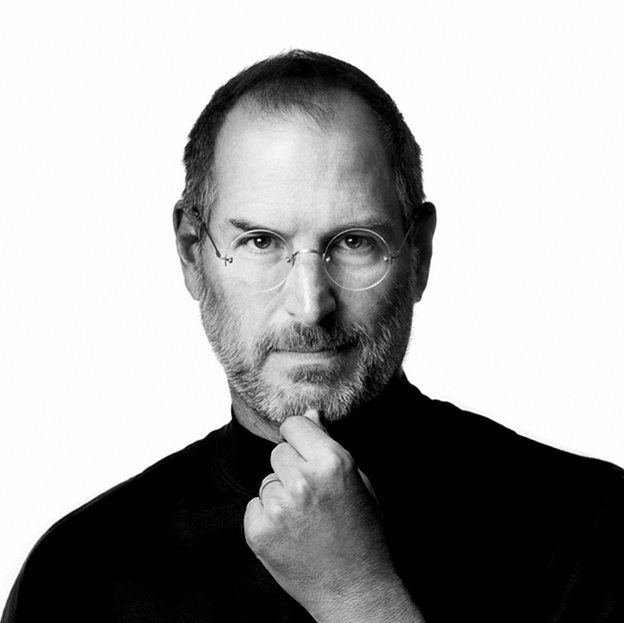
**CPSC 546 Modern Software Management**

**Prof: Lidia Morrison**

**Final Exam Summary Report**

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



**Chapter 1**

**Revolution**

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.

**Agile Business Objectives**

There are five key business objectives for a good exploration process such as Agile Project Management:

**Continuous Innovation:** *To deliver on current customer requirements.*

Developing new items and new administrations in the present complex business and innovation world requires a mentality that cultivates advancement. Endeavoring to convey client esteem, to make an item that meets the present client prerequisites, drives this consistent development process. Creative thoughts aren't produced in organized, dictator situations yet in a versatile culture considering the standards of self-association and self-restraint.

**Product Adaptability:** *to deliver on future customer requirements*

Regardless of how judicious a man, a group, or an organization, the future will dependably shock us. For a few items, changes in the market, innovation, or prerequisites happen week after week. For different items, the time allotment for consolidating changes fluctuates from months to years. With the pace of progress expanding and reaction time contracting, the best way to survive is to take a stab at item versatility a basic outline basis for an advancement procedure. Truth be told, in a deft venture, specialized brilliance is measured by both ability to convey client esteem today and make a versatile item for tomorrow.

**Improved Time-to-Market:** *to meet market windows and improve return on investment*

As the statistics for quickly contracting item development times demonstrate, lessening conveyance calendars to meet market windows keeps on being a high-need business objective for chiefs and administrators. The iterative, include based nature of APM adds to enhancing time-to-showcase in three key ways: focus, streamlining, and skill development.

**People and Process Adaptability:** *to respond rapidly to product and business change*

To be able to offer and attempt such individual attention to the Customer’s order, the internal operations have got to be geared up to deliver the same. The backend process has got to be flexible and adaptable to accommodate such orders and ensure perfect delivery of service or product. It is not only the process that should be adaptable and flexible, the management has got to build in the flexibility factor through empowerment of their teams and build a culture that is conducive for such operations.

**Reliable Results:** *to support business growth and profitability*

Reliable means meeting targets regardless of the impediments thrown in your way which means constantly adapting to meet a goal. Reliable processes focus on outputs, not inputs. Using a reliable process, team members figure out ways to consistently achieve a given goal even though the inputs vary dramatically. Because of the input variations, the team may not use the same processes or practices from one project, or even one iteration, to the next. Reliability is results driven.

**What is Agility?**

To answer this question, authors of this book explained a real-world case and gave 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.

**Agile Leadership Values**

Agility is more an attitude of how you handle the team and product. Visionary companies distinguish their timeless core values and enduring purpose, which should never change, from their operating practices and business strategies. We increase return on investment by making continuous flow of value our focus. We deliver reliable results by engaging customers in frequent interactions and shared ownership. We expect uncertainty and manage for it through iterations, anticipation, and adaptation. We unleash creativity and innovation by recognizing that individuals are the ultimate source of value, and creating an environment where they can make a difference. We boost performance through group accountability for results and shared responsibility for team effectiveness. We improve effectiveness and reliability through situationally specific strategies, processes, and practices. 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. That is, while there is value in the items on the right, we value the items on the left more.

