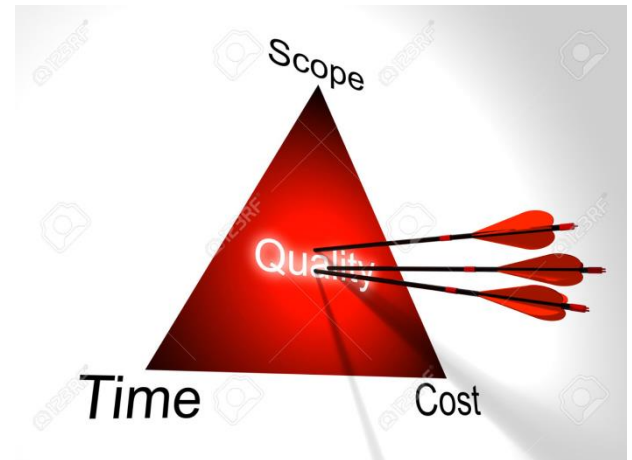


Project Scope Management

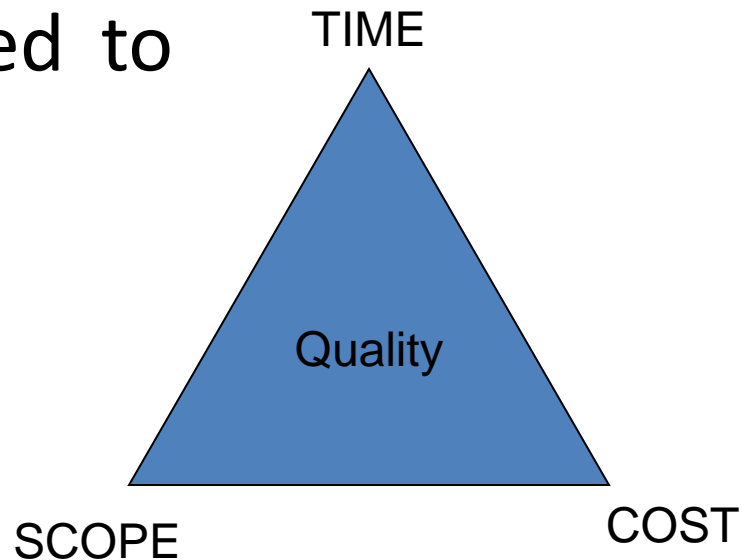


Project Scope Management

- “The processes required to ensure that the project includes all the work required, and only the work required, to complete the project successfully”
- Answers the question – “What will the project produce at the end”.

Why Do We Manage Scope?

- Can't manage schedule and budget if scope is out of control (Triple Constraint)
- Scope documents are used to manage expectations



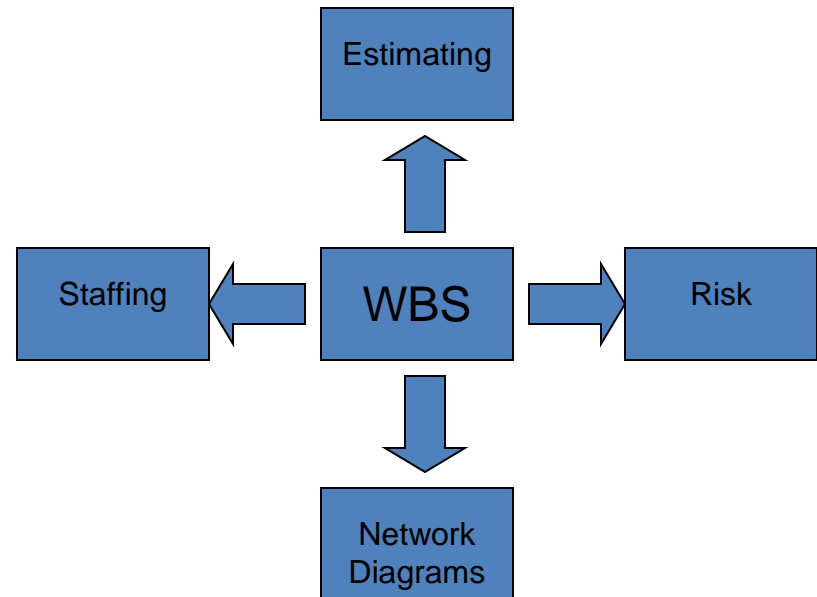
Scope Management Key Points

What is scope management

- Checking to ensure that one is completing work
- Saying No to additional work not in the charter
- Preventing extra work/gold plating

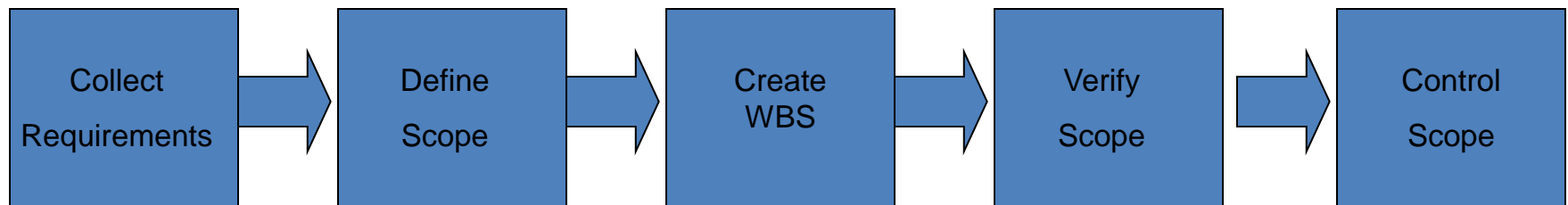
Work Breakdown Structure (WBS)

- Foundation of the project, all planning and controlling is based on the WBS
- Identifies all work to be performed, if it is not in the WBS it does not need to be done
- Graphical picture of work



How Do We Manage Scope?

- Five processes
 - **Collect Requirements**
 - **Define Scope**
 - **Create WBS**
 - **Verify Scope**
 - **Control Scope**



Project Scope Management Summary

Planning

Process: **Scope planning**

Output: Project scope management plan

Process: **Scope definition**

Output: Project scope statement, requested changes to the project, updates to the project scope management plan

Process: **Create WBS**

Output: WBS, WBS dictionary, scope baseline, requested changes to the project, updates to the project scope statement and project scope management plan



Monitoring and Controlling

Process: **Scope verification**

Outputs: Accepted deliverables, requested changes, recommended corrective actions

Process: **Scope control**

Outputs: Requested changes, recommended corrective actions, updates to the project scope statement, WBS and WBS dictionary, scope baseline, project management plan, and organizational process assets



Project Start

Project Finish





Requirements Document

Collect Requirements

In this process, you find out all of the stakeholder's needs and write them down so that you know what to build and your requirements can be measured and tracked.

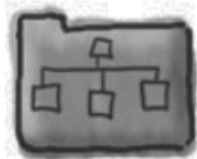


Project Scope Statement

Define Scope

Here's where you write down a detailed description of the work you'll do and what you'll produce.

When you do this right, the stakeholders are never unpredictable because you already understand their needs.



Work Breakdown Structure

Create WBS

The work breakdown structure (or WBS) organizes all of your team's work into work packages—or discrete pieces of work that team members do—so that you can keep the momentum of the project going from the start.

Pay attention to the WBS—there will be a lot of questions about it on the exam.



Change Requests

Control Scope

We already know how important it is to control changes on your project. When scope changes aren't controlled, it leads to the most frustrating sort of project problems. Luckily, you already know about change control, and now you can use it to manage your project's scope.

Remember Integrated Change Control from the last chapter? Now you'll see it in action.



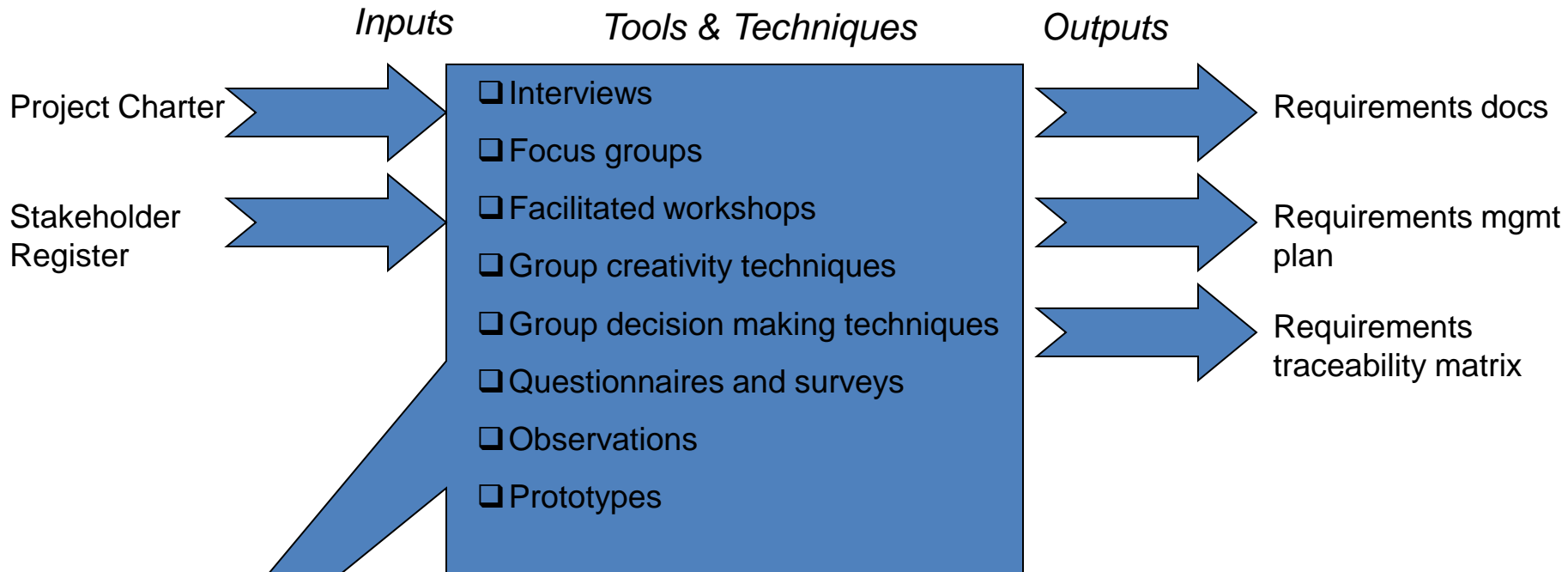
Accepted Deliverables

Verify Scope

Once the work is complete, you need to make sure that what you're delivering matches what you wrote down in the scope statement. That way, the team never delivers the wrong product to the customer.

On the exam, "customer" can mean the same thing as "client" and "sponsor."

