to maintain our achievement orientation and scope control. We'll apply several development methodologies, not with the aim of teaching them but with the objective of demonstrating how we manage them in a project. The same is true of process controls and documentation. We'll build them into our project plan but not provide detailed procedures.

MAINTAINING THE LINK TO THE USER AND THEIR BUSINESS RESULTS

Throughout this requirements process, we must maintain the link between the detailed requirements and functionalities and the business achievements we have already developed. Maintaining this linkage is a key to controlling the scope of the project. We keep the users' focus on their measured achievements and the supporting business events necessary to deliver them. It is this focus that allows us to filter out features and functionalities that do not contribute to the measured business results and should not be included in our project. As a result, the IT people won't "go off and do their thing" without the user. We'll use a Joint Application Design (JAD) approach in most of the sections of the plan. Working with the users, we'll take our achievement network and break down each of those measured business results into business events. Then we'll break each business event into its inputs and outputs, process models, the data models, external interfaces, etc. Likewise, we'll detail the sub-achievements that the user is accountable for delivering.

Copyright 2002, The Hampton Group, Inc. All rights Reserved

We'll seek user agreement and commitment to the feasibility of their achievement and their sign-off on the requirements "pyramid" that supports each business achievement. These pyramids will also incorporate our decisions about development process quality controls, documentation and checkpoints.

If we lose this linkage to the already approved measured business results, we open the floodgates to an endlessly growing list of new requirements. We revert to the old game of users signing off on technical requirements they don't understand. IT delivers those requirements but at the end the user says the systems don't meet their business needs. With no linkage to business results, we can also miss requirements that we uncover during user acceptance testing or even maintenance. Correcting requirements problems late in the game is hundreds of times more expensive than during our initial development.

Keeping the user engaged in the process requires that we continue to "speak their language" during the process. As well, we want to reinforce that the project is a joint effort. Also, we'll use an events-driven requirement approach in part of the case project. Sub-dividing each achievement into the business events the user will have to handle with the new application allows us to continue talking in a language they understand, as well as offering some advantages for later partitioning of the effort.

Beyond these two suggestions, it is not our intent to teach any particular requirements methodology or to delve into platform-specific techniques or specific CASE tools. You should be able to fit your choices in these areas into the project management framework we'll develop.

ELEMENTS TO BLEND INTO THE PLAN

Building our detailed project plan is tricky because there are so many diverse elements we have to include.

- First, the eventual work breakdown structure (WBS) has to reflect the hierarchy of business achievements that the user has approved.
- Second, the WBS has to provide for the functional and technical requirements of the system.
- Third, it has to include the system development checkpoints for software process improvement and quality control as well as hardware installation and networking..
- Fourth, our WBS must contain clear, measurable assignments we will hold our team members accountable for delivering.

Copyright 2002, The Hampton Group, Inc. All rights reserved

- Fifth, we need a project plan that is easy to keep current with actual results so we don't have to assemble the team to answer every question an executive asks.

Let's discuss each of the components that we need to build into the work breakdown structure (WBS).

CONTROLS ON THE DEVELOPMENT PROCESS

There has been a great deal of attention paid to software process improvement from many corners of the information technology world. One frequently used approach is the Capability Maturity Model[1] (CMM) developed by Software Engineering Institute. They have drawn particular attention to the evolution of software development processes in organizations. This CMM model highlights the cost and success rate benefits that come when organizations mature their development processes. For the purposes of our ongoing Ridgeway case, we will assume that this project is taking place in an IT organization that is in the first, or lowest, stage of maturity. That is, there has been very little standardization of the software development process. Development projects suffer from constant firefighting and poor control over requirements or software quality. The result is the frequent sacrifice of functionality and quality to meet deadlines. As IT organizations progress to higher CMM levels, the software development process becomes standardized, successes can be repeated and the organization has an increasing ability to measure the effectiveness of its processes and the quality of the systems produced.

In our project plan we will also apply several different software development lifecycles and control processes to different sections of our example project. In practice, your organization may mandate its own. Our discussion does not aim to change your organization's standards. Our purpose is to cover a number of fairly common alternatives in the hopes that every PM can adapt what they learn to their organization's process-specific requirements. Also, for those managing in organizations where software is developed on an ad hoc basis, the options discussed below will provide alternatives so that at least your project gains the benefits of development process controls, even if your organization has not yet developed them.

Project management obviously plays a key role in the software process improvement efforts. Our focus on developing project plans based on measured achievements supports these organizational efforts to make the development process more measurable, controllable and easier to track.

---

[1] Capability Maturity Model: Guidelines for Improving the Software Process, Carnegie-Mellon University, Software Engineering Institute, Addison Wesley November 1999

Copyright 2002, The Hampton Group, Inc. All rights Reserved

SYSTEMS DEVELOPMENT LIFECYCLE

The system development lifecycle has been around for a long while and most of the alternatives are built on the classic sequence of:

- Design
- Construct
- Test
- Maintain.

