ECON 180

Introduction to Principles of Microeconomics and Financial Project Evaluation

Lecture 32: Work Breakdown Structures

November 24, 2021

Version 1

Learning Objectives

- Understand how to construct a Work Breakdown Structure (WBS).
- Be aware of how to incorporate iterative and overly long tasks into a WBS.
- Be able to convert between indented list and flow chart WBS formats.
- Understand how to create and use a WBS Dictionary.

Required Viewing

- Girdler, A. (2020, September 2). Work Breakdown Structure [WBS EXPLAINED]. https://youtu.be/BVcd9uy9kuQ

Recommended Reading

- Engineering Economics Chapter 11, Section 11.3.1
- Engineering Economics Chapter 11, Sections 11.3.2 and 11.3.3 (excluding 11.3.3.1 and 11.3.3.2)
- Lanier, L. (2018, October 15). 'Red Dead Redemption's 100-Hour Work Weeks Spark Video Game Industry Outrage' [Web Page]. Retrieved from https://variety.com/2018/gaming/news/red-dead-redemption-2s-100-hour-work-weeks-spark-video-game-industry-outrage-1202980239/

Optional Reading: WBS and Crunch Time

- ea_spouse. (2004, November 10). EA: The Human Story [Blog Post]. Retrieved from https://ea-spouse.livejournal.com/274.html
- Feng, J., Zhang, F. & Li, M. (2009). Research on Work Breakdown Structure of IT Project. *2009 International Conference on Management and Service Science*. Retrieved from https://ieeexplore-ieee-org.ezproxy.library.uvic.ca/document/5303208
- Lanier, L. (2018, October 15). 'Red Dead Redemption's 100-Hour Work Weeks Spark Video Game Industry Outrage' [Web Page]. Retrieved from https://variety.com/2018/gaming/news/red-dead-redemption-2s-100-hour-work-weeks-spark-video-game-industry-outrage-1202980239/
- Lee, W., Hsu, K., Lee, J. & Kuo, J.Y. (2012). Applying Software Effort Estimation Model Based on Work Breakdown Structure. *2012 Sixth International Conference on Genetic and Evolutionary Computing*. Retrieved from https://ieeexplore-ieee-org.ezproxy.library.uvic.ca/document/6457256
- Press Release: 62% of Developers Indicate Their Job Involves Crunch Time [Web Page]. (2015, September 18). Retrieved from https://web.archive.org/web/20190803110350/https://www.igda.org/news/251411/Press-Release-62-of-Developers-Indicate-Their-Job-Involves-Crunch-Time.htm
- Ruskin, A. R. (2004). 100% product-oriented work breakdown structures and their importance to system engineering. *2004 IEEE Aerospace Conference Proceedings* [IEEE Cat. No.04TH8720]. Retrieved from https://ieeexplore-ieee-org.ezproxy.library.uvic.ca/document/1368217

Case Studies: WBS (I)

- Khera, R., Ransom, P. & Speth, T. F. (2013). Using work breakdown structure models to develop unit treatment costs. *Journal of the American Water Works Association*, 105(11), E628-E641. https://www-jstor-org.ezproxy.library.uvic.ca/stable/jamewatworass.105.11.e628
- Latief, Y., Nurdiani, D. & Supriadi, L.S.R. (2018). Development of work breakdown structure (WBS) dictionary for the construction works of lower structure steel bridge. *International Conference on Sustainable Civil Engineering Structures and Construction Materials*, *258*, 02003. https://doi.org/10.1051/matecconf/201925802003
- Li, D. & Lu, M. (2017). Automated Generation of Work Breakdown Structure and Project Network Model for Earthworks Project Planning: A Flow Network-Based Optimization Approach. *Journal of Construction Engineering and Management, 143*(1), 04016086. https://ascelibrary-org.ezproxy.library.uvic.ca/doi/full/10.1061/%28ASCE%29CO.1943-7862.0001214
- Potonski, M. (2015). Application of the Work Breakdown Structure In Determining Cost Buffers in Construction Schedules. *Archives of Civil Engineering*, *61*(1), 147-161. https://doi.org/10.1515/ace-2015-0010
- Rahman, W. A. Z. W. A. & Zaki, N. I. M. (2019). Work breakdown structure application for man-hours calculation in hull construction shipbuilding in Malaysia. *Cogent Engineering*, *6*(1), 1599524. https://doi-org.ezproxy.library.uvic.ca/10.1080/23311916.2019.1599524

