2. Introduction to Program Design

What does program design mean?

Add the following two numbers

Multiply the following two numbers

A program has the following basic stages

A program has the following basic stages

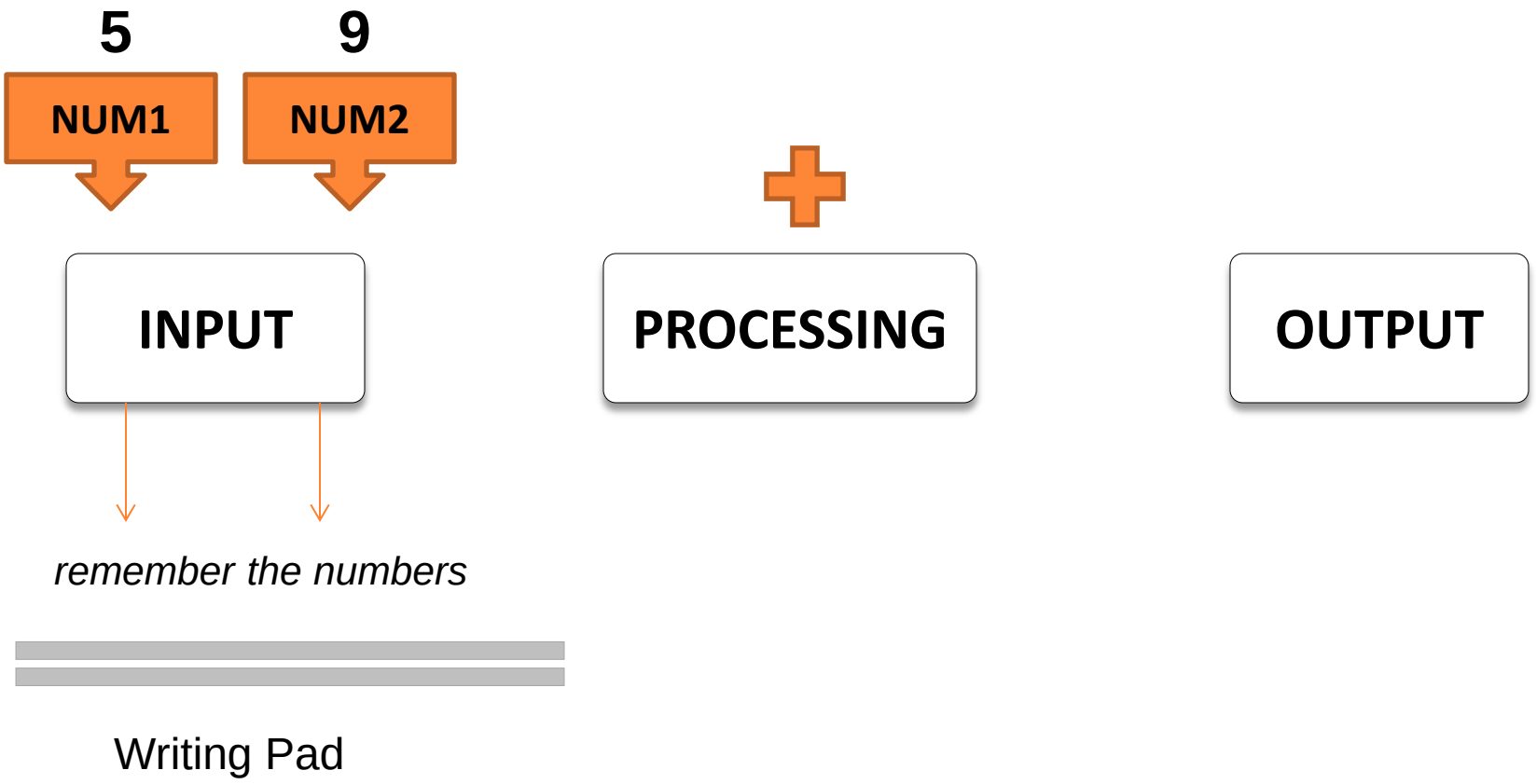
INPUT → **PROCESSING** → **OUTPUT**

Writing Pad

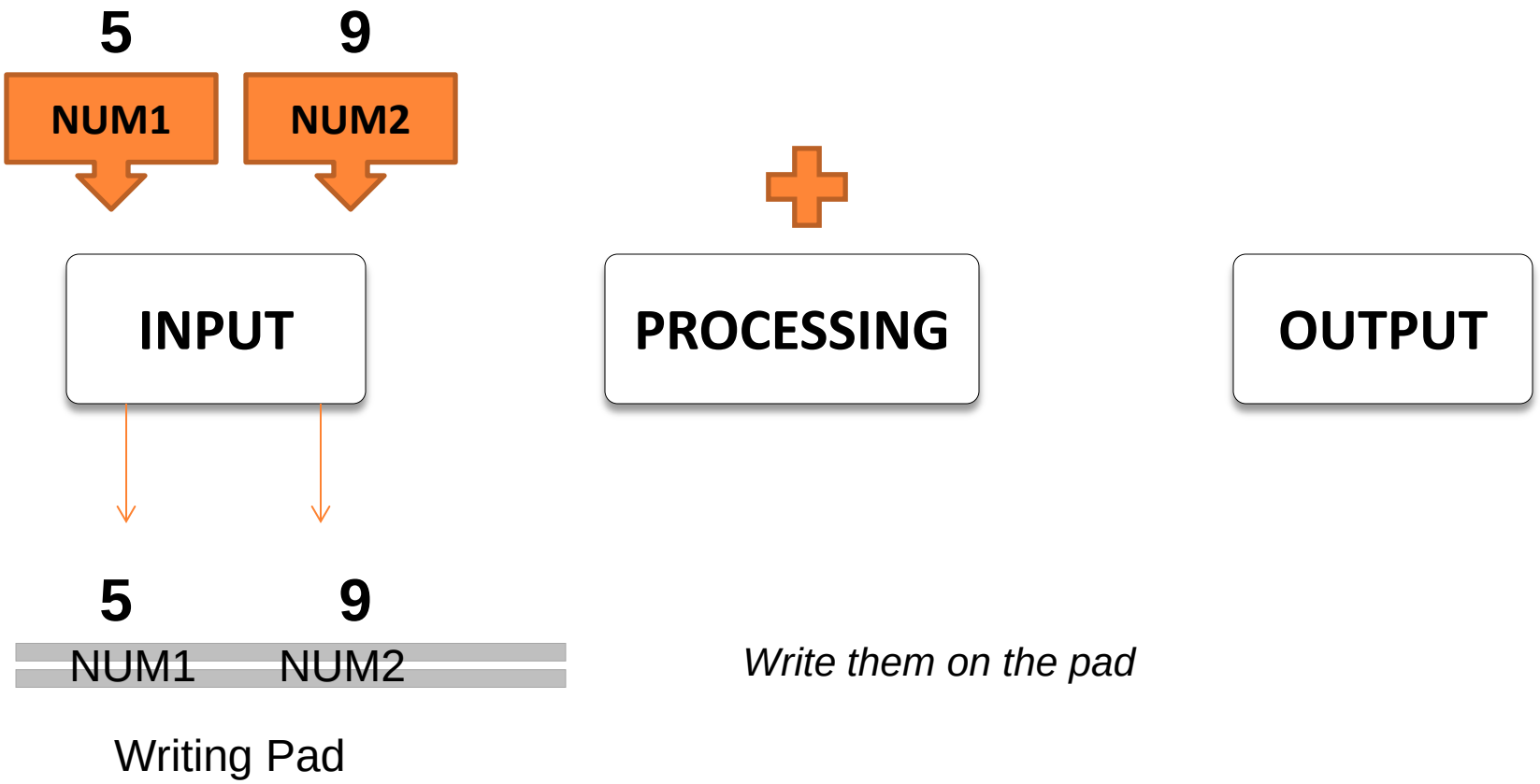
A computer needs to remember the numbers so it can work with them

So like you could have used your notepad to write down the large numbers I asked you to multiply together

