A Participative Visual Approach to Milestone Planning

Eduardo Miranda

Institute for Software Research, Carnegie Mellon University

This draft Nov. 11th, 2018

Direct correspondence to:

Eduardo Miranda

Institute for Software Research, Carnegie Mellon University

mirandae (at) andrew.cmu.edu

Abstract

This paper presents a participatory milestone planning method called the Visual Milestone Planning (VMP) Method. VMP promotes involvement and commitment, through the reification of the planning artifacts and its direct manipulation by team members who collectively create the plan. Once a project scope is defined via a work breakdown structure and relevant milestones identified, a novel construct called the milestone planning matrix is used to systematically and visually capture dependencies among milestones and map WBS elements to the milestones they help realize. The milestones due dates are later determined by accommodating sticky notes representing the work to be done on a resource and time scaled milestone scheduling canvas. The method is applicable to traditional as well as to agile projects.

Keywords

Milestone planning; participative planning; collaborative planning; milestone planning matrix; visual planning; agile project management

Introduction

This paper introduces a participatory approach to milestone planning method called the Visual Milestone Planning (VMP) Method. The method has been successfully used by teams of students in the Master of Software Engineering Program at Carnegie Mellon University for planning their capstone projects[1] and it has also been taught at a number of industry and government organizations.

Milestone planning is a planning process pioneered by Andersen [1] [2] and Turner [3] in which a project is planned in terms of outcomes, significant changes of state, external dates and customer commitments instead of the activities that make it up. In other words, the plan delineates the chosen strategy but does not dictate the activities or tasks that ought to be executed to achieve them which will be elaborated when needed. Typical milestones include: major decision points, key quality reviews, the delivery of key components or assemblies of the end product, completion of important process steps and the satisfaction of customer commitments towards the team. Responsibility for each milestone can be assigned to different organizations or team members and thus used to coordinate the efforts of different groups or individuals. Planning in terms of milestones instead of activities makes the process less onerous and the plan more robust and understandable.

Participatory planning is a planning process in which the people responsible for the execution of the plan is actively involved in its formulation. Successful examples of this way of working are numerous: the pull planning process in the "Last Planner System" used in the construction industry [4] [5] [6], visual planning [6] [7] [8] and the charrette method [10] [11]. The simplest reasons for using participatory planning is that it usually yields a better, more comprehensive plan and results in a better execution than an imposed plan. Paraphrasing Lewis [12], a plan that is developed in isolation to be later

[1] These projects range from 2,500 to 5,500 engineering hours and they are performed for real customer

communicated or simply handed down to those responsible for its implementation risks that its recipients would not: 1) get it, 2) agree with it, or 3) buy into it, because "it is their plan, not ours". This is even truer when it comes to milestone plans and self-organizing teams as the activities and tasks by which the milestones will be realized are unspecified and necessitate that those deciding them have a shared understanding of the plan's intent to achieve unity of purpose without the need to refer to a higher level authority.

These paper contributions are threefold:

- The integration of the milestone planning and participatory planning approaches through a visual planning process [6]
- A novel construct called the milestone planning matrix that systematically and visually captures: 1) temporal dependencies between milestones and 2) the allocation of work elements to the milestones they help realize
- The reification of work packages by means of sticky notes which must be physically accommodated on a resource and time scaled milestone scheduling canvas to derive the milestones due dates

While the article will provide a cursory explanation of all artifacts and techniques mentioned in it in an effort to make it self-sufficient, the focus would be on the proposed method and the novel constructs employed. We encourage the interested reader to consult the original sources on top of which VMP is built. The rest of the paper is organized as follows: Section 2 describes the milestone plans, Section 3 the proposed method, Section 4 the linkage of milestone plans to planning waves and iterations in agile projects.

Milestone Plans

In Andersen's work [1], a milestone is defined not *"as the completion of an activity, usually an especially important one"* but as *"a result to be achieved. A description of a condition or a state that the project should reach by a certain point in time. A milestone describes what is to be fulfilled, not the method to fulfil it"*. Figure 1 shows a typical milestone plan. As can be observed, a milestone plan is short. It is written in a language a project sponsor can understand and confined to a size which will allow it to be grasped at a glance. The plan is written as a sequence of states the project will go through to its successful conclusion and not the activities the team needs to perform. For example, the "Design concept approved" state, defines a situation where the project team has presented an idea that satisfies the needs of the sponsor and he has acquiesced to it. The plan does not estipulate how the team will get there. Will they build wireframe diagrams? Develop high fidelity prototypes? Make a PowerPoint presentation? Perform user testing? Employ focus groups? At some point, these issues will certainly have to be addressed by the team, but they have no place in a milestone plan.

The focus on states rather than on activities results in a more robust plan since independent of what tasks are performed to get there, when and by whom, the project sponsor would like to approve the design concept before it is implemented and this is unlikely to change.

The dependencies between milestones are typically “Finish to Finish” relations and not the most common “Finish to Start” used in activity plans. Finish to Finish relations are very easy to spot and provide great freedom as to when the activities leading to the realization of the milestone could start.

Milestones could be hard or soft. Hard milestones are milestones that if not accomplished by the set date lose all or most of its value or results in severe penalties. The date a government resolution related to the system under development goes into effect and the start of the holidays shopping season could be examples of hard milestones the project needs to satisfy. Soft milestones have completion dates that results from the planning processes. They might be associated with penalties or other liabilities after a statement of work is agreed, but in principle are discretionary.

Date	Client	Team	Milestone description
Oct-1			Project kick-off
Nov-15			Design concept approved
Nov-30			Infrastructure selected
Dec-15			Design completed
Jan-30			Release 1: CL*, BD, AC, CO, Data Base
Feb-15			Cloud infrastructure available
Mar-1			Release 2: CBL, RC, SM
Mar-10			Beta testing launched
Mar-30			Beta testing results reviewed
Apr-15			Release 3: PD, PP
Apr-20			Acceptance testing procedure approved
May-15			Acceptance test completed
May-30			System deployed
Jun-30			Customer sign-off

Figure 1. Typical milestone plan showing due dates, responsibilities and milestones’ descriptions

The Visual Milestone Planning (VMP) Method

The VMP Method is depicted by Figure 2. The first step in the process is to create an outcome oriented work breakdown structure delineating the project scope. Each WBS element will have associated with it the estimated effort required for its realization. These estimates will be later used to establish windows of opportunity in which the anticipated work could be performed. A project whose final outcome is not well defined could be scoped in terms of learning activities such as running a design sprint and planning packages that have a definite purpose and budget but whose exact content has not yet been decided.