Collect Requirements

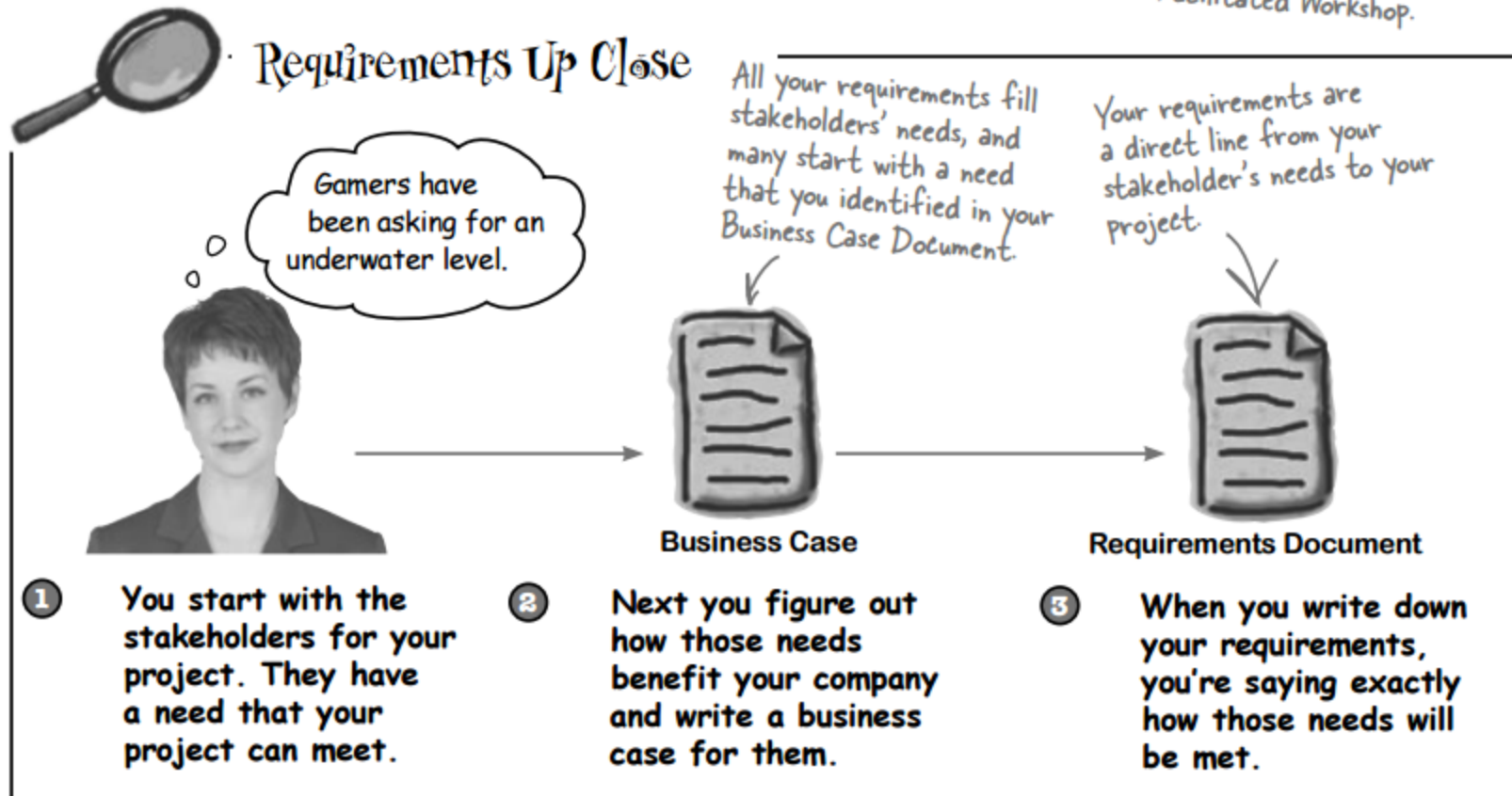


Interviews are important ways to get your stakeholders to explain how they'll use the product or service your project is creating. By talking to people one-on-one, you can get them to explain exactly what they need so that you can be sure that your project can meet its goals.

Focus Groups are another way to get a group of people to discuss their needs with you. By letting a group discuss the end product together, you can get them to tell you requirements that they might not have thought of by themselves.

Facilitated Workshops are more structured group conversations where a moderator leads the group through brainstorming requirements together. In facilitated workshops, misunderstandings and issues can get reconciled all at once because all of the stakeholders are working together to define the requirements.

← If you've ever done a Joint Application Design (JAD) session where users and the development team work together to define requirements, it's considered a Facilitated Workshop.



Make decisions about requirements



A big project usually has a lot of stakeholders, and that means a lot of opinions. You'll need to find a way of making decisions when those opinions conflict with each other. There are four major decision-making techniques you can choose from.

Unanimity means everyone agrees on the decision.

Majority means that more than half the people in the group agree on the decision.

Plurality means that the idea that gets the most votes wins.

Dictatorship is when one person makes the decision for the whole group.



You'll need to know the difference between the four different decision techniques for the exam. Here are the minutes from a Facilitated Workshop that the CGW team held with all of its stakeholders. Identify which of decision-making technique was used in each case.

1. The group voted on the CCG (cud-chewer gun) five times but decided not to include it because they couldn't get everyone to agree on it.

- Unanimity Plurality
 Majority Dictatorship

2. The VP of Engineering told everyone that they had to come up with a new character for Team Guernsey. Since he's the highest-ranking person in the room, nobody argued with him.

- Unanimity Plurality
 Majority Dictatorship

3. There were 10 new scenery suggestions up for approval, but only 5 could make it into the game. The team chose the top 5 in a general vote.

- Unanimity Plurality
 Majority Dictatorship

4. Over half the group wanted to see a new story that involved Farmer Ted. So that requirement was recorded as an absolute necessity.

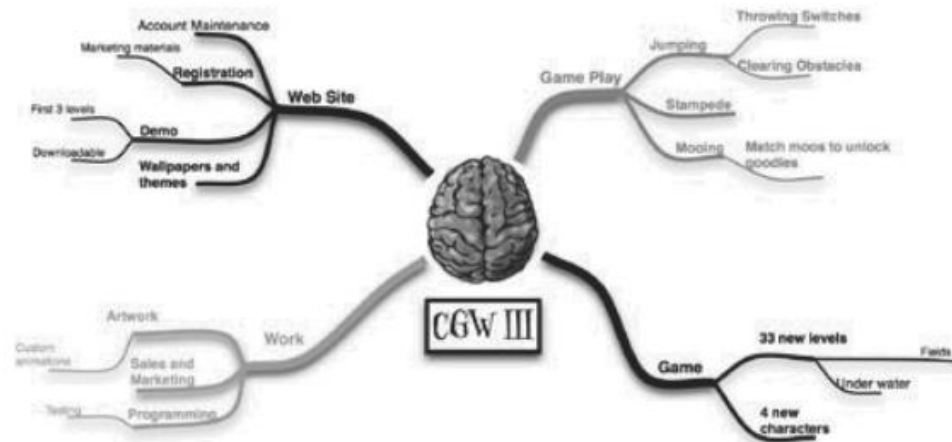
- Unanimity Plurality
 Majority Dictatorship

Help your team to get creative



Getting your team to think creatively can help you create a better product from the start. **Group Creativity Techniques** are all about getting those creative juices flowing while you gather your requirements.

Mind Maps are a good way to visualize the way your ideas relate to each other. When you've finished working through an idea, it sometimes helps to create a map of how you got there and show which ideas can be grouped together.



The name, Delphi Technique, comes from the Oracle at Delphi.

Here's the mindmap the CGW team made from their brainstorming session.

Delphi Technique is a way of letting everyone in the group give their thoughts about what should be in the product while keeping them anonymous. When you use the Delphi Technique, everybody writes down their answers to the same questions about what the product needs to do and then hands them into a moderator. The questions could be about specific features that the product should have.

When the CGW team used the Delphi Technique, here were a few of their questions:

CGW III Delphi Questionnaire

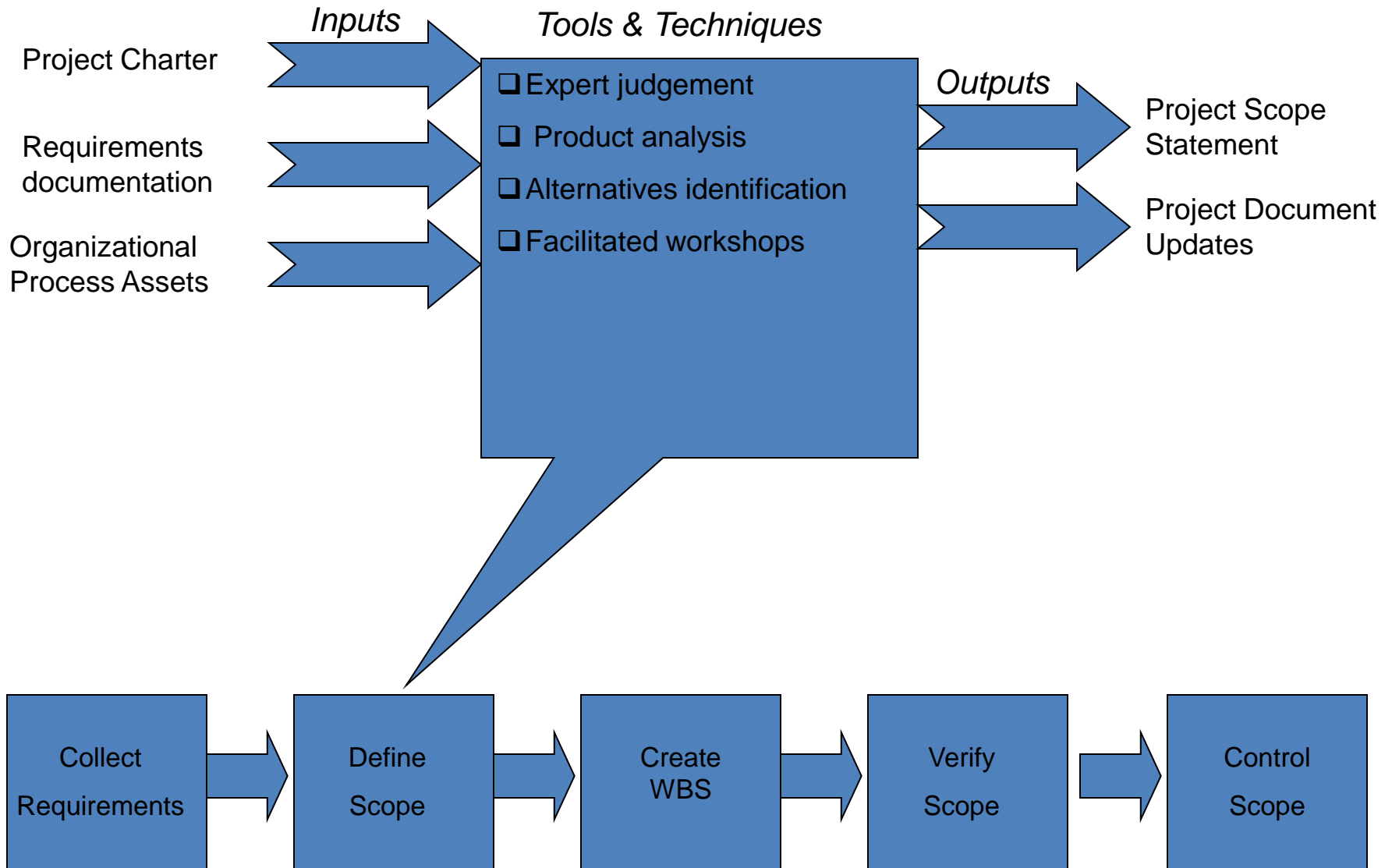
- What new levels would you like to see in the game?
- What new abilities should Bessie have?
- What should the story for CGW III be about?

The Delphi Technique can be used to estimate the work the team will need to do and how long it will take too!

Requirements Traceability Matrix								
Project Name:								
Cost Center:								
Project Description:								
ID	Associate ID	Requirements Description	Business Needs, Opportunities, Goals, Objectives	Project Objectives	WBS Deliverables	Product Design	Product Development	Test Cases
001	1.0							
	1.1							
	1.2							
	1.2.1							
002	2.0							
	2.1							
	2.1.1							
003	3.0							
	3.1							
	3.2							
004	4.0							
005	5.0							

Figure 5-6. Example of a Requirements Traceability Matrix

Define Scope



Project Scope Statement

- Product scope description
- Product acceptance criteria
- Project deliverables
- Project exclusions
- Project constraints
- Project assumptions

Remember the difference between
product and project.

The product scope is all about the final product—its features, components, pieces.

When people talk about scoping out their products, a lot of times they're talking about figuring out the features of the product, not the work that goes into it.

Product scope means the features and functions of the product or service that you and your team are building.

When we talk about scoping out a project, we mean figuring out all of the work that needs to be done to make the product.

Project scope is all of the work that needs to be done to make the product.

THIS is a big part of what the project manager is concerned with... the work the team has to do.

Scope creep means uncontrolled changes that cause the team to do extra work.

This means changes that just went in without anyone bothering to figure out what effect they'd have on the project's time, cost, scope, quality, risk, or resources.



Here are some attributes of Cows Gone Wild III. Which are project scope and which are product scope?

1. Programming

Project Scope

Product Scope

2. 34 levels in the game

Project Scope

Product Scope

3. Graphic design

Project Scope

Product Scope

4. Four playable characters

Project Scope

Product Scope

5. Great graphics

Project Scope

Product Scope

6. Testing

Project Scope

Product Scope

7. Mac and PC compatible

Project Scope

Product Scope

8. A "boss battle" milk fight level at the end

Project Scope

Product Scope

—————→ **Answers on page 216.**



Here are some attributes of Cows Gone Wild III. Which are project scope and which are product scope?