Case Studies: WBS (II)

- Sharon, A. & Dori, D. (2015). A Project-Product Model-Based Approach to Planning Work Breakdown Structures of Complex System Projects. *IEEE Systems Journal, 9*(2), 366-376. https://ieeexplore-ieee-org.ezproxy.library.uvic.ca/document/6748857
- Siami-Irdemoosa, E., Dindarloo, S. R. & Sharifzadeh, M. (2015). Work breakdown structure (WBS) development for underground construction. *Automation in Construction*, *58*, 85-94. https://doi-org.ezproxy.library.uvic.ca/10.1016/j.autcon.2015.07.016
- Sutrisna, M., Ramanayaka, C.D.D. & Goulding, J.S. (2018). Developing work breakdown structure matrix for managing offsite construction projects. *Architectural Engineering and Design Management, 14*(5), 381-397. https://doi-org.ezproxy.library.uvic.ca/10.1080/17452007.2018.1477728

Relevant solved problems (required text)

- WBS Theory: 11.5, 11.6, 11.7, 11.8
- WBS Applications: 11.27, 11.28, 11.29
- Gantt Charts: Example 11.2, Review Problem 11.3, 11.9, 11.10, 11.11, 11.12, 11.30, 11.35, 11.37, 11.40.a., 11.41 (first part)
- AoN Diagrams: Example 11.3, Review Problem 11.2, 11.13, 11.14, 11.15(b), 11.16(b), 11.24(a), 11.31(a), 11.36(a), 11.38(a), 11.39 (first part), 11.40(b)

ESSENTIALS (20 slides)

‘Red Dead Redemption 2’s’ 100-Hour Work Weeks Spark Video Game Industry Outrage

By LIZ LANIER

October 15, 2018
(From the required reading.)

A crisis in game software engineering

- According to the 2015 Developer Satisfaction Survey by the International Game Developers Association…
- 62% of developers indicate their job involves crunch time – a period of team-wide overtime to meet a deadline.
- During crunch, >50% work >60 hours a week, and 17% work >70 hours.
- 37% of developers undergoing crunch are not compensated for the extra time.
- In 2004, ‘EA Spouse’ blew the whistle on Electronic Arts, which had mandatory 9 am – 10 pm 7-day work weeks without extra compensation during peak crunch. Crunch was a normal, expected part of the job.

This is a job for… economics?!?

- Economics is the study of the allocation of scarce resources.
- The time available to software engineers is a scarce resource.
- Anecdotes suggest that this resource is currently being inefficiently allocated. (Overworked employees are often inefficient.)
- Let's see if we can do something about that!
- Today, we'll take a first look at the most fundamental project planning technique: the work breakdown structure.
- This will be the first step on the path to Critical Path Management, a technique that will provide useful insights into how to improve on current software engineering practices in the video game industry.

