



## Schedule Slippage: Root Causes

Mukesh Soni and Madhusudan Acharya

*“The single most important task of a project: setting realistic expectations. Unrealistic expectations based on inaccurate estimates are the single largest cause of software failure.”*  
- Futrell, Shafer

### Introduction

With global and competitive market, it is very important to launch a product or service in the market on time and ahead of competitors. A timely launch definitely depends on on-time-completion of the product development projects. Project planning has lots of challenges to overcome in order to finish the project on time – right from

- Schedule predictability
- Envisioning future/possible risks and
- Coming up with mitigation plans.

This article talks about some of the challenges often faced in the Software Product Development industry that cause schedule slippage.

### Schedule slippage:

Delay in the project completion from its initial estimated date of completion.

Each project plan will have a planned completion date (NRA, RA), and a bounding box or upper limit in the schedule. Nowadays, it is common practice to have three dates associated with any project plan:

- **Non-Risk Adjusted (NRA) Date:** Project completion date assuming no hurdles – Ideal conditions.
- **Risk Adjusted (RA) Date:** Project completion date assuming some risks will come on the way and we will need extra time to attend to them.
- **Bounding Box (BB) or Upper Limit:** The upper limit on the project plan before which the project has to be finished under any circumstances – Generally decided by top management based on product/services roadmap and launch in the market.

The relation between these dates on a time scale can be explained by figure 1.

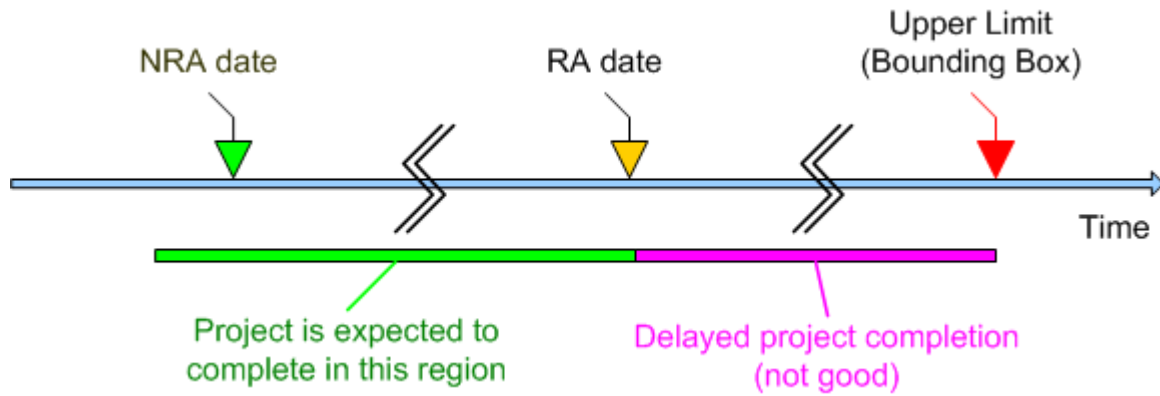


Figure 1: Relationship of various dates for project completion (NRA, RA and upper limit) on time scale.

Under ideal circumstances, a project is scheduled to complete by the NRA date. Considering risks that may come along the way that will eat some time out of the schedule, the project should be over by RA the date. If the risks are not envisioned and hence not planned for, then the project may get delayed and will complete after the RA date. Project completion crossing the RA or upper limit is neither good nor expected in a well-planned project.

### Root Causes

We always plan for a project to be over before the RA date however it is seldom the case that it happens as expected. There are multiples reasons for schedule slippage, right from improper planning, lack of resources to unplanned requirements and rework that eat away vital time from the planned schedule.

A typical project development process is shown in figure 2. Each project will have a team (development, testing and other functions) that will work through a process (requirement analysis, schedule estimation, design, implementation and testing) to deliver a product to the customer/end user. Each entity that participates in the project – directly or indirectly - affects the schedule.