**Agile Performance Measurement**

Measuring performance is the key goal for every project and on-going project. But there is no such total accurate thing when you are measuring the performance. Issues and problems arises and to overcome that, agile triangle is introduce which is replacing the traditional iron triangle of project management. The traditional iron triangle of undertaking administration, the furthest left triangle, comprises of scope, schedule and cost. The second triangle speaks to an early perspective of measuring coordinated improvement where schedule was settled, and scope was permitted to change that is, time was utilized as a settled imperative. Tragically, this second triangle still complies with the current iron triangle measures. The third triangle measures here are esteem, quality also imperatives (scope, schedule, and cost). Limitations are yet critical undertaking parameters, yet they are not the venture's objective.

**Chapter 2**

**Values over Constraints**

It explains the way we measure the performance on projects. We all know that money and time are key things when thinking about a project but most important is value for customer. We cannot just focus on simple things and forget about the important factors. Agile development endeavors to change that inclination and concentrate on the most vital things, and value is at the highest priority on that rundown.

**Continuous Flow of Customer Value**

Continuous flow mirrors the requirement for pioneers to see an incentive after some time amid both the present and future tasks. Programming in specific needs to change at the pace of business. Conveying variant 1 of the product in an opportune way is essential, yet significantly more vital might convey a top-notch item that is effectively versatile to future needs. Customers define capabilities that provide value and the business objectives that assist in quantifying that value. Today, value arises from implementing capabilities as they evolve over the life of a project.

**Innovation**

Innovation often offers the largest amounts of significant worth creation in an organization's undertaking portfolio. Innovation may come as new items, new plans of action, new forms, or new execution activities. Making new items and administrations contrasts from making minor upgrades to existing ones. 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.

**Execution**

As per authors, the basic venture administration approach is made from three procedures that trade data—arranging, controlling, and executing. In development ventures, they see a few disturbing parts of customary arranging. To begin with, the inspiration for arranging frequently originates from outside the undertaking; that is, plans are created to fulfill legitimate, administrative, or administration prerequisites rather than being founded on the work that should be expert. Second, the inspiration 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, since plans were sensibly right, and making an interpretation of the arrangement without hesitation was viewed as a basic process.

**Lean Thinking**

A considerable lot of the thoughts of the lithe development initially emerged 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. Though lean assembling thoughts have been utilized a while, lean item advancement thoughts, particularly those of an organization, have increased less acknowledgment.

**Delivering with Iteration**

Agile development and project management strain delivery of versions of the real product, or in the case of excessive-value substances, powerful simulations or models of the actual product. Finishing a requirements record verifies that a group has effectively accrued a hard and fast of necessities. Completing and demonstrating a hard and fast of working product capabilities verifies that the development group can clearly supply something tangible to the customer. Working functions offer reliable comments into the improvement method in methods that documentation cannot.

**Technical Excellence**

Project leaders must be champions of technical excellence; they must support and advocate technical excellence while maintaining a watchful eye on other project objectives. High quality ensures that companies will be able to deliver value in the future. 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.

**Delivery Versus Compliance**

Simplicity also means reducing overhead. APM is first and foremost about delivering value to customers. Too many project managers, too many project office members, too many organizations have drifted toward compliance as their primary, if often implicit, focus. 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**

**Leading Teams**

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. 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. 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. The agile movement supports individuals and teams through dedication to the concepts of self-organization, self-discipline, egalitarianism, respect for individuals, and competency.

**Building Self-Organizing Teams**

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 Declaration of Interdependence contains the principle, "We boost performance through group accountability for results and shared responsibility for team effectiveness." When an individual commit to delivering a feature during an iteration he accepts accountability for that delivery. Self-discipline enables freedom and empowerment. When individuals and teams want more autonomy, they must exercise greater self-discipline. One of the acute dangers of process-centric development and project management is that they reduce any incentive for self-discipline.

**Encourage Collaboration**

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.

**Customer Collaboration**

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. Accept accountability for results and adapting constraints.

Self-organizing teams are at the core of agile management, but the concepts have become corrupted—and counterproductive—in parts of the agile community. 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.

**Shared Space**

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**

**Adapting over Conforming**

Both traditional and agile leaders plan, and both spend a fair amount of time planning. But they view plans in radically different ways. They both believe in plans as baselines, but traditional managers are constantly trying to "correct" actual results to that baseline. The ability to respond to change drives competitive advantage. Think of the possibilities not the problems of being able to release a new version of a product weekly. 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.

