

How to Manage an FDA Controlled Software Project

Tips, Techniques and Best Practices

As a senior engineering manager, you are already keenly aware that people are the single most significant factor in project success or failure. You can assign the most challenging project to your best software project leader, give her a highly experienced, brilliant team of engineers and you can sleep at night knowing this team will produce great results for you.

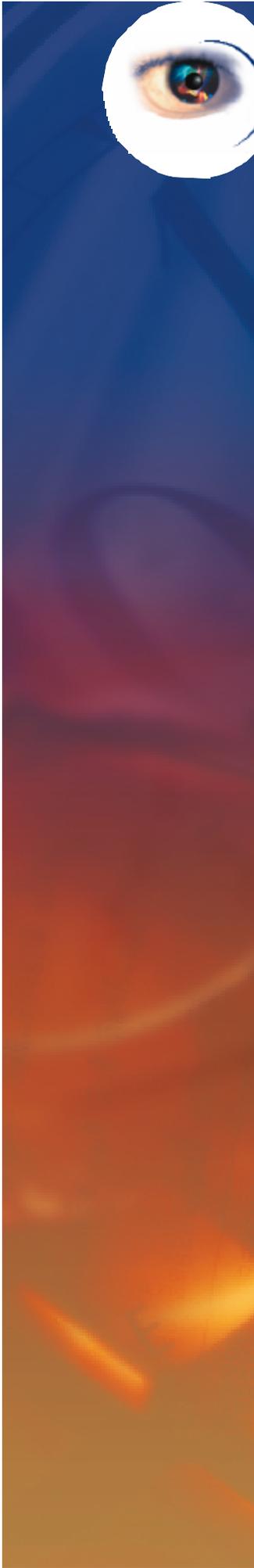
If only all our software teams were like this. Every senior engineering manager probably dreams of having an entire department of super star project leads and brilliant, experienced, highly disciplined software and QA engineers. For most managers, the reality is you have a range of experience, talent, and personalities. These are the resources you have to work with and produce high quality results.

A medical device software consulting firm such as Full Spectrum Software has the luxury, or one could argue the responsibility, of only hiring super star talent. That talent and experience is, in the final analysis, the only reason to hire a consulting firm. The client is paying precisely for the ability to use this type of talent.

Having developed software for many different medical device makers gives us a unique perspective on many of the different management styles, organizational structures and people with whom we have collaborated with over the years. It is from this experience that we can say confidently, what works well, what doesn't and what is the recipe for a total disaster from a project management perspective.

In this white paper we will share what we have learned that can assist senior engineering managers in helping them to structure their teams most effectively and review best practices. In addition, we can identify what are some of the elements of truly effective project management that are applicable to every organization. The paper is structured from the perspective of working with an outside team of consultants, but many of the same rules apply to managing an internal team of software and QA engineers.

Some of the lessons learned are difficult, even painful, to implement - particularly when it comes to dealing with people. We are human beings and we all have feelings and sometimes it is necessary to hurt someone's feeling in order to achieve project success. However, if done correctly and with great tact, those negative feelings can be minimized and in the best case, turned into a positive experience.



Full Spectrum Software

Ultimately, the following five topics are the most important elements that represent the keys to success and we will explore each one in a practical, non-theoretical manner in an effort to provide maximum value to our readers.

- Technical Aptitude
- Communications
- Attitude
- Methodology
- Experience and Organizational Structure

