SEPM-001

The Dynamic Balance Between Cost, Schedule, Features, and Quality in Software Development Projects

William S. Junk

Computer Science Department
College of Engineering
University of Idaho
Moscow, ID 83844-1010

April 17, 2000

The Dynamic Balance Between Cost, Schedule, Features, and Quality in Software Development Projects

Introduction

In today's business world getting winning products to market is a significant challenge, but one that an organization must learn how to do in order to survive. Primary responsibility to see that product development is successful rests with the project or program manager. This paper discusses the dynamic interaction that takes place between product features, development cost (resources), quality, and development schedule in a "system" environment. The system that we are referring to is the abstraction created by feature, resources, quality and schedule and their interactions. The premise on which this paper is based is that the "system" will always seek an equilibrium point among these four dimensions. Unfortunately, the equilibrium point that balances the four factors may not correspond with what senior management defines as acceptable for a successful product development. A project manager must actively manage the project for which he or she is responsible and must do this with an understanding of the constraints placed on project control options. Using this model we discuss what is likely to happen if management does not properly plan and control the project.

Basic Principle of Project Balance – Staying in Equilibrium

A fundamental premise on which this paper is based can be stated as follows:

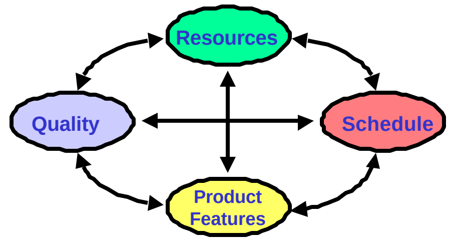
> *Cost, schedule, feature set, and quality are the four most significant project dimension that project managers must deal with. Cost, schedule, feature set and quality from a four dimensional self-regulating system that seeks a balance or equilibrium among the four dimensions. A balance will be achieved with or without project manager intercession.*

To be successful, the project manager must effectively control one or more of these dimensions and maneuver the project to achieve the desired results. We'll use the figure to the right to illustrate the interrelationships that exist among these dimensions and to help understand what an equilibrium point is in this system. A balanced system exits when there are sufficient resources and time available to deliver the required feature set at the expected quality level. Of course there are an infinite number of possible combinations of the four dimensions that can create an equilibrium state.

By definition a balanced system represents a system in equilibrium. It also reflects a point of local optimization. A system in equilibrium occurs when there is alignment of the dimensions producing a joint minimum of time and resources needed to deliver the required features at the expected level of quality. Note that a joint minimum of time and resources does not necessarily imply that either or both are at an absolute minimum, but rather, together they represent the best we can do given the desired outcome for the feature and quality dimensions. There can also be several different combinations of time and resources that can achieve the equilibrium point. To further illustrate, if three of the dimensions are set (e.g., by senior management decision), then the fourth dimension will adjust to its optimum value that makes the set feasible. Setting schedule, resources, and quality will drive the system to a point where delivered features creates a feasible solution. In this context, *feasible* is used in a mathematical sense. This solution may not be viewed as acceptable by senior management because it does not result in a viable product.

A balanced system, meeting an organization's expectations, will exist when the desired product feature set can be delivered with the required quality level using the allocated resources and performing to an acceptable and agreed upon schedule. Notice that

Interrelationships Between Major Project Management Dimensions



the emphasis here has shifted from that used in the basic definition of a balanced system. The definition put emphasis on having sufficient resources and time. When we include organizational expectations our system becomes more constrained. We no longer have complete freedom to balance the system by adjusting just any variable of choice. For example, we might not be able to added resources. Under these restrictions it may not be possible to reach an equilibrium point that is considered acceptable in all four dimensions.

Of course systems are not static. An “event” may occur in one dimension, moving that dimension’s current operating point. For the system to regain balance, a compensating adjustment in one or more of the remaining dimensions will take place to again establish equilibrium. Reaching equilibrium after a disturbance is not an instantaneous reaction. There’s delay from the time the trigger event occurs until the system has fully compensated. In some situations the adaptive behavior of the system may be invisible because the responses that indicate changes are taking place aren’t observable.