1. Programming

Project Scope

Product Scope

2. 34 levels in the game

Project Scope

Product Scope

3. Graphic design

Project Scope

Product Scope

4. Four playable characters

Project Scope

Product Scope

5. Great graphics

Project Scope

Product Scope

6. Testing

Project Scope

Product Scope

7. Mac and PC compatible

Project Scope

Product Scope

8. A "boss battle" milk fight level at the end

Project Scope

Product Scope

The scope statement tells you what you have to do

After you have done your scope planning, figured out as much as you could using stakeholder and product analysis, and identified all of the possible ways of doing the work, you should be ready to add any new findings to the scope statement.



Outputs

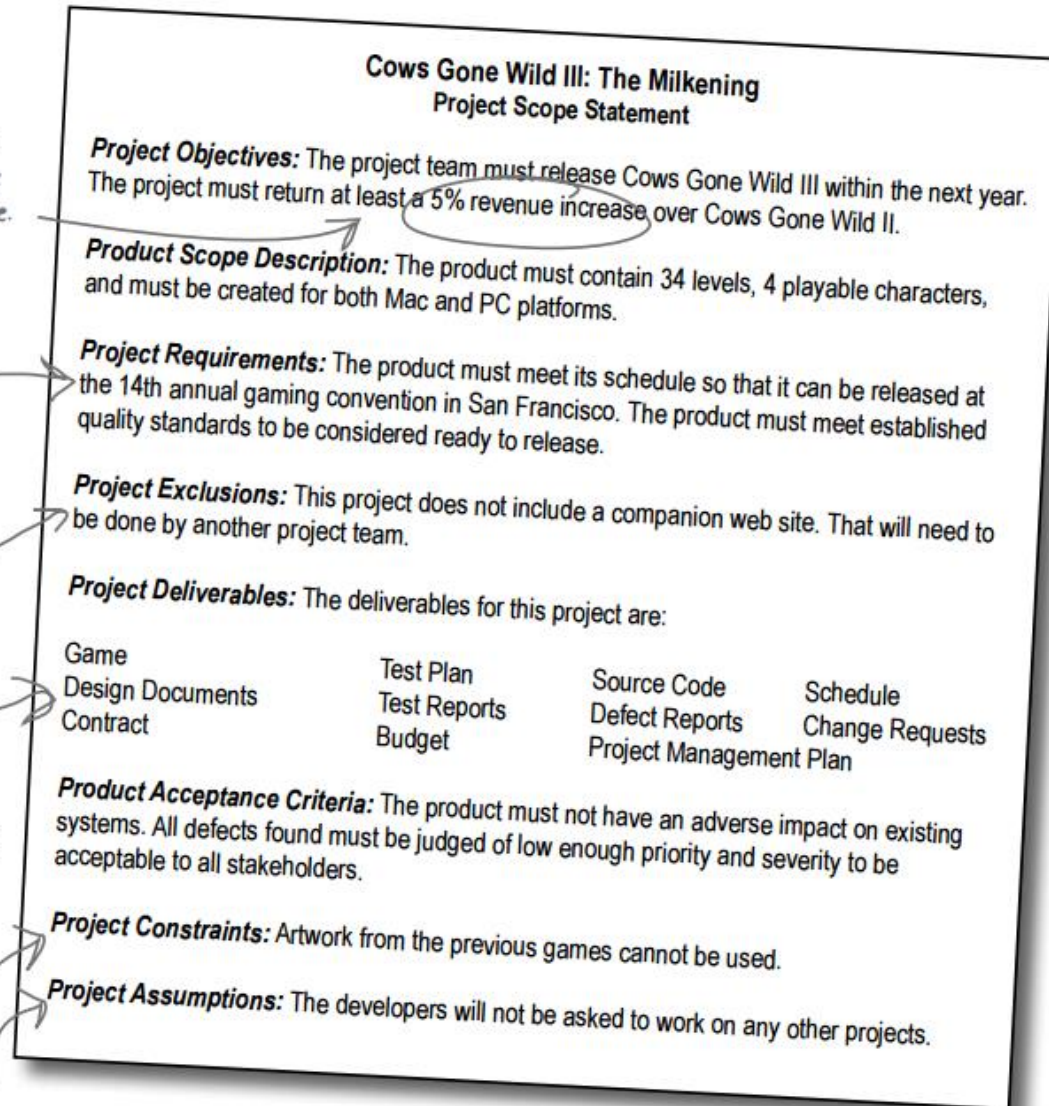
All the project objectives need to be measurable.

Even though you probably can't fit ALL of the requirements here, there should be enough detail to let you keep on planning and refer back to it later.

This means looking for all the work the project DOESN'T include.

The deliverables listed here are EVERYTHING the project creates, including project management stuff.

Constraints are known limitations. Assumptions are things you think are true.



1. The game must have fewer than 15 defects per 10,000 lines of code.

2. There will be four graphic designers reporting to the art director, and six programmers and four testers reporting to the development manager.

3. No more than 15 people can be allocated to work on the game at any time.

4. Scenery artwork.

5. The product shall reduce tech support calls by 15%.

6. The game needs to run on a machine with 1 GB of memory or less.

A. Project objectives

B. Project deliverables

C. Project constraints

D. Initial project organization

E. Project requirements

F. Product acceptance criteria

—————> **Answers on page 217.**

The project scope statement tells what work you are—and are not—going to do to do in the project.

1. The game must have fewer than 15 defects per 10,000 lines of code.

2. There will be four graphic designers reporting to the art director, and six programmers and four testers reporting to the development manager.

3. No more than 15 people can be allocated to work on the game at any time.

4. Scenery artwork

5. The product shall reduce tech support calls by 15%.

6. The game needs to run on a machine with 1 GB of memory or less.

A. *Project objectives*

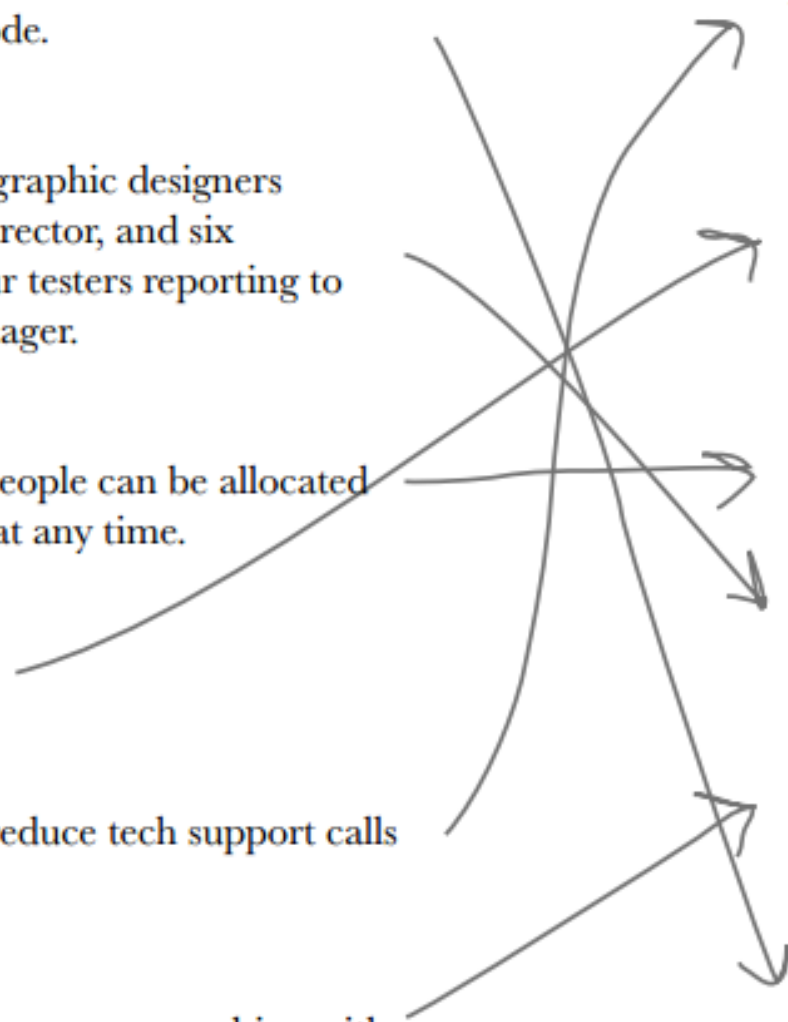
B. *Project deliverables*

C. *Project constraints*

D. *Initial project organization*

E. *Project requirements*

F. *Product acceptance criteria*





You'll need to know the difference between defining the scope and collecting the project's requirements for the exam. Which of these things is part of the Project Scope Statement, and which is part of the Requirements Document?

1. The work required to create the graphics

Requirements Doc Scope Statement

2. New characters in the game

Requirements Doc Scope Statement

3. 33 new levels

Requirements Doc Scope Statement

4. The performance requirements for the product

Requirements Doc Scope Statement

5. A description of how the WBS is created

Requirements Doc Scope Statement

6. How the software will be tested

Requirements Doc Scope Statement

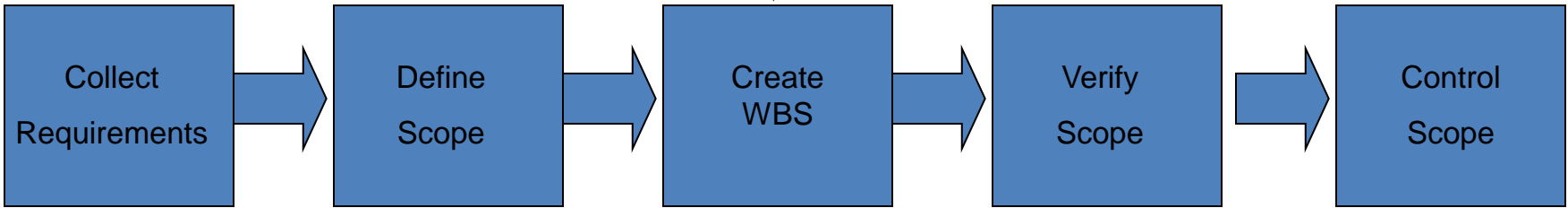
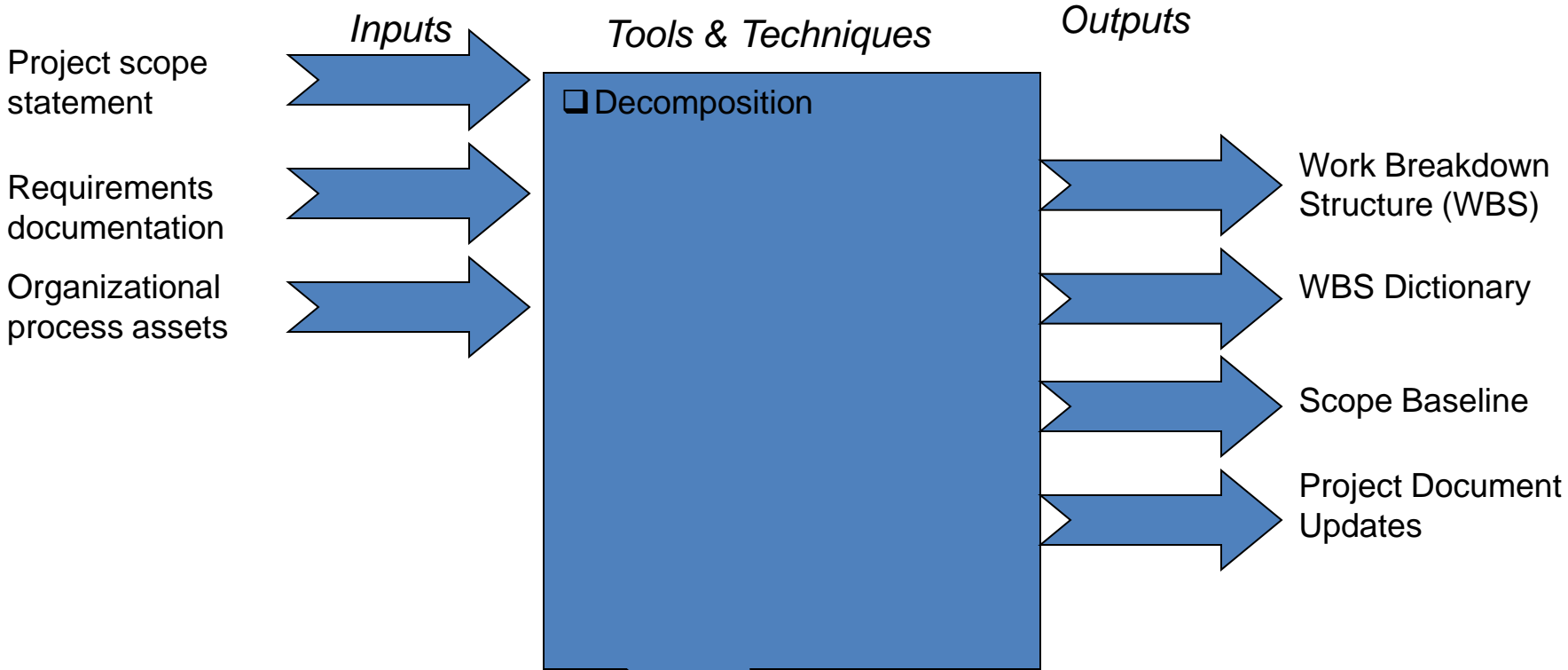
7. How the stakeholders will verify the deliverables