Technical Aptitude

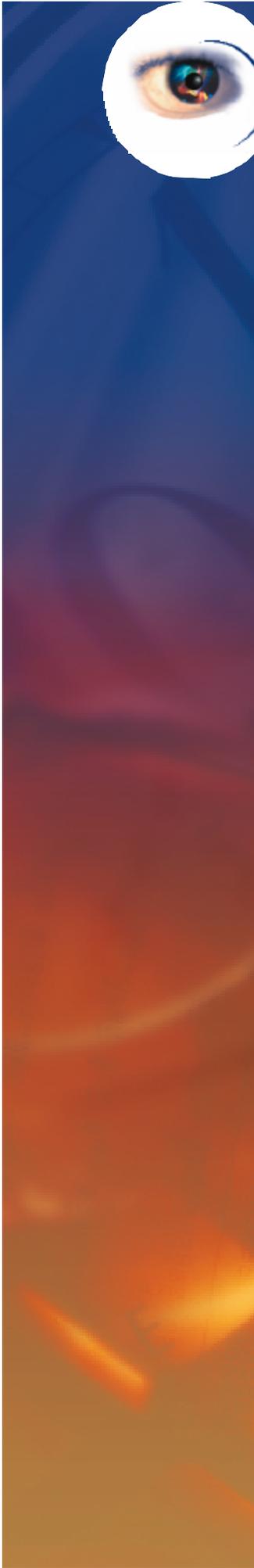
Fundamentally, the team must be able to meet the technical challenges of the project. FDA controlled projects necessitate the use of certified, trained and skilled personnel. While there may be positions within an organization for junior software engineers just out of college, junior engineers should not be contributing directly to production code as this can ultimately result in risk to a patient. Similarly, Software Quality Assurance Engineering tasks should rarely be performed by “testers” who are simply following written test protocols. A “tester” may observe what is indicated as correct behavior but may miss the subtleties of malfunction. Software engineering is a constantly evolving science.

Talented engineers should be capable of learning new tools, languages and techniques. Projects need not be populated in totality with “gurus” but the team should have experience in design patterns and core technologies that will allow them to adapt to new facilities.

Communications

Clear communication within the client organization, with the consulting firm and within the consulting firm’s software engineering and QA team are crucial for success. The client must have a team that clearly understands every component of the system and exactly how it should function. Every member of that team should share the same vision for the system or product. One obvious way to do this is through the process of writing the specifications either internally or in collaboration with the software consulting firm.

However, that does not guarantee that the entire team shares the same vision for the product. Vision is the ability to think through the evolution of the product well in to the future. What features will need to be added to stay competitive? What other applications and enhancements could be supported?



Full Spectrum Software

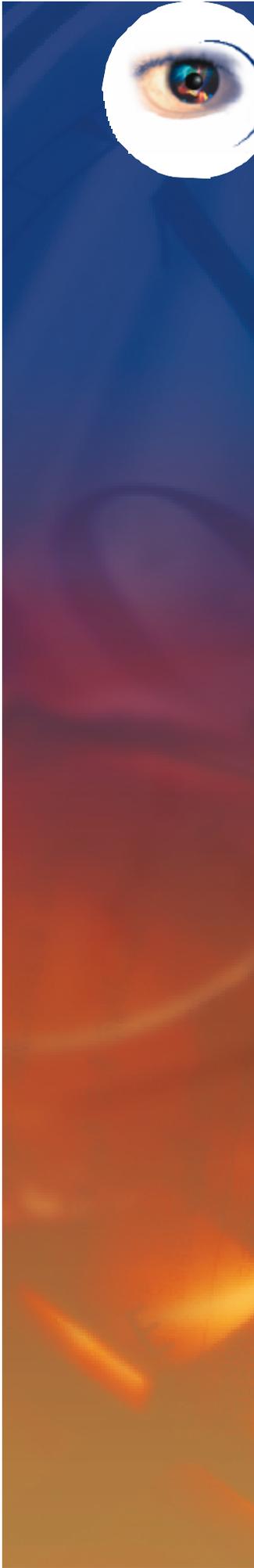
Perhaps one of the most important technical questions, is the systems architecture designed in such a way that it can easily support the changes envisioned by the product manager or team leader? Certainly the system has to meet the marketing requirements that customers are asking for today. Real vision meets those requirements and takes into account the flexibility for what customers might want in the future but may not have asked for or may not even have thought of yet.

The vision for the product needs to be articulated by the product manager and technical team leader first in informal brainstorming sessions and later in formal meetings. The team leader needs to listen to her team and be open to accepting new ideas or different strategies. Brainstorming sessions should be conducted in a fashion such that everyone, even the shyest engineer on the team feels comfortable opening up and contributing to the vision. Those meetings help to refine the vision and insure that the entire team “buys in” to the vision.

This vision also needs to be shared with the software consulting firm. They can also contribute valuable ideas. In fact, a highly skilled and experienced software firm should be able to help drive the specification process and contribute to the technical vision for the product.

There is one last point regarding specifications. Many engineering managers feel that they must develop exquisitely detailed specifications in order for a consulting firm to provide an accurate proposal.