To illustrate this dynamic behavior consider the specific case in which, after the project has been initiated, there’s a need to increase product features. In order for this system to regain equilibrium after the increase in features, there must be a concomitant adjustment in one or more of the other project dimensions. One acceptable adjustment might be to increase the project resources so that, along with the original commitments, the additional work can be completed as originally scheduled. Another alternative might be to allow additional time, generally accompanied by a corresponding increase in total expenditures. It is generally not an acceptable response to allow quality to fall below the organization’s norm, that is, one should not ship an immature product full of defects to the customer. In this situation what happens if the project manager doesn’t make any overt adjustment in the system dimensions? We contend that the system has been placed in an unstable state and will autonomously make adjustments to regain equilibrium. Specifically, without intervention, we expect to see an adjustment in schedule or quality or both. Without additional resources, the increased scope of development will increase the work that has to be done. This in turn eventually causes a delay in completion of previously scheduled tasks. The expanded feature set won’t be delivered by the original completion date. Something else might happen as well. When it is realized that the original schedule isn’t doable, the team may react by taking short cuts. Taking short cuts will invariably lead to a decrease in product quality.

It would be a difficult enough task to mange a product development if the project manager could be assured of beginning a project in a balanced state. Whether or not the project begins in a balanced state is dependent on the thoroughness and accuracy of the project planning process. Unfortunately it is never possible to know with certainty whether or not the initial plan creates a balanced system. In fact, the initial project planning is rarely perfect and it is likely to produce a plan that is out of balance somewhere. The more thoroughly we can plan the more we can lower the risk. In addition, selecting an appropriate development strategy that’s compatible with the level of uncertainty will also help to create a project environment that is more manageable.

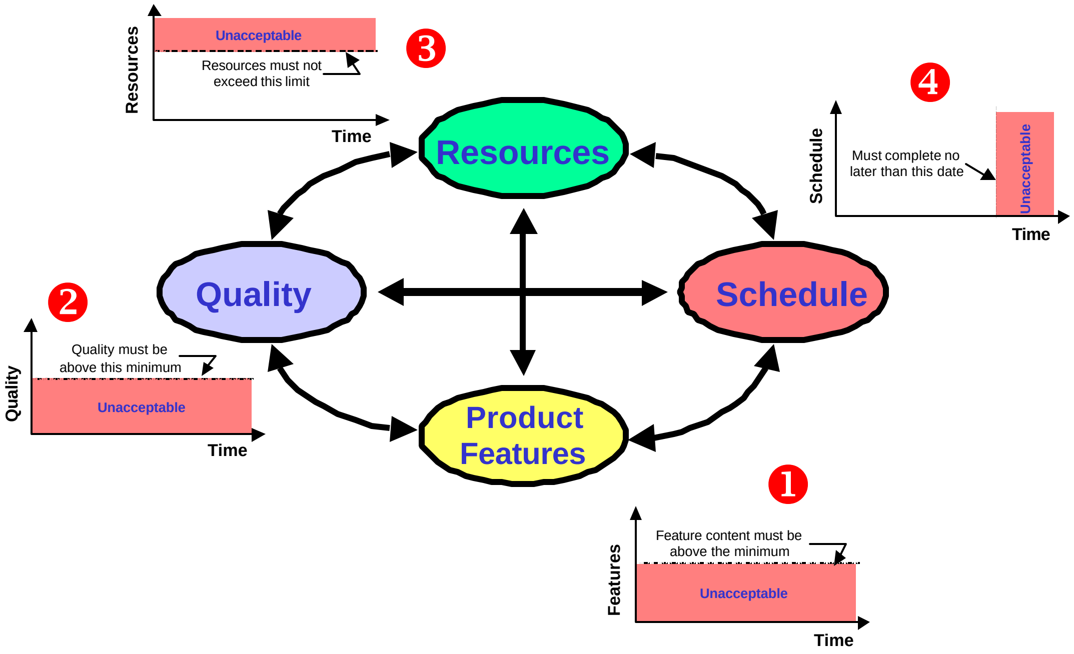
A project manager would like to be able to make the assumption that during the course of the project nothing will change in any of the four management dimensions. In reality, change is an expected part of most projects and something the project manager must be able to accommodate in order to be successful. Having a sense of how much change to anticipate is also important during initial project planning. Whether or not the project manager has the freedom and authority to make the required adjustments is another matter. Often they do not.