**Responding to Change**

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. Another factor that impacts project management style is the cost of an iteration; that is, 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.

**Product, Process, People**

The barrier to agility in many software organizations is their failure to deal with the technical debt in legacy code. The failure is understandable because the solution can be costly and time consuming. However, failure to address this significant barrier keeps many organizations from realizing their agile potential. It took years for legacy code to degenerate; it will take significant time to revitalize the code.

**Barriers or Opportunities**

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**

**The Envision Phase**

The authors 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). Each of these artifacts is simple in concept, powerful, and low ceremony (informal), and operates on the principle of limited "real estate." 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.

In product development, as in most endeavors, getting the right people involved is the

critical success factor. "Right" consists of both having the appropriate technical ability (or domain expertise) and exhibiting the right self-disciplined behavior. Getting the right people doesn't necessarily mean getting the most talented and experienced people, just the most appropriate people for the job.

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. Although an organization might insist that project teams follow a guiding framework, there should be flexibility as to what individual practices are used to meet the needs of each phase. The team should discuss what practices are going to be used or not used. Just because a practice is a good one doesn't mean it needs to be used in every project.

**Chapter 7**

Agile teams plan, but they recognize that reality always intrudes. Plans in this context serve as a vehicle for embracing change, not blocking it. 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. Plans are both conjectures

about the future our best guess at what will occur given the information we have and guides to the future determining what we want to occur and making it happen. Development generates new information that in turn creates the need to plan again.

**Speculating on Product and Project**

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. Many hardware product development efforts are driven first by the creation of a product structure and then an extensive list of the features. In addition, because more and more products include embedded software, hardware and software features are both candidates for this feature-driven approach.

**Product Backlog**

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. The Speculate phase product backlog expands and refines the one developed in the Envision phase identifying and listing the features and stories from feasibility or marketing studies, requirements-gathering efforts, and product visioning. 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.

**Creating Backlog**

In product development, lack of or poor customer involvement in the process has been a recurring problem. However, the root of the traditional problem wasn't as much specialist intervention as poor collaboration between the product and development teams. In a waterfall environment, business analysts interact with customers and create documents that are passed on to development teams. 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. There are many approaches to business/product analysis and story definition. Many organizations utilize Use Cases, some in concert with business process analysis techniques. Whatever the technique, and at whatever level of analysis, three things are key to understanding: the people involved (that is, identifying the roles or personas that exist in the customer's environment), identifying the functions performed by these personas, and breaking that functionality down into implementable chunk stories.

**Release Planning**

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. Although experienced project leaders concentrate first on deliverables and then on the tasks necessary to create those deliverables, work-or task-driven plans often degenerate into very detailed, prescriptive plans. Any product has a set of features that customers use for some purpose. The more quickly we can link those customers to the features they have requested and get feedback on them, the more likely the product development effort will be successful.

**First Feasible Deployment**

This issue of deployable versus deployed has been confused by many aspiring agile people. Iterative development creates done-done, deployable pieces of a product. Those deployable pieces may or may not be implemented incrementally depending on circumstances. Actual

deployment is preferred, but not required.

**Estimating**

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. Estimating items again and again while they are in the backlog can also be wasteful.

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 replanning 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**

The Explore phase delivers running, tested, accepted stories. But rather than concentrate on the technical details of how to accomplish this goal, APM focuses on the agile leaders' tasks of creating self-organizing, self-disciplined teams that can deliver a releasable product. For small projects one person usually fulfills both project leader and iteration manager roles, whereas for large projects the roles may be split out to separate individuals. Even if these roles are filled by different people, both project leaders and iteration managers help in building the project community.

**Project Leadership**

Agile project leaders focus on adding value to a project. Unfortunately, many development engineers consider project management to be a roadblock—a hindrance, not a help. Project managers are viewed as administrators who put together detailed task schedules, create colorful resource profiles, bug team members about micro-task completions, and write reams of status reports for upper management, but don't serve as direct contributors to delivering value to customers.

**Iterating Planning**

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 team then re-estimates the work effort and adjusts the stories planned for the iteration if necessary. The entire project team product manager, product specialists, customers, developers, testers, iteration manager, project leader should participate in the iteration planning session as it provides everyone with the context for work to be accomplished during the iteration. Functional manager participation can help the team better understand strategic priority issues and demonstrates commitment to and support of the project. 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, especially when a team is in an agile learning mode, is that repetition breeds fast learning. Having to do things frequently improves learning, plus high frequency forces teams into learning how to do things quickly. For example, if a team is forced into going from daily to hourly builds, they will find ways to automate the process that they may not have been forced into otherwise. So, for immature agile teams, try shorter iterations in the beginning to force learning.