Some managers feel that this is a downside to outsourcing, that they must spend additional (and highly valuable) time developing detailed software specifications, whereas their in-house teams could work from looser specifications. While this is true for off-shore development companies, US-based software consulting firms who work on FDA controlled software are also experts in developing technical specifications. Therefore, all that's needed as a starting point is a solid Software Requirements Specification, and the willingness to collaborate with the software consulting firm to develop the final specifications including a Functional Specification. This approach can actually significantly accelerate the process of developing the required, detailed specifications. Time invested in this approach is offset by the more intimate understanding of the system that the software firm gains by participating in the specification development as well as new concepts and ideas that come from an experienced team of medical product developers.



Full Spectrum Software

Communications between Client and Software Consulting Firm

In any consulting engagement, there must always be a designated single point of contact on each side. This insures that no conflicting information is allowed to pass in either direction. All telephone conversations should be immediately summarized in an email, so there is effectively an “audit trail” on all technical decisions. A weekly status should be provided in writing by your consulting firm.

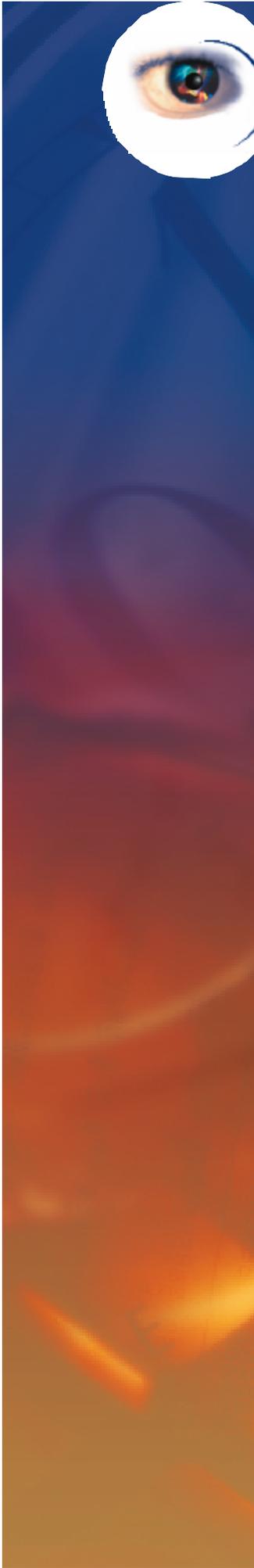
This should include progress from the previous week, expected work for the following week and new risk items that have been identified along with risk mitigation plans. There should also be an agreed upon plan for review meetings. These meetings should consider the project as a whole along with recent project status reports. They can be ad hoc, formal or based on certain milestones, but in no event should review meetings happen less frequently than once per month. Generally speaking, a milestone delivery accompanies a monthly review meeting.

Depending on the size, scope and complexity, one formal review meeting per month is generally sufficient. The purpose of a review meeting is to determine progress, identify problem issues and their resolution and agree upon the goals and objectives for the next review meeting. Multiple stakeholders from both companies can participate in these meetings, allowing for input and collaboration. All decisions taken during such meetings must be summarized by the consulting firm’s project leader and emailed to the client’s project leader. The client must acknowledge and agree to the meeting summary and decisions via email or alternatively provide clarification or modifications.

It is also incumbent upon the client to assign a technical contact with very significant technical expertise in the project domain and the ability to clearly articulate that expertise and vision. This individual must be capable of making technical decisions that are consistent with the goals of the project.

This is not a job for a marketing person or a product manager (unless they are extremely technical). Projects can fail if the client does not assign a highly experienced engineer or highly technical product manager to be the point of contact. In addition to being highly experienced, they must be both highly articulate and able to make decisions on key technical issues without consulting with their colleagues for weeks on end. This would, of course, be true even for internal software teams.

The software consulting firm must also assign a very similar individual to both lead the project team and have excellent communication skills. This is often less of an issue for a consulting firm, as they tend to have program managers or project leaders who are trained specifically for this role. Lastly, both of these individuals must develop a healthy respect for the other person’s technical skills. In other words, they must be able to develop a productive working relationship. If that does not become apparent over the first month,



Full Spectrum Software

one of the two individuals must be replaced. If the two technical leaders cannot work with each other, or cannot communicate effectively, the project is almost certainly going to fail.

