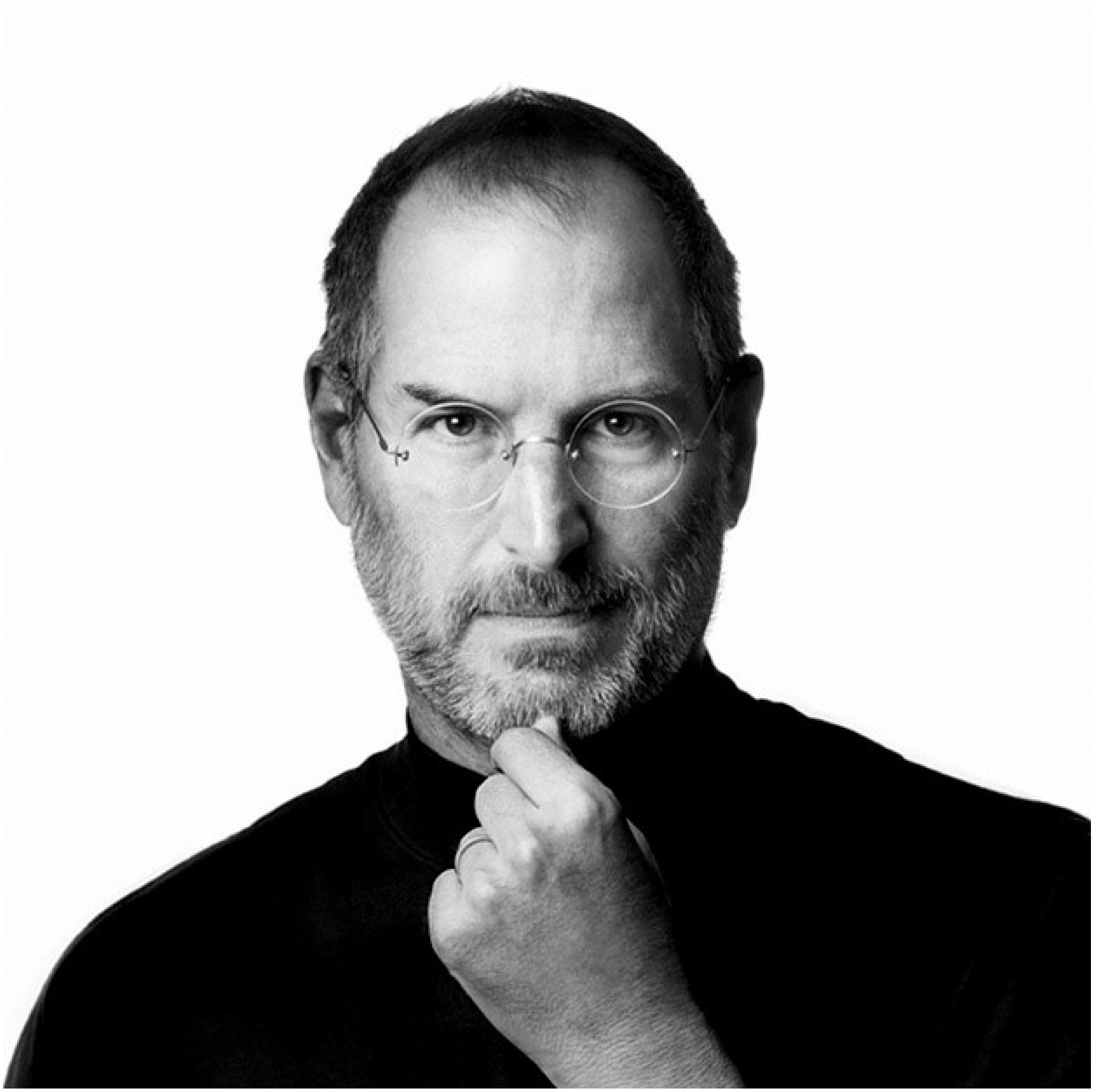
CPSC 546 Modern Software Management

Prof: Lidia Morrison

Final Exam Summary Report

In Memory of One of The Finest Agile Leaders Ever: Steve Jobs



Chapter 1

Change is a vital angle in bringing some better approaches to build up an item. Everyone needs something new something creative and financially savvy. Influencing an item to savvy and same time to get some development that makes a few issues and expands time for the two specialists and generation group to adapt up to the due dates. Nevertheless, progression and speedier change aren't adequate. Companies need to pass on better things furnished to what customers require at the period of shipment, which might take after what the gathering conjectured they required when the endeavor was begun. Associations that can quickly and monetarily propel a thing closest to the complete of the change lifecycle will have a huge high ground.

