

Creating The Work Breakdown Structure

By Kim Colenso, Managing Principal, Artemis Management Systems

The Work Breakdown Structure (WBS) is the foundation for project planning and control. It is the connecting point for work and cost estimates, schedule information, actual work effort/cost expenditures, and accountability. It must exist before the project manager can plan these related and vital aspects of the project, and they all must be planned before the project manager will be able to measure progress and variance from plan. In order to perform this vital function, the WBS is at its core a hierarchy of deliverables or tangible outcomes. This article describes a proven approach for creating and implementing a WBS.

Before we get started, let's establish some terms to facilitate communication within this article¹.

- Phase – The Phase is a major component within the project life cycle. When creating a process-based WBS, the Phase is the highest level within the WBS. When creating a deliverables-based WBS, the term is probably not appropriate. For the purpose of examples, the Phase is at Level 1.
- Activity – The lowest level within the WBS. Activities are where deliverables are assigned to an individual. The individual will perform a work process to create the deliverable. Therefore, the Activity is a combination of deliverable and process. For the purpose of examples, the Activity is at Level 3.
- WBS Entry – A generic term for any level within the WBS, but always representing a deliverable. WBS Entries are decomposed into other WBS Entries, and at the lowest level, are decomposed into Activities. For the purpose of examples, the WBS Entry (at Level 2) will fall below the Phase and above the Activities at Level 3. In actual practice, there can be as many levels of WBS Entry as needed.

The WBS As Hierarchy

Before looking at how to build a WBS, it is generally a good idea to create a mental model for what we are going to build. The mental model I have used many times in the past is the Outline. If you go back to your time in grade school (and if you're as old as I am, that can be a challenge), you will remember that your teacher always wanted you to create an outline as the first step in writing a report. The outline had some very specific characteristics and rules, and if you didn't follow them, the teacher would apply 'corrective discipline', and you would make the necessary changes. Well, the Work Breakdown Structure has many aspects in common with the outline.

Topic				Project	Level 1	Level 2	Level 3
	I.				Phase 1		
		A				WBS Entry 1-1	
			1				Activity 1-1-1
			2				Activity 1-1-2
		B				WBS Entry 1-2	
			1				Activity 1-2-1
			2				...
	II.				Phase 2		
		A				WBS Entry 2-1	
			1				Activity 2-1-1
		

¹ Note: the naming of levels within a WBS is not standardized within the project management discipline, nor is it standardized within many industries. The terms I am using in this article are often used, but within your organization you may have different terms for these same concepts. Other terms that are often used to represent WBS entry levels include stage, step, task, and work package.

The Outline

The outline begins with a topic. It is the subject that the rest of the work will describe. The outline is a tree structure, and at its highest level the sections are numbered with Roman Numerals. Within each Roman Numeral heading, there are sub-headings with capital letters. Within each of the capital letter sub-headings, there are sub-sub-headings identified by Arabic numbers.

The rules for outlines are as follows:

- At any level, if you have one entry at that level, then you must have two or more, otherwise the level itself is meaningless.
- At any level, the content of the lower level entries is conceptually equivalent to the higher level entry that they are a part of.

The WBS

Like the outline, the WBS is also a hierarchical tree structure. At the highest level is the project. The project is not part of the WBS, just like the topic is not part of the outline. Within the project is a hierarchical structure that defines the content of the project, just like the outline defines the content of the topic. Therefore, the following general statements are valid:

- Although the WBS content does not have to roll-up conceptually like an outline does, this is a very good idea as it provides the project manager with the ability to validate the WBS before committing to a completion date.
- Any WBS entry, when decomposed into components, should have two or more components defined, otherwise the decomposition itself is meaningless.
- For any WBS entry that is decomposed into components: the content of the lower level entries should be conceptually equivalent to the higher level entry that they are a part of. In other words, the higher level entry should be a deliverable, and the lower level entries should be components of that deliverable.

Like the outline, when not done properly there will be corrective action applied, only you won't have to stay after school. In all likelihood, the project will be late, over budget, and not meet expectations...

Creating the Work Breakdown Structure

The WBS can be structured in either of two ways. The first approach structures the WBS primarily from a deliverables perspective, in that the highest level (Level 1) entries represent the major deliverables that the project is committed to create. The second approach is from a life cycle perspective, in that the highest level entries in the WBS correspond to the major phases of the life cycle.

Steps to create a deliverable-based structure:

1. Take the committed deliverables from your project charter, statement of work, or other project concept documentation. This list of deliverables becomes your Level 1 (highest level) entries within the WBS. All WBS Entries that directly correspond to deliverables should be named as noun deliverables or adjective/noun deliverables. Examples include "Specification" or "Design Specification".
2. Take each of these highest level entries, and decompose them into their component parts (each becoming a WBS Entry). Each component must be *logically distinct*, as everyone who sees the WBS needs to understand what the deliverable or outcome will be from each WBS Entry.