Let’s now look at the four dimensional space the project manager is trying to control. In the diagram that follows each of the major dimensions is shown along with a graph that we’ll use to discuss the management objectives and the dynamics that are associated with each.

When it comes to product features (see ❶ in the figure), the project manager may have some room to navigate by removing planned features from the product, but there will be a point at which the feature set reaches those required for the minimum viable product. Once the minimum viable product point is reach, further reduction in features will result in a product that has such limited utility that it is unlikely to be a market success. We have to stay above the threshold and out of the red zone.

In the quality dimension (❷) there‘s generally a minimum acceptable quality level that must be reached before it is safe to release the product. This minimum level is set by a combination of factors, among them the customer’s general expectation for products in this general class. The customer may have additional expectations for a specific company that are largely based on previous experience with their products. It’s akin to brand preference. So

Goals and Limits
in the Major Project Management Dimensions


Resources
Unacceptable
Resources must not exceed this limit
Time
❸
Schedule
Must complete no later than this date
Unacceptable
Time
❹
Resources
Quality
Schedule
Product Features
❷
Quality
Quality must be above this minimum
Unacceptable
Time
❶
Features
Feature content must be above the minimum
Unacceptable
Time

again we have to be above the threshold and out of the red zone at project completion.

Projects are normally initiated with expectations that the work will be completed using some allocation of budgeted corporate resources (❸). The goal of the project manager is to bring the product in at some point under the target budget. So here we need to be below the threshold and out of the red zone. Of course, if something bad happens in another dimension there may be alternatives that allow increasing the resources available to this project. But there's a consequence to be paid by other ongoing projects or future projects. The opportunity cost associated with increasing the resources given to our project means something else can't be done.

Finally, there's the schedule dimension(❹). Probably the most significant objective given to the project manager is to get the project completed on or ahead of schedule. Here we must not exceed the threshold and must say out of the red zone.

At this point it is worthwhile to digress momentarily and discuss what a schedule really represents. The project schedule dimension is an interesting and unique one. A schedule isn't a map of what is going to happen during the project. It is an estimate of what we think will happen in the future. A schedule contains uncertainty and is probabilistic. When we start thinking about managing schedule, we're forced to do it indirectly but manipulating one or more of the other dimensions (features, quality, or resources). It is a misconception to think of this dimension as one that can arbitrarily set or controlled. Thinking that a project can be controlled by manipulating the schedule is just asking for trouble. This should not be construed as suggesting that a schedule isn't a valuable project planning and management tool. It is. But we have to consciously separate the tool from the project dimension. In effect, the project is done when it's done, but not necessary done when someone decrees that it needs to be done or when the schedule predicts it will be done.