Requirements Doc Scope Statement

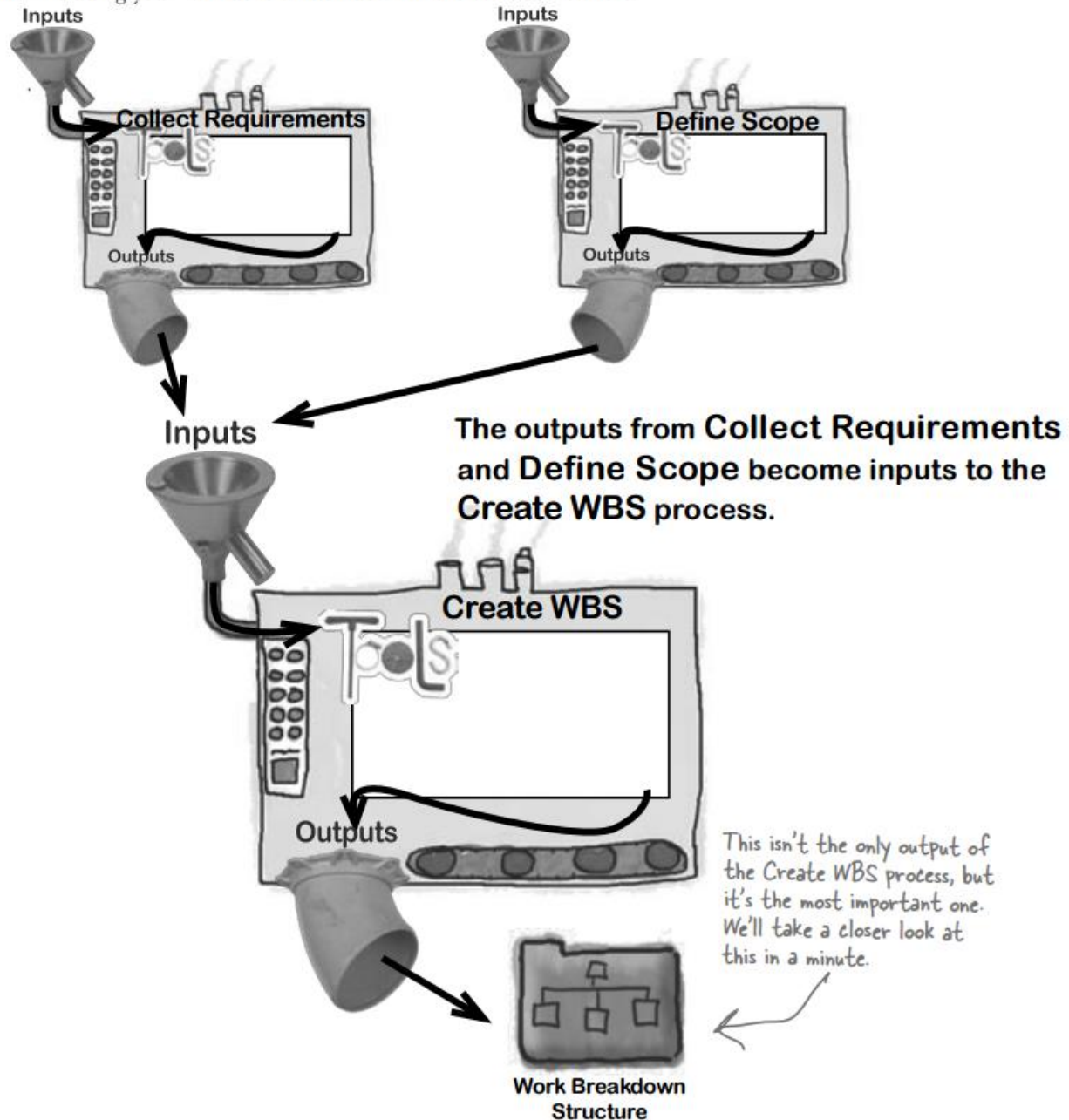
8. A list of all artwork that will be created

Requirements Doc Scope Statement

Create WBS



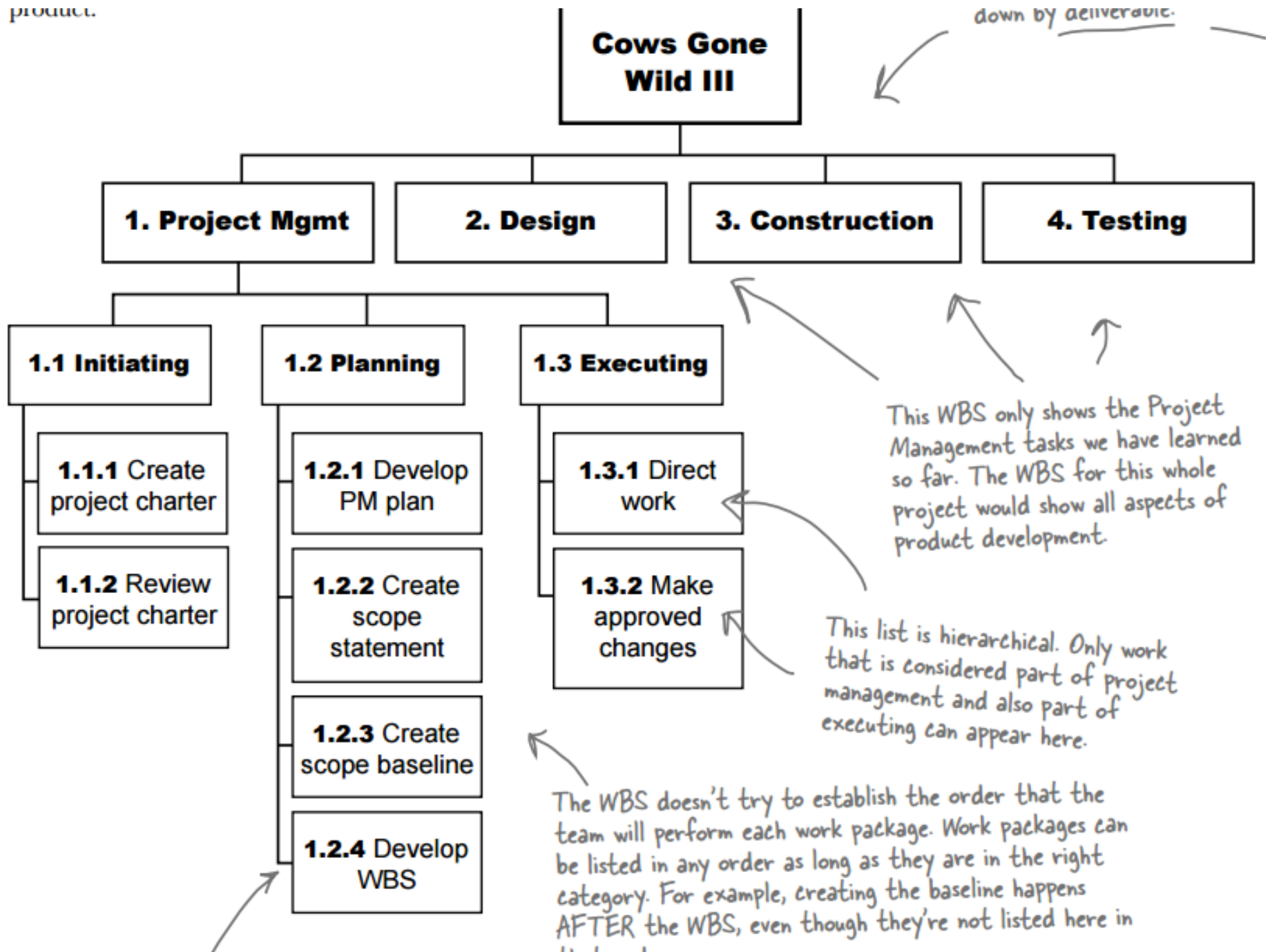
The **Create WBS** process is the most important process in the Scope Management knowledge area because it's where you actually figure out all the work you're going to do. It's where you create the **Work Breakdown Structure** (or WBS), which is the main Scope Management output. Every single thing that anyone on the project team—including you—will do is written down in the WBS somewhere.



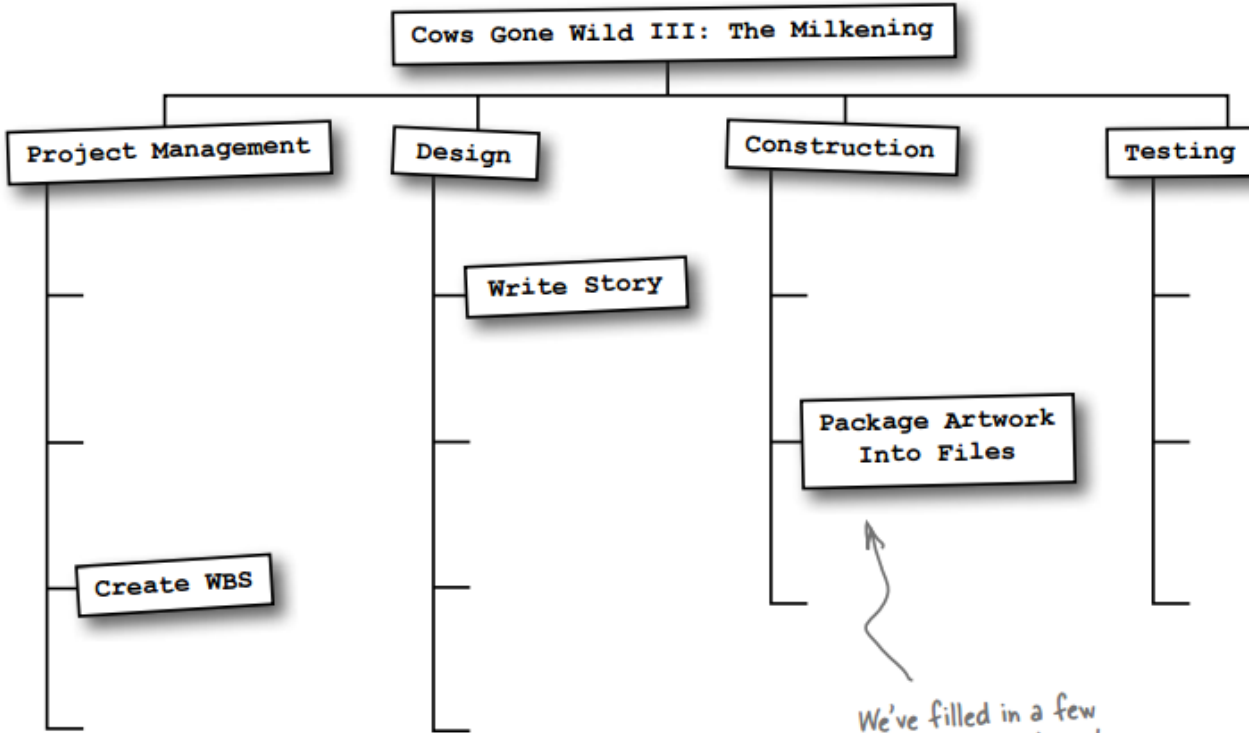
Work Breakdown Structure

- Start with major project deliverables or phases
- “Decomposition” is breaking down the deliverables into more manageable parts
 - Not all branches need the same level of decomposition!
- Decompose into “Work Packages”

product.