What logically distinct means is that the breakdown of a higher level deliverable to its lower level components must make sense. Each of the lower level components must be distinguishable as unique, and they must be recognizable as part of the higher level deliverable.

Continue the decomposition until you reach an appropriate level of detail.

This last very vague statement is intentionally so, as the level of detail in a WBS should be based on the complexity of the project, the level of risk in the project, and the level of control that the project manager needs to plan and manage the project. However, a general guideline in many IT organizations is that each lowest level entry in the WBS should be assigned to a single individual, and that individual should be able to complete it in 1 to 10 working days. This lowest level of decomposition is the Activity level. Activities should be named as active verb / adjective / noun deliverables. Examples include "Create Design Specification" or "Update Design Specification". By adding the active verb, you better communicate to the assigned team member not only what the outcome is (the deliverable) but you also communicate what kind of process the assigned person is going to perform (create or update). Note: You should never use terms like "perform", as they do not communicate what is expected.

3. When all committed deliverables have been decomposed to the appropriate level of detail (becoming Activities), examine each WBS Entry and Activity to see if there are required deliverables that are not already in the WBS but that will be needed to create something that already is in the WBS.

As an example, you may have a deliverable defined for a software component (system, subsystem, or function). However, to deliver this into the production environment, you may also need preceding deliverables such as test results, compiled code, design documentation, and requirements documentation. These preceding deliverables, even though they haven't been committed to, still must be created and therefore must be in the WBS.

Take all these required deliverables, and decompose them to the appropriate level of detail, just as you did for the committed deliverables.
4. Level the hierarchy to the extent that it is possible. At this stage of development, the WBS may have some Activities at level two, some at level three, and so on. See if the hierarchy can be modified so that the number of levels that Activities fall into is reduced to a short range.

One way to do this is to examine the number of Activities falling within a single WBS Entry. If the number is less than three to five, see if these Activities can be merged with another WBS Entry's Activities. If the number is more than 10, see if the WBS Entry can be split into two logically distinct components, each with its appropriate Activities. The general idea is to attempt to have each WBS Entry that decomposes into Activities have approximately 7 plus or minus 2 (5 to 9) Activities.

Do this for every WBS Entry, attempting to get each entry in the WBS to decompose into 5 to 9 lower level entries. This should be considered as a nice to have, and not a requirement. You should never make these changes if the merger or split of a WBS Entry does not make logical sense.

 - When evaluating whether to merge two WBS Entries, the question to ask is, "are these two deliverables really part of one deliverable, and is that deliverable distinct from all others?" If the answer is yes, then you should combine them, otherwise don't.
 - When evaluating whether to split a WBS Entry with too many components, the question to ask is, "does this WBS Entry deliverable have two or more major components, and can the already defined lower level deliverables be combined into these proposed higher level ones?" If the answer is yes, then you can split the WBS Entry. If the answer is no, then leave it as is.
5. When you think you have a completed WBS, validate it using a bottom-up approach. A bottom-up validation works like this:
 - For each WBS Entry that decomposes into Activities, ask yourself the question: "If I had all the deliverables from each of these Activities, would my WBS Entry deliverable be complete?" If the answer is yes, move on to the next WBS Entry. If the answer is no, add in the missing Activities.
 - Once the evaluation of the lowest level WBS Entries and Activities is complete, examine the next higher level of WBS Entries. Keeping with our three-level example, for each Phase ask: "If I had the deliverables from the WBS Entries that are part of this Phase, would the Phase deliverable be complete?" If the answer is yes, move on to the next one, if the answer is no then add in the missing WBS Entries or go back to step 4 and rebalance the hierarchy, or both.

Note: Validating the completeness of the WBS is extremely important, as a major reason for projects being late and over budget is the originally planned scope of the project was incomplete, and there was a significant amount of unplanned work that had to be done. Since this unplanned work was not part of the original plan, and it consumed resources that were originally scheduled for other project work, the schedule inevitably slips.
6. When you have completed your bottom-up validation, it is now appropriate to re-evaluate the entire WBS one last time by comparing the currently defined WBS deliverables to the originally defined objectives for the project. Ask yourself the question, "If I had all these deliverables, would I achieve the planned objectives for the project?" If the answer is yes, you can move on to the next step. If the answer is no, you still have a lot of work to do.

An Example WBS

The following example has the terminology used in this article in the leftmost column. In the middle column, is an example numbering system that is typical of many that are used, and the far right column provides the names of the entries as they would appear in your project plan.