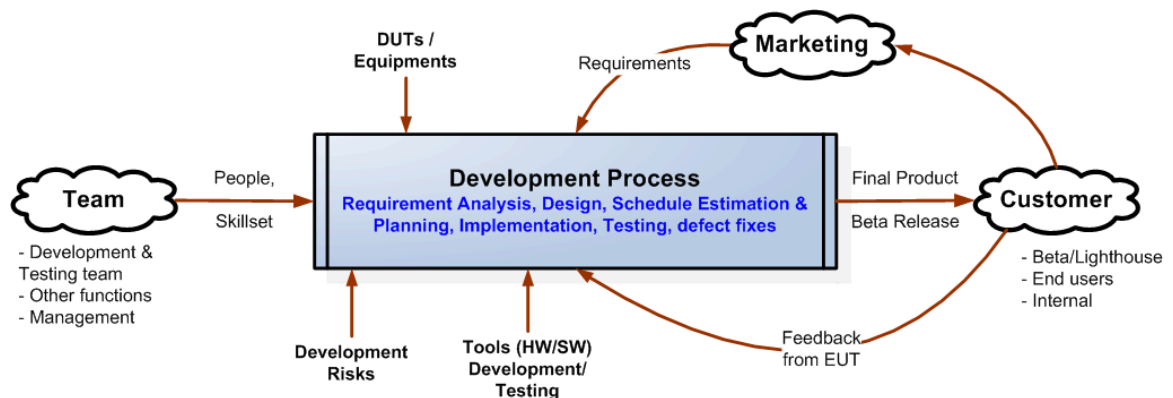


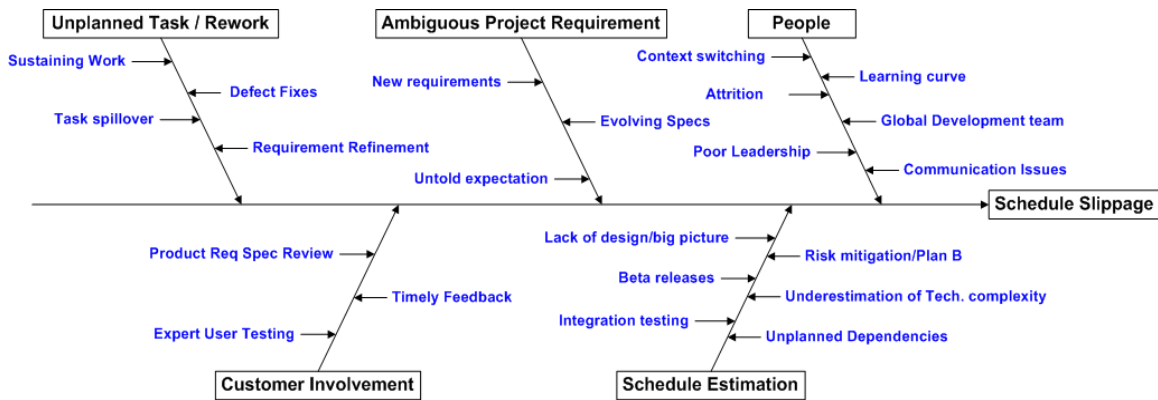
Figure 2: A typical project development process and various entities involved.

From the development process, we can identify the items that can cause delay in the execution of the project – for example:

- Misinterpreted or unclear requirement can add to completion time

- Unavailability of development tools or resources can prolong the project duration.
- Various processes like schedule estimation, detailed design and product development if not executed skilfully, may significantly blow out the project timing estimate

For a better understanding of all these possible causes that may result in schedule slippage, they can be categorized and represented in terms of a fish-bone diagram shown in figure 3.



**Figure 3:** Cause and effect (Fish Bone) diagram representation of possible causes of schedule slippage.

Let’s have a detailed look at the root causes of schedule slippage category wise.

### 1) Schedule Estimation:

*“The key is not to prioritize what's on your schedule, but to schedule your priorities.”*  
 - Stephen Covey

For a project to be executed on time, it is very important to have it planned very well. Any mistake in project schedule estimation reflects as delay in the project completion from its deadline. There are several factors that contribute to improper schedule estimation:

#### Underestimation of technical complexities:

At the start of the project, many of the team members may not have thorough knowledge of technical complexities and hence their estimation could be incorrect. Sometimes it may so happen that the person giving estimates for a particular task has no idea about the technical challenges involved in carrying out that particular task. You might hear, towards the mid/end of the project life cycle when the task is not finished on time –

“Oh, I didn’t know that this feature also requires 5 more tasks to be done!”

Or

“I was thinking this task to be so simple, but I under estimated it!”