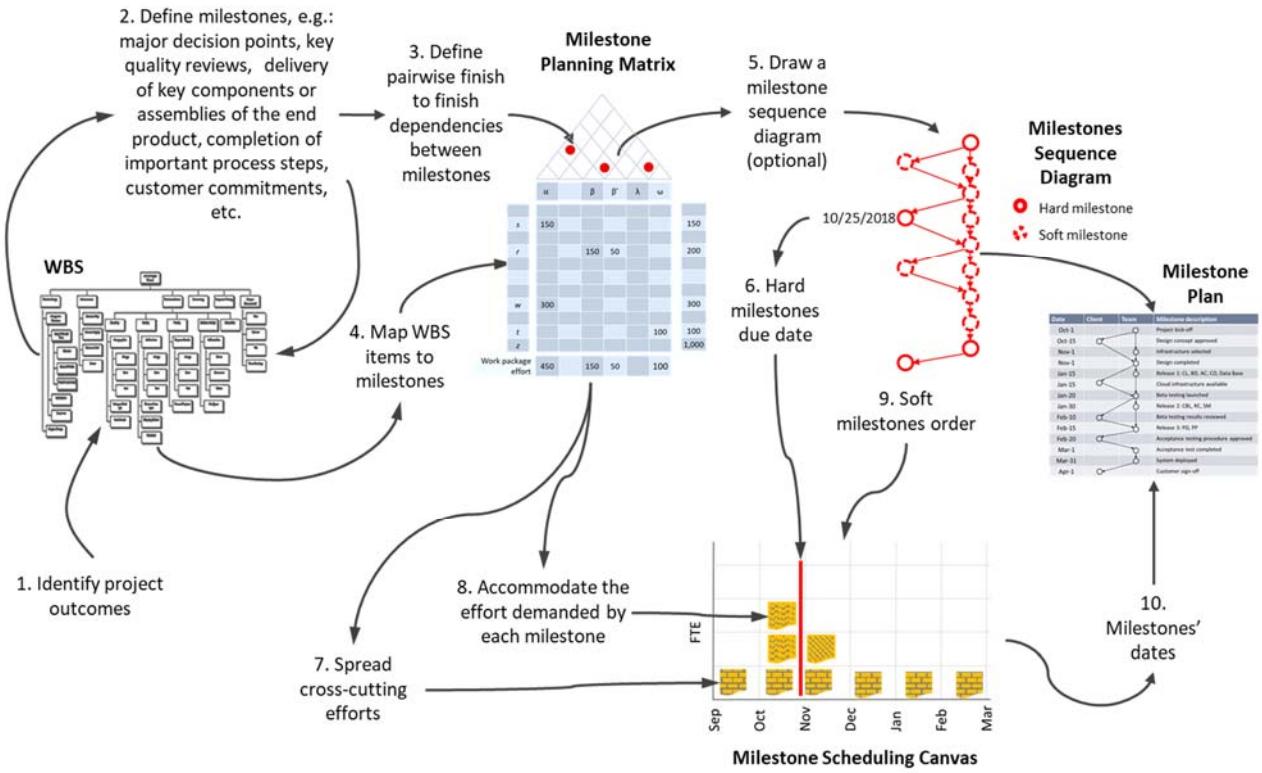


Figure 2 Participatory approach to milestone planning

The second step in the process is the definition of milestones. Milestones are chosen to signal the taking of a major decision, the delivery of key components or assemblies, the completion of important process steps or to mark a commitment made to the team, e.g. a customer makes proprietary equipment or technology required by the project available to the team. Notice that in the diagram there are arrows back and forth between steps 1 and 2. This is so because although the WBS will normally inform the choice of milestones, sometimes the selection of a particular milestone might result in the creation of a new task or outcome that must be incorporated in the WBS or leads to its rearrangement.

In Step 3, the identified milestones are first written down from left to right as column labels in the header row of the milestone planning matrix, ordered according to a logical sequence of completion. Beware that in the presence of multiple paths through the project this sequence might not be unique. Finish-to-finish non-redundant pairwise dependencies between milestones are captured by placing a dot at the intersection of the diagonals corresponding to the related milestones in the roof section of the planning matrix. The roof of the matrix could then be used to unravel the list into a milestone diagram, Step 5, which communicates the milestone sequence in an easier to read format for the team to validate.

In step 4, the WBS elements are associated to the milestones they help realize via the body of the milestone planning matrix. The association is done by labelling a row in the planning matrix with the name of the top most WBS element whose descendants all contribute to same milestone, and recording the effort required by it at the intersection of the said row and the column corresponding to the milestone with which the element is associated. A milestone can have multiple WBS elements associated with it, i.e. several WBS elements must be completed to realize the milestone. In most cases

a WBS element would be associated with a single milestone, there are however a few instances in which is convenient to allocate fractions of the total effort required by the WBS item to multiple milestones that would be discussed later. The set of WBS elements associated with a milestone is called its work package.

In Step 6 we mark hard milestones in the scheduling canvas and black out non-working dates such as holidays or known vacation periods. Hard milestones will act as anchor points for the plan.

In steps 7, 8 & 9 we build the project's staffing curve by posting sticky notes on an empty space in the milestone scheduling canvas such as all effort required by a milestone's work package could be fitted in a time box to the left of it and the milestones' order is respected. Time boxes cannot overlap as this would imply that somebody is actually performing, not multitasking, two tasks at the same time. The layout of the time boxes will determine the earliest date in which a milestone could be achieved can be reached. If somehow the plan is not feasible, e.g. there are not enough resources or the hard milestone dates cannot be met, the project scope should be renegotiated, the work approach reformulated or the constraints lifted.

In Step 10, we complete the plan by assigning due dates to the soft milestones, adding other missing information and properly formatting it. The approximate due date for each soft milestone is found by looking in the scheduling canvas for the date aligned with the right edge of the time box associated with the milestone. Hard milestones have, by definition, a set date.

The outcome oriented work breakdown structure

An outcome oriented work breakdown structure (WBS), see Figure 3, also known as deliverable or product oriented WBS, is a hierarchical representation of all the work a project needs to do constructed using the project's results as main decomposition criteria.

The importance of accurately identifying the scope of work in a project cannot be overstated. Accurately does not mean we need to know every last detail at the beginning of the project. Accurately means that if there are things we do not know now, we acknowledge them and make provisions to learn and perform the work with the understanding that if things exceed our allocations the plan will need to be revisited.