Pinning down a project

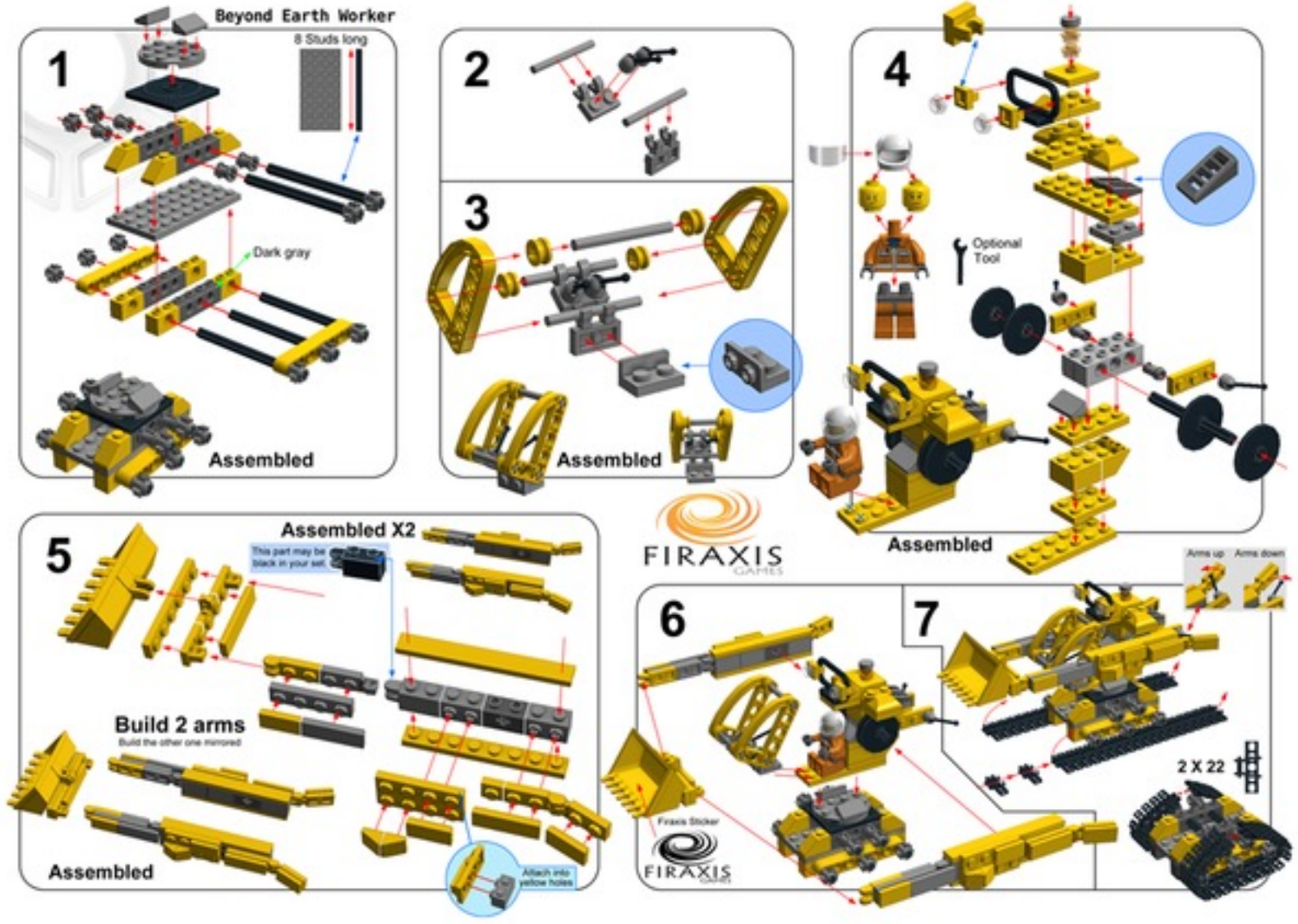
- Work breakdown structures (WBS): what tasks is a project made of?
  - What will the project cost?
  - How do we delegate responsibility?
- Gantt charts:what sequence do the tasks take place in?
  - How long will the project take?
- Critical Path Management: What are the dependencies between tasks?
  - Where's the wiggle room?
  - Where are 'crunch time' funds best spent?

Work breakdown structure (WBS)

- Can be thought of as a very thorough ‘to-do’ list.
- Breaks work to be done into a list of tasks.
- Use 1: Identifies activities and resources needed
- Use 2: Helps avoid duplication of resources

- What it looks like:
- Indented list or tiered organizational chart with each layer representing a level of detail.
- For larger projects, a list rather than a diagram is used.

A very simple work breakdown structure.

