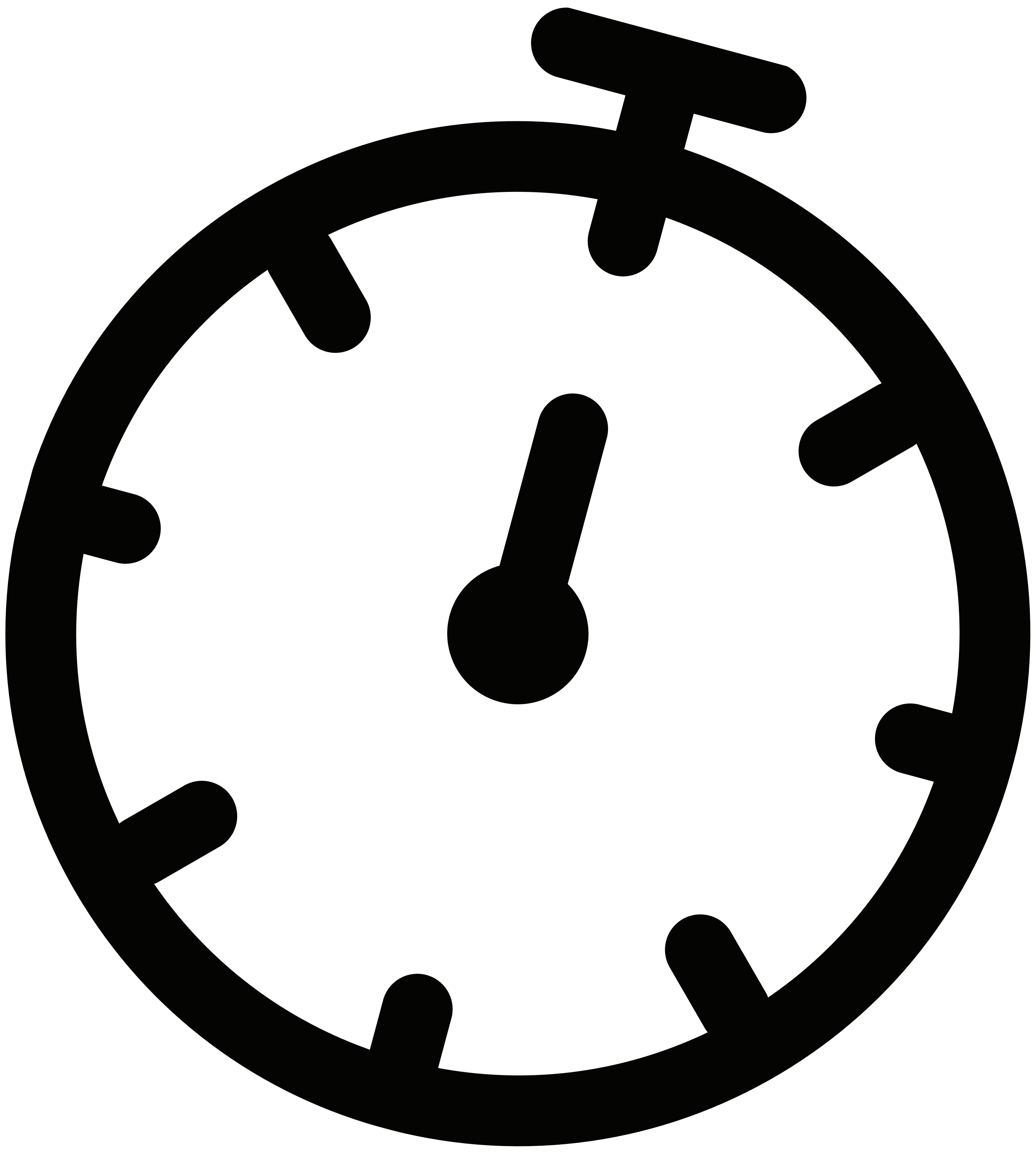
orangehill

The Ultimate Guide to Successful Project Estimation

Helping you on your quest for successful project estimations



f http://facebook.com/orangehilldev

http://twitter.com/orangehilldev

http://www.orangehilldev.com

office@orangehilldev.com

Thank You

Thank you for downloading this guide. We're looking forward to helping you on your quest for successful project estimations.