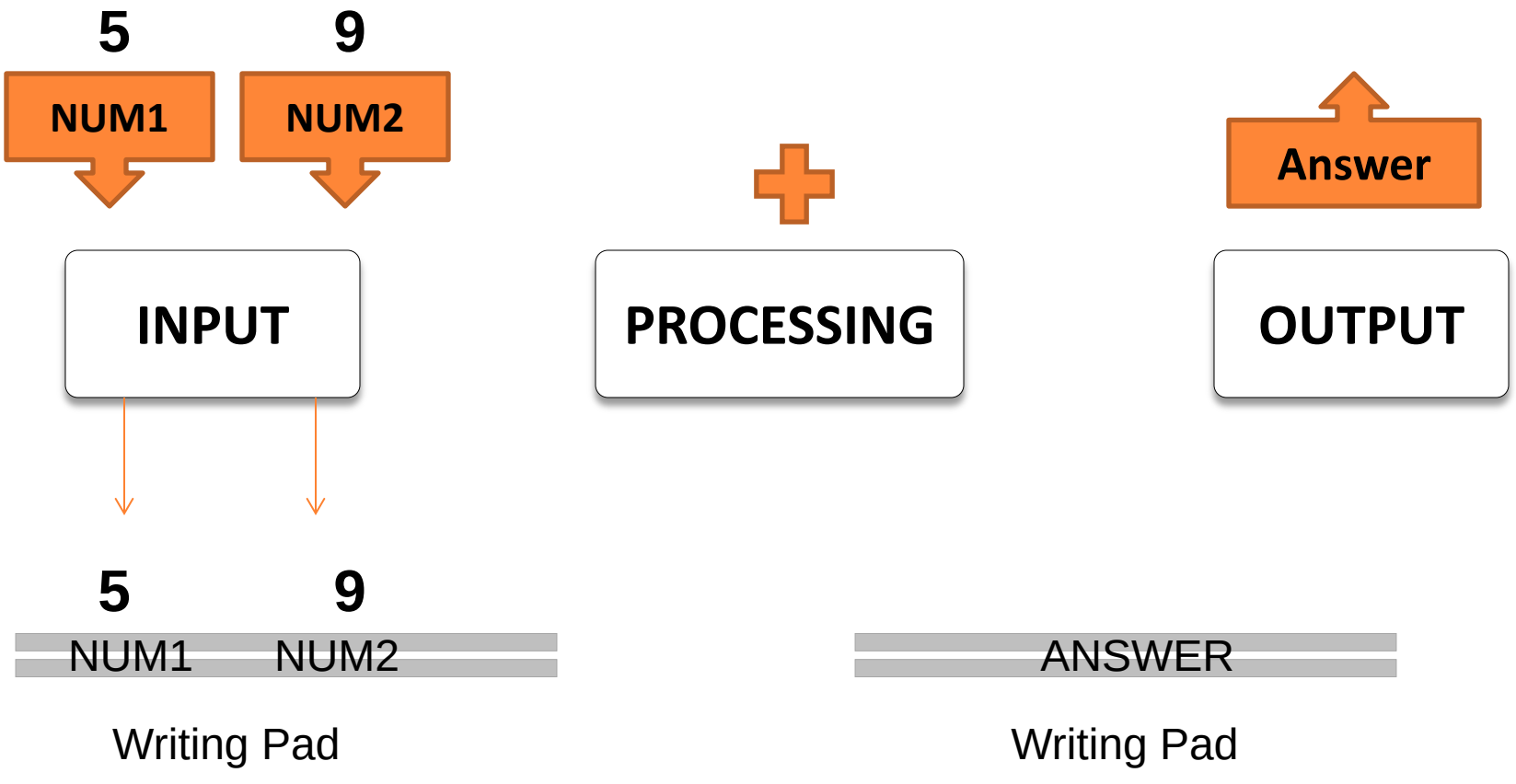
Need to get the numbers we're adding



Need to remember the numbers we're adding

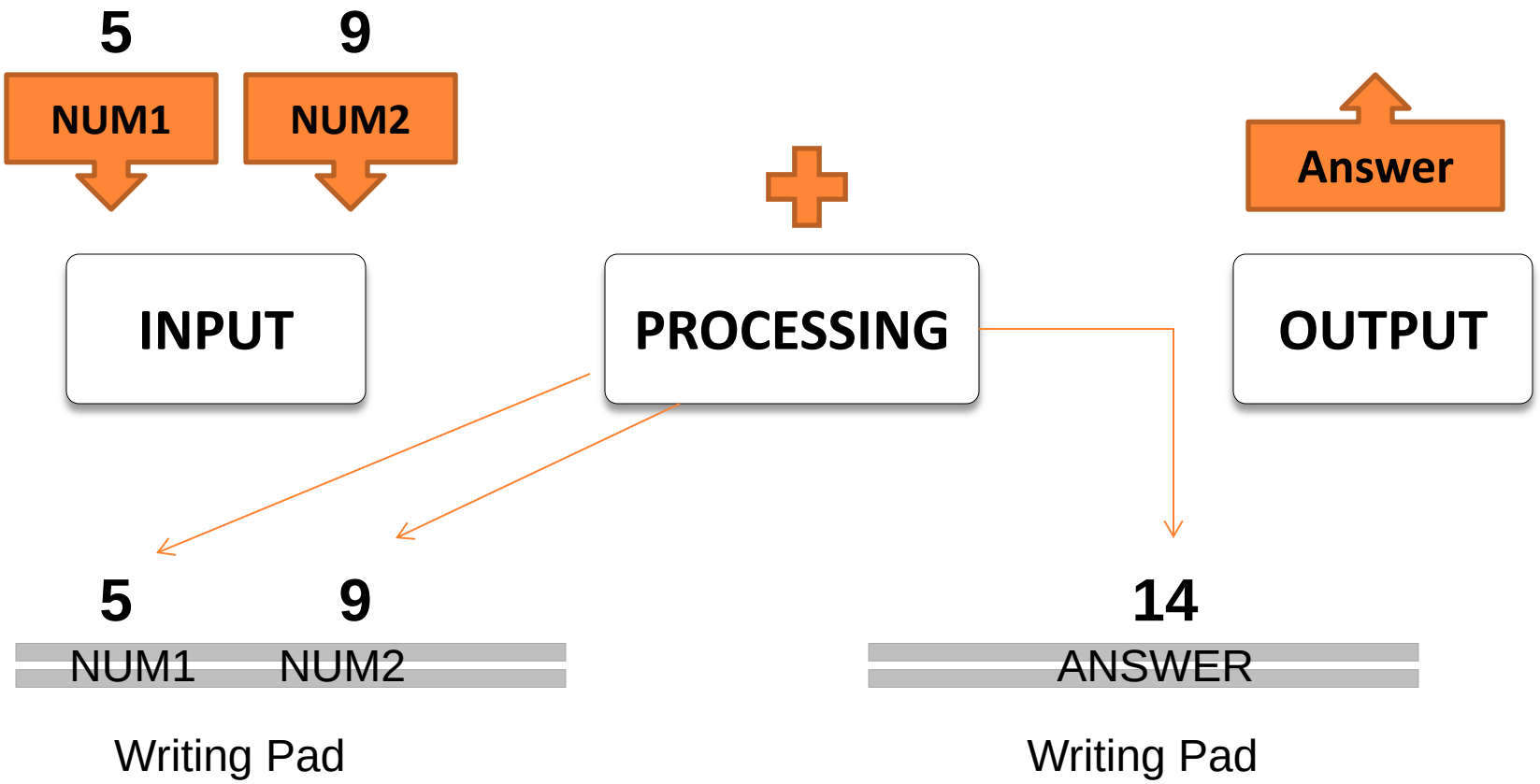


We'll need to remember the answer too



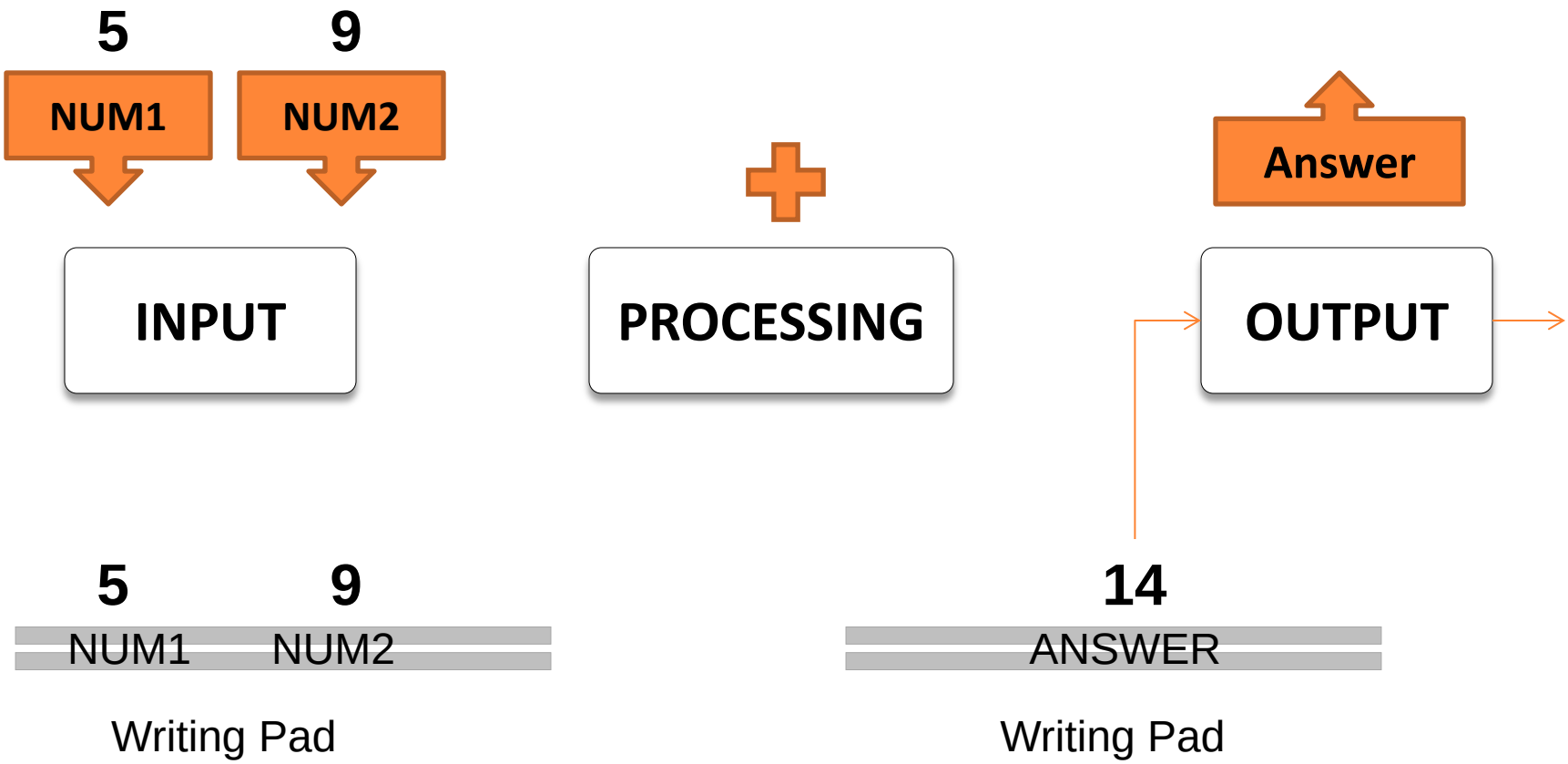
We'll need to remember the answer so we can give it back