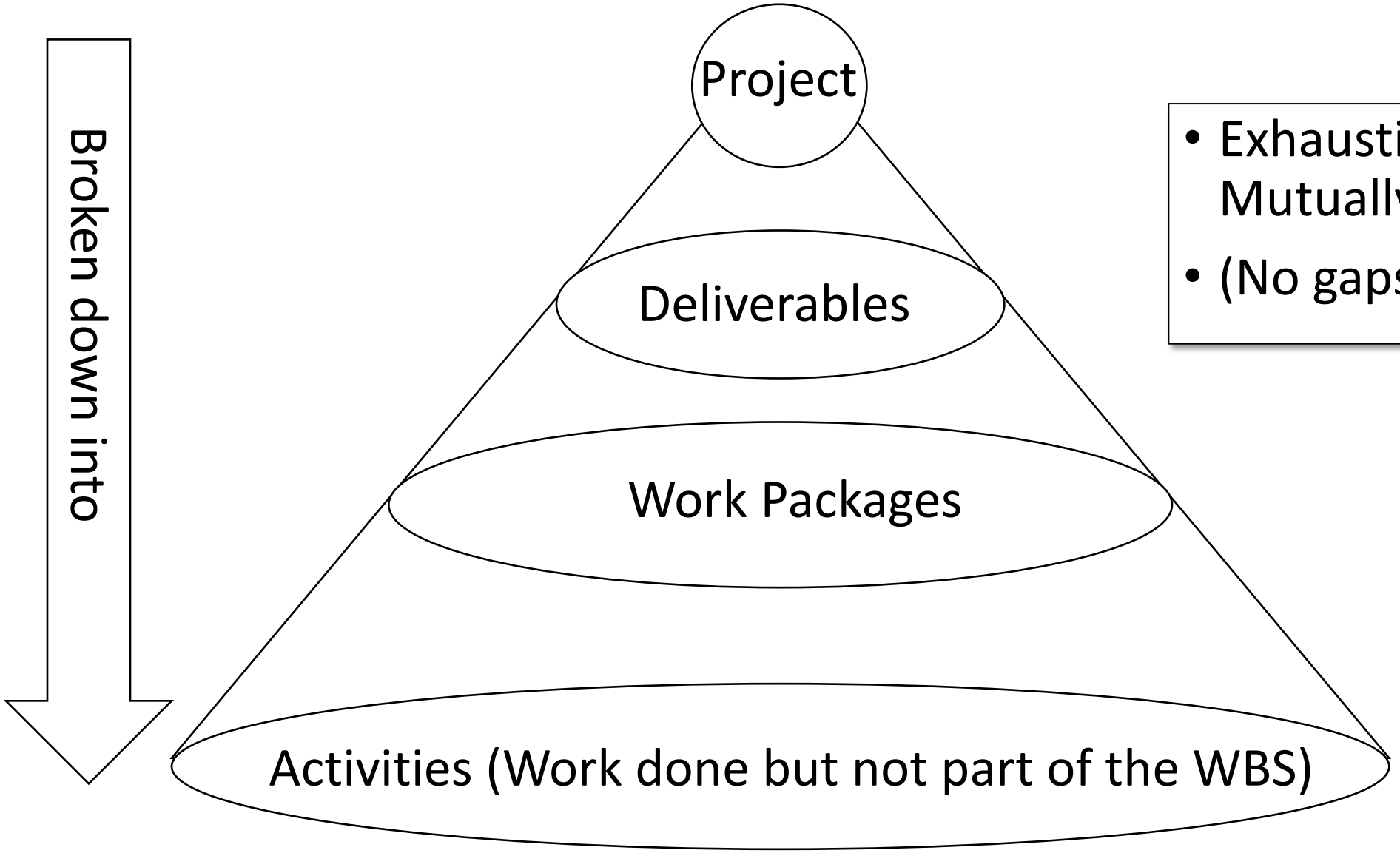


https://ideas.lego.com/projects/83016

Benefits of using an explicit WBS (Ruskin)

- Tasks are identified and arranged in terms of inputs, outputs and sequence.
- Minimizes chances of overlapping or overlooking tasks, and of over-committing individuals.
- Accountability: each element of the WBS has an accountable individual responsible for negotiating with the element's 'customer', the individual responsible for the relevant super-element.
- The WBS with the accountable individuals' names becomes the project organizational chart.
- →The WBS facilitates a clear chain of communication.

