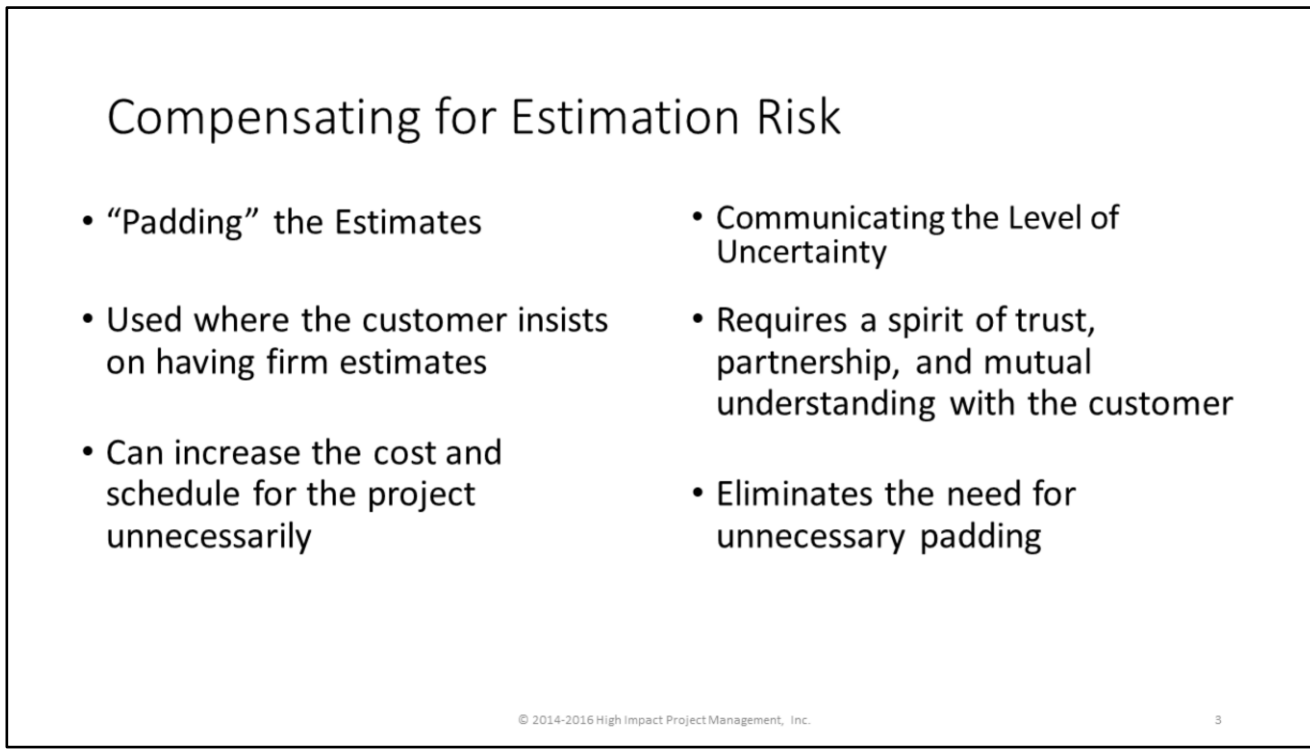
Mastering Agile Project Management

Learn to how to blend Agile and traditional project management to create an adaptive approach to project management