Phase 1 – WBS Level 1	1.0	Requirements Phase
WBS Entry - WBS Level 2	1.1	Business Objectives
Activity – WBS Level 3	1.1.1	Create Draft Objectives
Activity – WBS Level 3	1.1.2	Review Draft Objectives
Activity – WBS Level 3	1.1.3	Update Objectives
Activity – WBS Level 3	1.1.4	Approve Objectives
Milestone – WBS Level 3	1.1.5	Business Objectives Complete
WBS Entry – WBS Level 2	1.2	Draft Requirements
Activity – WBS Level 3	1.2.1	Interview Stakeholders
Activity – WBS Level 3	1.2.2	Write Draft Requirements
Activity – WBS Level 3	1.2.3	...
WBS Entry - Level 2	1.3	Final Requirements
...		
Phase 2 – WBS Level 1	2.0	Design Phase
...		
Phase 3 – WBS Level 1	3.0	Development Phase
...		
Phase 4 – WBS Level 1	4.0	Test Phase
...		

This example is a typical waterfall type of project plan. Each of the Phases are major process steps, and as such usually also happen to be major deliverables.

Steps to create a life cycle-based structure:

1. Take each of the major Phases from the life cycle, and use them as the highest level entries in the WBS (Level 1).
2. Take the committed deliverables (as in the deliverables approach) and use them to create the next level WBS Entry (Level 2) under the Phases. Place these committed deliverables within the Phase where they will be created.
3. Decompose the rest of the WBS just as in the deliverables approach.

Templates

Both the deliverables-based and the life cycle-based approaches can take advantage of using standard WBS structures and standard project templates. There is considerable time savings for the project manager when he or she does not have to spend the time to develop a WBS from scratch, not to mention all the additional planning information that gets attached to the WBS like estimates, dependencies, and generic resource assignments.

Completing The Work Breakdown Structure

Once the deliverables hierarchy has been established, detailed, and validated, it is time to move on to the final steps in completing your WBS. First up is compensating for the uncertainty that exists in your project. This is usually referred to as Risk Planning. My favorite technique to compensate for uncertainty within the WBS is as follows.

Contingency Planning

1. Take those risk events that have a schedule or cost impact from your project charter, statement of work, risk management plan, or wherever you have risk events documented, and create a contingency activity that is named for the risk event.
2. Examine the WBS, and determine which Activity or Activities would be impacted by this risk event, should it occur. If there is only one Activity that will be impacted, insert the contingency activity immediately after the impacted Activity in the WBS. If there is more than one, insert the contingency activity immediately after the impacted Activity that you think will be scheduled first.
3. Using the documented risk event impact and probability, calculate the duration of the contingency Activity. As an example, if the risk event had an estimated impact of four weeks, and an estimated probability of 25%, then the scheduled duration for the contingency Activity would be 1 week. ($25\% \times 4 \text{ weeks} = 1 \text{ week}$).
4. When you define your dependency network, you will want to create a finish – start dependency link from each impacted Activity to the appropriate contingency activity. The normal successor to the impacted Activity should be defined as a successor to the contingency activity. When scheduled, you will have incorporated a compensating factor for the schedule uncertainty in your project.
5. As for compensating for the cost uncertainty, that is even easier. Assign to each contingency Activity the type of resource that reflects the cost in the impact, and the estimated amount. As an example: if the impact of the risk event is \$100,000, should it occur, and the likelihood is 40%, then you would assign a budget resource to the contingency activity, and plan for a value of \$40,000. ($40\% \times \$100,000 = \$40,000$).

Milestones

The final step in completing your WBS is the inclusion of Milestones. Milestones are major events that occur during the course of the project, and should have the following characteristics:

- They are a point in time, and therefore have no duration. In most project management software, you define a milestone by setting the duration of an Activity to zero.
- Milestones represent major points of progress within the project, and will provide a high level means of communicating. Therefore they should represent events that have significant importance for project stakeholders. As an example, if the project charter or statement of work included committed deliverables that the stakeholders felt were important to the success of the project, then each of these committed deliverables should have a milestone representing its completion.
- Milestones should be named as past tense events. As an example, “Acceptance Test Completed”. This communicates two facts: the life cycle phase or the major deliverable, and the fact that it is completed. Milestones can also represent the beginning of a process, as in “Acceptance Test Started”, but the completion milestones should always be included in the WBS whereas the start milestones are optional.

Conclusion

This process for creating a Work Breakdown Structure is a major, but not the only, planning deliverable a project manager needs to complete when planning a project. It is true that you can't complete your project planning without a WBS, but it is not true that the WBS is the only planning you must do. Once the WBS is completed, you must develop a dependency network, estimate Activity duration, acquire and assign resources, calculate and refine the schedule, and baseline the plan. Many of these steps can be performed somewhat in parallel, but it is absolutely true that you cannot complete the dependency network, the estimates, the work assignments, or the schedule until the WBS is complete.

Author

Kim Colenso is the Managing Principal of the Methodology and Project Management Process consulting practice with Artemis Management Systems, with over 20 years experience in IT and over 15 years experience in project management. He is PMP certified by the Project Management Institute (PMI), and is a Certified Software Quality Engineer by the American Society for Quality (ASQ). Currently he is serving as the volunteer project manager for PMI's Practice Standard for Work Breakdown Structures project, which will provide guidance to project management practitioners worldwide on using the WBS as a project management tool.