Basic Work Breakdown Structure (WBS) hierarchy



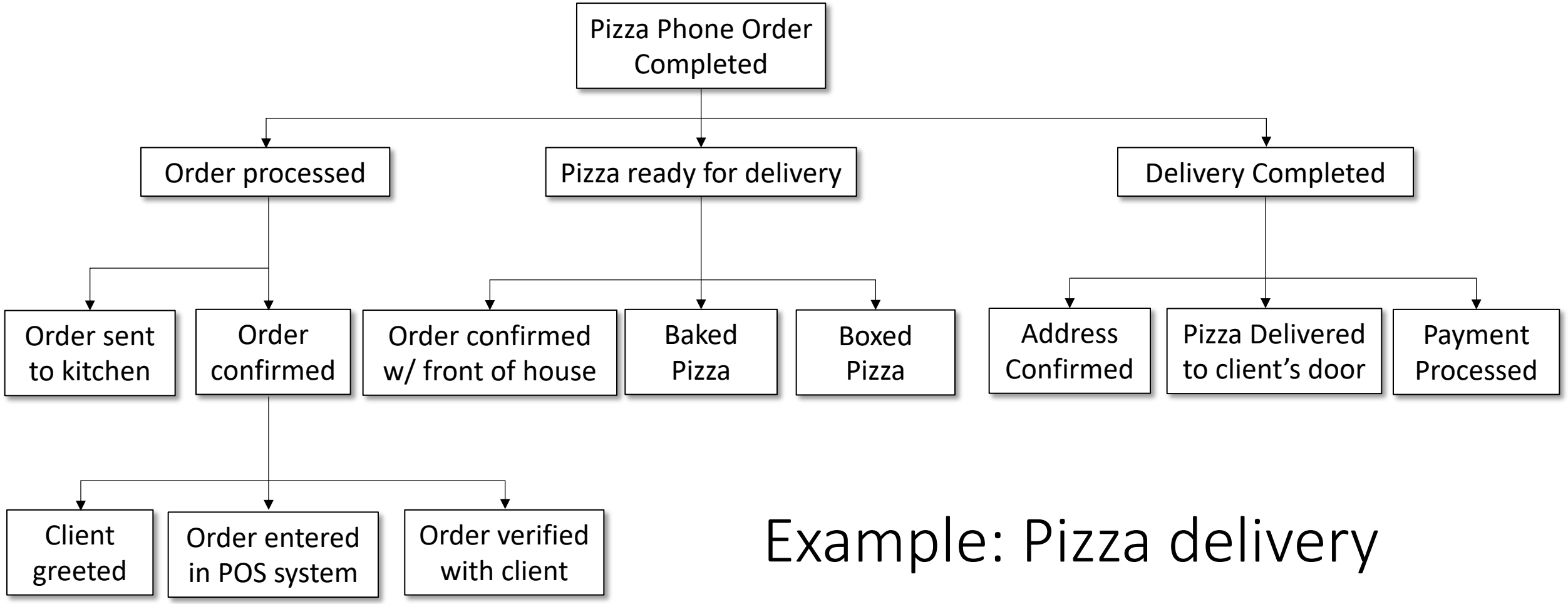
- Exhaustive and Mutually Exclusive
- (No gaps, no overlap)

What is a deliverable?

- An item of work that someone can take responsibility for.
- It is an *outcome*, not an action.
- *Responsibility* is very important. A deliverable should always include…
- Required characteristics for delivery to be complete.
- Person(s) responsible for ensuring delivery (e.g. pizza delivery person).
- Person(s) responsible for accepting item (e.g. client who ordered pizza).

Deliverables, Work Packages and Activities

- These differ mostly in who the work is delivered to.
- A **deliverable** is usually delivered to a client or supervisor.
  - e.g. Pizza at the door, delivered by delivery person to client.
- A **work package** is a deliverable at the lowest level of the WBS.
  - Work packages are delivered internally within the team/firm.
  - e.g. Pizza delivered by the kitchen to the delivery person.
- For most **activities**, the same persons are responsible for delivery and acceptance. Activities are not part of the WBS.
  - e.g. Tossed Pizza dough delivered by the kitchen to the kitchen as part of making a pizza.



Example: Pizza delivery

Indented List format

- The flow chart format above is very useful for seeing at a glance how a project is structure, BUT blows up very quickly.
- In practice, only the top 2-3 levels are usually graphed.
- For more detail, deliverables etc. are listed as an indented list.
- WBS items are often labeled using an **index** that of their place in the project, e.g. 1.2.1 – Intermediate deliverable 1 of deliverable 2 of Project 1.
- This index is extremely important, as it provides a summary of a task's place in the project. You should always include it in a WBS, even if the task has a separate 'Task ID' assigned to it.