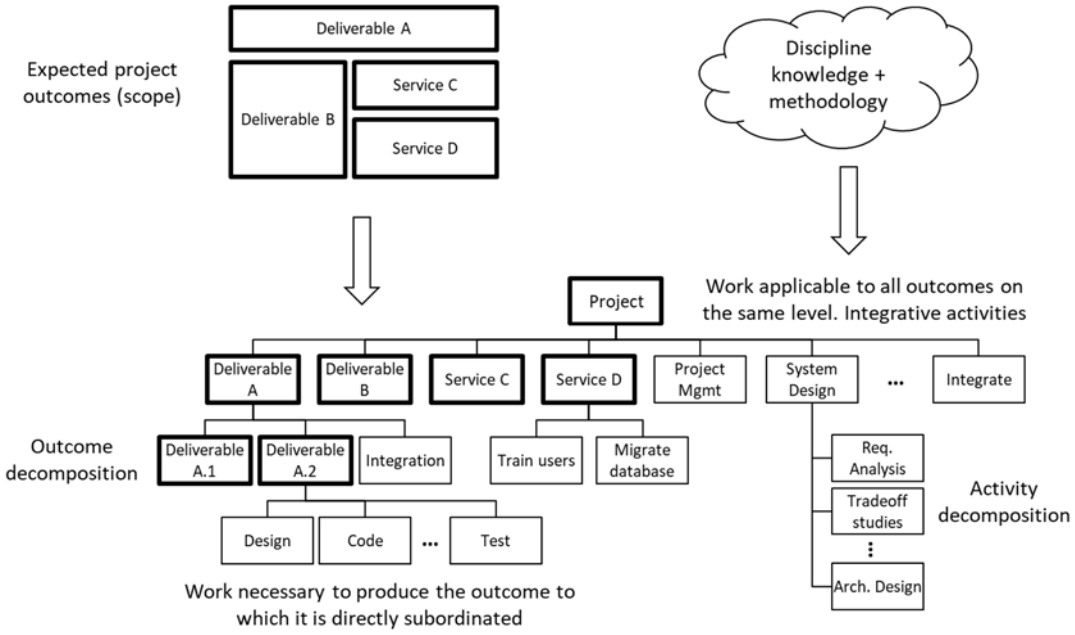


Figure 3 WBS example (Adapted from NASA's System Engineering Handbook, 2007)

We prefer to call this type of WBS outcome and not deliverable or product oriented to force teams to think that a project could be expected to produce results that are not products, for example a project might be about organizational change, that is after the project is successfully concluded the organization performs at a new, higher level. We find useful thinking in terms of four types of outcomes a project might be expect to deliver: products, services, processes and capabilities. From the point of view of the WBS construction, all outcomes are treated the same.

An outcome oriented WBS will contain the expected project results and the activities necessary to realize them [13]. The first level of the hierarchy defines a set of outcomes and activities that collectively and exclusively represent 100% of the project scope. Outcomes might be broken down into lower level outcomes and activities or tasks. Activities can only have other activities and tasks as descendants. Tasks are not decomposable. The convention used in building the WBS, is that integrative and common activities support all the WBS elements with the same ancestor and that, whenever and activity or task applies only to one element it should be subordinated to it.

In the WBS, activities and tasks should be positioned such as the most realistic effort associated with producing a WBS element could be established as this information would be used to support scoping, staffing and scheduling decisions. For example, if an architectural activity supports the whole project it should be positioned at level 2 of the WBS, however if the system under planning is made up of a number of subsystems which could be implemented or not depending on the available funding or some other factor, each of them including significant architectural work we will include the subsystem's architectural activities under each of them, so in case a decision not to implement a subsystem is made, removing the subsystem from the WBS will also remove all the direct costs associated with it without affecting other parts of the WBS.

The Practice Standard for Work Breakdown Structures [14] defines two types of activities:

- Discrete Effort. Work effort that is separate, distinct, and related to the completion of specific work breakdown structure components and deliverables, and that can be directly planned and measured.
- Level of Effort (LOE). Support-type activity (e.g., seller or customer liaison, project cost accounting, project management, etc.), which does not produce definitive end products. It is generally characterized by a uniform rate of work performance over a period of time determined by the activities supported.

The distinction is important, because as will be explained later, it affects the way in which WBS elements are mapped to milestones.

There isn't a hard and fast rule as to how far should a WBS element be decomposed, but as there is a cost associated with having a greater level of detail, any additional decomposition has to be purposeful. Among other reasons, one should breakdown a WBS element, when the decomposition:

- Supports scoping decisions, i.e. we can do this but not that; scheduling decisions, i.e. we can do this now and that later or allocation decisions: i.e. we can do this and outsource that
- Helps isolate riskier elements, i.e. elements that somehow should be treated differently
- Makes estimation of the effort required by WBS component easier or more accurate, i.e. by estimating the decomposed elements and aggregating them
- Have to be separated for administrative or compliance reasons, i.e. CAPEX vs. OPEX, industrial benefits, etc.

All these while considering the materiality of the newly created items, i.e. a 1-hour task is not material in a 10,000 hrs. project and any gain obtained by decomposing would be probably faded by the cost of creating and maintaining the new decomposition.

The milestone list

Correctly identifying the set of milestones to include in the plan is essential to its acceptance as these milestones would become the vocabulary that will be utilized to explain the work logic to the project sponsors and other team members as well as to gauge its progress. A good milestone set will include, as a minimum, the things the sponsors care about. For example, if the project sponsor wanted to have a review of the user interface before moving forward with the rest of the of the software, it would make sense that the milestone set included a "UI Approved" milestone.

A milestone describes an event that needs to occur not later than a certain date at risk of delaying other milestones or the whole project. Typically, milestones fall in one of three categories: the realization of an outcome, the attainment of a relevant project state or the satisfaction of a commitment made to the project team by an external party. The first two types of milestones are achieved upon the completion of all work items included in its work package. In the case of a commitment to the team, the milestone is achieved when the party responsible for it fulfils its obligation. These milestones tend not to have a work package associated with it and are an excellent instrument to synchronize work across multiple teams, each working according to their own plans.

The following are examples of the different types of milestones:

- Outcomes: a document, a partial or total system capability, a prototype, the results of a survey
- Desired states: a major decision, an approval, an attainment of some kind, e.g. number of transactions per second, number of users trained
- Satisfaction of commitment: delivery of proprietary equipment necessary to test the software under development a special hardware is delivered to the project team, publication of an API specification by another team

The criteria by which to judge the realization of a milestone is known by different names in different contexts: exit criteria, definition of done and conditions of satisfaction but they are all about the same thing: having an objective test to determine whether the milestone has been reached or not.

Typically, a completion criteria would include the list of work items to be finished, a description of its state and quantity if applicable, demonstrated performance such as transactions per second or power efficiency, and a definition about the quality those things need to be completed at, e.g. defects counts, tolerances, weight budgets, level of coverage, etc.

Many times writing the completion criteria will bring up or force the breaking down or the re-estimation of activities already on the WBS. This is a good thing because the early identification of these gaps help prevent problems later in the project. Working on the definition of done also helps the team develop a shared understanding of the work.

The number of milestones chosen must balance visibility with robustness and ease of understanding. Depending on the size of the project 10 to 50 milestones will satisfy the needs of most small to midsize projects. Given the visual nature of the method every effort should be made to confine the plan to a size which allows to have it in plain sight at once.

Once the milestones have been identified, the team will order them in the approximate sequence in which they must be completed.

Milestone planning matrix

The Milestone Planning Matrix (MPM) resembles the matrix known as the “House of Quality” in the Quality Function Deployment method [15]. Strictly speaking, the MPM is made up of two matrices: A triangular matrix, the roof of the house, which captures finish to finish dependencies among milestones and the body of the house, which is a multiple domain matrix [16] [17], mapping work items in the WBS to the milestones signaling their completion. See Figure 4. The beauty of the approach is that the construct provides a straightforward mechanism to make visible these relations to everybody involved in the planning process. Visibility is the key to prevent gaps and overlaps in the plan and in the achievement of its shared understanding by stakeholders.