Generally the consulting firm will attempt to re-assign their project leader. However, if the client has assigned an inappropriate contact person on their side, it is the particularly awkward duty of the consulting firm to request that a different individual be put in charge of the project. This is not only awkward, it is sometimes seemingly untenable for the consulting firm to ask for. The client, after all, is the client. However, a really top notch software engineering consulting firm will get their executive team involved and meet with the client's upper management with an un-biased and clearly documented list of reasons that mandate a change of their technical lead or risk project failure.

Communications within the Software Consulting Firm

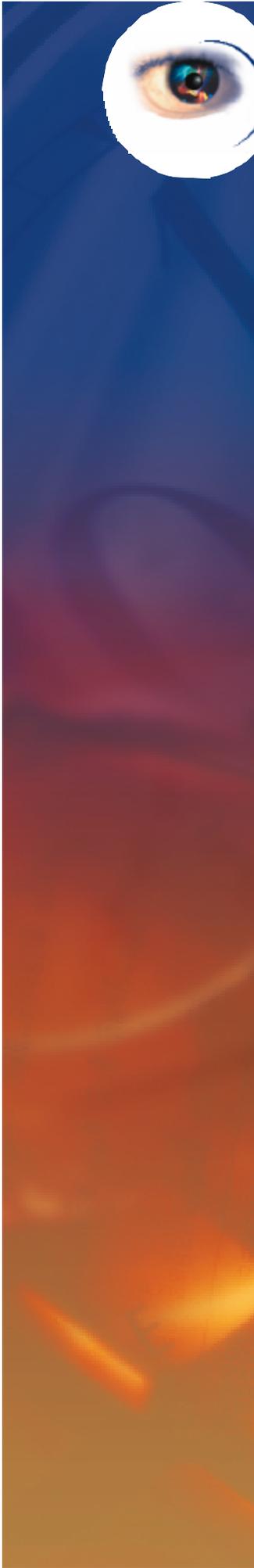
This is, or should be, somewhat of a given. A highly skilled and experienced software consulting firm will have internal systems for team communications as well as systems for collaboration and communications with their clients. Much of this should be documented in their Quality System. However, if the software firm has not actively participated in the specifications development, there is a chance that specifications can be misinterpreted and that can lead to problems in development later in the project.

In order to avoid misinterpretations, a specification review should be conducted. In this meeting the QA teams from both parties should participate. This is a quality review and QA must have visibility into this process. There can be variations on this process, but for each item you should ask the questions "Is there anything in this task/feature that is ambiguous? Do both parties understand exactly what this task means? Can it be verified and mapped to a QA traceability matrix?"

It is important to note that for a requirement to be verifiable, the QA team must be able to describe how it will test the feature, what will the test criteria or protocol be and how to define success and failure of the requirement. If any of these questions cannot be answered, repeat steps one and two. If QA can't describe how they will test a feature, it is not clearly defined. This is another process that should be executed internally, even if there are no outside parties involved with the project.

Attitude

While this is an intangible, it can impact the success or failure of a project. Since it can't be empirically measured, it needs to be monitored. While it is most important for the project teams, it also applies to management and upper management. In the ideal world, everyone from the CEO on down would have boundless enthusiasm for the project, be truly excited



Full Spectrum Software

about collaborating with other team members and be highly motivated to produce a truly innovative, superbly high quality system. Unfortunately, we don't live in an ideal world.

Sometimes executive management sees software development as a necessary and expensive evil. Senior managers are stressed out or burned out and only care about meeting budget and project team members sometimes would prefer to be working on anything but the current project. The reality is that it's usually somewhere in between.

No person or consulting firm is perfect. Maintaining a positive attitude, when things go off track, is a challenge. However, team leaders need to view problems as simply that; a problem that requires a solution. Getting emotional or pointing fingers does not solve problems.

Asking and probing for solutions does solve problems. This can involve assembling the entire technical team, holding a meeting or series of meetings, using the open brain storming technique discussed earlier and attempting to brain storm a solution.

If there is a software consulting firm that is leading the major aspects of the software project, it's very tempting to simply blame them and use that as an explanation to management. However, that also does not solve problems.