Next step, have numbers, do processing



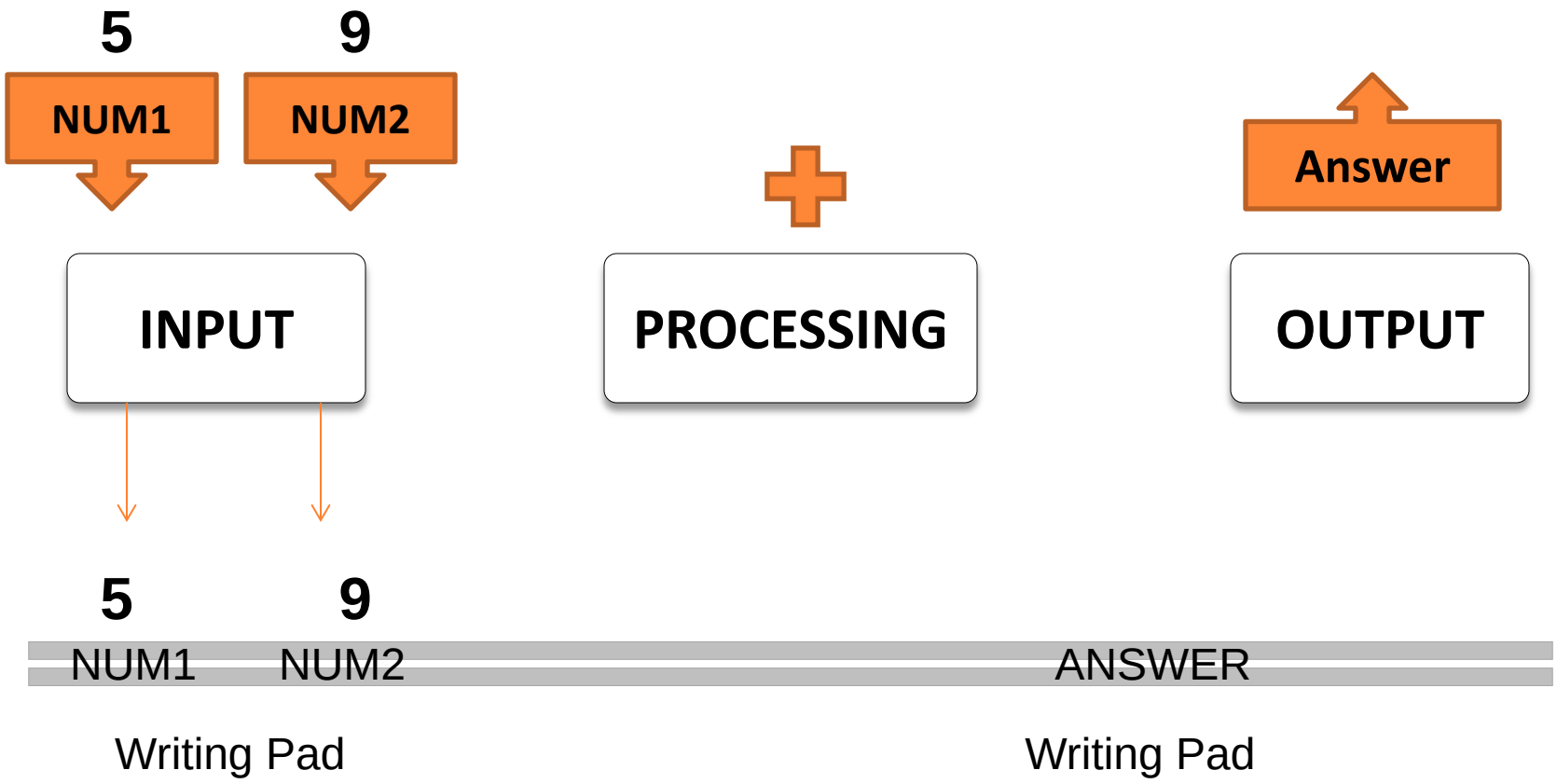
Step 1: Get numbers
Step 2: Add them
Step 3: Write the answer on the writing pad

What is the answer?



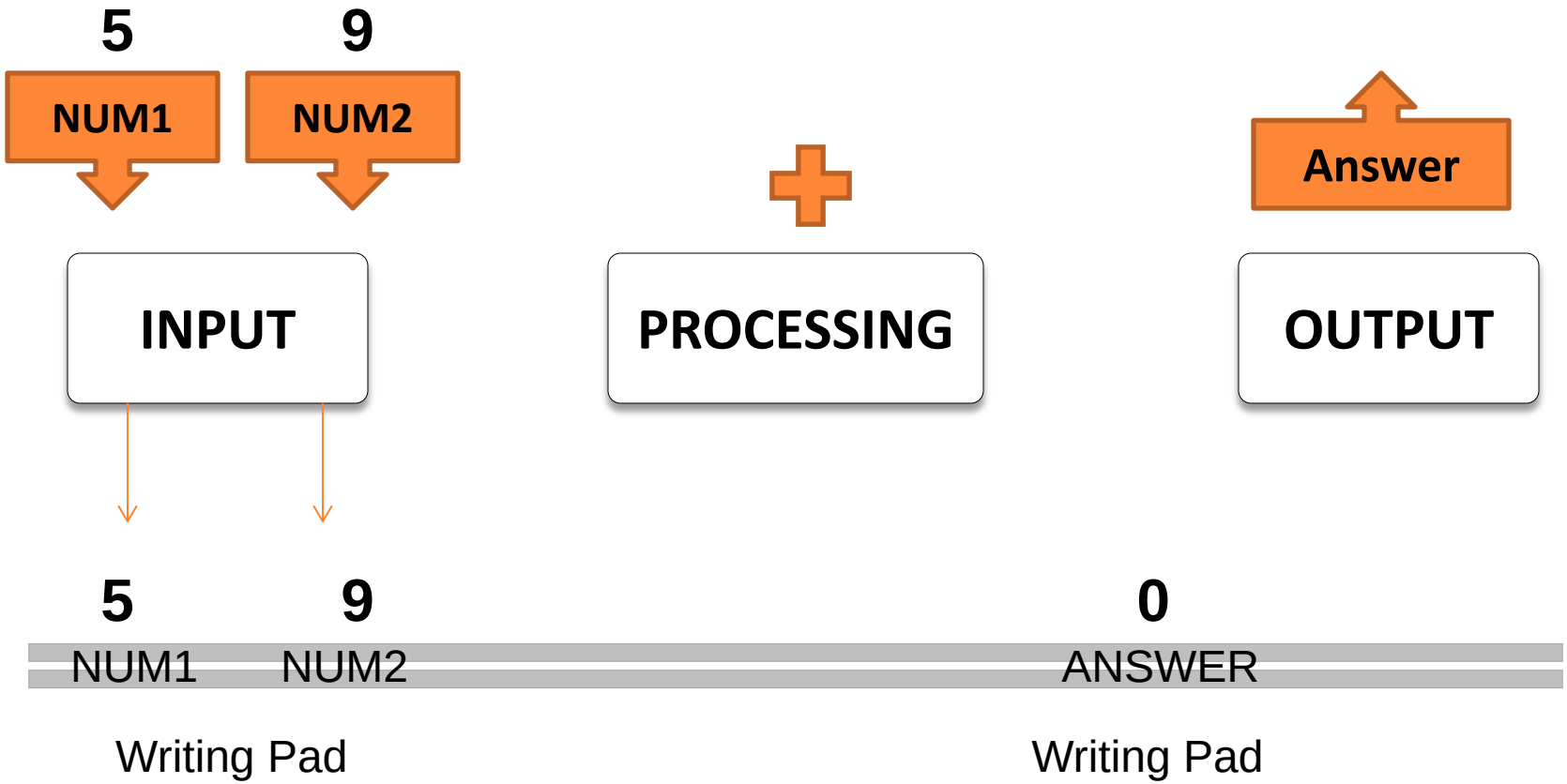
Step 1: Get the answer
Step 2: Display it, Say it .....