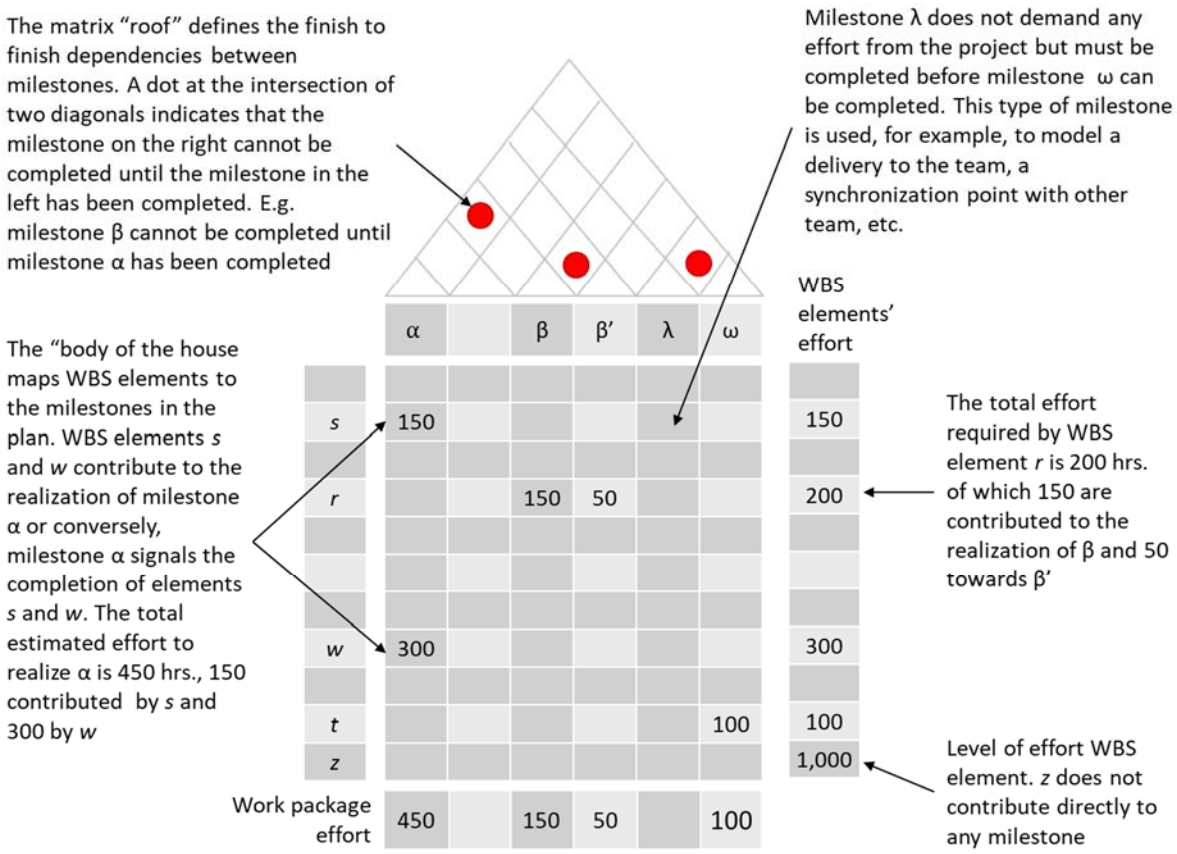


Figure 4 The Milestone Planning Matrix

Most of the time, work elements will map naturally and on its entirety to a single milestone, but in practice the author has found two situations in which this is not the case: level of effort activities and activities involving a preliminary and a final delivery. Of course, it is always possible to breakdown the "offending" activity into as many WBS elements as milestones it maps to and associate each of them to the corresponding milestone but this could be seen as something that unnecessarily complicates the WBS just to serve the choice of notation.

For the level of effort activities, we employ two tactics: When dealing with an activity such as integration in which the level of effort task concurs to the realization of some milestone but not others, we allocate a fraction of the WBS element's effort to each milestone to which it contributes. In the case of activities such as project management and quality assurance whose effort contributes to all milestones we do not allocate hours to any milestones.

For activities that have a preliminary and a final outcome, e.g. the delivery of a draft document followed by a review and some later work to obtain final approval, and for which is important to highlight both events: the delivery of the preliminary version so the receiver can prepare its own activities and the acceptance of the final one so the performer can receive a payment or proceed with other activities. In this case do one of two things: Allocate a fraction of the WBS element's budget to each milestone like we do for the level of effort tasks or assign all the hours of the work package to the last milestone in a given milestone chain and discretionarily position the precursor milestones on the staffing curve.

Milestone Scheduling Canvas

Figure 5 below shows a typical milestone scheduling canvas. The canvas is a key piece of the process since it is there that the plan materializes. As the planning involves the physical positioning of sticky notes on the canvas, there has to be a correspondence between the work hours represented by each note and the canvas' physical dimensions. If for example, we choose each 3"x 3" sticky note to represent 40 hours of work, each three inches on the date axis of the canvas will correspond to a week and three inches in the resources axis will correspond to a full time equivalent (FTE) resource. Had we chose the sticky note to represent 150 hours of work, for example for a larger project, each three inches on the time axis would correspond to a month instead of a week. If necessary, sticky notes might be ripped off to express fractions of effort or time.

In constructing the staffing curve, we will assume the distribution of competences in the plan matches the tasks' needs. If one wanted to know or was somehow limited by the resource availability, it would be possible to break the work into competency lanes and assign the corresponding effort to each lane.