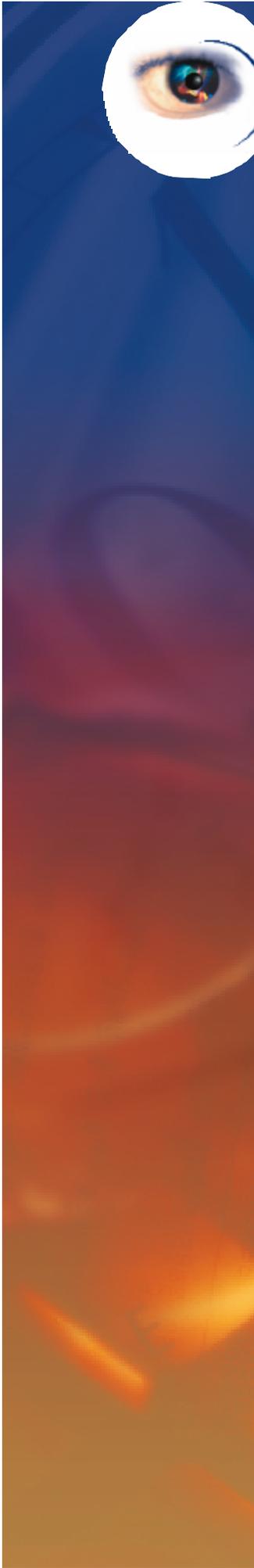
A good consulting firm will employ a variety of techniques to solve a technical problem. Some common techniques include adding a systems architect to the team, or adding another systems architect, dedicating a senior resource to researching the problem and holding internal round-table meetings with their most senior software developers and systems architects to see if they can identify a solution.

Solving the problem however, requires a positive, collaborative attitude. Rather than allow the consulting firm to spin their wheels and burn budget, it is far more constructive to meet with them and try to solve the blocking issues. If there are internal resources, such as domain experts or senior software engineers with related experience that could be of assistance in solving complex problems, it may be extremely helpful to request their assistance. This type of positive attitude and the willingness to tap into corporate expertise will almost always lead to a better, if not a truly good outcome.

Note the caveat "almost always." If the blocking issue is determined to be that the consulting firm has completely underestimated a set of tasks, then a different type of meeting needs to be called.

For the purposes of discussion, assume that the consulting firm has made an honest mistake and honestly wants to remedy it (bear in mind that someone on the client side signed off on those estimates). They should be open to a some form of financial concession.

This discussion needs to be handled in a delicate fashion and with the right attitude. The right attitude is that while the consulting firm has made a mistake, you still want a



Full Spectrum Software

successful outcome. Demanding huge price cuts may be tempting and may even feel justified. However, if your real goal is to have as successful an outcome as possible, then this type of financial punishment may result in the consulting firm developing a very negative attitude towards your project and impacting the value of your end product.

Since a positive attitude is one of the key ingredients for success, this is a highly undesirable situation. Best practices in vendor management would be a negotiated agreement such that the consulting firm reduces their hourly rate preferably to cost until the underestimated task(s) are completed. The client receives a price reduction and the consulting firm is now exceptionally motivated to complete the underestimated task(s) as fast as they possibly can. A good consulting firm will put in extra hours that will not be billed to the client for several reasons. They know that the client has treated them extremely fairly, even though they are taking a financial hit. They also know that the faster they get the tasks completed, the faster they can return to making a profit, and therefore working some non-billable nights and weekends is in their best interest.

Methodology

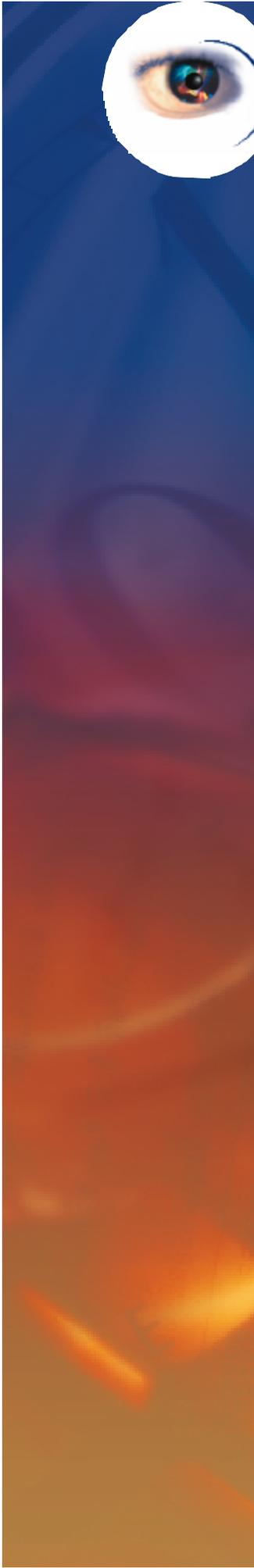
As anyone who has been involved in FDA controlled software development knows, methodology in both QA and development is crucial. Since we assume the reader knows the FDA guidelines for software development and QA, it's not necessary to review them here. It is, however, worth noting that an important element of surviving an FDA software audit is documented traceability. In the course of updating software Quality Systems, it is prudent to pay close attention to traceability in all of the documentation.