**Monitoring the Progress**

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. In any case, if a burndown chart is used, I recommend a unitary task burndown chart, one that shows whole tasks completed, rather than the burndown of total iteration hours that many use.

**Simple Design and Integration**

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. Adaptation means waiting until requirements or design issues arise and then building them into the product.

Simple design means valuing adaptation over anticipation. The objective of continuous integration is to ensure that product features fit together into an integrated whole early and often during development to reduce both the high cost of late misalignment and the burden of testing.

**Coaching and Team Development**

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. Team members want a boost every now and then, but they don't want meaningless rah-rah speeches. Team members want the facts, even negative ones, so they can help figure out how to deal with the situation.

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. Teams are groups of people, who respond to emotions and whose emotions may experience wide swings from despair to euphoria over the life of a project. 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.

**Decision Framing**

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. Every team and situation are different, so there isn't a quantitative answer to the question of how many unilateral decisions are too many. However, even though presenting absolute numbers risks misinterpretation, I think the following guidelines may help define appropriate "levels" of management decision making that will continue to foster self-organization.

**Scrum Meetings**

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. When dealing with uncertainty, risk, fluid requirements changes, and technological frontiers, product managers need to be fully involved in identifying stories, specifying requirements, determining priorities, making key tradeoff decisions (cost, schedule, etc.), developing acceptance criteria and tests, and more. Being the "customer" for an agile project is not an insignificant job, but it may not need to be full time.

Exploring is accomplished by competent, self-disciplined teams led by competent leaders who create self-organized environments. Team members work in a semiautonomous fashion, striving to meet iteration plans that they themselves have had a hand in constructing, managing their own workload, collaborating to generate innovative ideas, and applying specific technical practices aimed at building adaptable products that in turn facilitate the very exploration process that they are employing.

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. Adaptations can take many forms. A team that rushes to deliver features but creates defects hurrying rather than being quick needs to adapt its behavior. A creeping design degeneration gives rise to additional "refactoring" activity in the next iteration. Cost overruns are highlighted by the project status review, and appropriate action can be taken. The objective of the review and adaptive action practices is to ensure that frequent feedback and high levels of learning occur in multiple project dimensions. There are two main reasons for conducting review and adaptive action sessions at the end of an iteration. The first reason is obvious: to reflect, learn, and adapt.

**Customer focus group (CFG)**

Here the authors demonstrate 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. For example, whereas one or two individual customers from manufacturing might be involved in the day-to-day work with a team on a manufacturing software application, six to eight might be involved in a CFG. This wider audience participation helps ensure that features don't get overlooked, the product encompasses more than the viewpoint of a few people, confidence in the product's progress increases over time, and customers begin to become familiar with the product before actual deployment. The intent of a CFG session is to stimulate discussion about the product to generate customer and product team feedback. The sessions are designed to encourage participation, questions, and change requests from the product team.

**Project Status Reports**

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 and Risk Status**

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.

**Closure**

A project close is both a phase and a practice. Organizations tend to spend too much time initiating projects and too little time closing them. During one client engagement, I encountered the "failure to close" problem again. The customers of an IT project considered IT delivery to be less than stellar, partially because customers thought the project had been underway for years. The application in question had been installed for several years, but the initial production and ongoing enhancement releases were not differentiated from each other. The most important closing activity is conducting a project retrospective. Teams using APM have done mini-retrospectives each iteration. These minis help the team learn about its own processes and team dynamics as the project progresses and as such are intra-team learning activities.

Frequent iterations that deliver working features allow agile teams to make frequent adjustments based on verifiable results rather than documentation artifacts. This can create an un comfortable situation for managers and customers who don't want to deal constantly with either reality or the need to make tradeoff decisions.

The Adapt phase provides a short respite from the intensity of short-cycle iterative development. In a serial project, in which a working product may be months or even years in the future, it is very difficult to maintain high levels of work intensity there is always tomorrow. Agile projects sometimes have the opposite problem they can be overly intense. The brief review, adapt, and plan activities of the Adapt phase give team members time to catch their breath, and their mental faculties, before rushing off to the next delivery iteration.

**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**