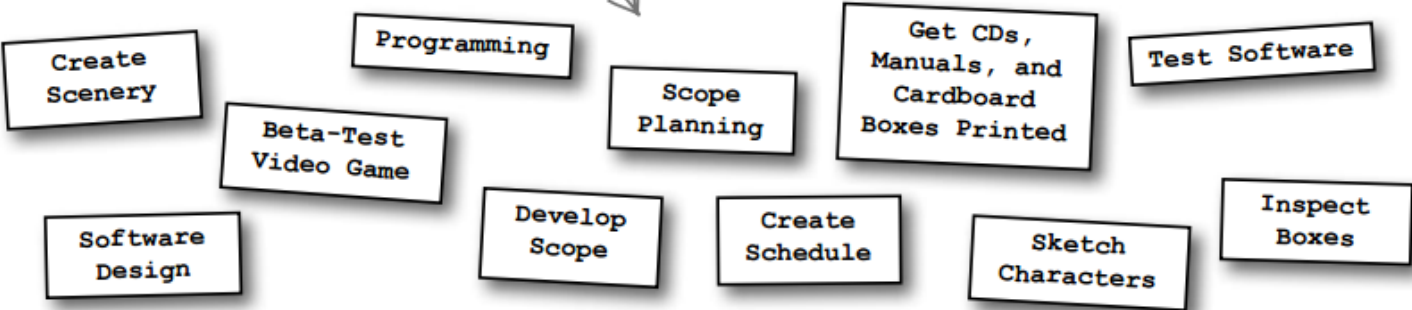


On this page, create a work breakdown structure broken down by project phase.

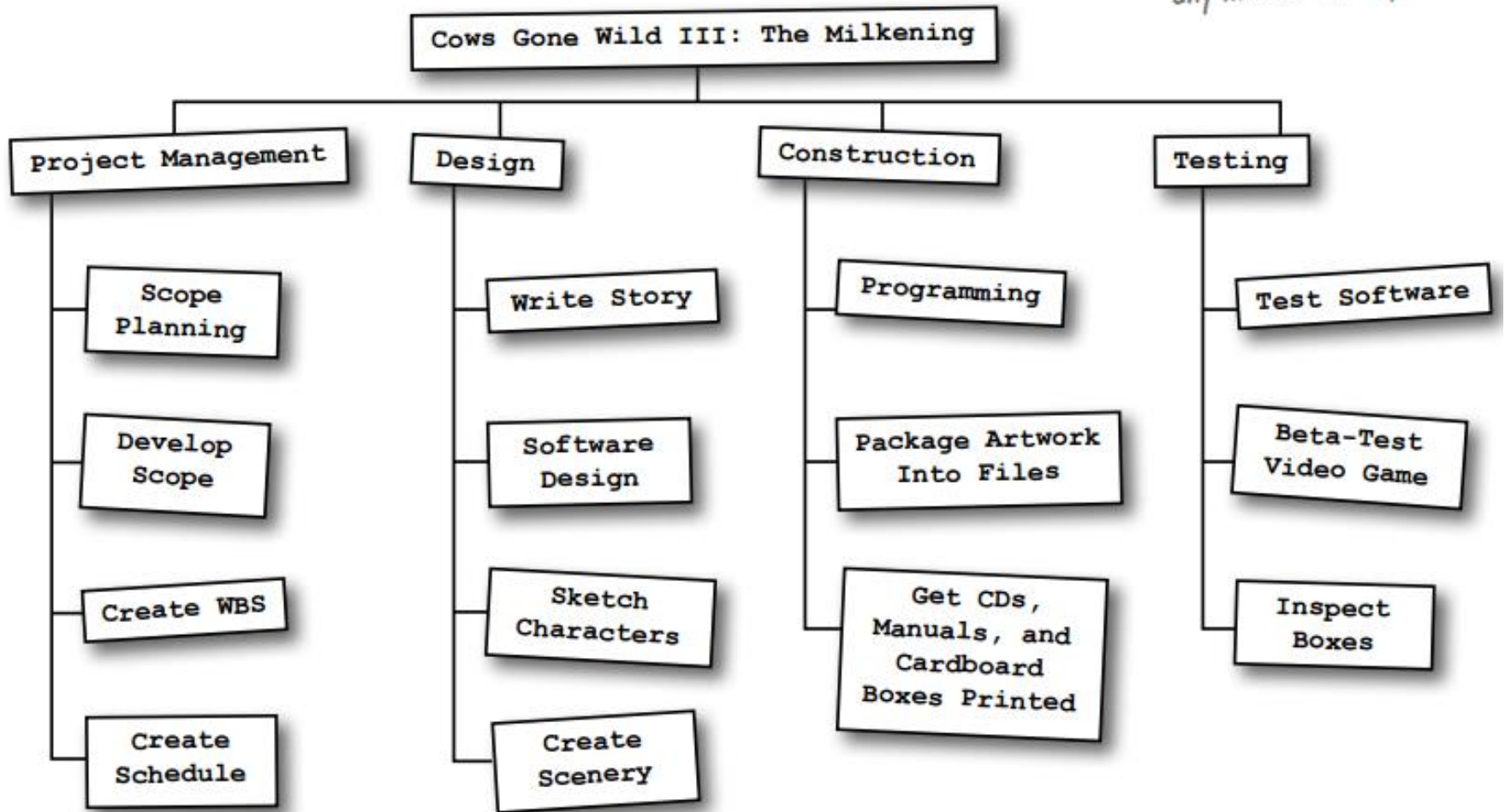


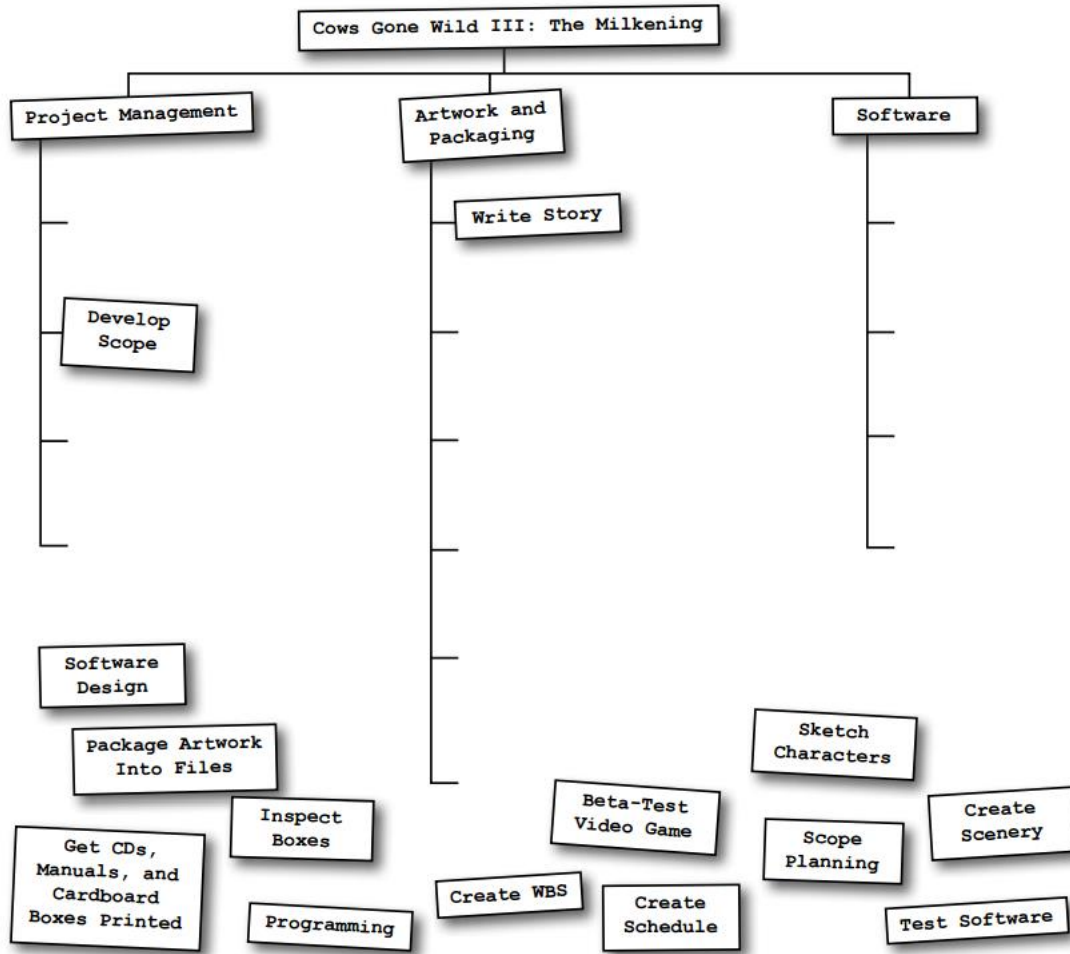
We've filled in a few work packages to get you started.

Use these magnets to fill in the blank WBS.



any number of ways.

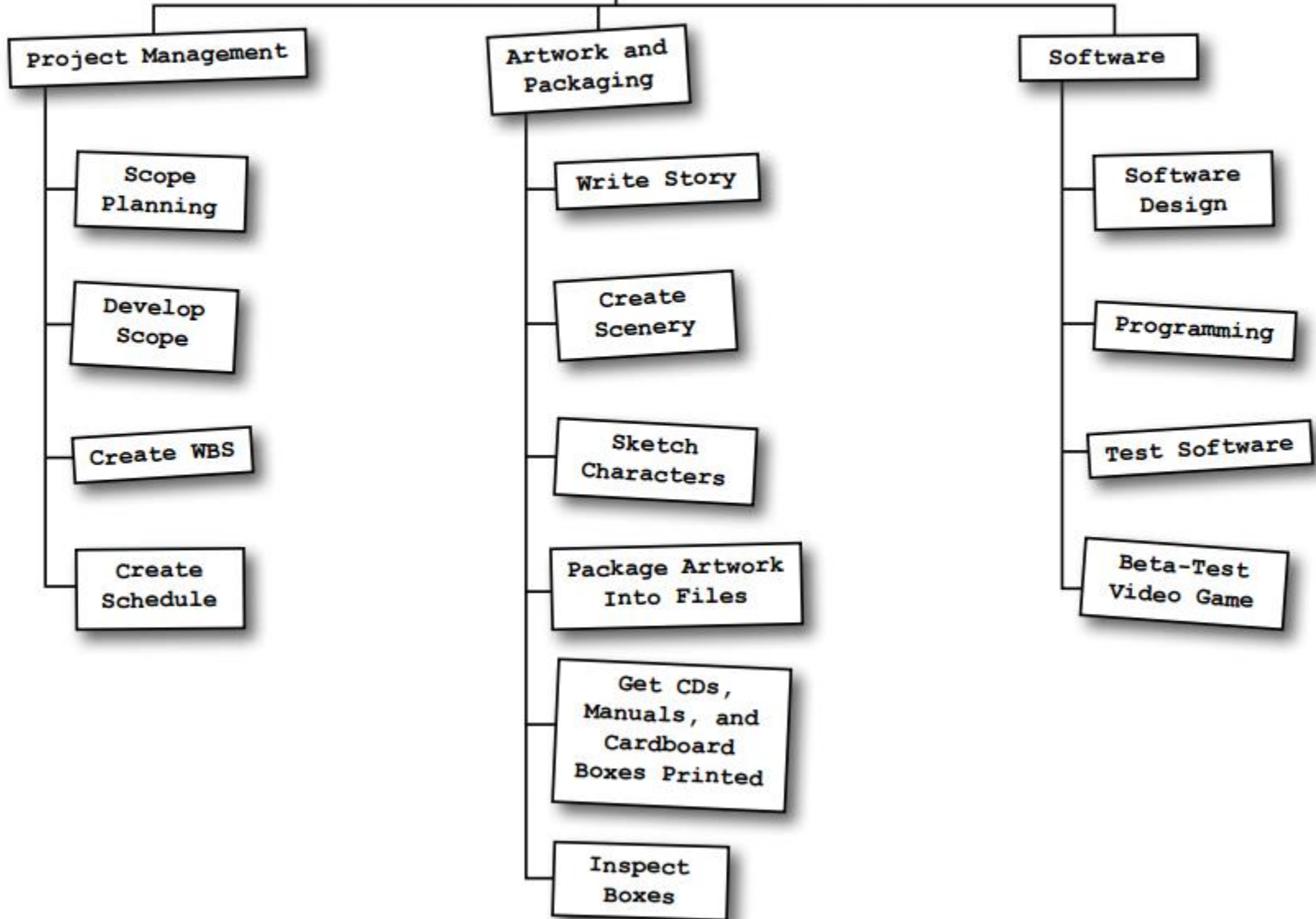




This WBS has the same work packages, but they're broken down differently.