As organizations add process improvement methodologies, the basic lifecycle expands with sub-divisions of the four basic steps and additional checkpoints, approval and documentation requirements:

- Requirements definitions
- Requirements analysis
- Preliminary design
- Detailed design
- Implementation
- System testing
- User acceptance testing
- Maintenance and operation.

We might also see even more elaborate lifecycle models with the stages of:

- Funding decision
  - Business case
  - General business requirements to approval of funding for technical requirements
  - Feasibility study
- Project launch
  - Technical requirements approved
  - Enterprise architecture approved

Copyright 2002, The Hampton Group, Inc. All rights reserved

- Detailed design complete
- Construction/unit testing
- Component integration testing
- Systems test
- Interoperability test
- Site readiness

- Deployment
  - Site system test
  - Installation/deployment
  - Maintenance
  - Retire system.

We'll use several variations of these processes as well as applying Rapid Application Development (RAD) and Spiral development processes. Whatever option you use, we need to include:

- Clear specification of control points and approvals before work starts
- Inclusion in the work estimates of the effort required to meet the development standards
- Tracking of progress using the selected control process
- Treating changes to our process control points with the same process used for other scope changes.

The intent is to make our development controls part of the project work, not something to be completed after we're done or if we have time. Nor should they be the easy target for elimination when the due date is slipping and we're over budget.

EVENTS-DRIVEN REQUIREMENTS ANALYSIS

The development of detailed systems requirements is a key component to project scope control and the system development lifecycle. It also gives us our foundation for work and cost estimating. Just as organizations adopt a wide variety of systems development processes, there are also a wide variety of choices in terms of how the requirements analysis is conducted. We will utilize a relatively straightforward events-driven requirements process, which can include some or all of the following steps:

Copyright 2002, The Hampton Group, Inc. All rights Reserved

- Identify business events
- Identify inputs and outputs for each business event
- Determine relationship between inputs and outputs
- Determine the precedence relationships between events
- Gather information on external interfaces
- Gather information on operating environment requirements
- Develop process model
- Develop logical data model and set up the data dictionary
- Prepare the software requirements specification document
- Prepare acceptance criteria
- Obtain user sign off on the test criteria.

The main advantage to an events-driven approach is that it maintains our linkage to the business achievements better than many other ways of breaking up the development effort. An events-driven approach lets us speak the users' language very deeply into the process so it fits the joint development and achievement oriented approach quite well. Thus, we continue with both high levels of user involvement and the exercise of scope control.

Let's begin our requirements analysis for the Ridgeway project by detailing the high-level achievement that addresses improving the ability of the Customer Service Representatives to answer customer inquiries about the status of their order. In the strategic planning, the PM developed the HLA™ and the supporting sub-achievements below:

- Increase customer inquiry calls per Rep from 8 to 12/hour
- Shipping/order history data converted at 99.5% accuracy by sample
- <10% of order status calls require transfer or call to Shipping
- 95% of Customer Service Reps score 80% or higher on test of using shipping status information.

We will focus on the second of the above sub-achievements, giving the Customer Service Reps the information to answer 90 percent of the order status phone calls without having to transfer the customer to Shipping or call Shipping themselves. The key issue here is that because of the way the project manager has structured the achievement network, our requirements analysis does not have to cover every possible question the customer could ask about their order's shipping status. To reach the productivity increase described by the high-level achievement, the project manager "cherry picked" the

Copyright 2002, The Hampton Group, Inc. All rights reserved

problem with a 90 percent solution. Thus the analyst doing the requirements analysis works within already defined constraints. The analyst must be concerned with scope control but we've made the job easier with the strategic planning.

For this achievement, we'll see how the analyst would develop the detailed requirements using an events-driven approach.

BUSINESS EVENTS AND SYSTEM RESPONSE

Meeting with the users, we would take these sub-achievements and develop a list of business events with which the system must cope. Because we don't need to solve every possible situation, we need information on the volume or frequency of each type of inquiry. We might start the session by discussing the achievement, stressing that we need to begin by identifying the business events that cover 90 percent or more of the customer shipping status inquiries. The analyst is going to work very hard to limit the discussion to just that subset of events. From a sample of customer inquiries or from the users' knowledge of their business, we might discover that the customer inquiries about shipping status break down as follows:

- 68% call asking if their order has been shipped
- 15% call asking when the order will ship
- 10% call asking when they will receive their order
- 3% call asking about international shipping & customs
- 1% calls asking other questions.

With this information, it's clear that if we address just the first three kinds of inquiries, we can reach the 90 percent level required by the achievement. As an aside, it is very often the case that adding the fourth and fifth event types might double the cost of the systems development effort in this part of the project.

Now we take the next step, listing the business events to which the system must respond. Here's where it's important to continue our scope control process. The users are typically not happy when we ignore part of their "problem." But the analyst has to stay within the confines of the achievement and not let the session drift to other features or functionalities which, no matter how desirable, do not contribute to handling the first three events listed above. The achievement is that our Customer Service people can answer 90 percent of the inquiries, leaving 10 percent that will still have to be transferred and which need not be addressed in our project requirements.

After discussion with the users, we might have an event list like this:

Copyright 2002, The Hampton Group, Inc. All rights Reserved