Indented List Example

WBS Code	Title	Responsibility
1	**Complete Pizza Phone Order**	
1.1	**Process Order**	
1.1.1	**Confirm Order**	
1.1.1.1	Greet Customer	Front of House
1.1.1.2	Take down order	Front of House
1.1.1.3	Verbally verify order	Front of House
1.1.2	Send order to kitchen	Front of House
1.2	**Ready pizza for delivery**	
1.2.1	Verbally confirm order	Kitchen
1.2.2	Bake Pizza	Kitchen
1.2.3	Box Pizza	Front of House
1.3	**Complete Delivery**	
1.3.1	Confirm address with POS	Delivery Person
1.3.2	Deliver pizza to address	Delivery Person
1.3.3	Process payment	Delivery Person

It is standard practice to write **summary tasks** in **bold** text. Summary tasks are those which are made up of intermediate deliverables, and do not have their own assigned responsibility. (They inherit that of their component tasks.)

While baking the pizza is a long process that includes kneading the dough, assembling toppings, etc., all of these steps are delivered by the Kitchen to the Kitchen. They are therefore ***Activities*** and not included in this WBS.

Is the top level always coded as ‘1’?

- Nope! Consider a car company with MANY projects going on at once…
- …also, the lower levels don’t *have* to be coded as numbers (it’s just convenient).

- WBS codes such as SEDAN2018.A.A.F.M are just as valid as 1.1.1.6.13
- It’s the *dots* that matter: they tell you what level you’re on.
- Between dots, pick something meaningful and easy to keep track of:
- No duplicate labels on any one level!

		Name	Duration	Resource Names	Main Task	Start	Finish
1		**REQMS**	9.75 days			2007/1/1 ...	2007/1/...
2		**Use Case 1: Edit Requirements**	9.75 days		YES	2007/1/1 ...	2007/1/...
3		**Implementation**	7 days			2007/1/1 ...	2007/1/...
4		Service Module	3 days	Daniel		2007/1/1 ...	2007/1/3 ...
5		GUI	4 days	Tim		2007/1/2 ...	2007/1/5 ...
6		Web Interface	4 days	Daniel		2007/1/4 ...	2007/1/9 ...
7		**Test**	4.75 days			2007/1/8 ...	2007/1/...
8		Integration Testing	2 days	Albert		2007/1/8 ...	2007/1/9 ...
9		System Testing	1.75 days	Albert		2007/1/11 ...	2007/1/12 ...
10		**Use Case 2: Edit Use Cases**	10 days		YES	2007/1/1 ...	2007/1/...
11		**Implementation**	8 days			2007/1/1 ...	2007/1/...
12		Service Module	5 days	Daniel		2007/1/15 ...	2007/1/19 ...
13		GUI	3 days	Tim		2007/1/17 ...	2007/1/19 ...
14		Web Interface	3 days	Daniel		2007/1/22 ...	2007/1/24 ...
15		**Test**	5 days			2007/1/2 ...	2007/1/...
16		Integration Testing	1 day	Albert		2007/1/22 ...	2007/1/22 ...
17		System Testing	4 days	Albert		2007/1/23 ...	2007/1/26 ...

Software Engineering Indented List WBS Example (Lee et al.)

The Top Down Approach and Consistency

- Define the major deliverable
- Divide into intermediate deliverables
- Divide each deliverable into work packages
- *How* you divide the work depends on which point of view you choose.
- Components? (Interface, Engine) Functions? (Plan, Launch, Review, Test) Phases? (Design, Construction, Cleanup) Location? (Front/Back of House)
- Pick ONE approach and stick to it.
- Mixing them can lead to overlap and confusion.
- Regardless of approach, the bottom level of the WBS should be the same.

WBS Best Practices

- Items in a WBS should represent deliverables.
  - In particular, deliverables that someone can be responsible for.
  - This helps in determining roles and responsibilities.
- Items should be mutually exclusive and exhaustive (no gaps, no overlap)

- IF best practices are followed, the WBS can be ‘rolled up’ from the bottom to obtain a cost estimate for the project.
- If one or more of the three best practice qualities are not satisfied, this estimate will be off.

Yes, you need a WBS Dictionary.

- The WBS dictionary is a document where team members can look up information about any given task.
- This information is crucial for scheduling and cash flow management.
- It should include…
- WBS Label and Task ID (these can be the same, but are often not)
- Estimated duration, predecessors and milestones.
- Required characteristics of the deliverable, person(s) responsible for delivery and acceptance.
- Assigned resources and costs of the same, or a link to the relevant resource dictionary (reference list of resource details and cost.)
- A description of the work, including component activities, or a link to the relevant paragraph in a longer ‘statement of work’.
- Any assumptions made.