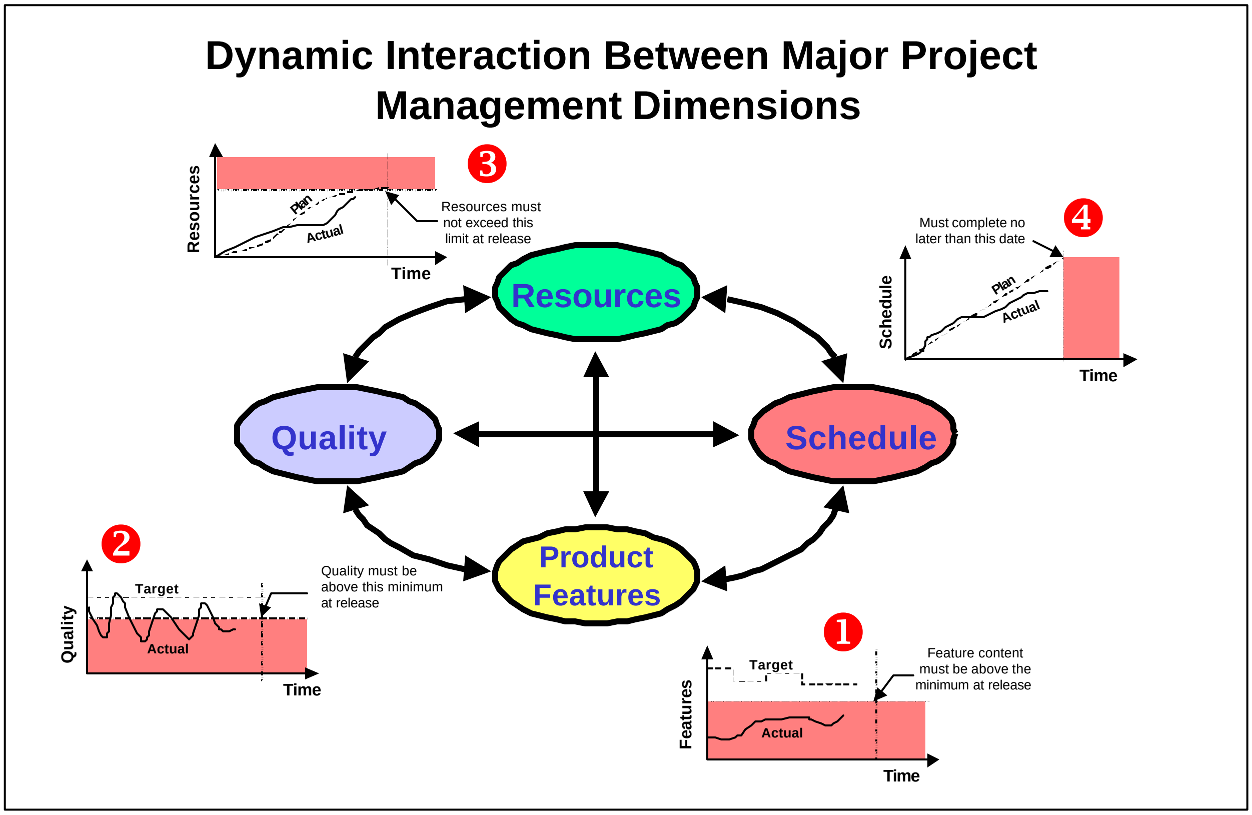
A Closer Look At Reality

Initial project plans are generally prepared using preliminary information. In essence we make an informed guess! As a consequence there is a significant likelihood that some portion of the system will not be understood completely, and therefore initial plans contain errors. In this case errors are not really the same things as mistakes, but rather reflect an uncertainty about future needs and outcomes. It might be more appropriate to think of these errors as system noise that will introduce a degree of randomness within our system. We can only hope that these random movements aren't large enough to knock us from an acceptable state into and unacceptable one.

In most cases the project manager and team don't have complete freedom to navigate in all four dimensions with equal agility. It is very common for the team to be given general marching orders and a completion date. The completion date is often set well before there's a solid understanding of what the product or system must do. That is, major commitments are made before the requirements for product features are set. This effectively removes one degree of freedom from the program management space. Once the delivery or release date is set it is very difficult to get additional time for the project.

The following figure illustrates the general trends that one might expect to observe during a project. Let's assume that the project is using and incremental delivery approach and that the system is generally kept in a functional state. That is, we don't introduce massive changes that disrupts previously completed and validated functionality. As a general rule, when we complete the project we want to be on or ahead of schedule and have product features, quality, and total resource expenditures under control. Having product features under control means that we have at least delivered the feature set required by the minimum viable product, but generally expectations will be substantially higher. In the figure the red shaded area represents an unacceptable final state. We must deliver a product that places our final state safely within the white area on each dimension, otherwise we compromise the success of our project.