There is just one writing pad

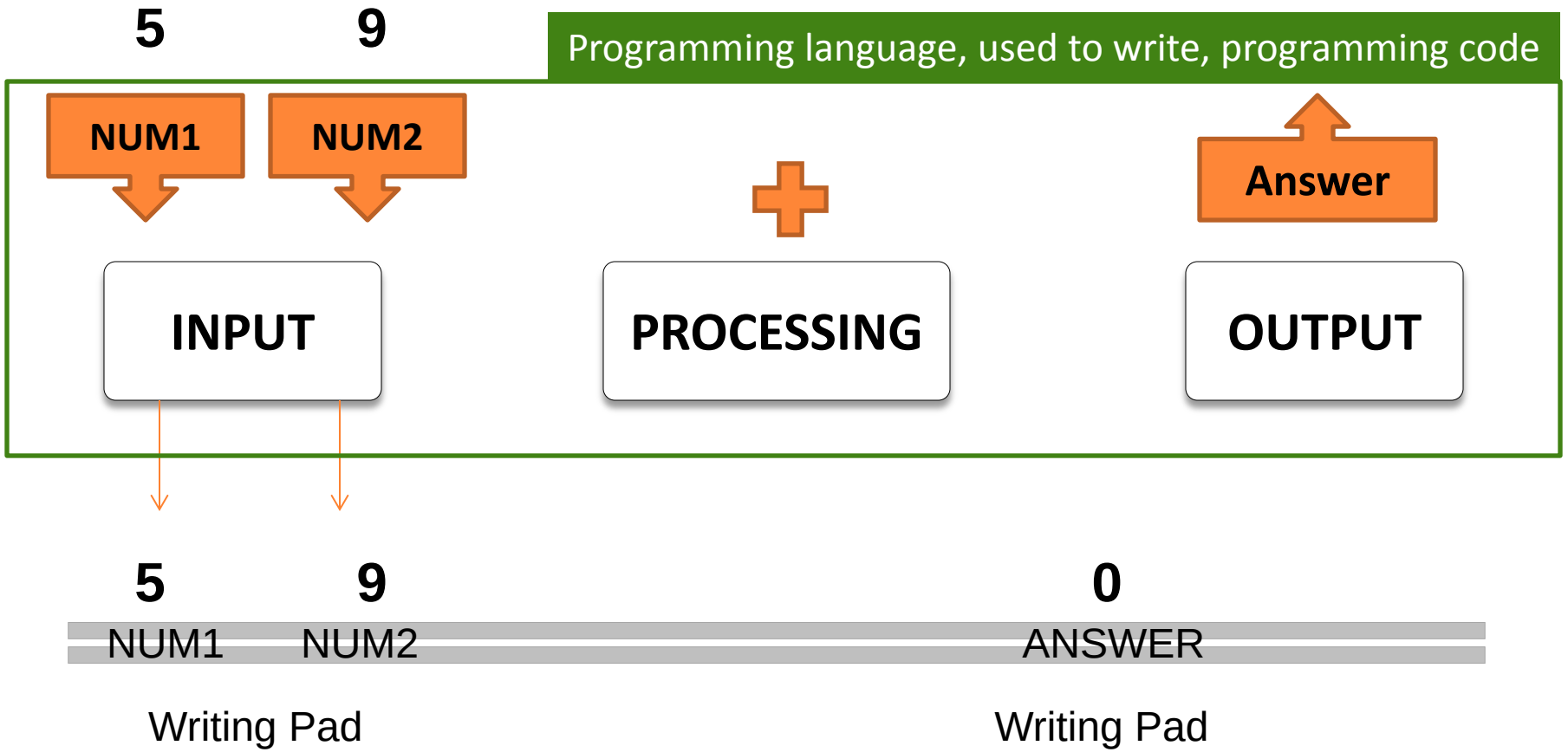


There isn't two writing pads just one

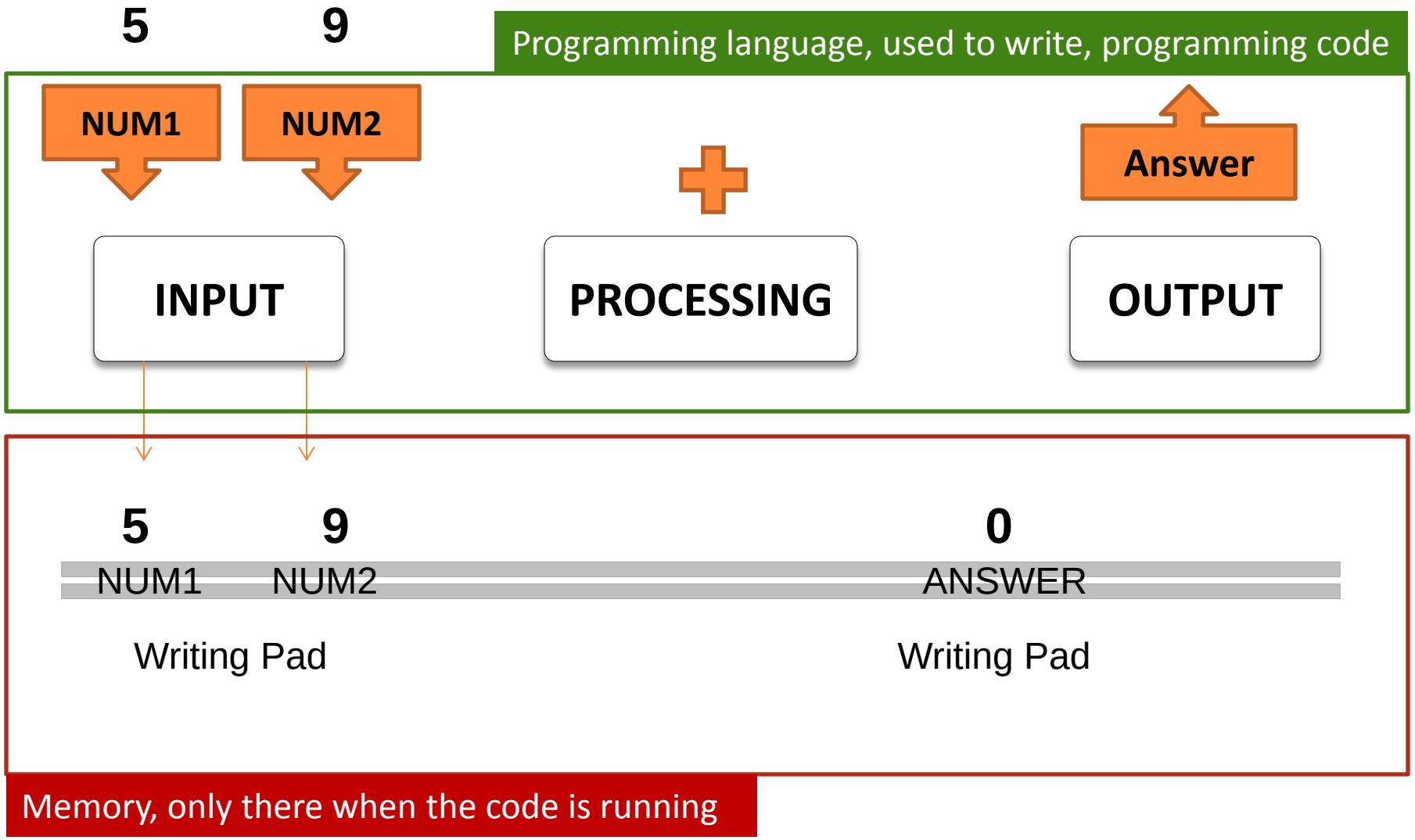
One pad so lets make answer ZERO before we start processing



This is exactly how a computer program works – the code

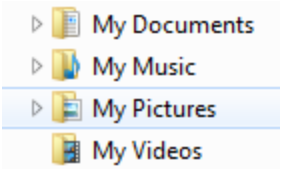


This is exactly how a computer program works – memory (your writing pad)



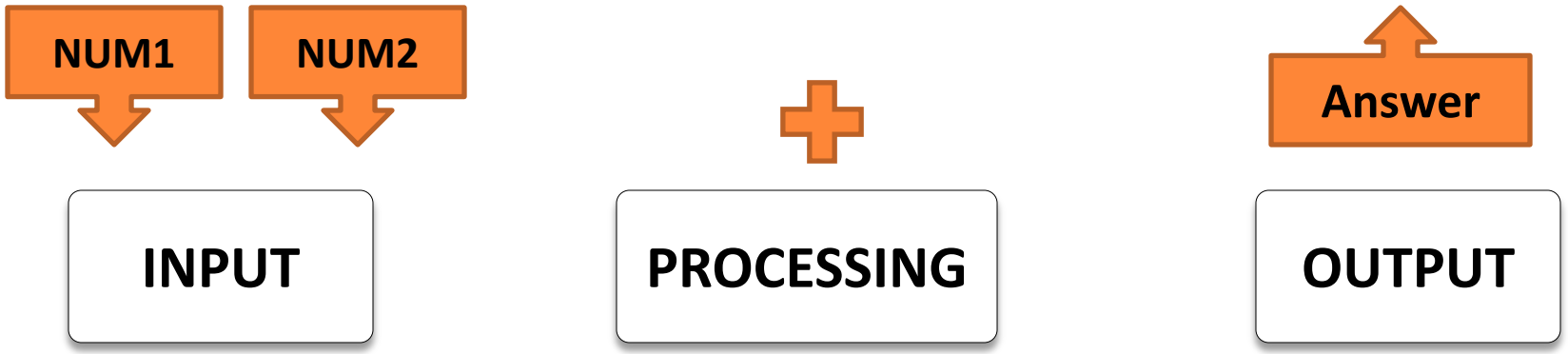
This is your program as stored

If a computer has power then ...