Project Estimation Best Practices

Project Estimation Best Practices

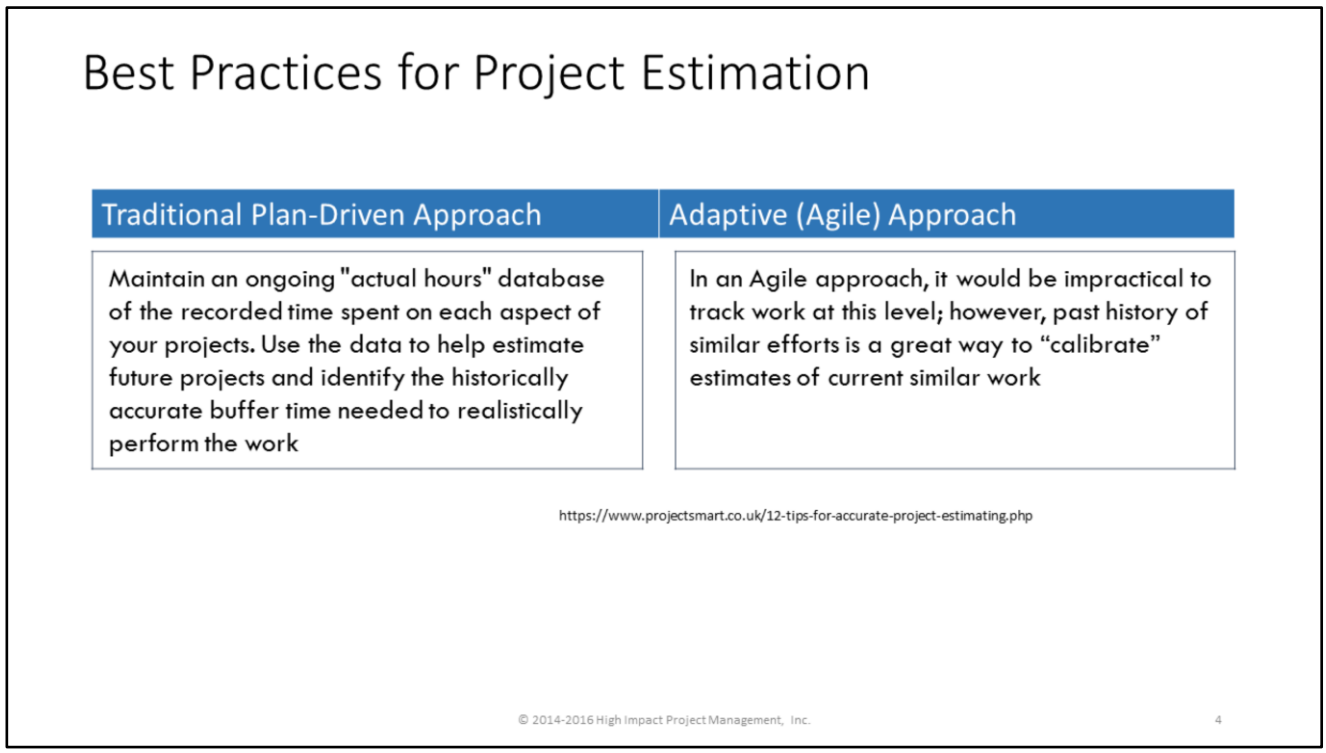
In this lesson we're going to discuss some general best practices for project estimation and how they're different between an Agile environment and a traditional, plan-driven environment.



There are two primary approaches for compensating for the risks that are inherent in estimation:

- The first approach that is commonly used is to pad the estimates to add a buffer to compensate for the level of uncertainty in the estimate. This would become necessary if the customer insisted on having firm, fixed-price estimates. The downside of this approach is that people tend to add padding at multiple levels:
    - The developers add padding to their estimates to protect themselves
    - The project manager may add padding a multiple levels to cover the project-level risk

    The overall impact of that might be significant – you might have padding on top of padding and the net impact might be to significantly increase the cost and schedule of the project because once a plan has been agreed on, there is not too much of an incentive to come in under budget and ahead of schedule
- A better approach, if you can do it, is to just be open and transparent about communicating the level of risk and uncertainty in the estimates to the customer and both the customer and the project team agree to work collaboratively in a spirit of trust and partnership to manage the risks as the project is in progress. This is a much more realistic approach and is the preferred approach in an Agile environment but it obviously requires a spirit of trust and partnership with the customer. If it can be done, it has a big benefit of eliminating the need for unnecessary padding of the estimates and eliminating that padding offers a potential opportunity to significantly streamline the project costs and schedule.



I want to briefly walk through some best practices that are commonly used in a traditional, plan-driven environment and discuss how they might be done differently in an Agile environment. Here's the first one:

- In a traditional, plan-driven environment estimates are generally based on detailed task-level estimates and it would be a good practice to keep track of previous work that is similar to the work being done. For example, in an automotive body shop that does repairs of automobile body damage, there might be a history of what it takes to remove and repaint a bumper on a car since it is a task that is done often and repeatedly

- In an Agile environment, estimates are typically done at a much higher level than individual tasks so it probably wouldn't be done at a task level and the work might not be exactly the same from one project to the next; however, there is a lot of value in making comparisons to similar work to calibrate estimates. It just has to be done at the right level and it has to be understood that it may not be an exact comparison.