#### Lack of Design/Big picture:

It is important to have a big picture / overview of the complete project to understand how a particular module/feature would fit in to the complete project. Product or system level design helps in understanding the interfaces among other modules and the required coordination for product assembly and hence, a better insight into the

work involved. Often, estimates without focus on detailed design tend to deviate more from the actual time taken for finishing the job.

**Integration Testing:**

While making a project plan, testing also needs to be accommodated in the schedule. At times, the unit testing or testing done by individual contributors on their module is taken into account but not the system level testing. Toward the release, when all the individually tested modules are brought together, system level or integration testing is a must. Having the time for integration testing not accounted for in the overall project schedule will cause delay in the project completion.

**Unplanned dependencies:**

Project planning is not only about breaking the project into minute tasks and managing them. A well-planned project schedule also needs to consider certain unplanned dependencies. Some of these are:

- **People:** Optimum utilization of human resources calls for the same set of people to be concurrently working in multiple projects. A person may not be available to work for currently planned/assigned project due to extended/unplanned work in another parallel project. Another issue related to people could be unplanned/unexpected attrition that will affect the project plan. Time is also lost in mentoring of new member by a senior (more experienced) person which goes unaccounted if not planned.
- **Tools & Equipments:** Project can be delayed if the team is waiting for the release of upgrades or procurement of any vital tool (hardware or software being used in the project) or if the equipment required for development and testing is not available.

*“We had a 3-months project for validating our existing solution on new product platform using customer DUT (device under test). We had to wait for the DUT for nearly 1.5 months as it got stuck in customs. After getting the DUT, we realized that it’s been damaged partially during transportation. As a result we had to ask for another DUT and whole project took more than 5 months to get finished.”*

I am sure that such cases will be quite familiar to many organizations.

Other reason for timely unavailability of tools / equipments is that they are shared among various projects to reduce the operating cost. Any unplanned dependency on their usage or wrong assumption about availability of these shared resources could cause a delay in the program. Team members might have to work on shifts to optimize the usage of shared resources which can cause reduced work hours and/or productivity loss and results to schedule slippage.

*“I was waiting for Matlab license to be released by another person in the team but he left the office without doing so and I lost 3 hours figuring out what to do?” – is it something you faced before?*

- **Other Programs:** If multiple programs have deliverable dependencies, then a delay in one project will have a cascaded effect on other projects, which directly or indirectly depend on its deliverable.

*“We got delayed because we had to wait for a critical UI component from the framework project team”*

Or

*“We didn’t plan for bug fixes for a component which was supposed to be delivered defect free for our usage”*

These are the common scenarios for delays in program which are dependent on other program deliverables.

Parallel programs may affect the schedule of your program in a different way as well. Sometimes, management change the priority of the programs running in parallel. If your project is considered as a low priority one then there might be lack of resources assigned to your project that may result in schedule slippage.

### **Beta releases:**

How many times do we seek feedback on our product during development? And how often do we allocate time for it? It’s important to plan beta releases if we desire to have our product validated by expert users or lighthouse customers during development. Getting feedback from beta customers becomes important especially when their requirements echo that of a mass customer base. The process of giving workable releases to customers, collecting their experience, having their feedback analyzed, and then incorporating in the final product version takes significant time.

### **Risk mitigation and plan B:**

Every project will have some or the other risks. These risks can be of varying severity and probabilities. While making the project plan, it is important to treat the risk individually based on their severity and probability of occurrence. If risks of both high probable and higher severity are not planned for with their own mitigation plan (or plan B), they will have a huge impact on schedule deviation from planned.

As in one of the previous examples quoted, getting a DUT on time for validation was a risk. Had there been a mitigate plan (plan B) like – Validate with other DUT or if DUT is not available here, let one developer travel to customer’s place and finish the validation on time, the schedule slippage would have been avoided.

## **2) People:**

Ultimately, projects are executed by people who may not be skilled or talented. Hence, looking for perfection in projects involving human beings may not be a feasible thought. Certain unpredictable and hence unavoidable issues under this category are:

### **Poor leadership:**

Before thinking of project execution, it is project planning that actually sets the platform for success. Execution of the project depends on its team while planning is taken care by the project leader. The project leader is expected to have enough

technical know-how to understand the project goals and the details of the tasks involved. Poor leadership and superficial knowledge of assignments often results in invalid effort estimation and ad hoc task delegation causing stress and possible delay in project execution.