Cows Gone Wild III: The Milkening



Inside the work package

You've probably noticed that the work breakdown structure only shows you the name of each work package. That's not enough to do the work! You and your team need to know a lot more about the work that has to be done. That's where the **WBS Dictionary** comes in handy. It brings along all of the details you need to do the project work. The WBS Dictionary is an important output of the Create WBS process—the WBS wouldn't be nearly as useful without it.

This is one of the WBS Dictionary entries for the Cows Gone Wild III project. It goes with the "Test Software" work package in the WBS.

This is just a description of the work that needs to be done.

Don't forget that the WBS doesn't show dependencies among work packages.

Test Software
WBS Dictionary Entry

Work Package ID and Name: 3.2.4 – Test Software

Statement of Work:
The goal of software testing is to verify that the Cows Gone Wild III software implements all of the requirements. Each requirement will be fully tested by a team of quality engineers.

Responsible Organization: Ranch Hand Games QA Team

Schedule Milestones:

- 4/26 – Programming team delivers software
- 6/18 – Functional testing and graphics testing completed
- 8/10 – QA approves software for beta testing

Quality Requirements:
The software must meet the requirements defined by the Ranch Hand Games QA team's quality standards document ("RHG QA Standards.doc")

Code of Account Identifier: RHG-236

Required Resources and Cost Estimate:

- Test planning – One QA lead and two QA analysts (\$8,500)
- Functional testing – 2 leads, 3 analysts, 11 testers (\$36,000)
- Monitor beta testing – 2 leads, 1 analyst (\$6,000)

The WBS Dictionary contains the details of every work package. It's a separate output of the Create WBS process.

Each work package has a name, and in many WBSes the work packages will also have ID numbers.

Here's what the WBS entry would look like with this ID number.

3.2.4 – Test Software

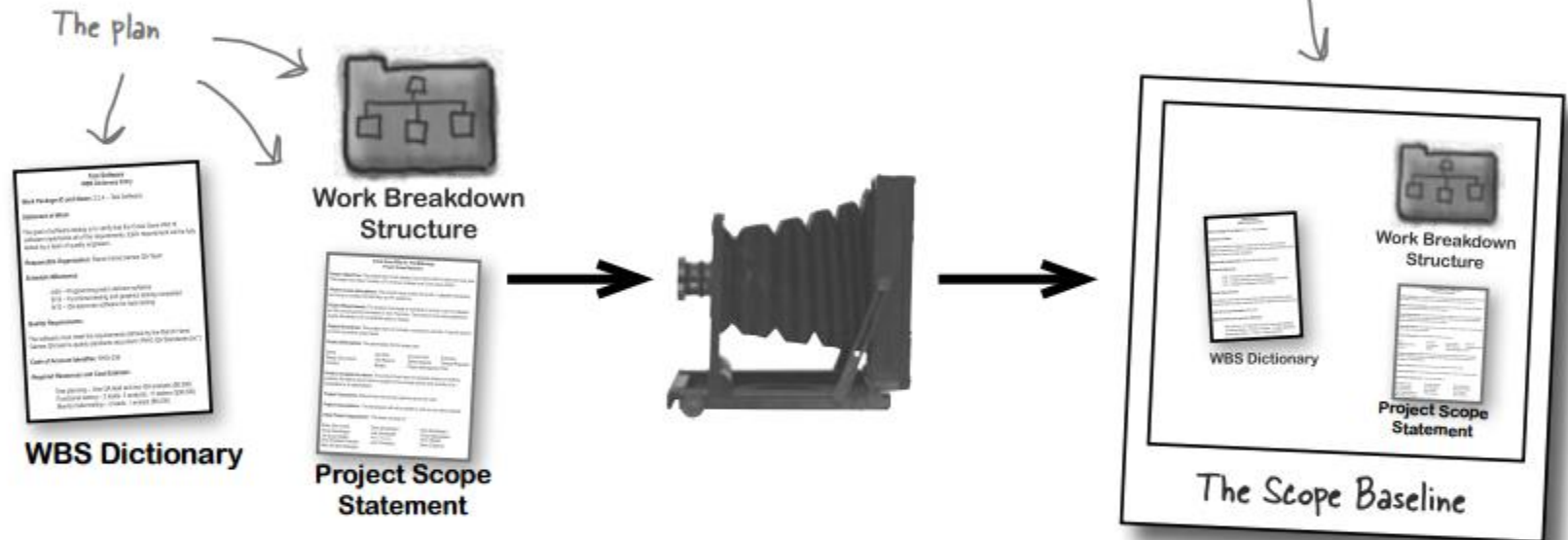
Each work package should be small enough to make cost and resource estimates.

*This account identifier is important—it's how you hook your WBS into your company's accounting system. That way you can make sure all of the work is paid for.

The baseline is a snapshot of the plan

As the project goes on, you will want to compare how you are doing to what you planned for. So, the **scope baseline** is there to compare against. It's made up of the scope statement, the WBS, and the WBS Dictionary. When work gets added to the scope through change control, you need to change the baseline to include the new work packages for that work, so you can always track yourself against the plan.

The scope baseline is a snapshot of the plan, and it's an important output of Create WBS.



So, if someone wants to add or remove work packages, that's a change?

Putting together a baseline just means making copies of your project documents so you can compare them with later versions after you put your project through change control.

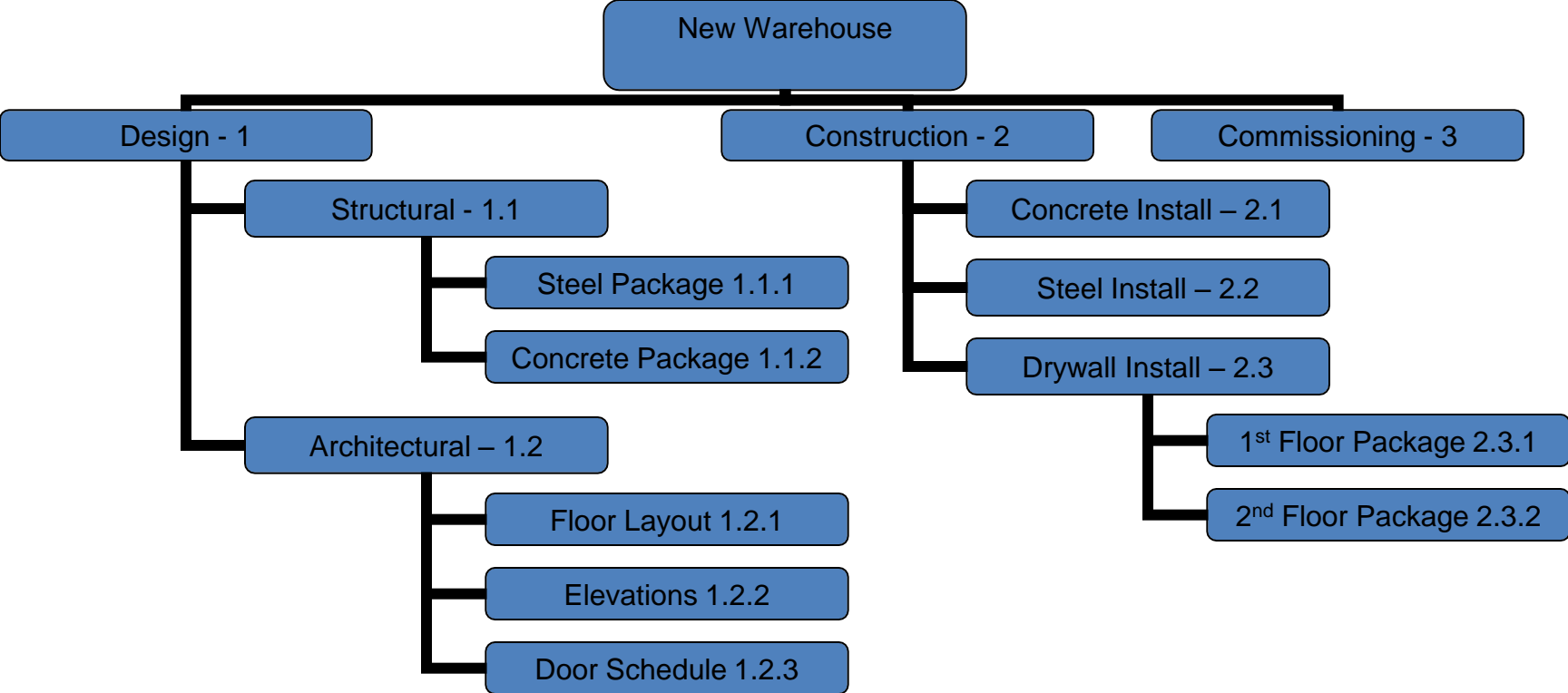
Yes. When there's a change you need to take a new snapshot.



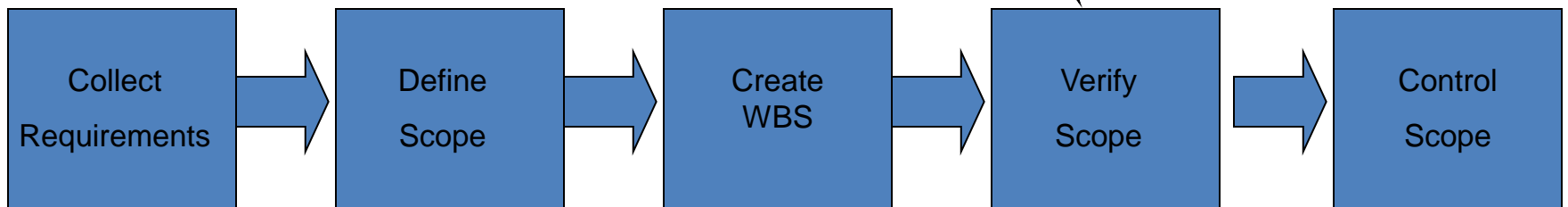
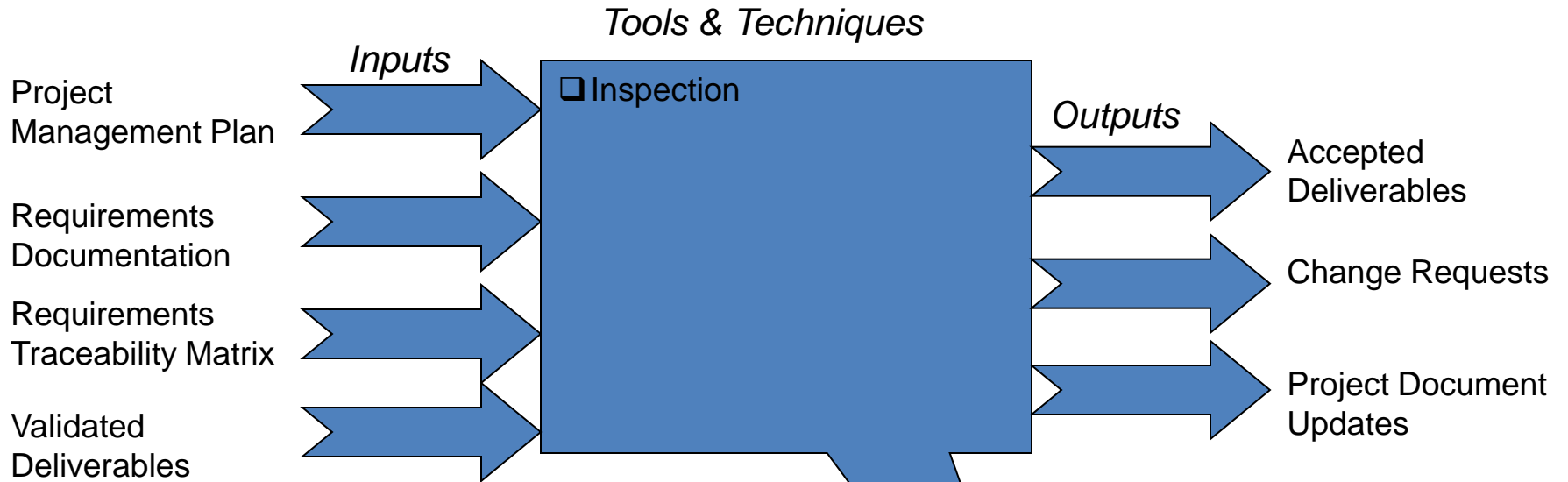
BULLET POINTS: AIMING FOR THE EXAM

- The **Create WBS** process is a really important process on the PMP exam.
- The WBS is created by **decomposing** large work products into **work packages**.
- To finalize the WBS, **control accounts** are established for the work packages.
- The **WBS Dictionary** is a description of each work package listed in the WBS.
- The inputs to WBS creation are the outputs to the Define Scope and Collect Requirements processes: the Requirements Document, and the Project Scope Statement.
- As you decompose the work, you find new information that needs to be added to the Requirements Document and the Project Scope Statement. That information is treated as a change and goes through change control. Once it's approved, it can be added into the document, and that kicks off the planning cycle again.