Best Practices for Project Estimation

Traditional Plan-Driven Approach	Adaptive (Agile) Approach
Create and use planning documents, such as specifications and project plans	In an Agile approach, detailed planning documents and specifications are not used
Perform a detailed task analysis of the work to be performed	In an Agile approach, it may be futile to perform a detailed task analysis of the work to be performed if the requirements themselves are not well-defined

https://www.projectsmart.co.uk/12-tips-for-accurate-project-estimating.php

Here's another best practice:

- In a traditional, plan-driven environment estimates generally rely heavily on having well-defined plans and specifications for the project deliverables and how they will be developed and it's important that these documents accurately define the deliverables and the work to be done in detail so that there is no misunderstanding of what is being estimated.
- In an Agile environment, that really isn't practical because an Agile project uses much more of a rolling-wave, just-in-time planning approach to further elaborate the plan as the project is in progress. As a result, the estimation approach needs to be consistent with that environment so an Agile project might start out with a rough high-level estimate of the project that gets further refined as the project is in progress but it probably never gets as detailed as a traditional plan-driven environment.

Another best practice would be to do a detailed task analysis of the work to be performed. The alternative in a plan-driven environment might be to shortcut that process and just put together a SWAG for an estimate without doing a detailed analysis to support it. Doing a SWAG like that might not be considered an acceptable practice in a traditional, plan-driven environment where the estimates are expected to be detailed and fairly exact.

In an Agile approach it is probably futile and a waste of time to perform a detailed task analysis of the work to be performed if the tasks and the requirements for those tasks themselves are not well-defined. Any estimate can only be as accurate as the requirements that the estimate is associated with and it is foolish to try to develop a detailed and precise estimate for requirements that are not well-defined.

Best Practices for Project Estimation

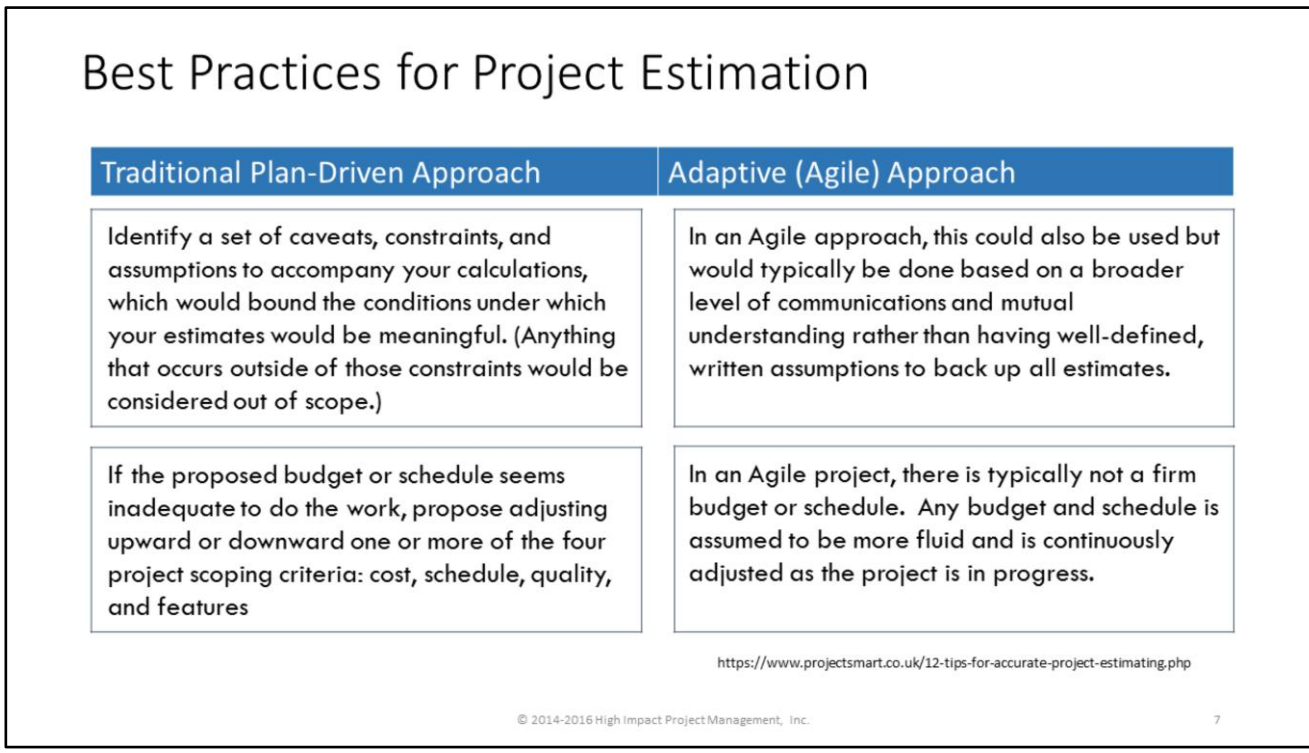
Traditional Plan-Driven Approach	Adaptive (Agile) Approach
Use a "complexity factor" as a multiplier to determine whether a pending project is more or less complex than a previous one	In an Agile approach, this could be a sensible approach
Use more than one method to arrive at an estimate, and look for a midpoint among all of them	This approach could also be used in an Agile environment to some extent