From a software development perspective, many organizations are either using or considering the use of Agile development. Agile development in its pure form, which is sometimes called "capital A" Agile, was never designed for FDA controlled software development.

The Agile Manifesto was published in 2001 is considered by many in the Agile community to be the ultimate definition of Agile development and accompanying Agile principles. The Agile Manifesto states:

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan



Full Spectrum Software

That is, while there is value in the items on the right, we value the items on the left more.”

Agile development was created in response to the traditional “Waterfall” methodology which was perceived as too slow and cumbersome.

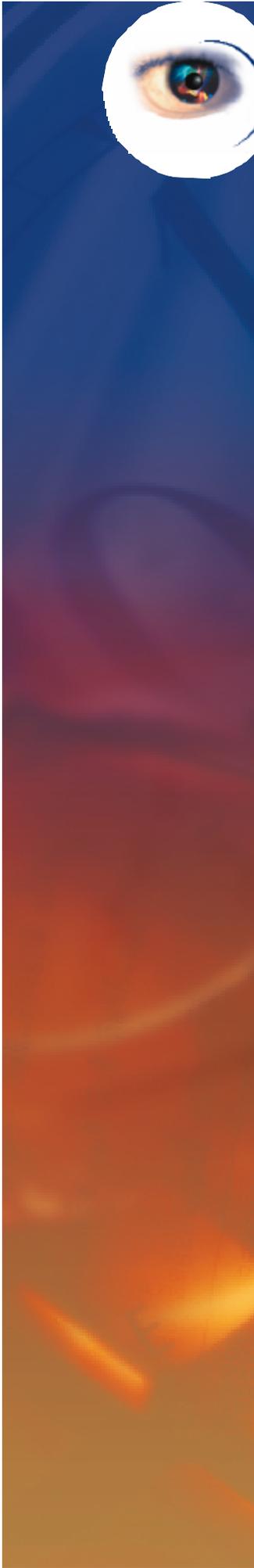
Even though the preceding statement from the Agile Manifesto would seem to directly contradict some of the characteristics of an FDA controlled software project, there are some aspects of Agile development that can be adapted to benefit an FDA controlled project. This is sometimes referred to as “small A” Agile development.

Inherent in the Agile methodology is a concept of “lean” development or “test driven” development. This is the notion that a highly experienced software engineer can take a small set of specifications and rapidly develop a small module of very high quality code through a process of continuously testing and documenting their own code. This code is then passed to the QA team who run the code against their documented QA Test Plan. This is the FDA extension to Agile. This is just one of example of how Agile methodology can be modified to be useful in an FDA controlled software project.

It’s also important to note that no two companies ever implement an Agile development methodology in exactly the same way. That’s because the adoption of Agile development is entirely dependent on the skill and experience of the software development and QA teams. Highly experienced software engineers don’t require nor appreciate a high degree of micro-management. This allows teams to meet frequently but informally and empowers them to very rapidly resolve technical issues. Highly skilled software engineers not only test their code continuously, they will use automated tools such as static analysis and run-time analysis, to enhance their own manual and unit testing.

Modified Agile development can speed up development in an FDA controlled project, but as previously noted it only works well with a highly experienced and highly skilled team. Any organization considering adopting Agile development should proceed at a measured pace of adoption and carefully analyze what techniques work with their own staff and increases productivity without impacting safety or quality. There are two approaches to adopting a modified Agile development methodology. One approach is almost always successful, the other approach is almost always a disaster.

The most successful way to introduce Agile development is to work with a software consulting firm that is already successfully employing a modified Agile and is highly experienced in the methodology. Be clear with the consulting firm that one of the goals of the project is to introduce Agile development into the organization. In order to facilitate the knowledge transfer it is helpful to assign one or more of the most highly respected software and QA engineers in the client organization to work collaboratively with the consulting firm. Those individuals will be able to recognize which aspects of Agile will be most useful to the client organization. As highly respected software and QA engineers, they can more



Full Spectrum Software

effectively lead the changes in methodology due to their stature in the organization.