Estimated duration, predecessors and milestones?

- Estimated duration: At first, you won't know exactly how long tasks will take. Start with an estimate, and revise as more info comes in.
- Predecessors: WBS items that must finish before the WBS item in question can start. e.g. The pizza must be baked before boxing it for delivery.
- Milestones: dates by which the WBS item should reach a certain level of completion. e.g. Debugged demo version of software by trade show.
- These can be thought of as *tasks with a duration of zero* (and in fact, that's how they're often implemented in project planning software).

An Abbreviated WBS dictionary

(Enough information for a Gantt chart, below)

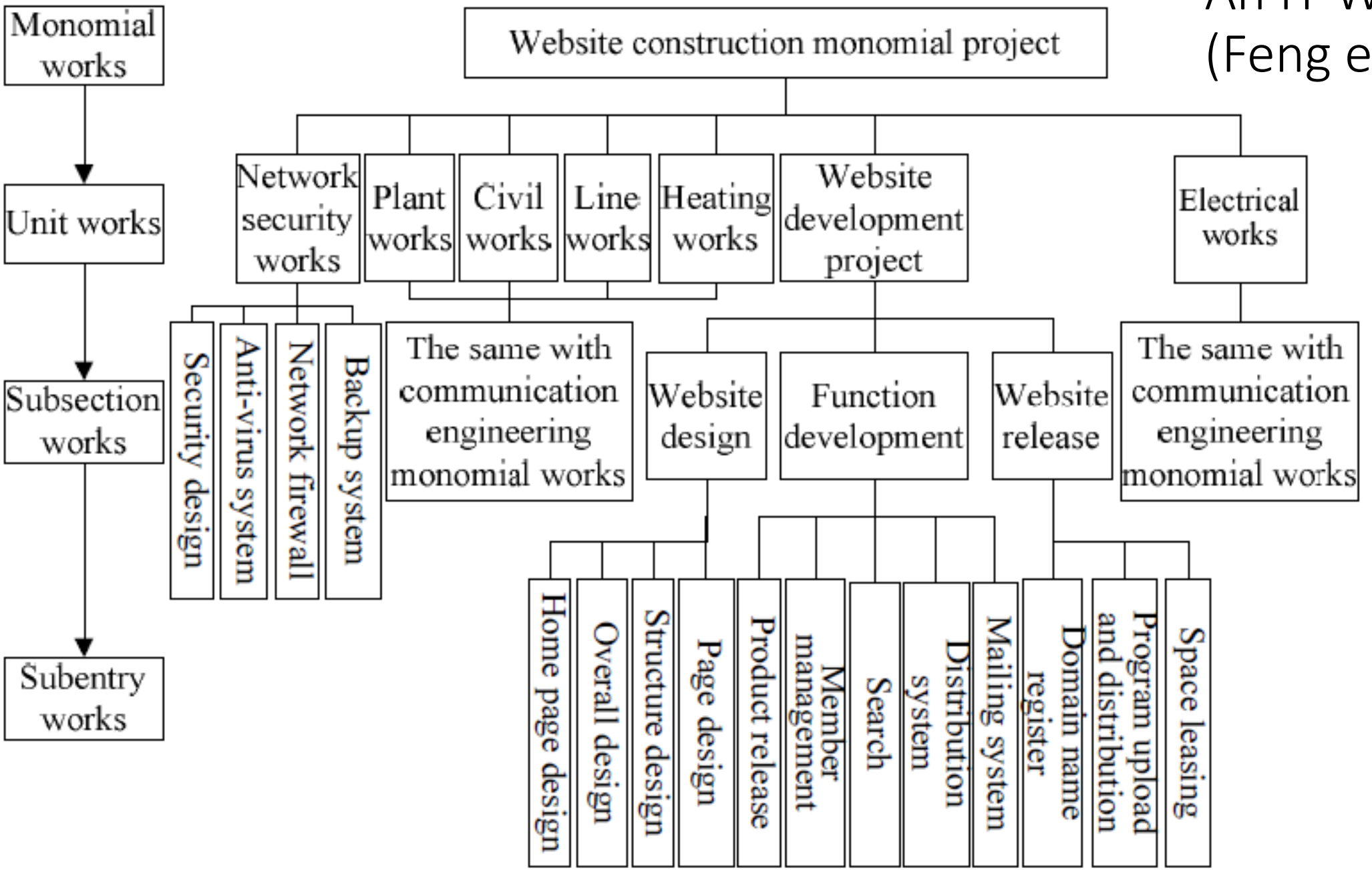
ID	WBS Code	Title	Responsibility	Duration (Minutes)	Predecessors
1	**1**	**Complete Pizza Phone Order**		*40*	
2	**1.1**	**Process Order**		*10*	
3	**1.1.1**	**Confirm Order**		*8*	
4	1.1.1.1	Greet Customer	Front of House	2	-
5	1.1.1.2	Take down order	Front of House	4	4
6	1.1.1.3	Verbally verify order	Front of House	2	5
7	1.1.2	Send order to kitchen	Front of House	2	3
8	**1.2**	**Ready pizza for delivery**		*14*	
9	1.2.1	Verbally confirm order	Kitchen	2	7
10	1.2.2	Bake Pizza	Kitchen	10	9
11	1.2.3	Box Pizza	Front of House	2	10
12	**1.3**	**Complete Delivery**		*30*	
13	1.3.1	Confirm address with POS	Delivery Person	2	7
14	1.3.2	Deliver pizza to address	Delivery Person	10	8
15	1.3.3	Process payment	Delivery Person	6	14