There are five key business objectives for a good exploration process such as Agile Project Management:
Continuous Innovation: *To deliver on current customer requirements.*
Product Adaptability: *to deliver on future customer requirements*
Improved Time-to-Market: *to meet market windows and improve return on investment*
People and Process Adaptability: *to respond rapidly to product and business change*
Reliable Results: *to support business growth and profitability*

The author of this book, Mr. Jim Highsmith III explains a real-world case and proposes two definitions of agility. In a questionable and turbulent world, achievement has a place with organizations that can make change, and possibly bedlam, for their rivals. Making change disturbs competitors reacting to change prepares for aggressive pushes. Making change requires advancement: growing new products, making new deals channels, diminishing item improvement time, redoing products for progressively littler market portions.

Jim says agility is more an attitude of how you handle the team and product. Visionary companies distinguish their timeless core values and enduring purpose. He says such companies should retain their core values with adopting Agile as a form of Servant Leadership which ultimately benefits the company in tackling many domains such as customer engagement, market adoption, adjusting to change and so on.

The introduction of Agile triangle replacing the traditional iron triangle of project management enables better management of imperatives (scope, schedule, and cost).

Chapter 2

Jim explains the vital aspects while measuring the performance on projects. Money and time are key things when thinking about a project but most important is value for customer. Agile development helps to strike a perfect balance between **Values vs Constraints** with value delivery being the core deliverable.

Customers define capabilities that provide value and the business objectives that assist in quantifying that value. Jim says that nowadays value arises from implementing capabilities as they evolve over the life of a project.

Innovation offers the largest amounts of significant **worth creation** in an organization. Innovation may come in a variety of aspects ranging from new items, plans of action, forms, or new execution activities. Productivity and enhancement are fitting drivers for a generation venture, while innovation and inventiveness should drive an investigation sort venture. Enhancement suggests that we know how to accomplish something yet that we now need to enhance it. Innovation infers that we don't know how to accomplish something, and looking for that information is principal.

As per Jim, the basic venture administration approach is made from three procedures that trade data—arranging, controlling, and executing. Plans are created to fulfill legitimate, administrative, or administration prerequisites rather than being founded on the work that should be expert. To design frequently relates more to the want for control than the necessities of genuine work execution, maybe because the general population arranging the work are not associated with the doing of the work. Control has generally fixated on redress as opposed to learning

Jim quotes an example from Japan where in the lithe development car business initially emerged in Japan inside lean assembling, which started in the car business in Japan in the 1980s. One of the major precepts of lean assembling is the methodical disposal of waste; that is, any movement that doesn't convey an incentive to the client.

Jim talks about iterative delivery citing that the customer doesn’t really know what he wants. Iterative delivery helps him see small prototypes of the final product and the further prototypes are remodeled with the feedback the client gives. This way the focus is about making the right product and meet the client expectations.

To get all these ideals to work, Jim says that Project leaders must be champions of technical excellence; they must support and advocate technical excellence while maintaining a watchful eye on other project objectives. Many software products suffer from technical debt, the accumulation of problems caused by poor quality practices. Few products today, particularly industrial products, are one-shot wonders.

Jim mentions that keeping it simple also means reducing overhead. Agile is first and foremost about delivering value to customers. Compliance activities, at their best, attempt to mitigate the risk of mistakes, fraud, poor performance, and financial overruns. It is important to note that there are good reasons for some compliance activities. Every organization, every company, every team has legitimate compliance issues. For example, in the automotive industry, an enormous volume of documentation must be retained for potential product liability lawsuits and government crash investigations.