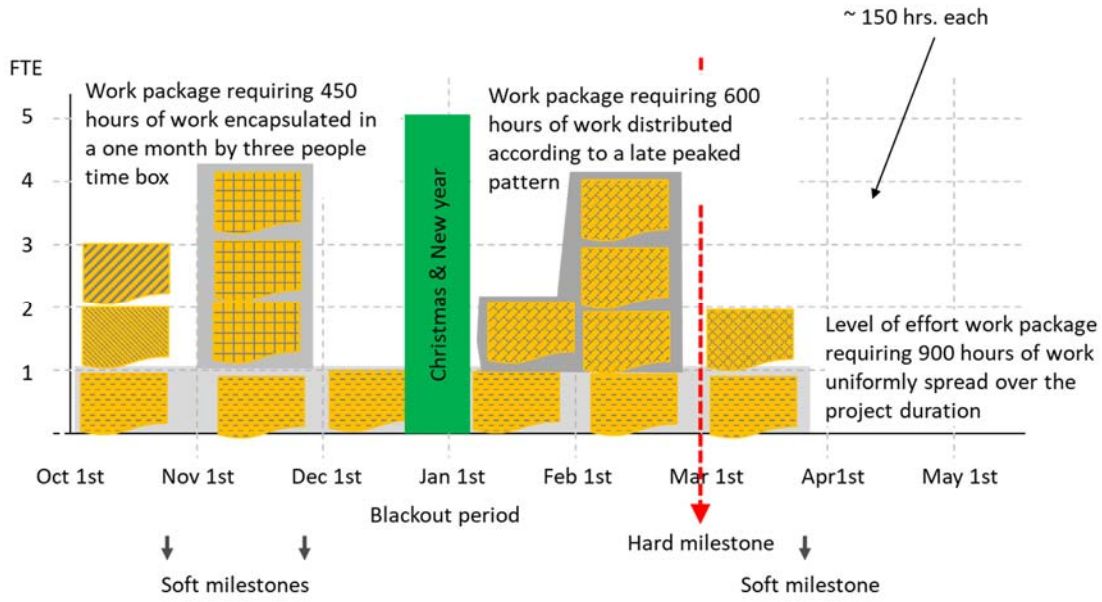


Figure 5 The Milestone Scheduling Canvas

All the effort required by a work package will be contained within a time box. The rightmost side of the time box will mark the earliest date in which the corresponding milestone could be achieved. Time boxes do not overlap and their shape does not need to be regular. As much as possible the time box contour should reflect the nature of the work performed, e.g. front loaded, back loaded, flat, early peaked, late peaked, etc. In any case, their height at any point cannot surpass the amount of resources available that could reasonably be applied to the execution of the work package and the cumulative height of any stacked time boxes cannot go above the number of resources available to the project. The time box length would be such, that the area enclosed by it is equal to the work package's effort. The process of populating the milestone scheduling canvas starts by accommodating the work required by

the level of effort activities such as project management and quality assurance that stretch over the length of the project or phase at defined manning levels. The next step depends on whether or not the project contains hard milestones. If it does, we first post the work corresponding to them and after we move backwards to the start of the project (back casting) accommodating the work packages corresponding to the soft milestones in the order imposed by their dependencies. If there are no hard dates, the easiest way to proceed is to accommodate the work packages from left to right, starting at the beginning of the project and moving forward to the end along the most logical path. Sometimes the team might feel the need to introduce schedule buffers, not covered here for reason of space, to protect important milestones. Once all milestones have been layout, the due date for each milestone could be found by reading the corresponding dates in the horizontal axis of the scheduling canvas.

Example

The example is organized around the method's steps shown in Figure 2. Imagine your company is bidding in a contract to develop an ecommerce site for a small book publisher and after some discussion with your customer you sketched the following notes:

a. The customer wants to include a beta testing period to validate the site design.

b. He will not accept deployment until a system wide acceptance test is satisfactorily completed.

c. Customer sign-off will follow satisfactory deployment of the system.

d. He would like to have at least three software releases: one to collect users' feedback via beta testing, another one to confirm the progress of the system towards the launch date and the final one to complete the system with minimum risk to the launch date.

e. In the first release he would like to include the following functionality: Category List (CL), Book Details (BD), Add to Cart (AC), Check-Out (CO) and all Data Base

f. In the second, the Category Book List (CBL), Remove from Cart (RC), Shipping Method (SM).

g. In the final release: Payment Details (PD), Process Payment (PP)

h. The customer is preparing to launch its business in April of next year so he would like the system to be ready at least one month before that.

For its part, your company:

i. Cannot start the project until the end of September and has only three developers and a project manager available to work on the project

j. To minimize the risk of rework, it does not plan to start programming until the infrastructure is selected and the user interface and information architecture design are well underway

Step 1

Based on the requirements above and its professional knowledge the development team created the WBS shown in Figure 6 describing the contract's scope of work and estimated the effort required for its execution. The software required software capabilities were grouped into four categories: Browsing, Buying, Paying and Data Base to increase the readability of the WBS as the team felt a grouping based on Releases would have difficulted the comprehension of system functionality. Website design, beta and acceptance testing could have been made part of the Website Software deliverable but the team chose to put them at the first level of the WBS to highlight its understanding of the customer wishes.

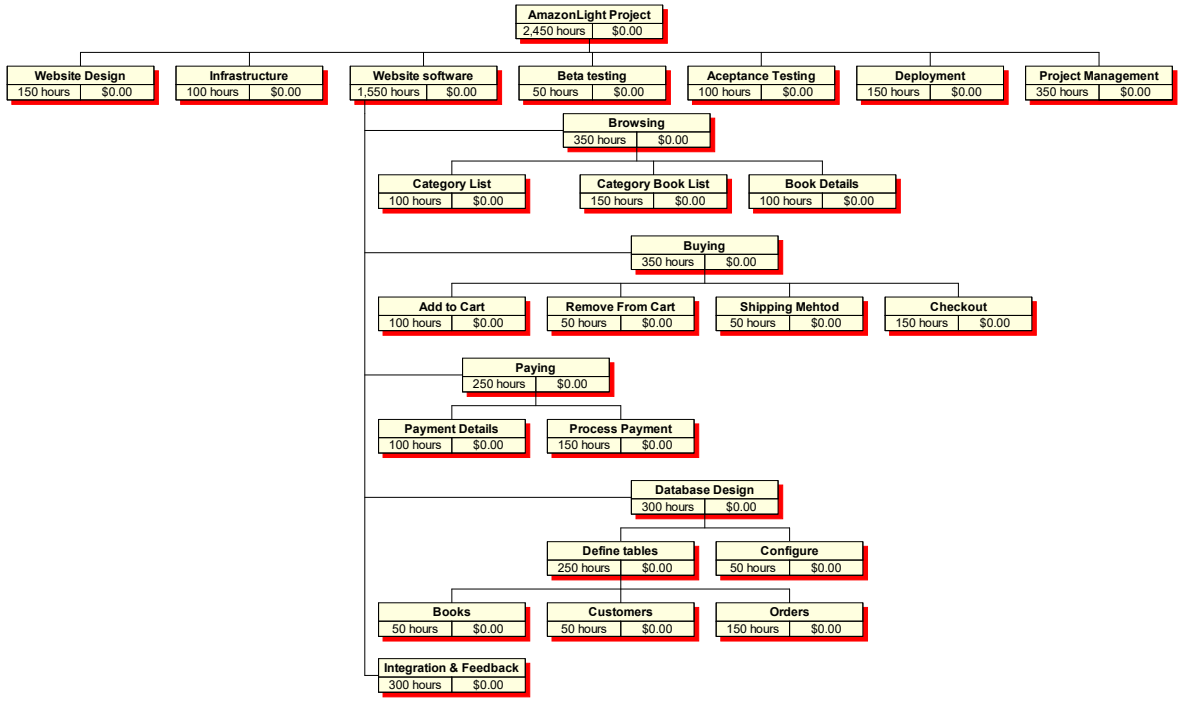


Figure 6 WBS fot the AmazonLight Project

Step 2

After obtaining concurrence for the WBS from the client, the team started choosing relevant milestones and produced the list in Table 1. Beware that the solution is not unequivocal. While there are self-evident milestones like project kick-off, software releases and the client request for a beta test, others are created by the team based on its best judgment as to what is important and what is not. The preliminary completion criteria provided defines the meaning of the milestone and helps identify which WBS elements should be mapped onto them. The milestones are organized in what seems like the most logical sequence to make the list easier to understand and search for.