We like keeping things short and to the point, so let's get started. Before we dig into the techniques you will be using to improve the accuracy of your estimates, we need to first make sure you're not making one of the most common mistakes when it comes to project estimation.

Contents

Common Project Estimation Mistakes

Project Estimation Techniques

Building a culture of accurate project estimation

Common Project Estimation Mistakes

MISTAKE #1

Miscommunicating your ballpark estimate as a final estimate

When a new project offer comes in, one of the first things to do is to establish a cost of the project. As a rule, in the early stage of the project, features are not defined in full or at all, the designs are also not provided or there’s only a vague idea on how they would look.

In this stage, you're not really making an estimate. You're making a guess based on (very) limited information. **It's extremely important to clearly make it known that what you're quoting is a ballpark estimate and in no way your final estimate.** As such it would be wise to avoid placing it on official documents to prevent miscommunication from happening. A stakeholder might read this number mentioned somewhere in the document and mistake it for a final estimate.

Why would you need such an estimate in the first place? Ballpark estimates do serve a good purpose. In the early stages of the project this estimate helps inform the stakeholders of a rough cost of the project. In cases where the cost is too high, the scale of the project can be reduced to better fit the budget. And this is best done in the early stages.

Common Project Estimation Mistakes

MISTAKE #2

Estimating a project without consulting the development team

This can be tempting if you're a manager. A project might seem very similar to some previous one, so you feel confident you can make the estimate on your own.

Don't do it!

Your estimate might be somewhat accurate since it at least uses some historic reference. However, there might be some hidden technical details you will miss, and your estimate might now be off by 50% or more. It's not a big deal if it's a small 10 hour project. The mistake can prove especially painful for a complex 200 hour project where making a wrong assumption can lead to missing a deadline by more than a couple of weeks.

It's especially dangerous for larger companies where you're not the only manager utilizing the development resource. If you're not familiar that they are already busy working on two projects this can turn into a catastrophe.
It goes without saying that once you (mis)quote a price to a client, the deal is sealed and it's extremely hard and unprofessional to renegotiate a price once the project has started. While making this kind of estimate is better than nothing, a smarter solution is:

Leave the estimates to the people who will be doing the actual work - your development team.

Common Project Estimation Mistakes

MISTAKE #3 Not using the right tools for the job

Your development team will likely do a much better job at estimating. But only if they have the knowledge of estimation techniques required to make the estimate. **As the size and complexity of the project grow, so does the importance of using one or more of the proven estimation techniques.**

If the development team gives their estimate by merely looking over a few JPEG mock-ups and a couple of pages of documentation for a few minutes, their estimate will likely be based on the time they "feel" it would take.

This feeling will often be over-optimistic by failing to notice details that will take much longer to complete. Or it may be over-inflated if the developers are uncertain of their abilities. The uncertainty usually comes with a project which requires a new technology which needs to be learned first.

It might be a good idea to exclude the learning time from the price you quote to a client if you feel the knowledge might be beneficial on other projects as well. This will also make your quote look better than the competitor's. On the other hand if it's something you won't be able to reuse, there's not much point covering the education cost yourself.

This vital stage is where estimation techniques come in to replace feeling with logic and greatly improve the accuracy of the project estimation process.

Project Estimation Techniques

Importance of project documentation

We have previously mentioned this, but will repeat it: **you need to have the project documentation and designs to make an accurate estimate**. The project estimation accuracy improves with the completeness of project documentation and designs.

Project Estimation Techniques

Analogous Estimation Technique

In the early stages of the project where only limited documentation is available a good technique to employ is analogous estimating, also known as top-down estimating. Analogous estimating uses historical data (similar past project) to estimate the duration of your current project. As with any other technique, **its best used by the developers who actually worked on the past project** and can make the best expert judgment on how the current project compares to the past one.

EXAMPLE

You need to develop a small eCommerce website similar in scale to the one you did two months ago. Except this one also has Facebook and Twitter sharing feature. The previous website took 40 hours to develop and you estimate based on prior experience that it would take about 4 more hours to implement and test the sharing feature. Your estimate using the analogous estimating technique is 44 hours.