In the file system it has our code

code

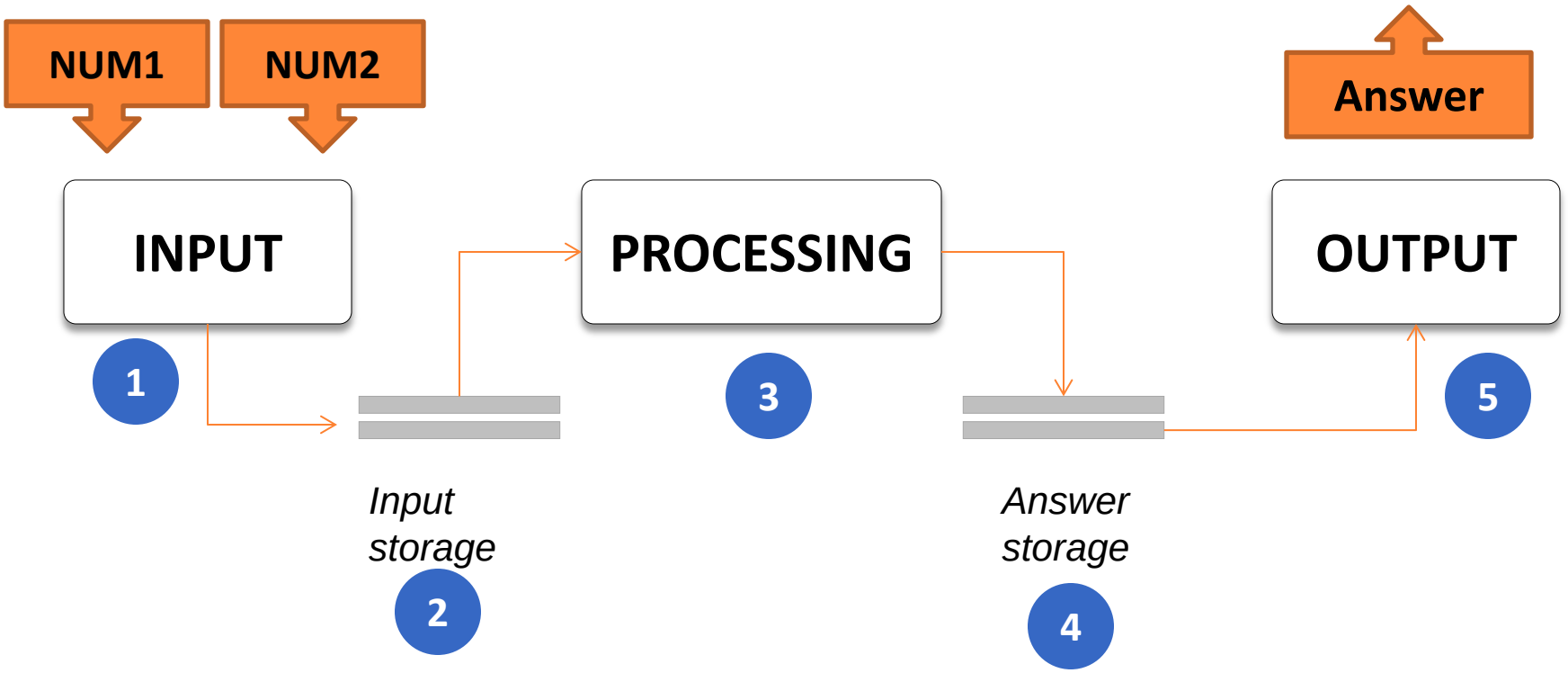


And it has memory ready to be used by any program that runs

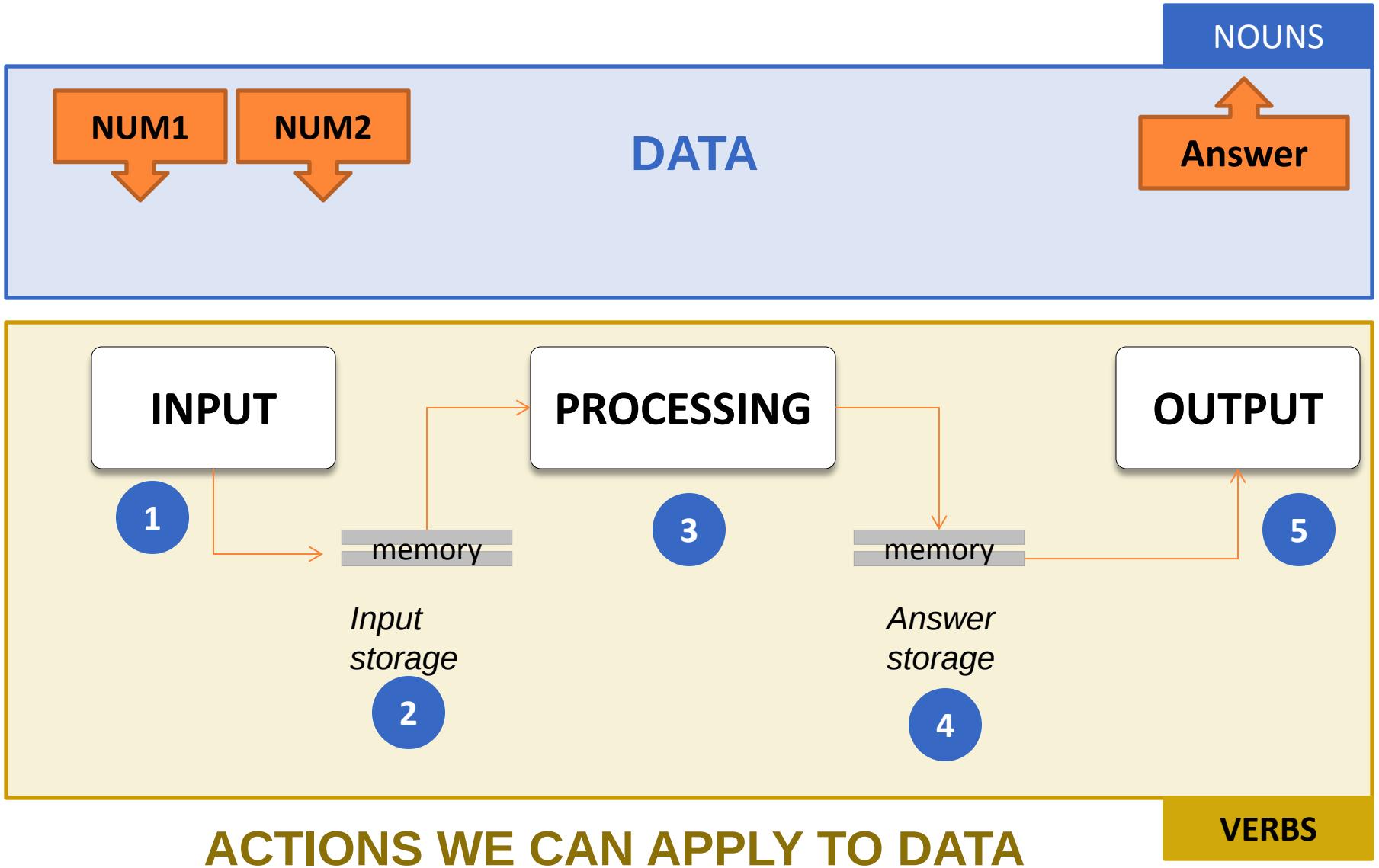
Memory

Writing Pad

The most basic program design is



In terms of a generic programming language



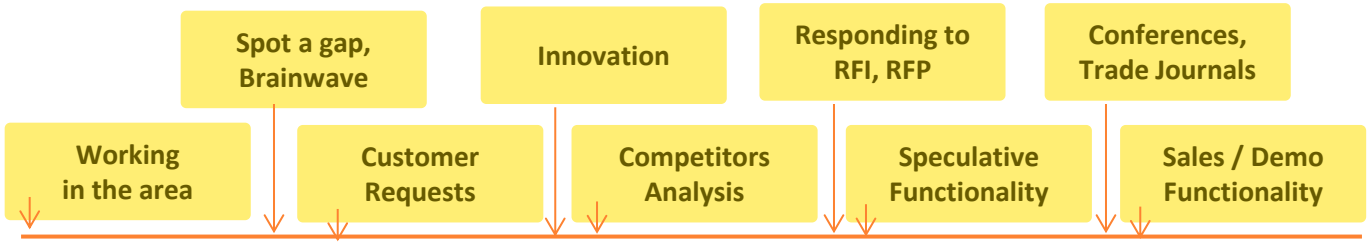
ACTIONS WE CAN APPLY TO DATA

Why do we write programs?

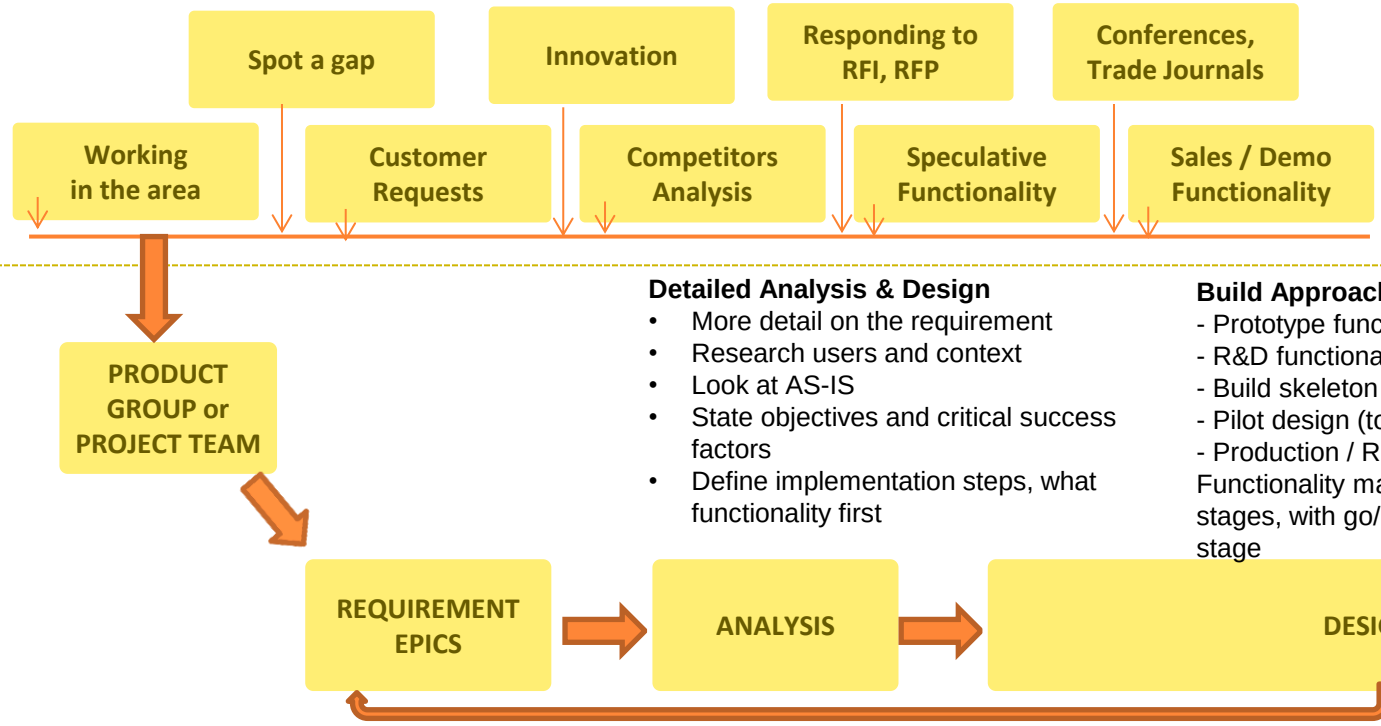
http://upload.wikimedia.org/wikipedia/commons/3/39/Aerial_view_of_a_US_helicopter_as_it_flies_over_a_Mogadishu_residential_area.JPEG

The Development Life Cycle

- The development life cycle for digital media is **long**
- Multiple streams of work are occurring in **parallel**
- **Multiple resources and roles** are involved
- New issues and requirements are **evolving** as the product or service is being built
- The target audience, the market context and the business need may be **changing** because
  - A competitor has changed the market or audience landscape
  - Economic conditions have changed
  - The pilot was a disaster!