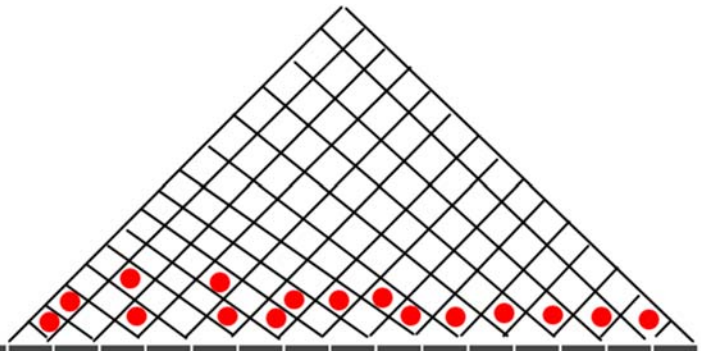
Table 1 Potential project milestones

Milestone	Rationale (see notes above)	Hard date	Completion criteria
Project kick-off	i	October this year	Development team assembled, meeting with project sponsor concluded
Design concept approved	j		Information architecture and UI designed and approved by sponsor
Infrastructure selected	j		Cloud provider selected. Consider AWS, Azure, Google Cloud and 2 others
Design completed	Team		Feedback from sponsor incorporated into design
Cloud infrastructure available	Team		Cloud production environment available
Release 1: CL, BD, AC, CO, Data Base	d		Indicated functionality is ready and tested at 90% coverage and working in production configuration. No broken menus or links
Beta testing launched	a		Release 1 software made available to beta users. User behavior hypotheses defined. Website instrumentation working
Release 2: CBL, RC, SM	e		Indicated functionality is ready and tested at 90% coverage and working in production configuration
Beta testing results reviewed	a		All insights arising from the beta testing disposed
Release 3: PD, PP	f		Indicated functionality is ready and tested at 90% coverage and working in production configuration. Changes resulting from beta testing implemented
Acceptance testing procedure approved	b		Acceptance test suite approved by sponsor. Includes at least one positive, one negative and one invalid test case for each functionality
Acceptance test completed	b		All acceptance test passed with no objection from sponsor
System deployed	c	May next year	All functionality running in production environment, operators trained. System must run for at least 15 consecutive days without a fault attributable to software
Customer sign-off	c		Customer accepts ownership of the software
Project closed			Project postmortem executed, all records archived

Steps 3 & 4

During step 3, the dependencies between milestones are defined and documented in the roof section of the milestone planning matrix and during step 4, the WBS elements are associated with their corresponding milestones. This is a pretty mechanical process whose value resides on the transparency that brings to the planning process. As discussed above, a WBS element can be associated with more than one milestone as shown by element 1.a in Figure 7. While this technique relief us from creating WBS elements just for the sake of having each element to map to a single milestone, abusing it diminishes the transparency of the mapping. Milestone “Cloud Infrastructure Available” has no WBS element associated with it, because although the work to select the infrastructure is part of the scope of the project, the effort to provision it, is not. The reason to include it as a milestone is that it represents a commitment made to the project team by the customer so that they could beta test the system and to signal that a delay in fulfilling this promise could affect the completion date of such test. Elements 1.c.5 and 1.g in in Figure 7 are Level of Effort tasks. In the case of “Integration & Feedback” a certain number of hours were allocated to each of the three milestones representing the completion of the software

increment according to the amount of functionality to be integrated. In the case of project management, the effort was not allocated to any milestone in particular to be later spread over the life-span of the project according to a uniform profile.



WBS Code	AmazonLight Project	Total Effort	Project kick-off	Design concept approved	Infrastructure selected	Cloud infrastructure available	Design completed	Release 1: CL, BD, AC, CO, Data Base	Beta testing launched	Release 2: CBL, RC, SM	Beta testing results reviewed	Release 3: PD, PP	ATP approved	ATP completed	System deployed	Customer sign-off	Project closed
1.a	Website Design	150		140			10										
1.b	Infrastructure	100			100												
1.c.1.1	Category List	100						100									
1.c.1.2	Category Book List	150								150							
1.c.1.3	Book Details	100						150									
1.c.2.1	Add to Cart	100						100									
1.c.2.2	Remove From Cart	50								50							
1.c.2.3	Shipping Mehtod	50								50							
1.c.2.4	Checkout	150						150									
1.c.3.1	Payment Details	100										100					
1.c.3.2	Process Payment	150										150					
1.c.4	Database Design	300						300									
1.d	Beta testing	50							40		10						
1.e	Aceptance Testing	100											50	50			
1.f	Deployment	150													150		
1.c.5	Integration & Feedback	300						150		75		75					
1.g	Project Management	600															
	Work Packages	2700	0	140	100	0	10	950	40	325	10	325	50	50	150	0	0

Figure 7 Milestone planning matrix for the AmazonLight project

Step 6

By definition, to be successful, a plan must satisfy its hard milestones, so they tend to act as anchoring points for the whole project. The accommodation of the work elements in the scheduling canvas starts by marking on it any hard milestones the project might have. In this step we also blackout any known non-working periods involving the whole team such as holidays, closings, and special vacation periods.

Steps 7, 8 & 9

The goal of these steps is to establish a time frame in which the work represented by each work package could be executed. To do this, the team labels a number of sticky notes proportional to the effort required by the work package and accommodates them in an appropriate empty space on the milestone scheduling chart. The team starts by accommodating the effort corresponding to all cross-cutting work packages, then that corresponding to the work packages connected to hard milestones and finally that corresponding to the rest of the work packages in the order dictated by the milestone dependencies. If necessary, the team might intersperse buffers to protect critical milestones.

Figure 8 shows a possible plan for the AmazonLight project. Notice that due to the holiday period, extending from late December to early January the effort for the Release 1 work package was spread over two months by splitting the sticky notes. Another interesting case is beta testing, where the effort distribution for the work package follows a double hump pattern, some initial work for part of the team members at the beginning of the testing, followed by a lighter period while the users exercise the system, followed by an intense period to analyze the data and dispose of any findings.

As shown by the figure, with the constraints put on the available resources – three developers and a project manager – it is unlikely that the system could be deployed by early April as the customer wanted. At this point the team could ask for additional resources, reorganize the work, e.g. relax the condition of not doing development work before the design concept has been approved, negotiate the scope, change the completion deadline or just take its chances.