THE GOOD

- Can be used with only limited documentation provided there is enough historic data
- Can be used in the early stages of the project to refine your initial ballpark estimate
- Relatively fast to calculate

THE BAD

- Accuracy is lower than other techniques and further
- Accuracy drops in case of insufficient or missing historic data

SUGGESTION

- **If you don't have sufficient internal historic data you can consult an external expert.** In this case keep in mind that external developer might be on a different skill level than your development team and adjust the estimate accordingly.
- **Always account for the missing documentation and add contingency to your estimate accordingly.** Your initial estimate of 44 hours would need to go up by 10-50% depending on the expected complexity and scope.

Project Estimation Techniques

Parametric Estimation Technique

Parametric estimating technique is similar in nature to the analogous estimating, but adds an additional layer of accuracy by dividing the work into measurable units.

EXAMPLE

You know you will be developing a five page website with a basic admin panel. You can use the information you have and compare it to the average time it takes to develop a new page template for a similar website and bind it to the admin panel. If that takes an average of 12 hours per page, your estimate using the parametric estimating technique would be 60 hours.

THE GOOD

- The same points as analogous estimating
- But also more accurate than analogous estimating

THE BAD

- The same points as analogous estimating but still more accurate

SUGGESTION

- Since this technique is very similar to analogous estimating the same suggestions apply. Due to increased accuracy you can reduce the contingency percentage if you're confident in your historic data.

Project Estimation Techniques

Three-point Estimation Technique

The project manager or the development team first decompose the project into a list of estimable tasks. This requires the documentation and designs to be available. The tasks can vary in scope, with more detailed tasks providing the best levels of certainty, but also take longest to estimate. Typically, a reasonable number is 20-30 tasks per project.

Three-point estimate requires three figures to be produced for each of the estimated tasks.

a – The **best-case estimate** for completing the task
m – The **most-likely estimate** for completing the task
b – The **worst-likely estimate** for completing the task

(All based on previous experience or educated best-guesses)

There are three formulas you can use to calculate the estimates for each task.

The **simplest one** is:

$ET = (a + m + b) / 3$

A more accurate **weighted formula** places more weight on the most-likely estimate:

$ET = (a + 4m + b) / 6$

And the most complex – **weighted that also accounts for standard deviation**:

$E = (a + 4m + b) / 6$
$SD = (b - a) / 6$
$ET = E + 2 \times SD$

You only need to pick one ET formula to use. You may also want to try all of them and see which one best fits you.

Final project estimate is calculated as a sum of all individual task estimates. Usually there's no need to add contingency to this estimate. You can compare the actual time taken to complete each task to the estimated time and add contingency based on what you find.

The three-point estimate works best when estimates are unbiased and based on previous experience on similar tasks.

Project Estimation Techniques

Three-point Estimation Technique

As such, some managers find it best to ask for a, m and b numbers for each task and calculate the ET themselves. In this way, the development team won't be tempted to alter the initial a, m and b numbers after seeing the final estimate.

To save you some time you'll find 3 tools to help you calculate each of these formulas on our website, downloadable as XLS formula sheets:

Simple Formula:
http://www.orangehilldev.com/resources/simple-formula.xlsx

Weighted Formula:
http://www.orangehilldev.com/resources/weighted-formula.xlsx

Weighted Formula With Standard Deviation
http://www.orangehilldev.com/resources/weighted-with-sd-formula.xlsx

EXAMPLE

You need to build an art gallery website which uses sliders to display works of art and show artist biographies. You split the work into following tasks (numbers are a, m and b in that order, meaning best-case, most-likely and worst-case time):

- *Responsive Layout: 2h, 3h, 5h*
- *Header: 2h, 4h, 7h*
- *Slider: 2h, 5h, 10h*
- *Biography section: 1h, 2h, 4h*

(Simplified to four tasks, in reality you would have much more front end and back end tasks)

Using the weighted formula, we calculate the averages for each task:

- *Responsive Layout: (2 + 4 x 3 + 5) / 6 = 17 / 6 = 2.8h*
- *Header: (2 + 4 x 4 + 7) / 6 = 25 / 6 = 4.2h*
- *Slider: (2 + 4 x 5 + 10) / 6 = 32 / 6 = 5.3h*
- *Biography section: (1 + 4 x 2 + 4) = 13 / 6 = 2.2*