Work Breakdown Structure



Verify Scope





Good change

A good change makes the product better with very little downside. It doesn't cost more time in the schedule or more money from the budget, and it doesn't destabilize the product or otherwise threaten its quality.

Good changes happen pretty rarely and nearly EVERY change has some impact that should be fully explored before you go forward.



Bad change

A bad change is one that might seem from the outside like a good idea but ends up making an impact on the project constraints. Here are a couple of examples:

Scope Creep

This happens when you think you know the impact of a change so you go ahead, but it turns out that *that* change leads to another one, and since you are already making the first change, you go with the next. Then another change comes up, and another, and another, until it's hard to tell what the scope of the project is.

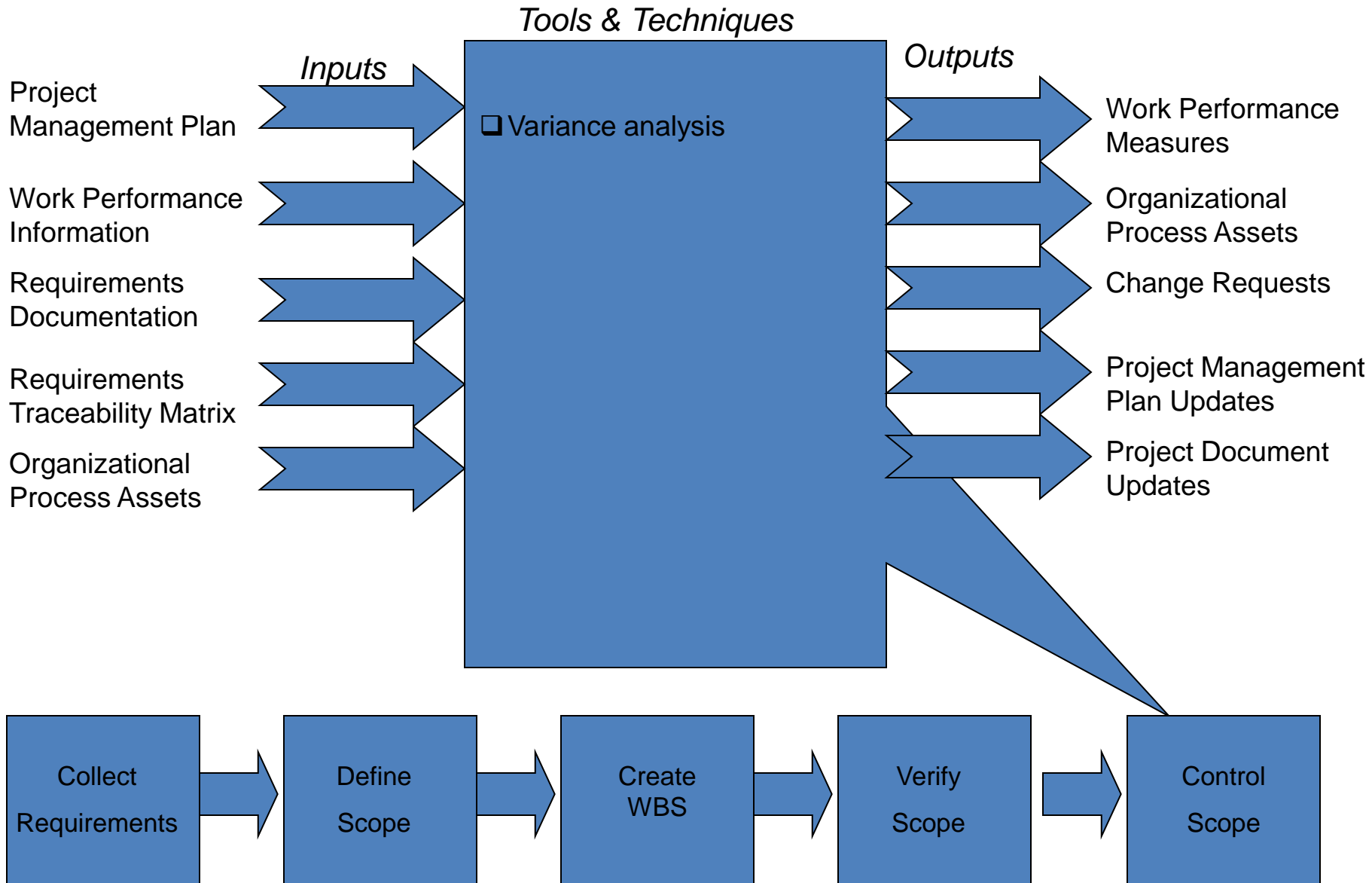
The way to avoid scope creep is to plan your changes completely.

Gold plating

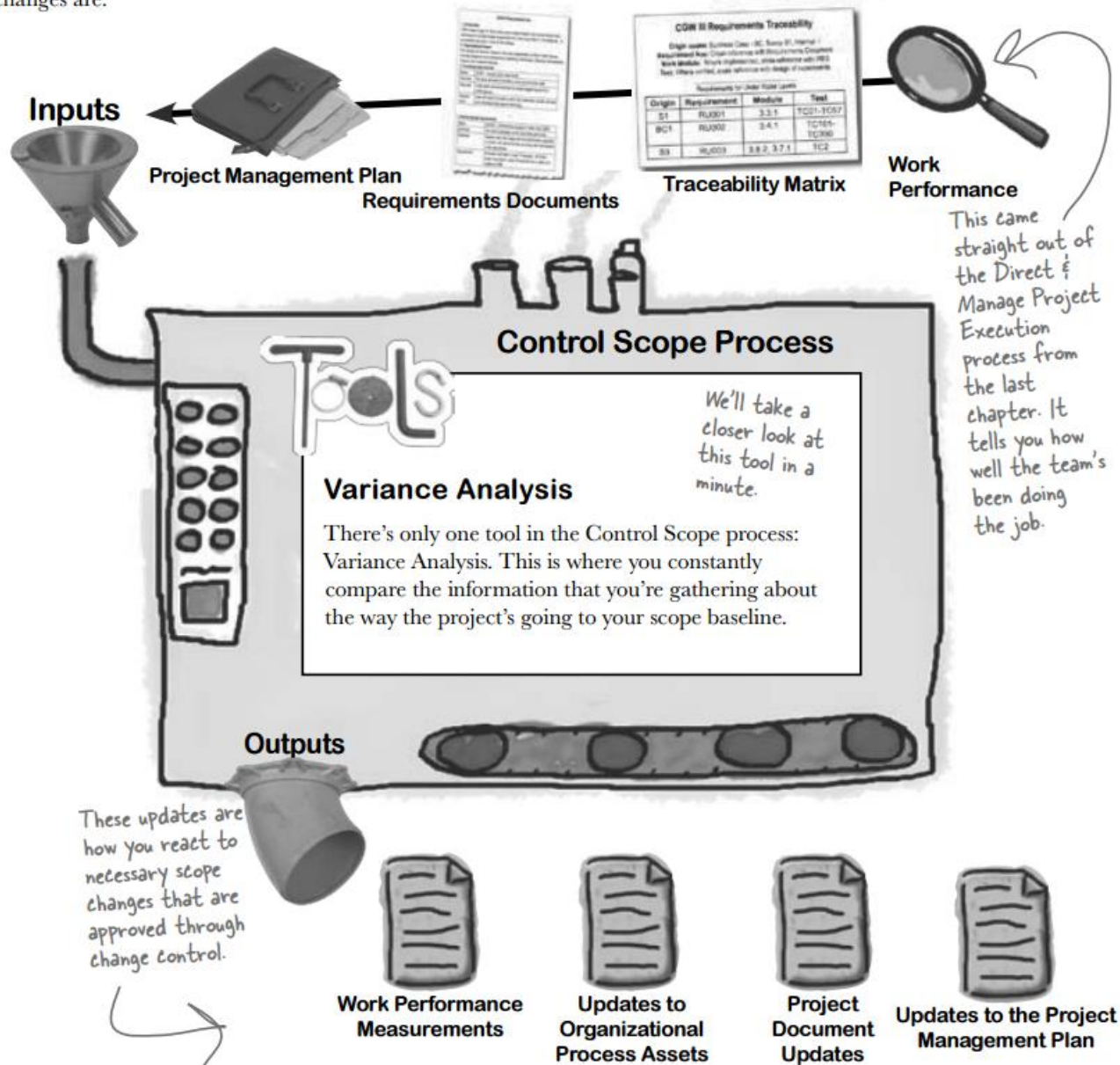
Sometimes people think of a really great improvement to the product and go ahead and make it without even checking the impact. In software, this can happen pretty easily. A programmer thinks of a way to make a feature better, for example, and just implements it, without talking it over with anybody. This may sound good, but it's not—because now you have to pay for these features you never asked for.

Be on the lookout for examples of scope creep and gold plating on the exam. Both are considered very bad and should never be done.

Control Scope



There's no way to predict every possible piece of work that you and your team are going to do in the project. Somewhere along the way, you or someone else will realize that a change needs to happen, and that change will affect the scope baseline. That's why you need the **Control Scope** process. It's how you make sure that you make only those changes to the scope that you need to make, and that everyone is clear on what the consequences of those changes are.



Anatomy of a change



Let's take a closer look at what happens when you need to make a change. You can't just go and change the project whenever you want—the whole reason that you have a baseline is so you can always know what work the team is supposed to do. If you make changes, then you need to change the baseline... which means you need to make sure that the change is **really** necessary. Luckily, you have some powerful tools to help you manage changes:

1 A change is needed

Every change starts the same way. Someone realizes that if the project sticks with the plan, then the outcome will lead to problems.



I know we're under pressure to get the game out the door, but we need to make a change. We only planned on making four meadows on level 3 and reusing them for level 6, but it's just not working for us. We've got to change this if we want the game to sell.

A change can come from anywhere — the project manager, a team member, even a stakeholder!

2 Create a change request

Before a change can be made, it needs to be approved. That means that it needs to be documented as a requested change. The only way to get a handle on a change is to write it down and make sure everyone understands it.



We'll write up a change request, and then put it through Integrated Change Control to get it approved.

3 Get the change approved

Remember Integrated Change Control from the last chapter? That's the process where the project manager takes a requested change and works with the sponsor and stakeholders to get approval to put it in place.

Think of Integrated Change Control as a kind of machine that converts requested changes into approved changes.