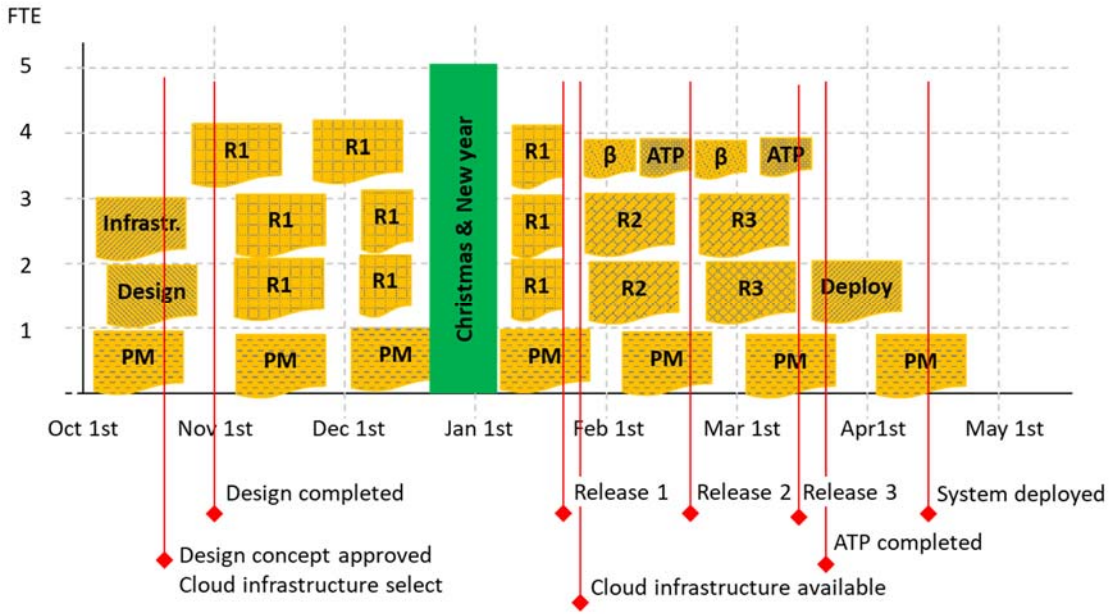


Figure 8 Staffing curve showing the chosen milestones for the AmazongLight project

Step 10

In Step 10, the milestone plan is completed by reading from the milestone scheduling canvas the approximated date in which the work associated with each milestone will be completed and assigning it as the due date for the milestone.

Detailing the milestone plan

A milestone plan is not directly "executable" as it does not prescribe the tasks to be done. To make the plan executable, the project team progressively refines each milestone's work package into the tasks necessary to complete it within the time and resource frames established by the package's time box. As work progresses, the milestone plan is updated to reflect new circumstances arising from the work completed or from changes in the project context, but since milestones are basically states or goals to be attained, the plan tends to be pretty robust.

Traditionally each of the refinement steps has been called a planning wave, see Figure 9, and the whole approach "Rolling Wave Planning" [18] [19].

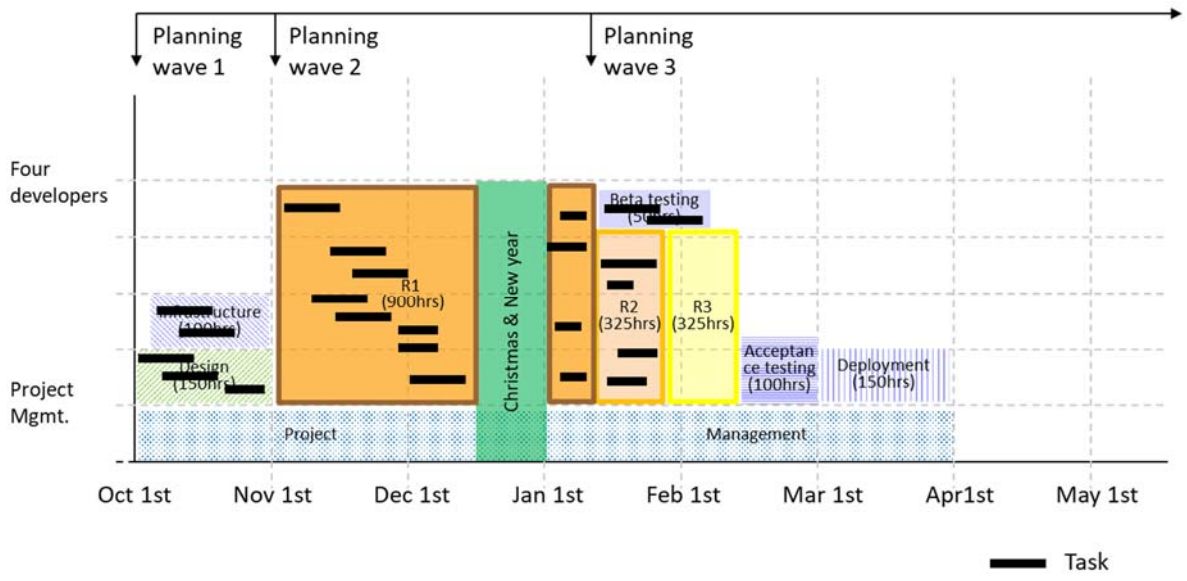


Figure 9 The rolling wave planning aproach

Milestone planning and agile software development

In an agile development context, milestone plans seat in between the iteration and the product levels of the agile planning onion, see Figure 10, but it does a little bit more than just providing a list of desired capabilities and a due date for them: a milestone plan provides an organizing frame that allows everyone to understand the intent of the work to be done and how it aligns with the overall project timeline.

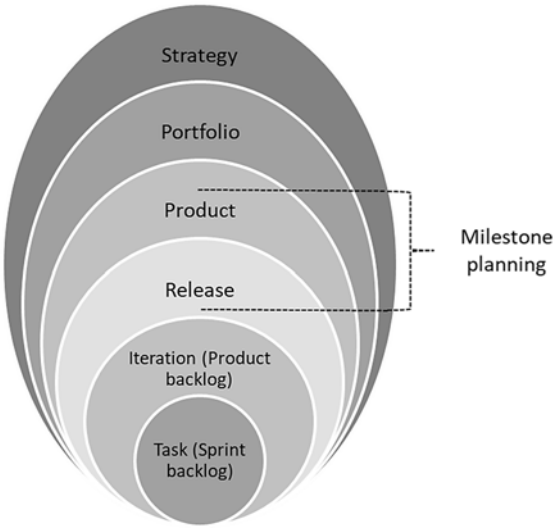


Figure 10 The Agile Planning Onion and Milestone Planning

The VMP method is just one of the techniques that can be used to produce a milestone plan, but its participatory and visual nature makes it a good fit for agile development, a process that relies heavily in socialization [20] [21] to achieve a shared understanding of the work to be done.

In the context of the Agile Planning Onion, the Release and Product plans and not the biweekly or monthly prioritization ascribed to the iteration planning meeting, drive the order in which the product backlog items are worked on. Similarly in the case of the milestone plan, see Figure 11, it would be the plan that drives the order of execution

To use milestone planning we need to do two things: rightsize the work packages before populating the product backlog and prioritize its items according to the milestone plan and not to whatever is deemed most important at a given time by the product owner and the team choices limited to those aligned with the plan. It goes without saying that the plan could be changed or abandoned if the need arises.

A fundamental tenet of agile development is that the development effort of any schedulable work item, e.g. user stories and tasks must fit within the confines of one iteration. This is what makes the inspect and adapt mechanism an effective vehicle to steer a project. If items were to span multiple iterations, the team will likely crash before even knowing it had to change direction. If any work item in a work package exceeded this prescription, it must be broken down before it can be moved into the product backlog as a set of product backlog items. This could be done during the backlog grooming sessions, a kind of look ahead planning, in which the work to be performed in the coming iterations is readied for the Sprint or iteration planning meeting.