Looking at the product features graph (see ❶ in the following diagram), this plot represents the magnitude of features that have been successfully completed at various points in time. Of course the ultimate goal is to deliver all of the planned features, but we may need to adapt to our circumstances and settle for delivering a product that's somewhat above the minimum viable product feature set line at the time the project is scheduled to complete (represented by the dotted vertical line on the right side of the graph). The project team may start from a point where some features initially work, as in the case where they are extending an existing product. As development proceeds, the amount of compliant functionality may increase or temporarily decrease as changes are introduced into the system, but there needs to be an overall upward (increasing) trend. In contrast, the plot of desired or required features may increase or decrease over time. It will increase at points when new features are added to the product. It


Dynamic Interaction Between Major Project Management Dimensions
Resources
Quality
Schedule
Product Features
3
Resources
Plan
Actual
Time
Resources must not exceed this limit at release
4
Must complete no later than this date
Schedule
Plan
Actual
Time
2
Quality
Target
Actual
Time
Quality must be above this minimum at release
1
Features
Target
Actual
Time
Feature content must be above the minimum at release

will decrease at points when the organization consciously has decided to decrease the feature content possibly to accommodate changes in other dimensions. Unfortunately, for a lot of products the general trend is to move from an early plan to develop an highly innovative and feature rich product to a position at release that is just slightly above the minimum viable product.

The graph of quality over time (❷) represents the current quality level exhibited by completed or delivered product features. It is possible that the quality at some intermediate point in time may exceed the minimum acceptable level and may even exceed the target level, but at these points the full functionality is not available. Quality is likely to oscillate over the course of the project as new features are developed and integrated. The magnitude of the oscillations will depend on how well the new features are tested before they are released for integration and how rapidly the system testing discovers additional defects . The general trend of the quality curve should move up and to the right as the project progresses, reflecting a gradual increase in the quality level. Since improvement occurs in response to the discovery and correction of defects, the earlier that defects can be detected and corrected the better. Similarly, the goal is to drive the quality above the minimum acceptable level before the scheduled completion time and keep it there.

One of the primary responsibilities of the project manager is to ensure that the quality target is reached or exceeded by project completion. In general the quality target is set based on the organizations historical performance and evolving customer expectations. Project quality targets are often rather subjective anyway. This makes them hard to define quantitatively and makes it difficult to know through meaningful measurements whether or not the project is on track to achieve the target. Despite the inherent difficulties in assessing quality, it is important to keep the quality level as close to the target level as possible. Deep depressions in the quality level will present significant challenges from which to recover.

The graph representing cumulative resource use (❸) will climb as resources are expended. A major responsibility of the project manager is to manage the resources so that all the planned work is completed as scheduled and that total resources stay below a limit authorized for the project. Authorized changes may be made the course of the project, moving either up or down in response to external events.

The final graph (❹) represents schedule progress as a function of time. The plot could represents the number of tasks or milestone completed vs. those scheduled to be completed. It could also be based on a more sophisticated techniques such as earned value that attempts to measure the delivery of product features. If the project is progressing on schedule, then actual progress should trace along the solid diagonal line. Under these circumstances, all tasks and milestones are completed by their scheduled completion date. If the project is ahead of schedule then the actual plot will be above the planned plot and just the opposite if the project is behind schedule. In practice it will be more typical to see movement above and below the diagonal reflecting actual accomplishments that are ahead of or behind schedule. This seemingly random behavior is the result of our inability to precisely foretell the future. Remember that schedules are only estimates. The project manager must be aware of the deviations from plan that are taking place and make appropriate adjustments that will bring the project to conclusion on time while also achieving the targets in the other three dimensions.