People leading the team are also responsible for keeping the team spirit high and motivation level upbeat. Poor personal commitment due to lack of motivation results in loss of productivity and may cause the schedule to slip.

Another reason that adds up to delays in projects is inability of the leadership team to track the schedule progress and take corrective action.

#### **Attrition:**

If the project duration is long and the job market is hot, it may be difficult to retain people in the project till its completion. Attrition may further delay the completion especially if the person leaving the job was on the critical path. A person leaving the organization would leave a gap in the project that a new person may not fill immediately, which in turn causes a sudden reduction in the task force.

#### **Learning curve:**

Whenever a new person or team member is included in the project, he or she may require some time to understand the project and to keep pace with other members. A learning curve is needed for new team members joining the team either due to attrition or due to any specific technical competency requirement.

#### **Context switching:**

In smaller organization or groups where people work on multiple projects simultaneously, it is important to have some buffer for context switching. A person planned to work in project 'A' for two hours after a gap of two weeks, would take more than scheduled time to complete that task. A gap of two weeks and the fact that he or she was involved in other project would require some time for the member to get back to the context of current project.

#### **Global development teams:**

In an era of globalization and outsourcing, it is common these days to have a development team distributed over different geographical regions. The project plan needs to account for different time zones and working culture. You might expect an input for your task on Monday morning your time, but it may be Sunday late evening for that person and finally when the input arrives, you might be on your way home after work.

Sometimes schedule estimation might go completely wrong if you have not understood the work culture of the region your teammate belongs to –

*“In my previous work, I was given a task to be completed with a heads up that its very critical task and needs immediate attention’. When I asked my project lead how many days/hours I have for it, I had been time for 2 weeks for high priority and ‘immediate-attention’ work.”*

Definition of ‘urgent’, ‘high priority tasks’ changes with culture and region.

#### **Communication Issues:**

People communicate differently. If important issues are not brought to the notice of the team members, or are not escalated in time, the entire project may suffer. Often

fear of embarrassment stops team members from reporting issues faced during execution. This might lead to more time being spent on that task than could easily have been spent on providing additional help.

### **3) Customer Involvement:**

These issues are quite serious if the customers or end users of the product are involved in the development phase. Understanding customer's priorities, defining your expectation from their involvement needs to be clear and in agreement between both parties.

#### **Expert user testing:**

In the beginning of the project, the expert user testing cycle needs to be planned. The process of giving builds or releases for testing and collecting their feedback, analyzing and incorporating them in your product takes significant time which, if not planned, can delay your program.

#### **Timely feedback:**

*"I got feedback from customers for features, delivered in development milestone-1, after milestone-5 towards the release. These feedbacks are critical but now I am worried how to incorporate them without affecting the schedule."*

It sounds like a common problem. Incorporation of feedback from customers needs to be planned well by getting a commitment from the customer.

#### **Product requirement specification review:**

Having a product requirement review planned and executed will keep you on the right track throughout the project. Reviewing the requirement specification will avoid requirement related defects fixing which otherwise would have delayed, the project.

### **4) Ambiguous Project Requirement:**

For any project to be initiated, the first thing is to have requirements for it. In the product development life cycle, the requirement phase acts like a foundation. Clear requirement or vision for the project navigates the team to success. However, requirements may not be clear at the time of estimation and may result in delay in the project completion.

Issues related include:

#### **Evolving specs:**

If you are making a product based on a standard which is not yet matured or still evolving, you are more prone to have this risk. Frequency changes in the specs will change the requirement for the project during different stages of product development and the team will continue to work on something that is not yet evolved. This results in rework that will delay the project if time for dealing with these changes is not accommodated in the schedule.

*"We developed an algorithm and hence measurement that was based on certain industry standard. Towards the release of the product, the specs changed and our measurement was no longer valid. We had to redo the algorithm to reflect the changes in the specs. This caused our product release to be delayed by 2 months."*

### **New requirements:**

Sometimes new requirements are added as the project evolves towards completion. Implementation of new requirements is not planned at the beginning of the project and hence is not accounted for in the schedule. Adding new feature without revising the schedule may result in delay.