https://www.projectsmart.co.uk/12-tips-for-accurate-project-estimating.php

Here's another best practice:

- In a traditional, plan-driven environment, a recommendation is to use a "complexity factor" as a multiplier to determine whether a pending project is more or less complex than a previous one. I can remember many years ago that there were some very sophisticated models for software estimation (an example was the COCOMO model) that allowed you to create an overall model to estimate the complexity of a software project based on a number of different factors. These models went a bit too far in attempting to setup a complete comprehensive model where it was a well-defined quantitative model that allowed you to simply enter a few parameters and it would crank out an estimate of the overall cost and schedule of the project. That kind of simplistic model isn't very reliable and isn't widely-used any more for that reason; however, the idea behind it isn't bad as long as it is done intelligently
- In an Agile environment, that could be a sensible approach as long as it is done intelligently. In fact, relative sizing is commonly used in Agile as an estimation approach.

Another best practice would be to use more than one method to arrive at an estimate and look for a midpoint among all of them. That approach also has some merit in an Agile environment to get different perspectives on an estimate from different sources to compare in order to improve the accuracy of the overall estimate. An example would be to ask each team member of an Agile team to independently develop estimates of the work to be done and then compare those independent estimates to reach consensus and then those consensus estimates could be compared against another completely different estimation approach based on relative sizing.



Here's another best practice:

- In a traditional, plan-driven environment, another recommendation is to Identify a set of caveats, constraints, and assumptions to accompany your calculations, which would bound the conditions under which your estimates would be meaningful. (Anything that occurs outside of those constraints would be considered out of scope.) A project plan in a traditional, plan-driven environment is many times like a contract and has to have detailed information associated with it to back up the contract assumptions so that there is no misunderstanding
- An Agile environment is different – instead of having more of a formal, arms-length contractual relationship with the customer, there is a less formal relationship based on more of a spirit of trust, open communication, transparency, mutual understanding. So the appropriate approach in an Agile environment might be having an open discussion with the client and reaching mutual agreement with the client on what assumptions make sense; however, those assumptions may not be as detailed and well-defined and may-or-may-not be formally documented. It is also understood that whatever assumptions that are made are likely to be much more fluid and likely to change as the project is in progress. Nonetheless, it is certainly still good practice to have discussion with the client and reach mutual understanding of these issues.
- Another best practice is "If the proposed budget or schedule seems inadequate to do the work, propose adjusting upward or downward one or more of the four project scoping criteria: cost, schedule, quality, and features". This one seems very straightforward; however, any times estimates will be way off from the project budget and plan and no corrective action is taken to resolve that.
- In an Agile project, there is typically not a firm budget or schedule. Any budget and schedule is assumed to be more fluid and is continuously adjusted as the project is in progress.