Where do ideas come from



Detailed Analysis & Design

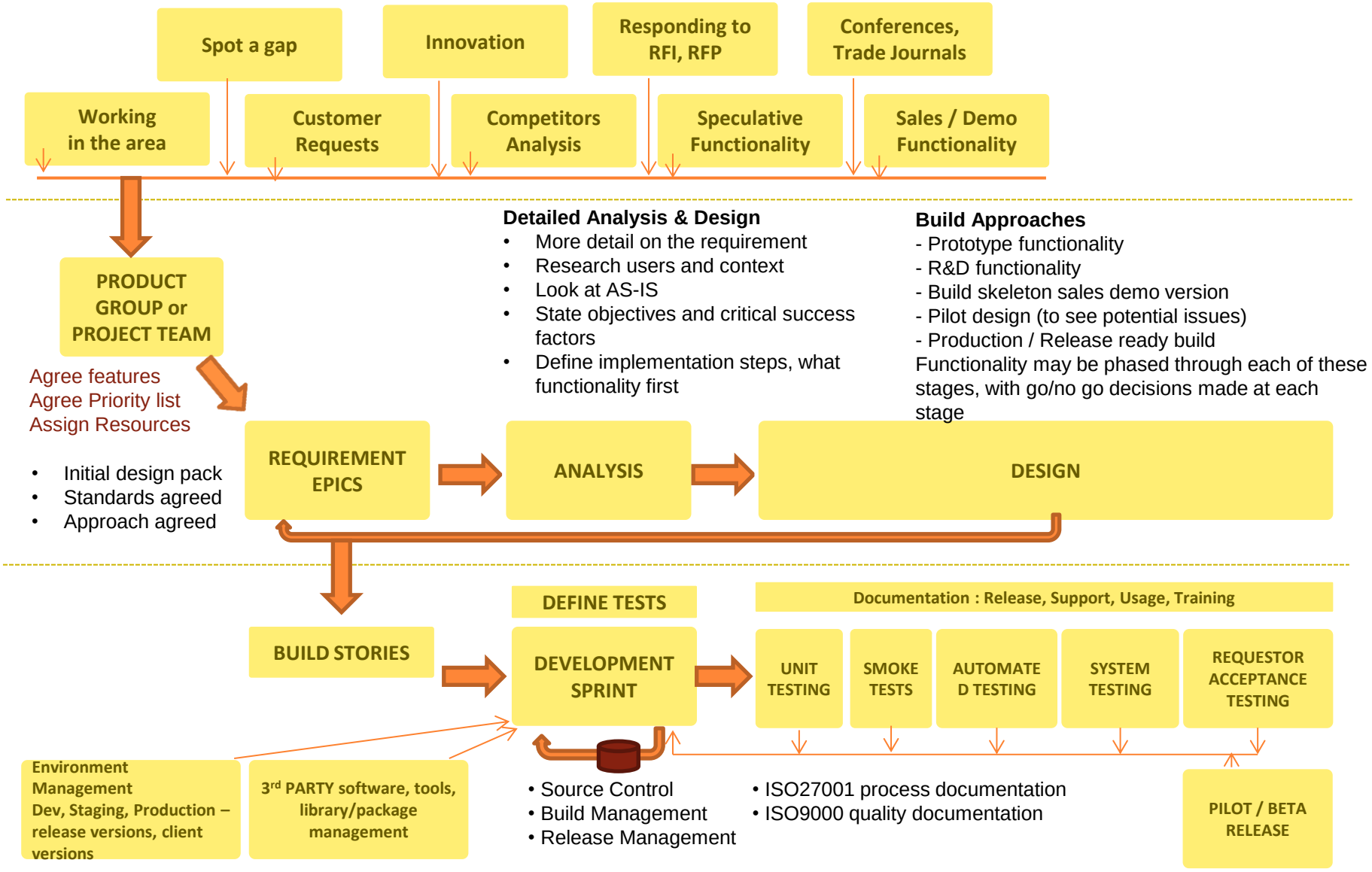
- More detail on the requirement
- Research users and context
- Look at AS-IS
- State objectives and critical success factors
- Define implementation steps, what functionality first

Build Approaches

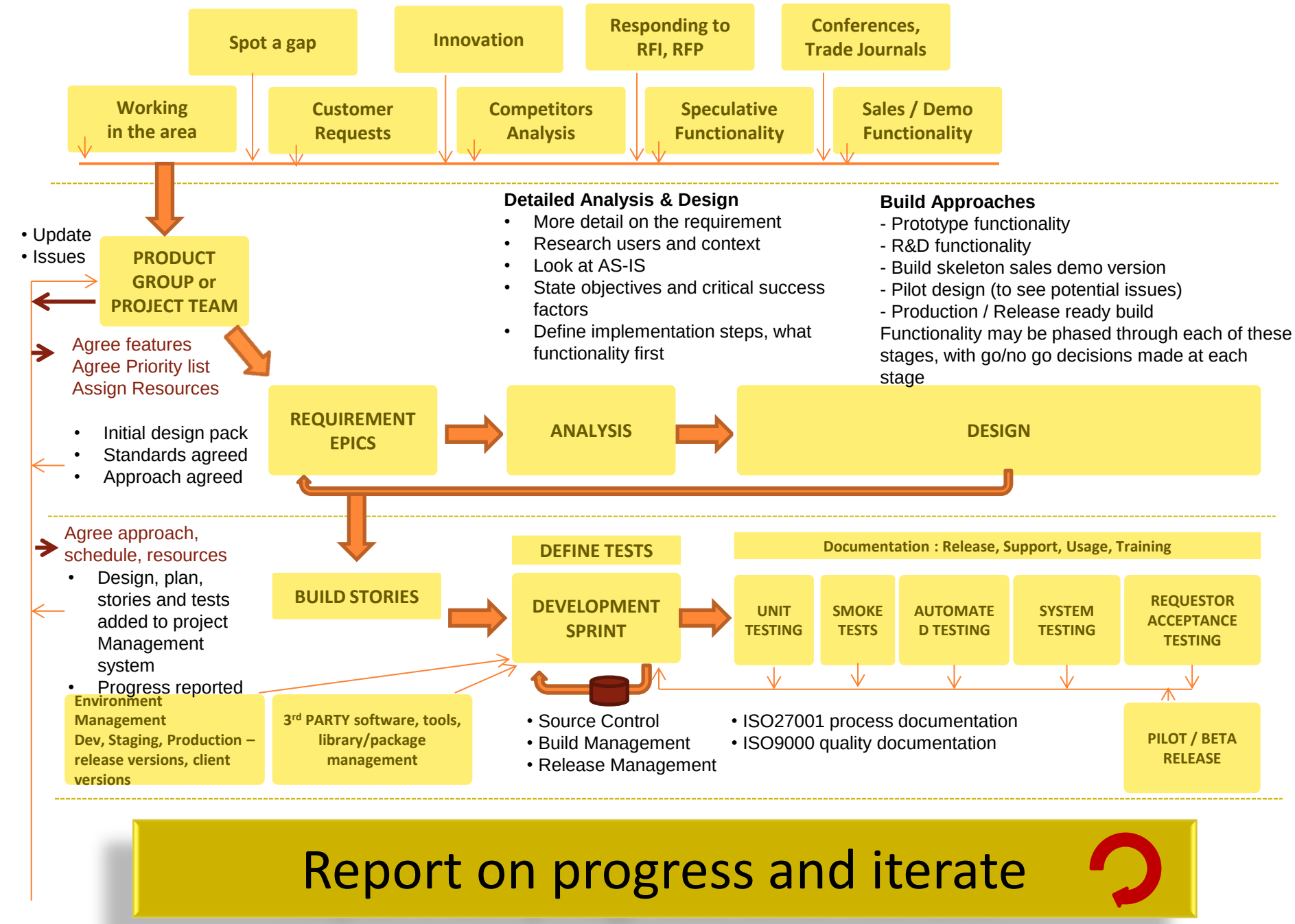
- Prototype functionality
- R&D functionality
- Build skeleton sales demo version
- Pilot design (to see potential issues)
- Production / Release ready build

Functionality may be phased through each of these stages, with go/no go decisions made at each stage