The other approach is to engage an Agile training firm to develop a full training program. The typical Agile training firm, with very few exceptions, are not experts with FDA controlled software development. Their expertise is generally in “capital A” Agile. The typical training program involves about 2 weeks of class room time for the entire software team. If you have a team of 25 engineers, two weeks of classroom training eliminates about 1 full person year from your team’s overall productivity. The class room training is usually followed by approximately 4 to 8 weeks of active support on a part time or full time basis. As previously noted Agile was never designed for FDA controlled software projects, so it will be up to the team leaders and other senior engineers to determine what to adopt and what to modify. This process will largely be by trial and error, a costly and potentially highly inefficient way to adopt a new development methodology.

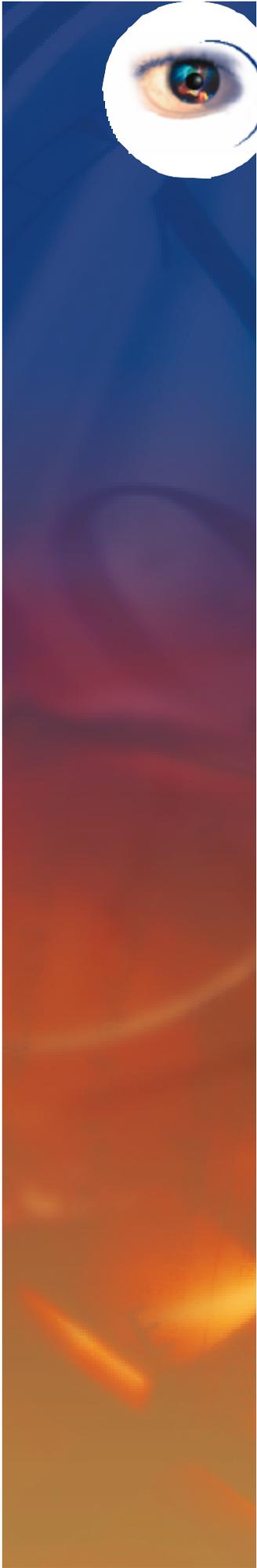
In the first approach, learning by doing is a highly efficient and productive way to introduce a new methodology. Formal training is time consuming and in some cases is actively resisted by some team members who may be far more comfortable with a different methodology, particularly if they have been trained in the traditional “Waterfall” methodology.

Experience and Organizational Structure

Optimizing Team Structure

Most software organizations have a mix of highly experienced and more junior engineers. High performing software teams are motivated to achieve a highly collaborative environment where effective, non-judgmental and open, frequent communications are highly valued. This motivation comes directly from the leader(s) of the software organization. Based on our experience, most experienced software engineering managers know how to balance effective communications against task oriented goals. Some degree of formal meeting time and informal meeting time enhances productivity. Too many meetings negatively impacts productivity. Having too few meetings can lead to time and cost overruns due to lack of understanding of project goals.

This is a delicate balancing act. In order to determine if you are achieving the correct balance, one good data point is polling the team regularly asking if they feel they are meeting frequently enough or not enough. Since software and QA engineers have widely varying opinions on meetings, this is not the sole data point. However, when specifically asked to express their opinions good engineers will at least offer an honest opinion. This data combined with “actual” versus “estimated” time to completion data, helps software managers achieve the optimal balance.



Full Spectrum Software

One technique for optimizing the performance of more junior engineers is to structure a formal mentoring program. Pairing a single very senior software engineer with a junior engineer can dramatically increase the productivity of the junior engineers. Having a defined mentor promotes a relationship to develop between the two and can also build a sense of “pride of ownership” for the senior engineer. Here again, there is a delicate balancing act. In theory, one could assign several relatively complex sets of tasks to a junior engineers based on the notion that a very senior engineer is there to guide them through the tasks. This approach has the effect of limiting the productivity of the highly valued senior software engineer, as they spend more time “mentoring” than driving team productivity. Note that not all senior engineers are great teachers or mentors, although most will give it their best effort.

