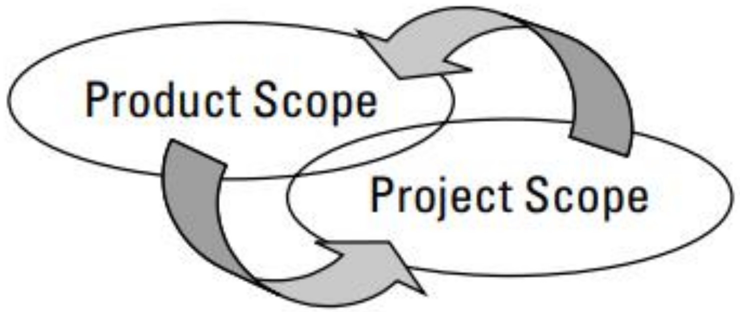
IT PROJECT MANAGEMENT

Muhammad Hamza Ihtisham

Creating the Software Scope

Understanding Product Scope and Project Scope

- The product scope is the summation of the attributes and features that will comprise the product you're creating for your customer. When the stakeholders agree on the product scope, then you can focus on creating the software project scope.
- The difference between the two is that the product scope describes the result of the project — the things the customer sees. The project scope describes the work that must be completed to complete the project — the things the project manager focuses on.



Completing stakeholder analysis

STAKEHOLDER ANALYSIS IS THE PROCESS OF DETERMINING WHO YOUR STAKEHOLDERS ARE AND WHAT THEIR INTERESTS AND CONCERNS FOR THEIR PROJECT ARE.

A STAKEHOLDER, TECHNICALLY, IS ANYONE WHO HAS A VESTED INTEREST IN YOUR PROJECT'S SUCCESS.

IN A LARGE ORGANIZATION, YOU MAY NOT IMMEDIATELY KNOW WHO ALL THE STAKEHOLDERS ARE.

Interviewing stakeholders now to avoid surprises later

- One of your most important responsibilities is to interview your stakeholders. This is a vital step because it ensures that you, the hub of the project, are in tune with what the project stakeholders really want.
- Ninety percent of project management is knowing how to communicate. Set expectations for communications early in the project management life cycle by asking your stakeholders lots of questions.
- When gathering stakeholder requirements and other information, be sure you have considered all stakeholders, not just those that are the most visible.

Interviewing stakeholders now to avoid surprises later

Stakeholder analysis isn’t just examining who the stakeholders are but also their demands and wishes for the project deliverables. You’ve got to ask lots of questions, for example:

- Can you describe the conditions this deliverable will operate in?
- What’s the opportunity this project will grasp?
- What’s the main problem this software will solve?
- How do you see the deliverable solving your problem?
- What other software will this deliverable interact with?
- What are the primary and secondary features of the software?
- How will this software make the end-users’ jobs better or easier?
- Are there other stakeholders that we should consider?
- How do you see this deliverable benefiting your customer?

Knowing the sources of common conflicts

- You are a software project manager. Chances are you’ve come up through the ranks as an IT professional, business analyst, or junior engineer.
- That said, sometimes you’ll have to resolve conflicts to move a project toward completion. The goal of conflict resolution is to resolve the problem, move the project along, and not make enemies.
- But regardless of the patience and leadership you demonstrate, some folks won’t be happy with your decisions, or the project objectives. Some people will blame you because their requests for the software features can’t be added to the project.

- Schedules: Think of all the different projects, responsibilities, and demands that carve out chunks of your day. Now think of the other people your software project involves and how their schedule is affected by your project. No wonder scheduling mayhem is at the top of the list.
- Priorities: What’s important to Jane isn’t always important to Bob. Stakeholders will have pet features, ideas, and components they’ll want your project team to build into the software. Some of these components, such as which database technology to use (SQL or Oracle), are mutually exclusive — you can’t have it both ways. Because everyone won’t get the component he or she wants, you should prepare for unhappiness.
- Resources: As a software project manager, you know your project team and the abilities of all its individuals. If demands spread your project team too thin they’ll never get their work done and their morale will plummet, which puts the crunch on your project’s success. Resources, especially good developers, are in high demand.
- Technical beliefs: If you’ve ever hung around software programmers for more than, oh, say ten minutes, then you know these IT folks can disagree over eight ways to accomplish the same task. Technical beliefs can be a real stumbling block for project team members with diverse backgrounds.