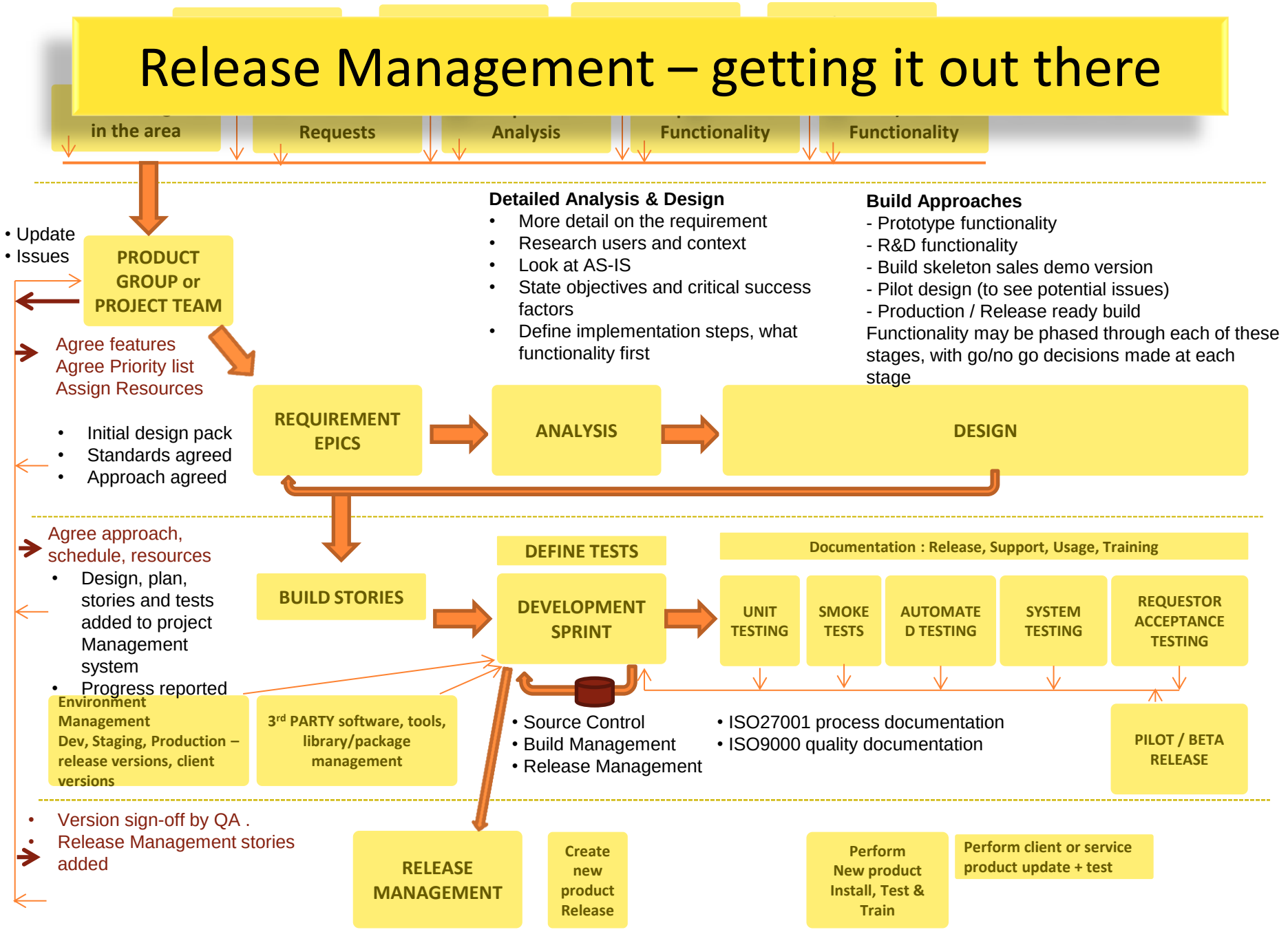
Moving from concept to plan and proto-type