Estimated project time is: 2.8h + 4.2h + 5.3h + 2.2h = 14.5h

Project Estimation Techniques

Three-point Estimation Technique

THE GOOD

- One of the most accurate estimation techniques
- Psychologically easier for developers to provide three numbers, less pressure of "getting it wrong"
- Identification of the worst case number leads to discovering otherwise hidden problem areas
- Reduces the risk of over-optimistic and over-inflated estimates
- There are tools available on our website to help reduce the overall time of calculation
- Estimate is solid even with limited previous experience

THE BAD

- Requires full documentation to be able to decompose the project to tasks (which actually isn't bad)
- Providing three estimates for each task requires more time than other methods
- Leaving the final calculated estimate visible might lead to development team adjusting their initial estimates

SUGGESTION

- Use the recommended tools to keep you from manually calculating estimates for each of the 20-30 tasks

Project Estimation Techniques

Bottom-Up Estimation Technique

Bottom-up estimating starts in a similar way to three-point estimation - the project manager or the development team first decompose the project into units or tasks and then proceed to decompose those to the smallest measurable chunks.

After that, each of these small chunks is estimated and once all the chunks for a given unit or task have known estimates, a sum of those chunks makes up an estimate for the entire unit.

You may wonder why not just estimate the entire unit or task in the first place if that's the point?

Well, here's where you really see why bottom-up estimating is the most accurate technique of them all. **By decomposing the unit to smallest chunks you are also analyzing the unit in more detail and noticing parts that are more complicated and would probably get missed otherwise.**

Also of great importance, **you're noticing parts or features which are not clearly documented**. A good indicator that something need to be documented better is that you start guessing how something should behave.

While very simple in nature, the bottom-up estimates do require a level of previous experience on similar projects. If there's not enough experience, the lack can partially be covered by contingency. That of course will never be as accurate as a skilled estimate.