Specific Scenarios

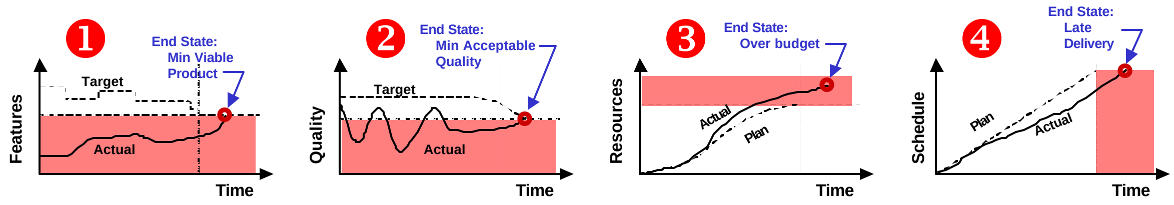
In the following paragraphs we present a more detailed discussion based on several scenarios that are typically encountered in real projects. Each scenario is accompanied by a diagram showing one possibility for the dynamic behavior of each management dimension. All scenarios assume that the project is modifying or extending the features provided by a successful predecessor product.

Scenario 1: The amount of work required to implement the features is underestimated.

Let’s consider the situation in which the initial project plan does not identify and provide for all the work that will be needed to complete the product. This can happen if optimistic estimates are used, if some of the work scope is not adequately defined, if the assumptions about available resources are invalid (e.g., counting on a specific skilled individual), and a number of other causes.

The consequences of poor initial planning can be multifold. We generally don’t realize there’s an imbalance until we’ve moved into activities, such as detailed design or coding, where the effects of the increased work scale are noticeable. They’re noticeable as a systematic failing to meet schedule or as additional tasks that have to be added to the schedule. This is indicated in graph ❹ of the following diagram where the actual schedule

Scenario 1: Underestimating the Amount of Work



accomplishment is shown lagging planned accomplishment. What will the system adjustment be in the other dimensions?

If the original feature set requirements are maintained, and if no additional resources are added to the project, then quality and schedule must absorb the shortfall. From a project management standpoint we can only allow quality to absorb the shortfall to the point where we've reached the minimum acceptable quality level. Additional schedule and resources will be required to compensate.

In most scenarios there is very little room to maneuver with respect to quality. There's a quality expectation that has been set by our prior products or similar products from competitors. Towards the end we may relax some of our own ideals (❷) but we can't go too far for fear that our product will be rejected in the marketplace.

If we are given the option of reducing features then we may be able to respond effectively (see ❶ in the graph below). The key is to have enough features beyond those required by the minimum viable product that we can sacrifice to adsorb the difference. If we can off load enough tasks (via feature elimination) we may be able to get back to an acceptable equilibrium point where all of the original project assumptions except for feature set are satisfied. If we can't reduce the feature content sufficiently to reach an acceptable equilibrium point, then another dimensions must absorb the adjustment. Again, it is likely to be a combination of resources (❸) and schedule (❹).

What this scenario really portends is a final state reflecting a minimum viable product, minimum acceptable quality, over budget, and late delivery. The dynamics leading to this final state are shown in the following diagram.

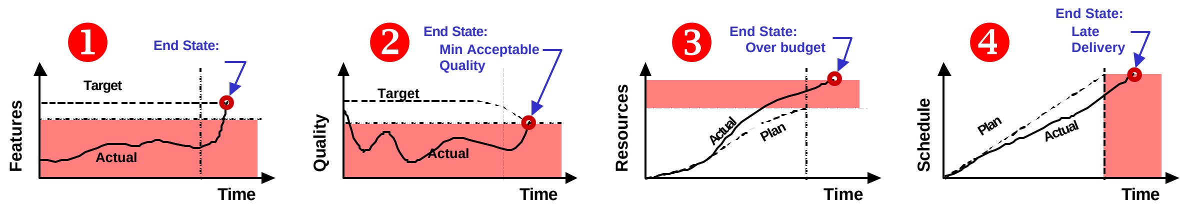
Scenario 2. Greater than expected number of defects.

We could find ourselves in a situation reminiscent of this scenario if we don't use good software engineering principles and practices, or if we try to take short cuts to improve schedule performance. Let's assume that we're heavily into the system level testing of our product. Up to this point things have seemed to be going fairly well, but now we are seeing many serious defects, many more than is typical. In addition, these defects are requiring extensive amount of our development resources to analyze and fix and the sheer number of them are making it difficult for the test team to make progress in running the system tests. What do we expect from our system – a system that wants to maintain its equilibrium?