### **Untold expectation:**

Requirements from the customers may be of two types – implicit or explicit. It is important to have the requirements well documented. Implicit requirements needs to be better defined and documented to avoid any confusion towards the end of the project. Customers may not describe their requirements related to system performance, memory issues, user interface quality and usability but they are very keen on providing feedback in those aspects once the product is given for expert user testing. If we are not clear about such requirements, our design might not address them. Addressing them towards the end of the project may call for design changes and extra work that would delay the project.

## **5) Unplanned Tasks / Reworks:**

The Bounding box for the project is set by higher management and often lacks a buffer for unplanned task(s). Having more unplanned tasks that creep up at different phases of project can cause schedule slippage.

The unplanned tasks or rework may arise due to:

### **Sustaining work:**

In smaller organizations, some of the project team may also be responsible for sustaining / customer support of existing products. These unplanned tasks, which come on an event basis, related to customer support are always of high priority. Excess or prolonged sustaining work may take resource out of the planned project causing a potential threat for schedule slippage.

### **Defect fixes:**

Defects are bad as they degrade the product quality and consume extra time/effort to fix them. It is good to have testing of the intermediate releases of the project to find and fix defects sooner in the development life cycle. If the fixing-cycle for such internal-milestone defects is not planned, then either the project is either going to slip or product is going to be of poorer quality.

Poor programming skill of the team, not adapting to modern programming practices and having ad hoc development processes may lead to higher number of defects which would take more time to fix than planned, and cause slippage.

### **Task spillover from previous milestone:**

Tasks that are not completed by previous milestone, due to whatever reason (inefficiency, vacation of the team member, resource crunch etc), will have to be completed by the next milestone thereby increasing the load on the team. If adequate buffer is not planned, these tasks spilled from previous milestone over to next, can delay the project.



### **Requirement change / refinement:**

Requirement changes during the product development will result in rework of what has been previously done with first version of requirement(s). Addressing changes in the requirements needs extra time and effort and may cause schedule slippage.

In some cases, the requirement from the customer is misunderstood resulting in wrong system design and implementation. Additional, unplanned time is lost in correcting the design/implementation which causes schedule slippage.

### **Conclusion**

On time delivery is the challenge software development companies are facing globally. To have complete control over the estimated schedule, it is very important to identify the elements in the development cycle that cause schedule slippage. This article uncovers and explains the root causes of delay in programs using examples from the real world. Having an insight to the root causes will help program managers make good decisions to avoid future schedule slippage.

### **Acknowledgements**

The authors would like to thank the team comprising of Nilashish, Manisha, George, Krishna, Ashish and Mukesh who participated in brainstorming session and generated significant points for this article. We also want to thank Hamsa and Prasanth for their inputs/examples.

### **The Authors**

#### **Madhusudan Acharya:**

For the past 4 years, Madhusudan has been working at Tektronix, India as a Senior Software Engineer for serial communication (Both low and high speed) advanced measurement solutions. His current role includes algorithm development for advanced measurements solution (which runs on window based oscilloscopes) in the area of HDMI standard.

Madhusudan has been awarded with “President’s Award” in April 2007 for his valuable contribution at Tektronix.

He holds a bachelor degree in Electronics and Telecommunication Engineering and a Masters degree in Electronics.

#### **Mukesh Soni:**

Mukesh Soni has more than eight years experience in different phases of hardware and software product design – including research, development, support and maintenance. He has worked for Bosch, General Electric (GE Global Research Center, Bangalore India) and L&T EmSys (Medical R&D) and is now employed by Tektronix.

He holds a diploma in Industrial Electronics (Instrumentation), bachelor degree in Electronics and Telecommunication Engineering. He completed his Masters in Biomedical Engineering from Indian Institute of Technology (IIT), Bombay.

Mukesh holds two patents (USPTO approved) while working at GE. He was awarded for his contributions at GE and Bosch with management awards. He also won several recognitions at Tektronix as well. Mukesh is a GE-certified Six Sigma Green Belt, and has co-authored three books in the area of electrical and electronics engineering.

Project Perfect is a project management software consulting and training organisation based in Sydney Australia. Their focus is to provide creative yet pragmatic solutions to Project Management issues.

Project Perfect sell “Project Administrator” software, which is a tool to assist organisations better manage project risks, issues, budgets, scope, documentation planning and scheduling. They also created a technique for gathering requirements called “Method H”™, and sell software to support the technique. For more information on Project tools or Project Management visit [www.projectperfect.com.au](http://www.projectperfect.com.au)