- Policies and procedures: Don’t you just hate it when your organization’s rules, procedures, and policies get in the way of progress? If you yield to temptation and cut corners, you’ll pay the price later. If you try to argue your way through the red tape and procedures, you may make enemies, anger management and waste time. Policies and procedures, both good and bad, exist for some reason — even if no one can explain what that reason is.
- Costs: When it comes to software project management, costs are usually tied to a timeframe for research and development, simulations, reworking kinks, learning, and productive coding. Stakeholders don’t always see the value of any dollars committed to time that isn’t directly attributed to creating productive code. Time is money, and software development takes time. Chapter 9 covers project costs and budgets in a lot more detail.
- Personalities: Some project managers find it hard to believe that personality is the least common source of problems on projects, but according to the PMI, it’s true. Most people can work together toward a common goal — the successful completion of the project. Personality conflicts, as a rule, only become a problem when they prevent the project from moving forward. In other words, annoying people may give you a few headaches, but they don’t usually prevent the job from getting done

Resolving common conflicts

To help you resolve conflicts, you've got five approaches you and other stakeholders ought to use:

- Problem solving: Utilizing problem-solving strategies is the pinnacle of conflict resolution. This approach requires both parties to work together for the good of the project. Both parties want the solution that works best for the project. Using problem-solving techniques means removing ego and politics from the scenario to create a win-win solution.
- Forcing: You've seen this approach to conflict resolution before. Forcing means the person with the power decides and there is no further discussion. And you know that just because someone has power doesn't mean that this person always makes the best decision for the project. That's why forcing usually results in a win-lose solution.
- Compromising: This sounds nice, but really, it's not. True compromising means that both parties in the disagreement have to give up something they want. Both parties get part of what they want, but neither gets everything. Compromising is different than problem solving because it's more confrontational, whereas problem solving has a spirit of cooperation for the best solution. Compromising is considered a lose-lose scenario because no one wins 100 percent

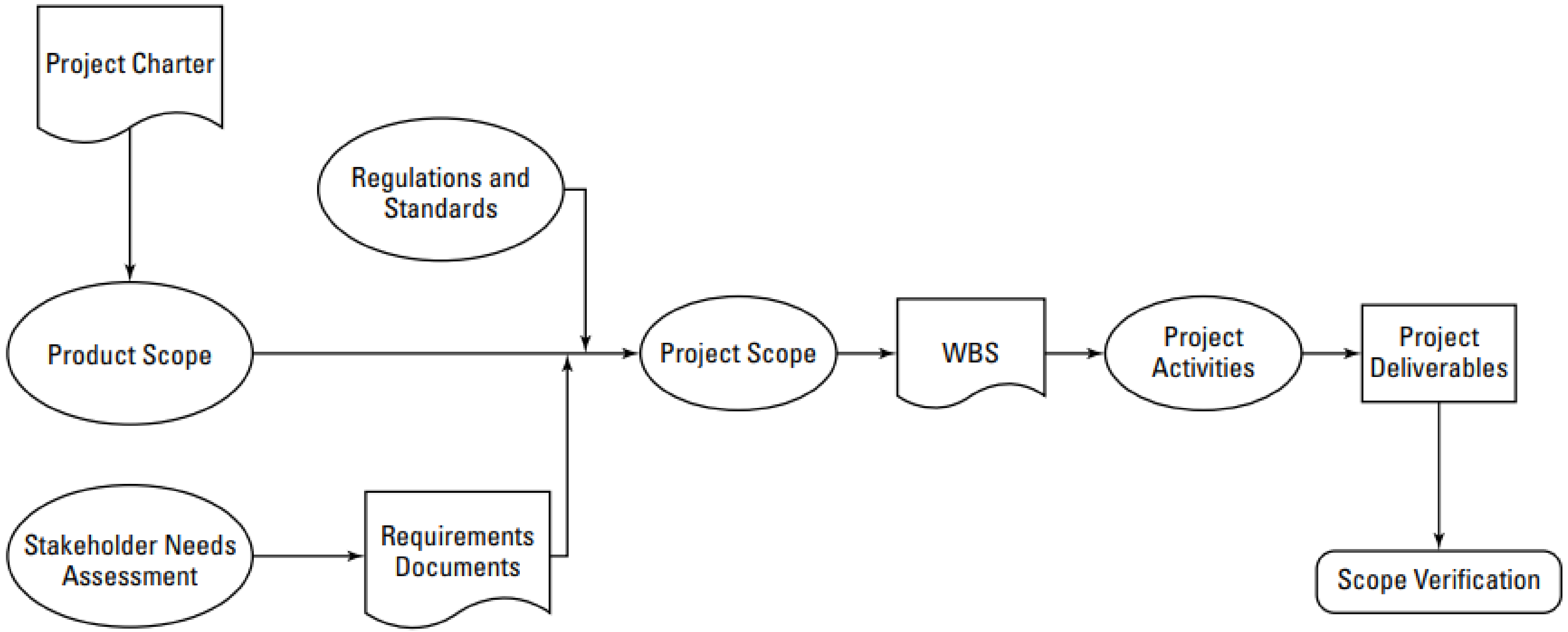
- Smoothing: This solution allows the project manager or other people in power to downplay the differences between the stakeholders and minimize the problem. The project manager smoothest the conflict without offering a solution. This is a lose-lose scenario because neither side wins. On the other hand, the project manager (or other stakeholder) could smooth the conflict while offering a solution, making it a win-win situation (which is what you should strive for). For example, if one group of stakeholders insists on software functionality that will support a particular naming convention, whereas another stakeholder would prefer a different naming convention, you could smooth the situation by allowing both stakeholders to air their arguments. You may all discover that the differences have more to do with politics and personalities than with a real preference for naming conventions. Giving people the opportunity to talk things through enables stakeholders to minimize problems on their own. Trial and error is the best approach to finding the right conflict resolution technique for each situation. No one method will work 100 percent of the time with 100 percent of your stakeholders on 100 percent of your projects. Arm yourself with as much knowledge about conflict resolution as possible and then let experience help you figure out which methods you're most comfortable with in each situation.
- Withdrawal: Ever been in a disagreement where the other person talked the issue to death? What'd you do? I bet you threw your arms in the air and surrendered just to get moving. That's a withdrawal — where one side of the argument takes him- or herself out of the discussion. This is considered a yield-lose scenario because one side of the argument yields to the other without anyone really considering what the best solution for the project may be.