Chapter 3

Project leaders should be both managers and leaders, with the importance of the latter escalating rapidly as the exploratory nature of projects increases. Good leadership contributes significantly to project success. Leaders depend for the most part on influence rather than power, and influence derives from respect rather than fear. Respect, in turn, is based on qualities such as integrity, ability, fairness, truthfulness. Leaders are part of the team, and although they are given organizational authority, their real authority isn't delegated top-down but earned bottom-up. Leaderless teams are rudderless teams. Leaders who want to create adaptive, self-organizing teams steer rather than control they influence, nudge, facilitate, teach, recommend, assist, urge, counsel, and, yes, direct in some instances. From the outside, a managed team and a led team can look the same, but from the inside they feel very different. Projects, like organizations, need both leaders and managers. Unfortunately, it is often difficult to find both skill sets in the same person. And since creating a project budget is more tangible than, say, resolving the ambiguity of trying to satisfy conflicting customer needs, project management training tends to focus on tangible practices and tools.

Self-organizing teams form the core of APM. They blend freedom and responsibility, flexibility and structure. In the face of inconsistency and ambiguity, the teams strive to consistently deliver on the product vision within the project constraints. There is a big difference between the terms "self-organizing" and "self-directing." Self-directing usually implies self-led, as various team members assume the leadership role depending upon the situation. The organization, through the project leader or other managers, has the responsibility to staff the project with the right people. Getting the right people means finding those with appropriate technical and behavioral skills. Responsibility and accountability create self-organizing teams that work.

The capability of self-organizing teams lies in collaboration—the interaction and cooperation of two or more people to jointly produce a result. When two engineers scratch out a design on a whiteboard, they are collaborating. When team members meet to brainstorm a design, they are collaborating. The quality of results from any collaboration effort are driven by trust and respect, free flow of information, debate, and active participation—bound together by a participatory decision-making process.

Create and manage the feature/story backlog. Set priorities in release and iteration planning. Identify and define features/stories. Define acceptance criteria. Review and accept completed features and stories. Interact on a continuous basis with the development team.

Although self-organizing is a good term, it has, unfortunately, become confused with anarchy. There are several issues in returning self-organizing to usefulness. First, we need to get away from the idea that agile teams are leaderless or that leadership revolves around the team depending on the situation (this type of situational leadership does occur, and often, but it does not replace a designated leader). There is just too much experience and management literature that shows that good leaders make a big difference. The anarchists want to eliminate leaders and merely go with situational leadership.

Innovation cannot be guaranteed by some deterministic process—innovation is the result of an emergent process, one in which the interaction of individuals with creative ideas results in something new and different. Demos, prototypes, simulations, and models are the catalysts for these clever interactions.

Chapter 4

X amount of time spent in planning helps to deliver 5*Y amount of result. Agile leaders need to be adapting to changes and not stick to baselines created during the very start of project.
Think of the competitive advantage of being able to package features so customers feel they have software specifically customized for them and the cost to maintain the software remains low.

In the last two decades a vanguard of scientists and managers have articulated a profound shift in their view about how organisms and organizations evolve, respond to change, and manage their growth. Scientists' findings about the tipping points of chemical reactions and the "swarm" behavior of ants have given organizational researchers insights into what makes successful companies and successful managers. For an agile project, the *ensemble* includes core team members, customers, suppliers, executives, and other participants who interact with each other in various ways. It is these interactions, and the tacit and explicit information exchanges that occur within them, that project management practices need to facilitate.

Jim says Every project has knowns and unknowns, certainties and uncertainties, and therefore every project must balance planning and adapting. Balancing is required because projects also run the gamut from production-style ones in which uncertainty is low, to exploration-style ones in which uncertainty is high.
A major factor that impacts project management style is the cost of an iteration; ie, the cost of experimenting. Even if the need for innovation is great, high iteration costs may dictate a process with greater anticipatory work. Low-cost iterations, like those mentioned earlier, enable an adaptive style of development in which plans, architectures, and designs evolve concurrently with the actual product.

Legacy code results to technical debt. Agile tackles problems about process and product but upgradation of tech needs a significant time to revitalize. However, failure to address this significant barrier keeps many organizations from realizing their agile potential.