Durations in *italics* are calculated from the component tasks.

AFTER HOURS

- An IT WBC (1 slide)
- When do you stop? (1 slide)
- Law of Large Numbers (2 slides)
- Two curve balls (2 slides)

An IT WBS (Feng et al.)



When do you stop?

- Do you require two or more intermediate deliverables? Keep going.
- Can you accurately estimate the time/resources needed?
- If not, keep going. Smaller, focused work packages are easier to estimate, AND the law of large numbers works in your favour.
- If you assign the deliverable to someone, will they know what to do?
- If not, keep going. This is dependent on who you have on your team – a team of specialists will need fine-grained tasks.
- Rule of thumb: No more than 2 calendar weeks or 80 person hours of work at the bottom (*activities* can be much shorter).

Zooming in: The Law of Large Numbers and WBS

- Estimates are by definition subject to error. A given estimate may be lower or greater than the true value.
- If you break down a project into its component work packages, and estimate the time taken for each work package…
- …even if for each individual work package your estimation accuracy is no better than for the project as a whole…
- …some of the errors will cancel out when you add up the work package estimates, leading to a better estimate of the project as a whole.
- (Assuming the estimates are independent enough that some overshoot, and others undershoot the true value.)

A Quick Example

	Duration	Estimate	Off By
Complete Pizza Order	**40**	**30**	**25%**
Process Order	10	5	50%
Ready Pizza for Delivery	14	21	50%
Delivery & Payment	16	12	25%
Sum of Task Durations	**40**	**38**	**5%**

Each individual task estimate is no more accurate than the estimate of the project as a whole, but some of the estimation error cancels out when adding the task estimates, yielding a more accurate prediction. For this to work, estimation errors can't be highly correlated across tasks.

Curve Ball 1: Drafts and Iterations

- How do you handle *iterative* deliverables with an unknown number of iterations? (e.g. debugging code, revising a draft report)
- Option 1: Define ‘debugged code’ as a deliverable and assume a duration.
- Option 2: Assume a number of iterations, and have ‘revised but not final’ versions as intermediate deliverables.
- Option 2 is usually preferred. The WBS is more easily revised as new information comes in, and the law of large numbers may work in your favor (unless all iterations are assumed identical).

Curve Ball 2: Very long work/wait times

- What if a deliverable legitimately takes more than 80 hours?
- e.g. waiting for grass to grow on the Hobbiton set of The Lord of the Rings, waiting for paint to dry, waiting for a custom part to be built by a supplier.
- Create intermediate *milestone* deliverables every two weeks or so.
- e.g. ‘Confirmed that grass is growing on schedule.’
- A good idea anyway: alerts your team to potential problems sooner.
- Suppose a sudden storm washed away the grass seed shortly after the set crew left…