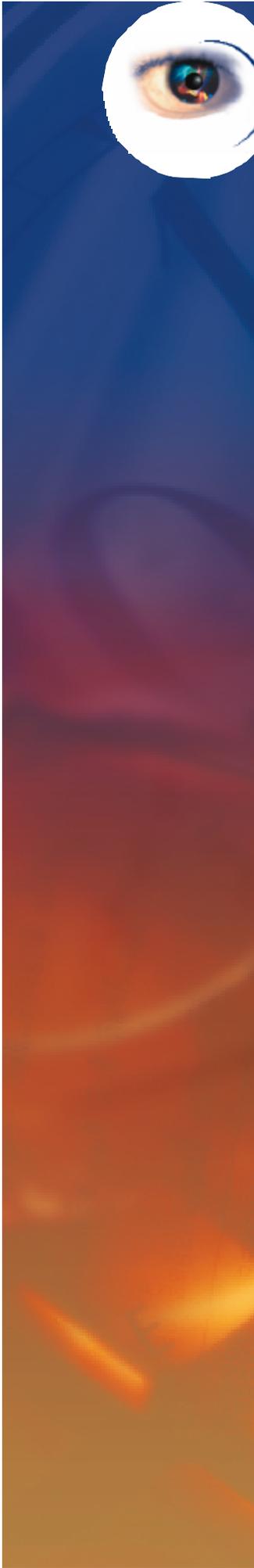
Generally a good trade-off is to continue to assign junior engineers simpler tasks with the senior engineer spending some amount of her time reviewing the code and doing real mentoring. In addition, the junior engineer should be assigned at least one or perhaps two tasks that are clearly challenging for their experience level. These tasks should be called out in the project plan and noted that the junior engineer should not attempt coding these features before a mentoring session has been scheduled with her mentor. All code produced by junior engineers should be code reviewed for accuracy, stability and good coding practices.

Promoting Technical Staff

It's always tempting to promote your best software engineer to team leader, based on their outstanding productivity and their technical accomplishments. However, unless they possess a natural ability to communicate effectively with both technical and non-technical people and are comfortable presenting to a small group, as well as other natural leadership skills, it may be worthwhile to take a few common sense steps.

The most obvious one is to determine if the engineer really does want to be a team leader. It's surprising how many software engineering managers loved being a senior software engineer or systems architect and in contrast are utterly miserable being a project leader. From a best practices perspective, it's ideal to have a management structure such that senior engineers can remain on a technical track and continue to be promoted and financially rewarded. Xerox pioneered this management structure back in the 1980's and this contributed to the stunning number of software and hardware breakthroughs achieved by the company. Unfortunately, in many engineering organizations the only way to promote and financially reward a highly productive engineer is to move her into a management role.

Once you have promoted an engineer to a management position, you must continue to monitor her progress to ensure success. The obvious litmus test for successful project



Full Spectrum Software

management is the on-time deliver of a software product. However, it is important to interview the manager's team members to see how she might improve. This can be an awkward conversation for individual contributors to have with senior management. An anonymous report gathered by a team member uninvolved in the project can be a solution to this awkwardness. The feedback to look for from the team relate to technical leadership, communication, leadership style and ability to resolve problems and interpersonal friction. Make sure to gather both positive and negative feedback in order to facilitate a constructive review between senior management and the project leader.

It's sometimes difficult to understand all of the responsibilities of a project leader from the perspective of a senior team member. Therefore it is important for any senior technical person transitioning to a management role to understand all of the responsibilities of a team or project leader. Prior to promoting someone to a technical leadership role, they should understand exactly what their new responsibilities will entail. This should be explained to them in great detail. If they are not entirely comfortable with any aspect of their new role, they can be teamed with a highly successful and experienced team leader in a mentoring relationship. Having a seasoned team leader as a mentor can make the transition smoother and more rewarding for the newly promoted leader.

Conclusion

We hope you found some tips or techniques that will be beneficial to your organization. Full Spectrum Software will be hosting a series of webinars to explore each of these topics in greater detail in the near future. If you would like to be notified of upcoming webinars, please send an email to ClientServices@FullSpectrumSoftware.com

Full Spectrum Software is a 14 year old consulting firm specializing in the development of embedded and applications software for medical devices. The company has helped their clients deliver over 100 commercial products and systems to market. Andrew Dallas, the firm's CTO, is widely considered one of the leading authorities on best practices in FDA controlled software projects. Andrew has published extensively in major trade and technical publications.

Contact Ken Carson, ClientServices@FullSpectrumSoftware.com

(508) 620-6400

225 Turnpike Road
Southborough, MA 01772

www.FullSpectrumSoftware.com