Most, but not all, of the time perceived barriers to change it costs too much really point out inefficiencies—opportunities to streamline the process and enhance the organization's ability to adapt. Agile development demands short-cycle iterations. Doing short-cycle iterations demands finding ways to do repetitive things quickly and inexpensively. Doing things quickly and inexpensively enables teams to respond to changes in ways they never anticipated previously. Doing things quickly and inexpensively fosters innovation because it encourages teams to experiment.
At best, a repeatable process can deliver only what was specified in the beginning. A reliable, emergent process can deliver a better result than anyone ever conceived in the beginning. An emergent process can produce what you wish you had thought about at the start if only you had been smart and prescient enough.

Chapter 6

Jim stated and explain the main four concepts which are,

What is the product vision?
What are the project objectives and its constraints?
Who will be included in the project community?
How will the team deliver the product?

The evolution of a project plan from vision, to scope, to release utilizing three simple but powerful practices: the vision box, the project data sheet, and the release plan (accomplished in the Speculate phase). The vision box exercise forces the team to condense information about a product vision onto two flip chart pages (front and back of a box). The project data sheet forces the team to condense key project scope and constraint information into a single page. The Envision phase defines the beginning of a project for which the kickoff event might be the approval of a feasibility study. Many companies conduct feasibility and marketing studies prior to initiating development projects, whereas others use only brief project requests.

A product vision (defined by a product vision box and elevator test statement) galvanizes members of the product team into focusing their often-disparate views of the product into a concise, visual, and short textual form. These two project artifacts, and the product roadmap, provide a "high concept" of the product for marketers, developers, and managers. Although a preliminary product vision may have been done during a conceptualization phase, few members of the delivery team are usually involved in that process.

Every product needs a marketing theme, a crisp visual image and capability description whose intent is to draw customers into further investigation. In this design the box exercises the project and product teams create a visual image of the product (vision does imply "visual," after all). For software and other small products, the image should be the product package.

The product vision establishes a baseline for what the product team, in conjunction with sponsors and the development team, desire in a product. Product visions outline a somewhat unconstrained goal, but a project also needs to establish objectives and constraints.

A project data sheet (PDS) is the second major envisioning practice in the evolution of a project plan. Although the product vision is an expansive view of what the product could become, the project vision bounds product development with objectives and constraints. Many managers and executives consider project success to be on time, on budget, and on scope. They define each characteristic with no tolerances and then fantasize that the team can respond to all manner of business change without project changes.

An exploration factor acts as a barometer of the uncertainty and risk of a project. Big projects are different from small projects; risky projects are different from low-risk ones. One issue in selecting project management practices and processes is the problem domain in which the team must operate. An exploration factor of 10 indicates a highly exploration-oriented (high-risk) problem domain, and a 1 indicates a very stable (low-risk) problem environment. It is important to identify the various problem domain factors, but it is even more important to tailor processes and practices to the problem and to adjust expectations accordingly.

The product interface differs between a product development effort that has external customers and a project that serves internal customers. Looking at the diagram from left to right, some representative of the end user or customer should be included in a feedback loop to the team, even if that person is not part of the product team.

Process and practice tailoring defines the approach that the team will use to deliver a product. Teams starting with the organization's standard framework and practices, and then tailors the framework and practices to their needs.

Self-organization strategy concentrates on how people work together, how they collaborate, and the mechanisms for that collaboration. Processes and individual practices concentrate on what people do. Although the strategy and process seem to overlap, they are complementary.

Project managers often perpetuate a hierarchical, command-control management philosophy in their project organizations by focusing on organizational charts, detailed process and activity identification, and documentation requirements. Thinking about a self-organizing strategy attempts to break this mindset by having the team focus first on interactions—how everyone associated with the project will play together. The strategy establishes the team's approach to communications, coordination, collaboration, decision making, and other individual-to-individual and team-to-team interactions.

Agile development and project management are built on an underlying premise that individual capability is the cornerstone of success, and furthermore, that individuals are unique contributors. It follows from these premises that rather than molding people to a set of common processes and practices, processes and practices should be molded to the team itself.

Chapter 7

Agile teams plan, but they must keep it real. Plans in this context serve as a vehicle for embracing change. Plans must adapt because the customers' understanding of their requirements changes, because estimates of work effort vary, because people arrive or depart from the team, and for a variety of other reasons. Development generates new information that in turn creates the need to plan again.