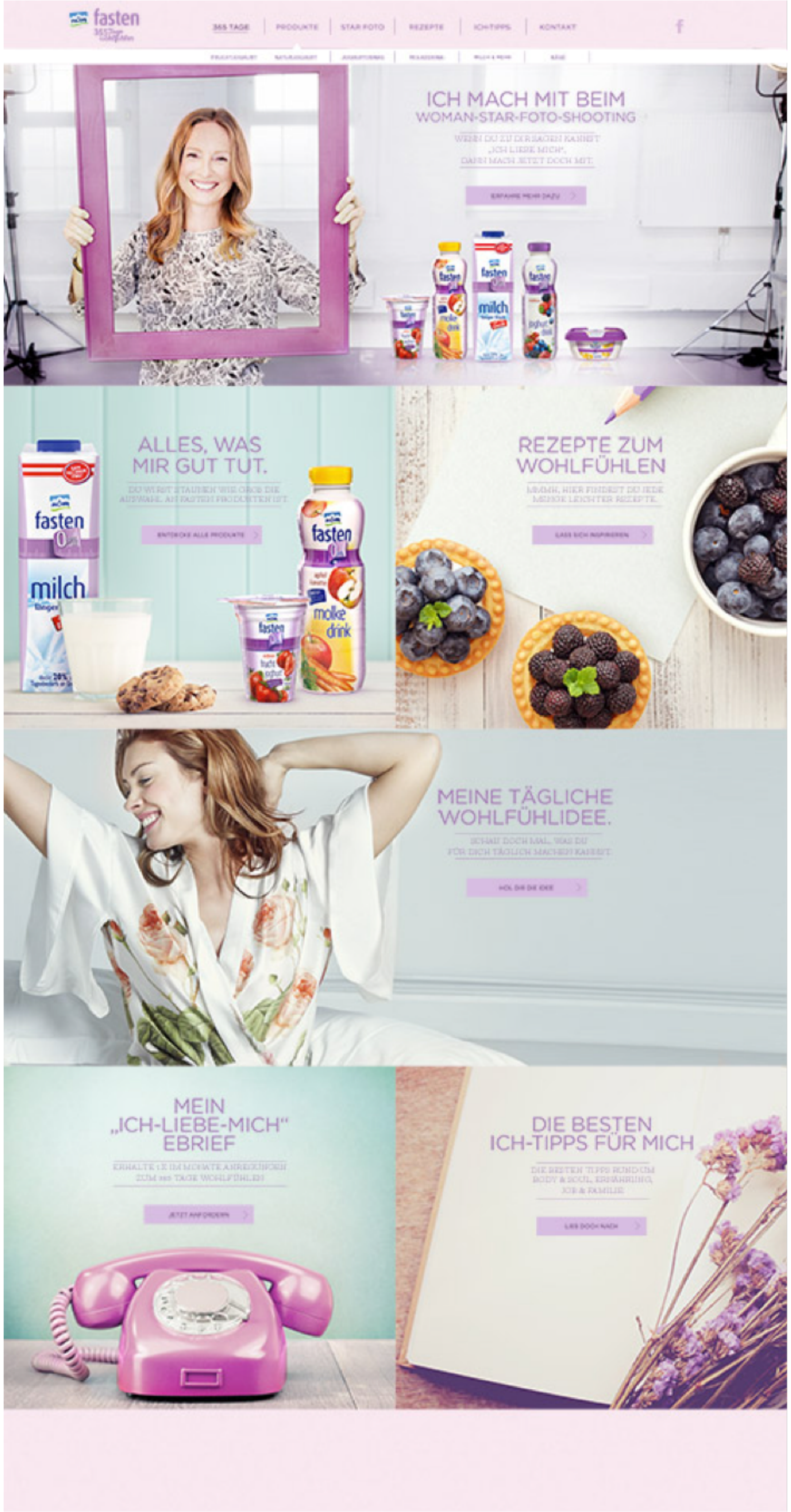
Project Estimation Techniques

Bottom-Up Estimation Technique

EXAMPLE

In our opinion this is the best technique if you have an experienced team so we will be giving a more detailed example based on a page from a real project.

Though all estimation techniques mentioned in this E-book serve well for any type of technical project, let’s take a look at how a senior front-end web developer would approach bottom-up estimation for this non-responsive home page (see example image):



Project Estimation Techniques

Bottom-Up Estimation Technique

EXAMPLE

What do we have on this page? There's a header there. There are six sections, two full-width and four smaller half-width sections. No footer, seems easy enough. Also, the page layout would be a unit of its own.

Did you notice that we just identified three units of this page?

1. **layout**
2. **header**
3. **inner sections**

(bigger and smaller ones, but basically very similar and can be combined as one unit)

We strongly recommend that you create tasks for these units and measure development time spent on each (using a project/task tracking software of choice). You can later compare the estimates to the actual time taken to complete a unit and if needed do a postmortem analysis of what went wrong with your estimate so you can adjust in the future.

We now decompose each of these units, let's start with the layout.

EXAMPLE

Layout

It's not responsive in a sense that it should support mobile and tablet sizes but since it's should cover the entire screen you need to use percentage values for the elements instead of pixels. OK, so we would have full width block header on the top, content wrapper we can reuse for other pages below it. In the content wrapper we can float:left all the sections where larger ones would have 100% width and smaller ones 50% width.

Assuming you're using a Grunt boilerplate, it can take roughly *5 minutes* to initiate it, install nodejs dependencies, another *5 minutes* to set up basic LESS CSS layout.

The site uses custom fonts and we don't have the web-font files so we need to convert them ourselves from the generic font file and import those using LESS CSS. That's usually no more than 10 minutes but if there's a problem with the file we have we might need to look at alternatives. Let's make it *20 minutes*.

Layout estimate: *30 minutes*

> *Notice that while the estimate for this unit won't be that high and might seem irrelevant, by splitting it into chunks you just performed an analysis of all the things you need to do to set up the website. In essence you just planned out the work you need to do for this unit. All while doing the estimate. You saved time in the execution stage and identified potential problems before you write a single line of code. That's just one more thing that's great about bottom-up estimation technique.*

Project Estimation Techniques

Bottom-Up Estimation Technique

EXAMPLE

Header

Moving to the next unit, the header. It has a repeat background that we would need to cut out of the PSD because the designer didn't provided it as a separate image file (as usual). Also, we need to cut out the logo and the custom Facebook button. Photoshop work alone will take *10 minutes*.

The logo and buttons can be positioned absolutely and would need to be links so the HTML and styling would be *5 minutes* total for both.