Best Practices for Project Estimation

Traditional Plan-Driven Approach	Adaptive (Agile) Approach
Consider simpler or more efficient ways to organize and perform the work	This is standard practice in an Agile environment
Plan and estimate the project rollout from the very beginning so that the rollout won't become a chaotic scramble at the end. For instance, you could propose using a minimally disruptive approach, such as a pilot program or a phased implementation	An Agile project can also benefit from this in the right context

https://www.projectsmart.co.uk/12-tips-for-accurate-project-estimating.php

Here's another best practice:

- In a traditional, plan-driven environment, another recommendation is to "Consider simpler or more efficient ways to organize and perform the work"
- This is a given and is standard practice in an Agile environment. Rather than having a static well-defined process that remains the same for the duration of the project and might be repeated from one project to the next, an Agile project is based on continuous improvement where the process is continuously improved as the project is in progress to try to improve the effectiveness and efficiency of the process.
- Another best practice is "Plan and estimate the project rollout from the very beginning so that the rollout won't become a chaotic scramble at the end. For instance, you could propose using a minimally disruptive approach, such as a pilot program or a phased implementation"
- An Agile project could also benefit from this but it may be more difficult to do because you may not know enough at the beginning of the project to plan a detailed roll-out; however, many enterprise-level Agile frameworks such as Scott Ambler's Disciplined Agile Delivery model do recognize the need for putting an adequate level of attention and planning on the deployment phase of the project. Another way an Agile approach helps in this area is by breaking up the project into iterations, there is typically much more of an incremental rollout which is much easier to plan and implement as opposed to a single, "big bang" roll-out for the whole project at once.

Best Practices for Project Estimation

Traditional Plan-Driven Approach	Adaptive (Agile) Approach
In really nebulous situations, consider a phase-based approach, where the first phase focuses primarily on requirements gathering and estimating	This is standard practice in Agile and Agile also uses “Spikes” to isolate significant areas of uncertainty
Develop contingency plans by prioritizing the deliverables right from the start into "must-have" and "nice-to-have" categories	This is standard practice in Agile and Agile goes much further into prioritizing all requirements by business value
Refer to your lessons-learned database for "20:20 foresight" on new projects, and incorporate your best practices into future estimates	This is standard practice in Agile and Agile goes much further by doing a retrospective at the end of each sprint

https://www.projectsmart.co.uk/12-tips-for-accurate-project-estimating.php

Here’s another best practice:

- In a traditional, plan-driven environment, another recommendation is to “In really nebulous situations, consider a phase-based approach, where the first phase focuses primarily on requirements gathering and estimating”
- This is also a given and is standard practice in an Agile environment where the requirements, by default, are assumed to be more nebulous and less-defined; however, the difference is that an Agile approach doesn’t attempt to resolve all uncertainty in the requirements prior to the start of the project, they are further elaborated as the project is in progress. Another approach that is commonly used in Agile to handle significant areas of uncertainty is the idea of a “Spike”. Dealing with uncertain requirements is routine in an Agile environment. It is considered the exception in a traditional plan-driven environment
- Another best practice is to develop contingency plans by prioritizing the deliverables right from the start into "must-have" and "nice-to-have" categories
- This is another best-practice that tis integral to an Agile development process and Agile goes much further into prioritizing all requirements by business value
- Another best practice is “Refer to your lessons-learned database for "20:20 foresight" on new projects, and incorporate your best practices into future estimates”. This is another practice that is actually much more well-engrained into an Agile development process and an Agile process goes much further with this by doing a retrospective at the end of every sprint rather than waiting to do a post-mortem at the end of the entire project.

There are big differences between the estimation approach in an Agile environment and a traditional plan-driven environment; however, many of the underlying principles still make sense

The key point I want to make is that “There are big differences between the estimation approach in an Agile environment and a traditional plan-driven environment; however, many of the underlying principles still make sense.”

We will go into much more detail into how the approach is different in an Agile environment in the rest of this module.

NEXT LECTURE…
TIME-BOXING AND TIME-BOXING ADVANTAGES

In the next lecture, we're going to focus on "Time-boxing and the advantages of Time-boxing" An understanding of time-boxing is fundamental to an understanding of how Agile estimation works