Dealing with project constraints

A constraint is anything that restricts a project manager’s options.

- Schedule. You and the project team must create the software deliverable by such-and-such date — or else.
- Budget. Of course, you’ve got a budget, but is it enough?
- Resources. You probably have your favorite programmers that you’d like to work with on every project because they’re just so gosh-darn good.
- Technology. When it comes to software project management, you have to deal with surprises.
- Competence. No one likes to admit to not knowing something — especially programmers.
- Management. Ah, here’s every project manager’s favorite scapegoat. Have you ever said
  - We don’t have enough time!
  - We don’t have enough cash!
  - We don’t have enough programmers!
  - Management is setting unrealistic expectations!

Creating the Project Scope



Creating a Work Breakdown Structure

- Cost estimating. The WBS allows you to create accurate cost estimates to create the thing the project requires.
- Cost budgeting. The WBS allows you to track actual costs against the estimates for the things your project will create.
- Resource planning. The WBS components require people and things to create. By creating the WBS, you can accurately capture everything you’ll need to complete the project.
- Risk management planning. Planning for risks when you can’t see what you’re creating can be tough. The WBS illustrates the things you’ll create and then you’ll have a clearer picture of where risks may be hiding.
- Activity definition. The end result of the WBS is to create an activity list. The activity list, or activity definition, lists all of the actions your project team will need to do to build the stuff in the WBS

8/80 Rule

- This is a general guideline that breaks down items into work packages. A work package is a unit of time allotted to a task or deliverable. The 8/80 Rule says that a work package should equate to no more than 80 hours of work and not less than 8 hours of work to create the deliverable.

Break down the scope into major buckets of things the project will create

- Project management deliverables
- Database deliverables
- Server deliverables
- End-user deliverables
- Education and documentation deliverables

- Decompose these deliverables again into smaller units or work packages
  - If you’ve decomposed deliverables down and the smallest item you have still equates to 400 hours of labor, break down the WBS some more.

Making updates to the WBS

- The WBS is your scope baseline, so any changes to the scope must be documented here. Otherwise, the following bad things could happen:
- Time and cost baselines may be skewed because they don’t match what’s in the WBS.
- Your customer may be confused as to why the WBS doesn’t match the deliverable you’ve provided.
- Project team members may be out of synch about what they’re supposed to be creating and what the WBS calls for.
- If someone in management reviews the WBS and the project deliverables don’t match up, you must have that conversation. Nobody wants to have that conversation. Especially you.
- Future projects based on your current project will have faulty information

Using a code of accounts

- You first identify a project number for your software project; let’s just say, in this example, you’re creating a piece of software to organize and access millions of corporate product photos. Your project is assigned the name PhotoBug 675. In your WBS, you abbreviate it to PB675. Each major component at the second level of the WBS also begins with PB675, but you append each stage or category of work with .1, .2, .3, and so.
- So, for example, say you’re starting with the database component of the software (makes sense; good choice). This item is called PB675.1 in the WBS. Your SQL Server 1 and SQL Server 2, at the next level of deliverables, are identified as PB675.1.1 and PB675.1.2, respectively