With high defect rates progress isn't being made as planned (see ❸ in the figure below). We are also experiencing a decline in quality and because our system is being stressed we are probably below the threshold of acceptable quality (see ❷ below). In the late stages of development it is difficult to effectively add resources (Brook's Law) because more people cause turmoil and interfere with whatever progress is being made. Realistically, about all we can do at this point is to ask our people to work harder and longer. The first tension point is likely to be schedule. If a schedule slip isn't considered acceptable what will happen? Well, a schedule slip will likely happen whether or not it is considered acceptable! And because all of our resources have been devoted to work that's delayed, resource requirements will exceed the original plan. We've assumed that we can't reduce features, but if we can, we might be able to mediate the impact in the other dimensions.

The end state for this scenario looks like the following: nearly full set of features delivered; quality is near, but above the minimum acceptable level; over budget; and late delivery. What could we

Scenario 2: High Defect Rate



do to improve this situation? Avoidance is the key. Do your up-front software engineering and don't cut corners.

Scenario 3. Late addition of features or change in features.

Requests for additional features or feature changes received late in the project (after major design is completed or code is written) are relatively common and arise from changes in the marketplace. Changes can be accommodated if compensating adjustments are made in other dimensions. Generally, it isn't acceptable to lower quality, although we might be able to get away with a little decline if we are substantially above our minimum threshold. What it really takes to respond to this situation is either additional resources, or additional time, or both.

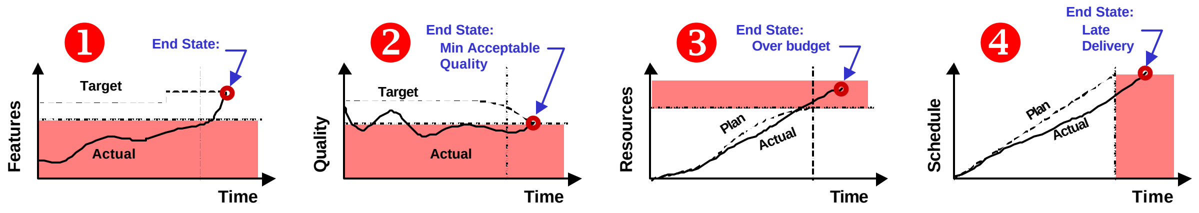
Unfortunately, there's a behavior that often surfaces in these situations. The project team is told to "get the job done" and aren't given additional resources or time. What should we expect to happen in our system? The system will naturally drive itself towards a balance. What we will see with features and resources fixed is a downward compensating adjustment in the target quality goal (see ❷ below). If the quality goal is lowered to a point very close to the acceptable minimum, but if this is an insufficient relaxation to complete all the rebalancing, then we'll also see a schedule adjustment, that is, a schedule slip (see ❹ below), to reestablish equilibrium.

The end state for this scenario looks like the following: full set of features delivered; quality is near, but above the minimum acceptable level; over budget; and late delivery. How do we avoid this scenario? Carefully manage features. Always perform detailed analysis of any new or modified features, not just analysis for technical feasibility, but a project management analysis to assess feasibility within the project constraints.

Conclusions

As we can see from these relatively common scenarios discussed in the previous section, if senior management insists on exercising control over critical project dimensions, the results can be undesirable. Since a project schedule is just our estimate of what we think will happen in the future, most of the time a change in the system is perceived as a schedule slip. Of course other things can happen too. Quality can decline, product features can decline, and resource requirements (costs) can

Scenario 3: Feature Set Change



escalate.

The real path to success is to have an initial project plan that is in equilibrium – the resources and time available to do the work are consistent with the features to be developed and the quality demanded. Good software engineering practices and feature management also contribute to success. But if the project gets out of equilibrium, compensation will be seen in one or more of the management dimensions that isn't being held constant.

Acknowledgements

I'd like to take this opportunity to thank the people who reviewed this paper during its preparation: Mike Ogush, Keith Moore, Sandy Lieske, Tom Irons, Deborah Owens, and Scott Matthews.