The menu has equal width items and borders between which could be tricky. These also have sub-menus for some so we best use nested ul > li elements. That usually takes 10-20 minutes, but let's make it *20 minutes* because of the borders and sub-menu positioning.

The sub-menu is similar and has an arrow on top so let's make it *20 minutes* as well. I have no idea if it should be shown on hover or click, let's check that but it doesn't change the estimate much.

Styling the spacing between the menu items and font-sizes on second thought might actually take *another 20 minutes* for menu and sub-menus.

What should FB button do? Does it like the website which would need JS or just links to the FB page? *Not documented.*

Header estimate: *1 hour 15 minutes*

> *We also planned out the header and identified two missing pieces of documentation.*

Project Estimation Techniques

Bottom-Up Estimation Technique

EXAMPLE

Inner sections

Inner sections next. The HTML is fairly simple for these, an h-something element for title, p element for subtitle and a button. That's no more than *20 minutes* for filling in the text for all six sections.

The LESS CSS is a different story. We need to set a covering backgrounds for each (*20 minutes*), style the fonts and also add weird and non-uniform borders on all subtitles (that might take *an hour*).

There's the custom button as well which will probably be used elsewhere so we should make it reusable (*30 minutes*). The buttons are links only so at least that's easy.

First we need to cut the background images from Photoshop, so *10 minutes* there too. Also we can cut the button arrow as SVG and make an icons web-font we'll probably use for other pages too. That's *20 minutes*.

Inner sections: *1 hour 40 minutes*

Project Estimation Techniques

Bottom-Up Estimation Technique

EXAMPLE

Total home page estimate

For the entire home page we have an initial estimate of 205 minutes. We have some missing documentation and if we account for unanticipated problems we would probably need to *add 20% contingency* as well.

That's 205 * 1.2 = 246 minutes, which rounds up to **4 hours final estimate**.

Hopefully this real example made you realize the benefits of bottom-up estimate techniques. This is the ideal case where you have an experienced web development team to work with. If that's not the case you can still use this technique. In that case I recommend that you combine it with the three-point estimation technique we discussed earlier.

In essence the process is similar, but instead of giving one estimate per chunk, the developer would need to give three, best-case, most-likely and worst-case estimates.

The worst case one makes the developer think about all the things that can go wrong with styling, JavaScript plugin implementation or similar.

Psychologically - having the freedom to put in any number he wants for the worst case gives you an additional layer of accuracy when calculating the estimate using one of the three-point estimation formulas.

It takes longer but works well to improve accuracy, especially for more complex projects. In addition to using the three-point estimation you might also think about adding a larger contingency budget to account for unanticipated problems.

Project Estimation Techniques

Bottom-Up Estimation Technique

THE GOOD

- The most accurate estimation techniques

- By decomposing the project down to smallest chunks there's almost no chance of missing important details which will take long to complete but would likely get unnoticed with other techniques

- Also helps identify holes in documentation so these can be filled early

THE BAD

- Requires more time to complete compared to other techniques, though it gets better fast as you use continue to use this technique

- Requires full documentation otherwise the accuracy can suffer.

- Most useful if you have at least somewhat experienced development team since it relies heavily on previous experience

SUGGESTION

- You can use this technique even for less experienced teams but make sure to add contingency to make up for lack of historic data

- This technique can be combined with three-point estimation add even more accuracy or to make up for lack of experience. Note that this will also require more time to complete an estimate.

A few more words on contingency budget

It should always be very clear that an estimate is just that - an estimate. **Do not expect it to be 100% accurate.** Even if it proves accurate in four projects, in the fifth one it may be off by 10-20%. The accuracy of any type of estimate can be low in cases of little to no previous experience.

In such cases you can manage the uncertainty with a contingency budget. You should also account for missing documentation, so it's reasonable to assume that if roughly 30% of documentation is missing, you'll need to add 30% to the estimate.

Building a culture of accurate project estimation

Mistake most companies don’t realize

More often than not we see situations where an estimate gets interpreted completely differently by the party who made the estimate and the party that requested the estimate. This is true for both estimates requested by project manager and estimates requested by the client.

What happens?