Requested Change



Approved Change Request

4 Do variance analysis

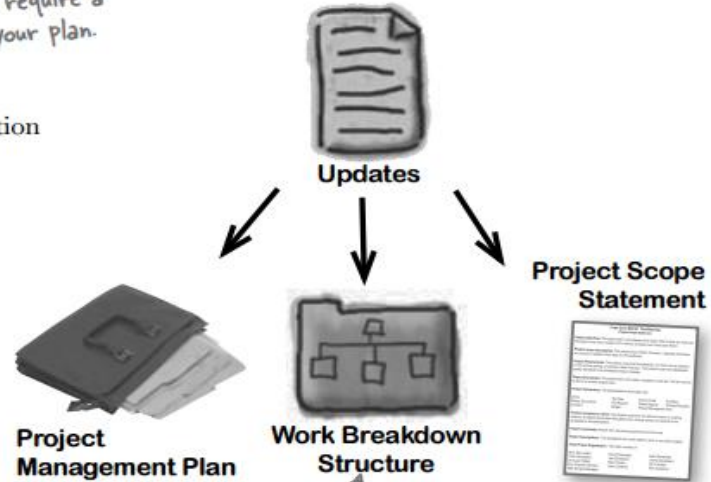
Take a look at the baseline and see how the change will affect it. This is where you decide whether you need to take some sort of corrective action. You compare the scope baseline against the change that you want to make, and figure out just how big the change really is.

You're weighing the change against the baseline to see if it's going to require a big change to your plan.



5 Replan the work

Now it's time to go back to the scope documentation and update it to reflect the change.



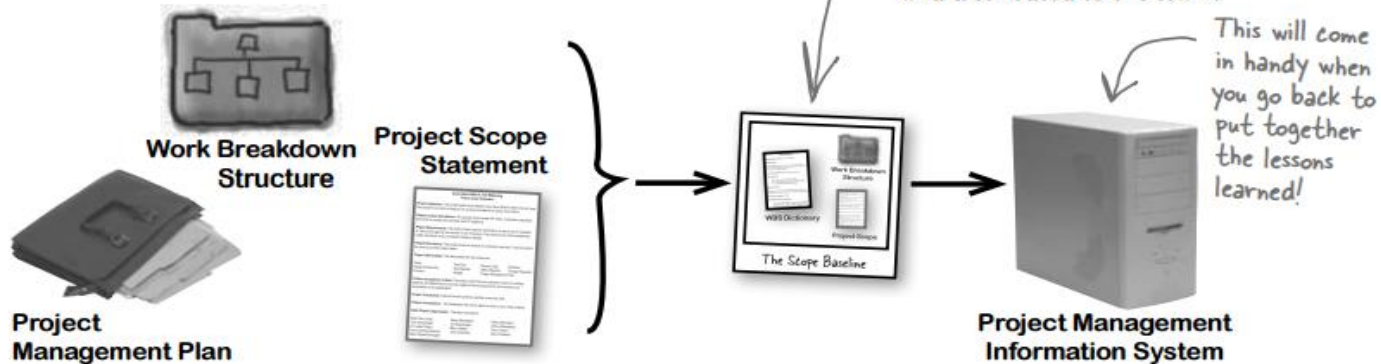
Don't forget to update the WBS Dictionary, too.

6 Create a new baseline

Now that you've figured out that you need to change the scope, it's time to update the baseline. Go back to the scope statement, WBS, and WBS Dictionary, and update them so that they reflect the change that needs to be made.

The change is done!

Now you can move on with the project using the new baseline that you saved and distributed to the team.



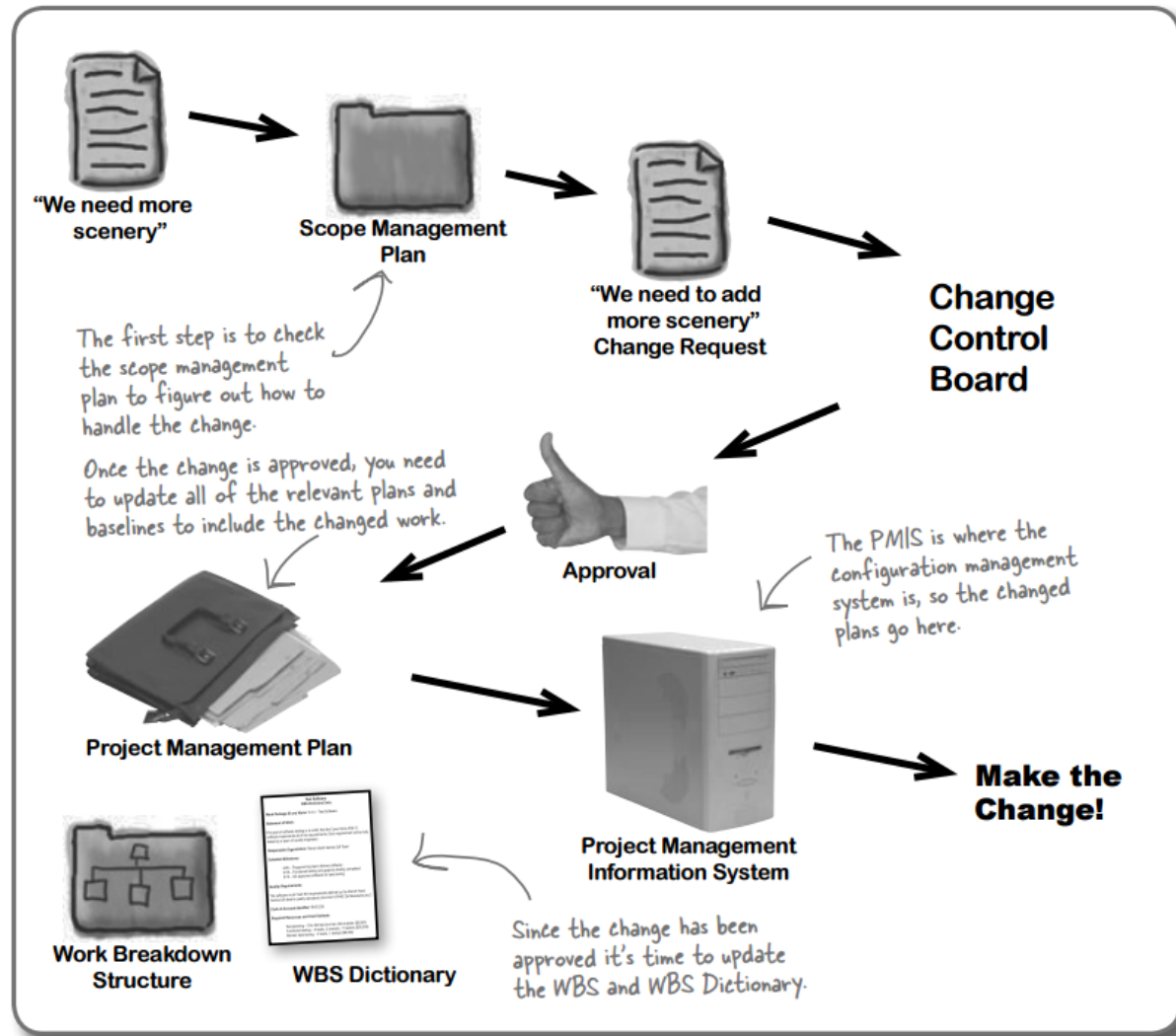
A closer look at the Change Control System



One of the most important tools in any Monitoring & Controlling process is the **Change Control System**. Let's take a closer look at how it works.

Since the folks at Ranch Hand need a change to add more scenery to *Cows Gone Wild III*, Mike takes a look at the scope management plan to understand the impact before forwarding it to the change control board. Once they approve the change, he updates the project management plan, checks it into the configuration management system, and changes the WBS and WBS Dictionary to include the new work packages.

Remember this from the last chapter? It's exactly the same change control system tool that we already learned about.



Just one Control Scope tool/technique

There's just one tool/technique in the Control Scope process. It's pretty intuitive: just take a minute and think of what you would need to do if you had to make a change to your project's scope. You'd need to figure out how big the change is, and what needs to change. And when you do that, it's called **variance analysis**.

Variance Analysis

This means comparing the data that can be collected about the work being done to the scope baseline. When there is a difference between the two, that's variance.

This tool of Control Scope is all about analyzing the difference between the baseline and the actual work to figure out if the plan needs to be corrected. If so, then you recommend a corrective action and put that recommendation through change control.

The goal of Control Scope is updating the scope, plan, baseline, and WBS info.



Control Scope Process Magnets

Whenever you make a scope change, you need to go through all of the steps of change control. So what are those steps? Arrange the magnets to show the order that you handle changes to the scope.

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____

Compare the change against the baseline

You figure out that you have to make a change

Store the updated baseline in the configuration management system

Now the team can change the way they do the work

Go back and plan for the new work

Get approval to make the change

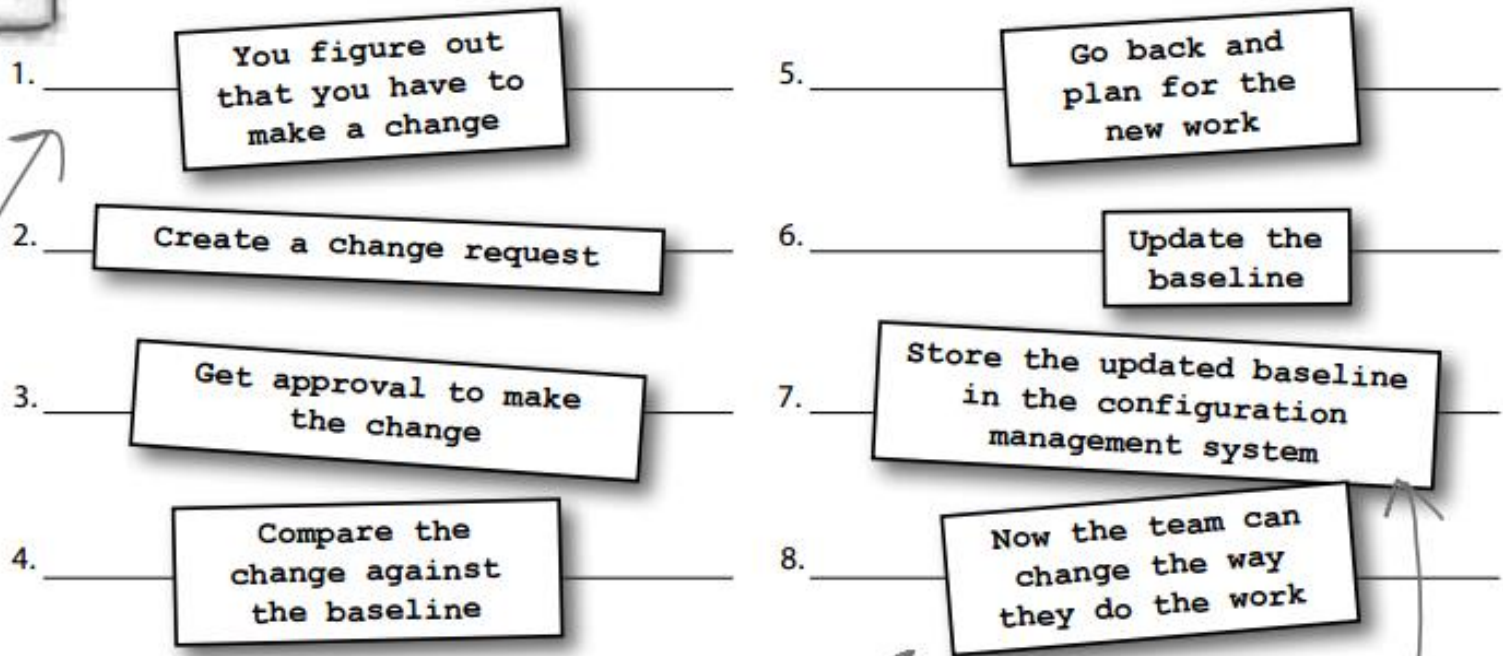
Update the baseline

Create a change request



Control Scope Process Magnets Solutions

Arrange all of the activities you do to control scope in the right order.



The whole idea behind change control is that you start by figuring out that a change needs to be made, you make sure the change is really worth making, and then you update your baseline so you can keep track of it.

Remember, Control Scope is all about changes to the scope baseline.

Make sure the team delivered the right product



When the team is done, what happens? You still have one more thing you need to do before you can declare victory. You need to gather all the stakeholders together and have them make sure that all the work really was done. We call that the **Verify Scope** process.