There are two crucial components of an iterative planning and development approach short iterative timeboxes and features. For software projects, iterations are generally two to four weeks in length. Hardware projects will have longer iterations and greater variation electronic devices will generally have shorter timeboxes than, say, automobiles. Short iterations act to accelerate projects. When they keep timeframes short, teams must figure out faster ways of accomplishing every aspect of development. Feature-based development is not a software-only technique.

The objective of creating a product backlog is to expand the product vision, through an evolutionary requirements definition process, into a product feature list, or backlog. For existing products, customers, developers, product managers, and customer support staff constantly make suggestions about product enhancements that add to the backlog. This backlog list, is maintained by the product manager and is the major inputs for release, wave, and iteration planning.

In planning a product, however, some items that need to be delivered may not sound at least to customers or product managers as though they provide direct benefit. An interface component deep in the bowels of an electronic instrument may have minimal interest for an end customer but be a necessary "technology domain" feature. For project planning and delivery purposes, teams need to include these technology stories in the plan.

Story cards themselves are important. They provide a mobile, tactile medium that team members can write on, shuffle about on the table, and have conversations around. After data gets entered a formal medium people are less likely to change it. The information on the cards becomes the product of the team's collaborative effort and a focal point for mutual understanding of the product at a detail level. A few key items of information should be recorded on story cards and supplementary documents for details.

In product development, lack of or poor customer involvement in the process has been a recurring problem. Development teams have little contact with customers or analysts. In a well-functioning agile team, collaboration among customers, product specialists, and developers supersedes documentation in importance and improves the entire requirements understanding process.
Three things are key to understanding: the people involved, identifying the functions performed by these personas, and breaking that functionality down into implementable chunk stories.

A release plan presents a roadmap of how the team intends to achieve the product vision within the project objectives and constraints identified in the project data sheet. Agile lifecycles are both iterative and story-driven—a significant change from traditional plans that are waterfall-and task-driven. The story-driven aspect changes the primary focus of planning and executing from tasks to product features. Most traditional project management plans utilize tasks to construct work breakdown structures (WBSs) to organize work.
Agile projects are planned by capabilities and stories, whereas many project leaders may be more familiar with task-based estimating. These leaders must learn to apply their experience in

estimating tasks to stories. Reducing waste, a lean thinking principle, involves looking at activities and eliminating or reducing those that don't directly generate customer value. Estimating can be one of those time-wasting activities. Detailed estimating of items too early in the project lifecycle wastes time because the items may be dropped anyway.

Three key activities need to be completed prior to beginning story development articulating a product vision, defining the project's objectives and constraints, and creating an iterative, story-based release plan. The release plan is constructed from the product feature/story backlog that evolves during the Envision and Speculate phases. After the release plan has been completed, other common project management planning artifacts such as expenditure budgets can be finalized.

While a Speculate phase may sound a little, well, speculative, in actual practices presented in this chapter have proven highly reliable in creating useful planning information early in development. Feature-based planning forces the engineering and product teams to understand the product in ways that task-based planning rarely does. With the preplanning that occurs at the end of iterations and waves, plans and the product evolve as information is gained through experimentation and constant feedback.

Chapter 9

Those investigate stage conveys running, tested, acknowledged stories. APM keeps tabs on the alarmed leaders' assignments from claiming making self-organizing, self-taught groups that could convey An releasable result. What's more cycle chiefs assistance clinched alongside building those task Group.

Alarmed venture pioneers concentrate on including esteem to an undertaking. Unfortunately, huge numbers advancement particular architects think about venture administration with make a barrier. Venture supervisors need aid seen as managers who assemble nitty gritty errand schedules, make bright asset profiles, bug cooperation parts around micro-task completions.

After the overall release plan has been established for the project, the team turns to developing a detail plan for the next (or the first, if it's the beginning of the project) iteration. The team takes each story card from the release plan and identifies a list of the technical and other tasks required to implement the story and records those tasks. The objective in agile planning is to match capacity with plans and not engage in wish based planning.

Many agile teams use story, or story-point velocity number of stories or story points delivered per iteration as a capacity number, but if the number of stories is small then velocity can vary widely. One key reason for short iterations is that repetition results fast learning. Having to do things frequently improves learning, plus high frequency forces teams into learning how to do things quickly.