At each Sprint or iteration planning meeting the items in the product backlog will be further broken down into tasks whose effort could be measured in hours and not days or weeks.

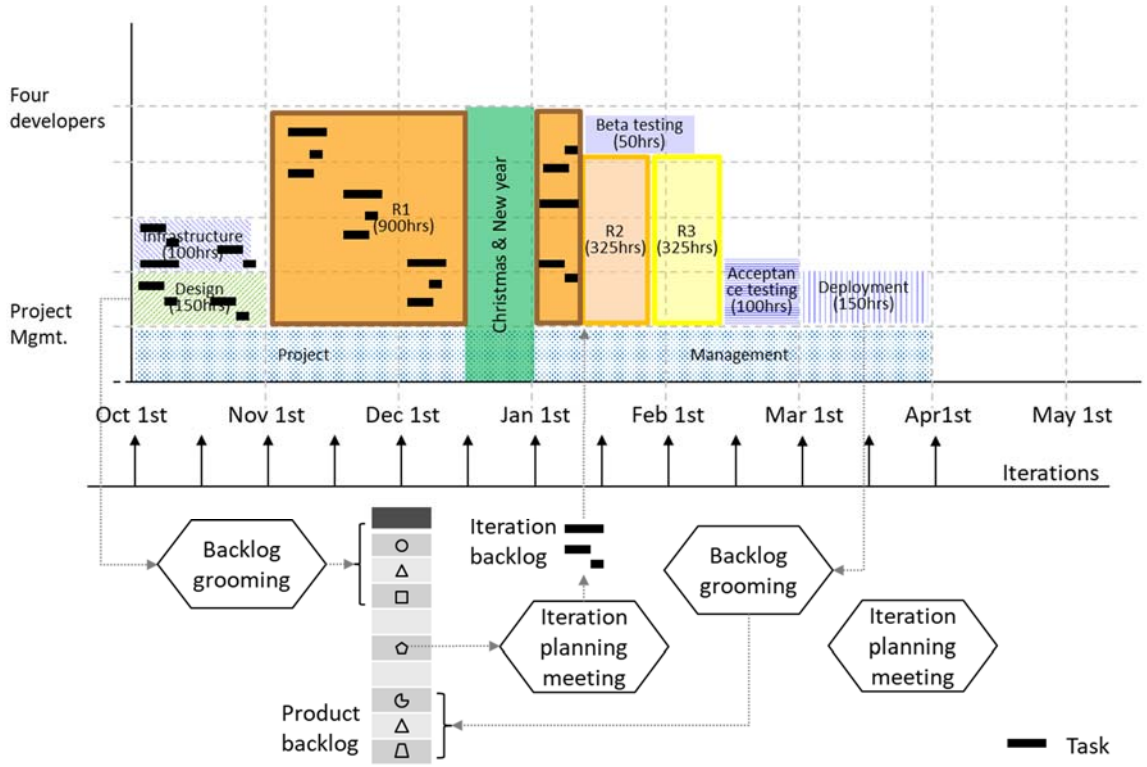


Figure 11 Milestone planning in the context of agile software development

Summary

Any project of any size or consequence needs an organizing principle which is shared, understood and accepted by its members. Without it, project members are likely to struggle in knowing what to do next or how to build whatever the project is trying to accomplish. An organizing principle helps the project members make congruent decisions in the absence of specific directions. While milestone planning provides the project with the requisite organizing principle, by facilitating the exchange of knowledge between individuals and accelerating team learning, the collaborative construction of the plan reinforces the well-known benefits of team members' engagement, buy in and ownership.

This paper improves on them by integrating both approaches through the introduction of a new construct, the milestone planning matrix, which makes explicit the allocation of WBS items to milestones and by compelling team members to participate in the planning process through the physically positioning of the reified work packages on the milestone scheduling canvas.

The method is simple and replicable, and as mentioned at the beginning it has been successfully taught to students in the Master of Software Engineering program at Carnegie Mellon University and at two organizational workshops led by the author. In all opportunities the participants manifested their understanding of the technique and were able to create a milestone plan by themselves after a two-hour tutorial. In the case of the students at Carnegie Mellon, three out of six teams trained in the method, chose it as their planning method for their capstone projects. These observations although anecdotal, point towards the method’s ease of use and its value in organizing a project.

Bibliography

[1] E. Andersen, "Warning: activity planning is hazardous to your project's health!".

[2] E. Andersen, K. Grude and T. Haug, Goal Directed Project Management, 4th., London: Kogan Page, 2009.

[3] R. Turner, The Handbook of Project Based Management, 3rd., McGraw-Hill, 2009.

[4] H. Ballard, *The Last Planner System of Production Control,* Birmingham: University of Birmingham, 2000.

[5] P. Brittle, C. Gaedicke. and R. Akhavian, "Perspective of the Last Planner: Effectiveness of the Traditional Critical Path Method in Comparison With the Last Planner System," *Journal of Supply Chain and Operations Management,* vol. 16, no. 1, 2018.

[6] A. Mossman, "Last Planner, 5 + 1 Crucial & Collaborative Conversations for Predictable Design & Construction Delivery," 12 2015. [Online]. Available: http://www.villego.com/wp-content/uploads/2017/02/Mossman-2015-Last-Planner.pdf. [Accessed 6 4 2018].

[7] L. Lindlöf, "Visual Management - on Communication in Product Development Organizations," Chalmers University of Technology, Gothenburg, 2014.

[8] M. Jurado, "Visual Planning in Lean Product Development," KTH Royal Institute of Technology, 2012.

[9] J. Tjell and P. Bosch-Sijtsema, "Visual management in mid-sized construction design projects," in *8th Nordic Conference on Construction Economics and Organization*, Tampere, 2003.

[10] Environmental Protection Agency - Conflict Prevention And Resolution Center, "The Charrette: Redevelopment by Design," EPA, 2010.

[11] R. Roggema, The Design Charrette, Springer, 2014.

[12] J. Lewis, Mastering Project Management, 2nd., McGraw-Hill, 2008.

[13] National Aeronautics and Space Administration, "NASA Systems Engineering Handbook," National Aeronautics and Space Administration, Washington, 2007.

[14] Project Management Institute, "Practice Standard for Work Breakdown Structures," Project Management Institute, 2006.

[15] J. Hauser and D. Clausing, "The House of Quality," *Harvard Business Review,* no. May - June, 1988.

[16] M. Maurer, "Structural awareness in complex product design," Tech. Univ. Munchen, Munich, 2007.

[17] T. Browning, "Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities," *IEEE Transactions on Engineering Management,* vol. 63, no. 1, 2016.

[18] G. Githens, "Rolling Wave Project Planning," in *Proceedings of the 29th Annual Project Management Institute 1998 Seminars & Symposium*, Long Beach, 1998.

[19] Project Management Institute, A guide to the project management body of knowledge (PMBOK guide, PMI, 2017.

[20] K. Schwaber and M. Beedle, Agile Software Development with Scrum, vol. 1, Prentice Hall, 2001, pp. 2-10.