The development team would estimate a project delivery in **two months**, which would fall on **March 15th**. To the development team, the meaning of that estimate equals the dictionary definition of the word estimate:

> *"An approximate calculation or judgment of the value, number, quantity, or extent of something".*

The weight here being on the word "*approximate*".
Project managers and clients see this differently. The two-month estimate becomes a **commitment and a deadline**.

Who is right?

The truth is, you can't put the blame for the situation described on just one of the parties involved.

Project managers and clients view the estimate as an educated projection based on the developer's past experience, skills and qualification. As such, this expert opinion will often cause them to recalculate their own projections, deadlines and finances in accordance with the time given by the team.

The estimate is now setting expectations and this is especially true the more precise your estimate sounds. So ironically, accuracy in cases like these can be a bad thing, especially if the price of the project gets calculated and delivered to the client based only on this estimate.

Building a culture of accurate project estimation

Mistake most companies don’t realize

What are some ways to prevent this from happening?

Imagine in our example, instead of saying the project is expected to be finished on **March 15th**, that the estimated project delivery is **in March with an 80%** likelihood that it will be finished in the first half of the month.

If this slips to March 20th, it only means that the team didn't hit the 80% likelihood, but is still in the range of the set expectations.

Managers should take this into account when communicating with the client and forming project price. Forcing the developers to be more exact, especially in situations where they are not experienced enough or don't have all the needed information and then treating that forced estimate as a deadline can lead to situations where the developers lose trust in the management (and vice versa) and start inflating the estimates to protect themselves. A very bad situation where the estimation process loses all meaning.

Building a culture of accurate project estimation

Mid-project scope changes

We’ve all had situations where the scope changes mid-project.

Communication in this situation becomes extremely important.

If the scope of the project gets expanded considerably mid-project, the original estimate will no longer be valid.

The development team **MUST** communicate this to the project manager and the project manager **MUST** communicate it to the stakeholder, so everyone is aware of the consequences of the decision to change the project in the execution stage.

Failure to communicate can lead to catastrophic consequences and deadlines missed by more than a couple of weeks. It's a situation where everyone loses, the client for missing the targeted launch, the manager for failing to do his job and monitor the project, and the development team for having to pull crazy amount of extra hours in a last-minute attempt to salvage the project.

Building a culture of accurate project estimation

Accepting projects with preset deadlines

Another sign of bad communication inside of a company is when a project manager asks for an estimate, the team says "Two months" and the manager replies "We need it in one month". There isn't much point in asking for estimates if the deadline has been preset **without ever consulting the team**.

While the team can be pushed to put in extra hours, this again leads to loss of trust between the developers and managers in the long run.

Another danger lurking here is the burnout effect when the extra hours achieve exactly the opposite of what they were intended to do. The project execution slows down due to fatigue and worse, the quality diminishes very rapidly.

How to better handle preset deadlines?

There are valid situation where the client insists on a preset deadline.

There are four options in such cases where your team's estimate doesn't fit the deadline.

One we already discussed, burning **extra hours** to make the deadline.

Alternately, you can communicate the situation to the client and try to reduce the project scope by drawing up a **minimum viable product**. Pushing some of the work post-launch is often the best solution if the client agrees.

Another solution would be to **hire more developers**.

And finally, you can also **outsource parts of the project** to external teams.

Building a culture of accurate project estimation

What happens when a deadline is missed?

If despite the team's best effort and extra hours put in, you still weren’t able to meet the deadline you committed to, it's of vital importance to focus on postmortem analysis of the tasks before you start blaming anyone.

Each task where the completion time exceeded the estimated time needs to be discussed in detail, so you can determine with certainty what the reasons behind it were. Only then can you be sure that knowledge is gained from the mistakes and that your next project will have a greater chance of finishing on time.

Contact us

If you liked this guide or have anything you feel was left out or need improving, feel free to contact us at office@orangehilldev.com

Orange Hill Development

A programming company and a tight-knit group of hard-working people, great at what they do.

OUR SERVICES

- Web Sites
- Facebook Apps
- Responsive Web Design
- Magento E-commerce
- Wordpress
- Newsletter Templating
- iOS App
- Android App
- Cross-platform App

GET IN TOUCH!

http://www.orangehilldev.com
office@orangehilldev.com
+381 64 167 7367
Djordja Stanojevica 11b, Belgrade, Serbia