Many agile practitioners use a task burndown chart, which shows the number of tasks completed each day of the iteration, to monitor progress. I think task burndown charts can hinder self - organization when used by project leaders or iteration managers. If a leader is monitoring tasks daily, it takes away from the team's self-management, although a team might want to use a burndown chart for its own internal use.

The objective of simple design is to keep the engineering team grounded in what is known rather than anticipating the unknown. There are two fundamental approaches to managing change anticipation and adaptation—and good design strategies encompass aspects of both. Anticipation involves planning for the future and predicting what kinds of change are probable.

Every team member gets mired in details and forgets the goal at least periodically. Good project and iteration leaders remind the team about the goals from time to time by revisiting the key constraints and by reinvigorating the group with the ultimate vision and objectives of the project. This is part of encouraging exploration, which might be considered a leader's cheerleader role, but it's a role that must be based rather than fantasy.

Respect comes from understanding other people's roles on a project. Engineers need to understand how product marketing contributes to project success, and product marketing likewise needs to acknowledge engineering's contribution. Frequent interactions help generate understanding, which in turn can lead to respect and trust. Encouraging appropriate moods and discouraging others can help create group interactions conducive to generating emergent results.

Just as the development team needs coaching in both technical and behavioral skills to meet their responsibilities, so does the product team. Product team members may not know how to write acceptance tests or participate in requirements specification sessions or take part in the decision-making process of setting priorities. Just as the project leader facilitates the smooth running of the engineering team, the product manager must facilitate the smooth running of the product team.

The often-overused term "empowerment" means to delegate decision-making authority to lower levels of organizations by changing who *makes* decisions. Decision framing focuses on who gets *involved* in the decision process. Managers who make decisions without input from subordinates and peers make poor decisions. Engineers who make decisions without input from managers and peers make poor decisions. Who makes the decision is less important than getting the right people involved in the decision process.

Daily interaction with the product team helps ensure that the development efforts stay on track to meet the needs and expectations of the customer. One of the key tenets of APM is close development team interaction with product managers, product specialists, and customers.

Exploring is accomplished by competent, self-disciplined teams led by competent leaders who create self-organized environments.

Project leaders and product managers are direct contributors to the team's exploration process. They encourage rather than motivate; they are demanding, but not arbitrary;

they empower the team, but make certain decisions themselves; they coach rather than criticize and they facilitate rather than command. Effective agile project leaders work hard to unleash the talent and abilities of their teams by focusing their efforts, molding individuals into jelled teams, developing everyone’s capabilities, providing resources to the team, working with customers and stakeholders, and facilitating a participatory decision-making process.

Chapter 10

Adaptation depends upon understanding a wide range of information, including an assessment of the project's progress, technical risks, the requirements evolution, and ongoing competitive market analysis. APM has the potential to save money through the early termination of projects, but only if the team and executives are willing to face reality early. Iterative projects are also prone to oscillation going back and forth without making progress.

Customer focus group (CFG) demonstrates ongoing versions of the final product to the product team to get periodic feedback on how well the product meets customer requirements. Although these groups are conducted at the end of iterations and waves, they should be scheduled early to ensure the right participants are available. Whereas customer team representatives work with the engineering team throughout a development iteration, a CFG brings a wider audience into the evaluation process.

Project status reports should have value to the project leader, the product manager, executives, other key stakeholders, and the project team itself. The reporting of information should drive activities aimed at maintaining control of the project and enhancing team performance. Developing the reports should help the project, and product leaders reflect on the overall progress of the project to separate the forest from their daily battle with the trees. The number and frequency of reports and the information in the reports need to match the size, duration, and importance of the project.

Schedule reports can take a variety of shapes, depending on the organization's standard practices. During the planning for each iteration, the team estimates, based on progress and story changes, the projected number of weeks for the entire project. Notice that the range of these estimates is wider at the beginning of the project (greater uncertainty) and narrower at the end (greater certainty). A range that isn't narrowing indicates that uncertainty and risk are not being reduced adequately and the project may be in danger.

References:

- **Agile Project Management: Creating Innovative Products, Second Edition, Jim Highsmith, Addison-Wesley Professional**
- **http://managedagile.com/was-steve-jobs-an-effective-agile-leader/**
- **Steve Jobs: Walter Isaacson**