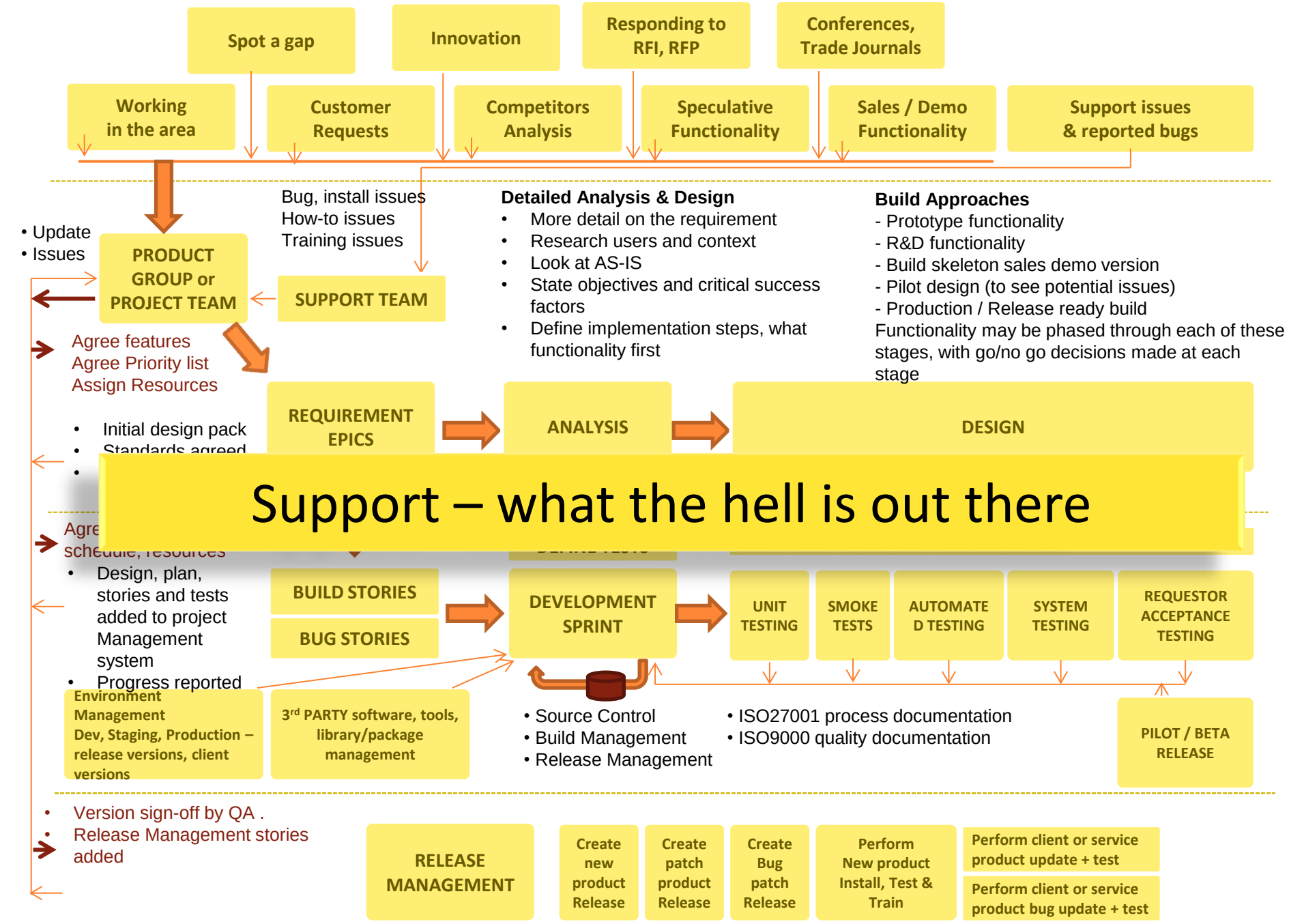


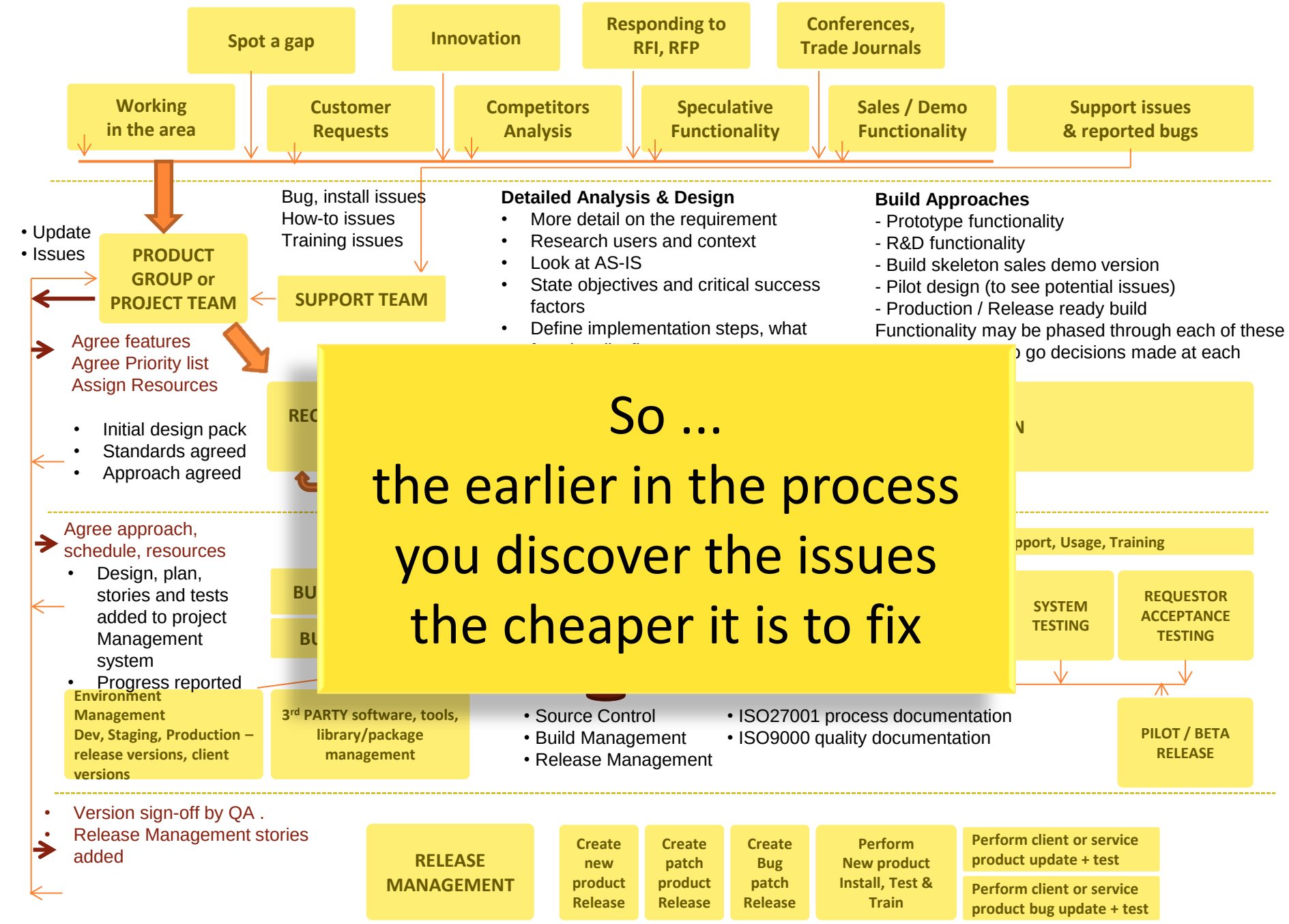
Moving from design to build


Spot a gap
Innovation
Responding to RFI, RFP
Conferences, Trade Journals
Working in the area
Customer Requests
Competitors Analysis
Speculative Functionality
Sales / Demo Functionality
• Update
• Issues
PRODUCT GROUP or PROJECT TEAM
Agree features
Agree Priority list
Assign Resources
• Initial design pack
• Standards agreed
• Approach agreed
Detailed Analysis & Design
• More detail on the requirement
• Research users and context
• Look at AS-IS
• State objectives and critical success factors
• Define implementation steps, what functionality first
Build Approaches
- Prototype functionality
- R&D functionality
- Build skeleton sales demo version
- Pilot design (to see potential issues)
- Production / Release ready build
Functionality may be phased through each of these stages, with go/no go decisions made at each stage
REQUIREMENT EPICS
ANALYSIS
DESIGN
Agree approach, schedule, resources
• Design, plan, stories and tests added to project Management system
• Progress reported
BUILD STORIES
DEFINE TESTS
DEVELOPMENT SPRINT
Documentation : Release, Support, Usage, Training
UNIT TESTING
SMOKE TESTS
AUTOMATE D TESTING
SYSTEM TESTING
REQUESTOR ACCEPTANCE TESTING
Environment Management
Dev, Staging, Production – release versions, client versions
3rd PARTY software, tools, library/package management
• Source Control
• Build Management
• Release Management
• ISO27001 process documentation
• ISO9000 quality documentation
PILOT / BETA RELEASE
Report on progress and iterate

Release Management – getting it out there




Spot a gap
Innovation
Responding to RFI, RFP
Conferences, Trade Journals
Working in the area
Customer Requests
Competitors Analysis
Speculative Functionality
Sales / Demo Functionality
Support issues & reported bugs
Bug, install issues
How-to issues
Training issues
Detailed Analysis & Design
• More detail on the requirement
• Research users and context
• Look at AS-IS
• State objectives and critical success factors
• Define implementation steps, what functionality first
Build Approaches
- Prototype functionality
- R&D functionality
- Build skeleton sales demo version
- Pilot design (to see potential issues)
- Production / Release ready build
Functionality may be phased through each of these stages, with go/no go decisions made at each stage
• Update
• Issues
PRODUCT GROUP or PROJECT TEAM
SUPPORT TEAM
Agree features
Agree Priority list
Assign Resources
• Initial design pack
• Standards agreed
REQUIREMENT EPICS
ANALYSIS
DESIGN
Support – what the hell is out there
Agre
schedule, resources
• Design, plan, stories and tests added to project Management system
• Progress reported
BUILD STORIES
BUG STORIES
DEVELOPMENT SPRINT
UNIT TESTING
SMOKE TESTS
AUTOMATE D TESTING
SYSTEM TESTING
REQUESTOR ACCEPTANCE TESTING
Environment Management
Dev, Staging, Production – release versions, client versions
3rd PARTY software, tools, library/package management
• Source Control
• Build Management
• Release Management
• ISO27001 process documentation
• ISO9000 quality documentation
PILOT / BETA RELEASE
• Version sign-off by QA .
• Release Management stories added
RELEASE MANAGEMENT
Create new product Release
Create patch product Release
Create Bug patch Release
Perform New product Install, Test & Train
Perform client or service product update + test
Perform client or service product bug update + test

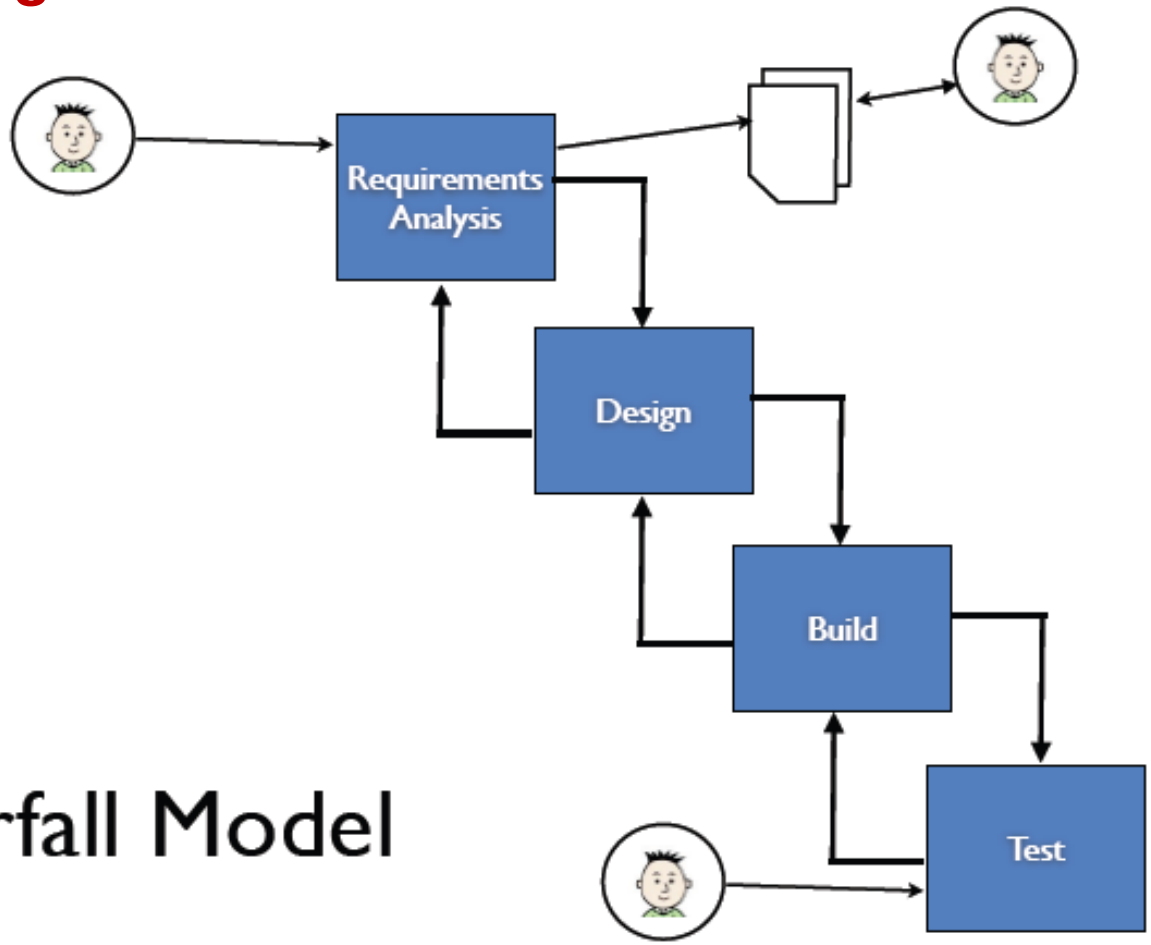

Spot a gap
Innovation
Responding to RFI, RFP
Conferences, Trade Journals
Working in the area
Customer Requests
Competitors Analysis
Speculative Functionality
Sales / Demo Functionality
Support issues & reported bugs
Bug, install issues
How-to issues
Training issues
Detailed Analysis & Design
• More detail on the requirement
• Research users and context
• Look at AS-IS
• State objectives and critical success factors
• Define implementation steps, what
Build Approaches
- Prototype functionality
- R&D functionality
- Build skeleton sales demo version
- Pilot design (to see potential issues)
- Production / Release ready build
Functionality may be phased through each of these
go decisions made at each
• Update
• Issues
PRODUCT GROUP or PROJECT TEAM
SUPPORT TEAM
Agree features
Agree Priority list
Assign Resources
• Initial design pack
• Standards agreed
• Approach agreed
So ...
the earlier in the process
you discover the issues
the cheaper it is to fix
Agree approach, schedule, resources
• Design, plan, stories and tests added to project Management system
• Progress reported
pport, Usage, Training
SYSTEM TESTING
REQUESTOR ACCEPTANCE TESTING
PILOT / BETA RELEASE
Environment Management
Dev, Staging, Production – release versions, client versions
3rd PARTY software, tools, library/package management
• Source Control
• Build Management
• Release Management
• ISO27001 process documentation
• ISO9000 quality documentation
• Version sign-off by QA .
• Release Management stories added
RELEASE MANAGEMENT
Create new product Release
Create patch product Release
Create Bug patch Release
Perform New product Install, Test & Train
Perform client or service product update + test
Perform client or service product bug update + test

Getting the user involved

- The **issues** typically begin at the source
  - The User
- Over the years there have been many different **approaches** aimed at getting the process right
- With **varying degrees of user contact** through out the life cycle

Waterfall Approach

User involved only
at the beginning and the end

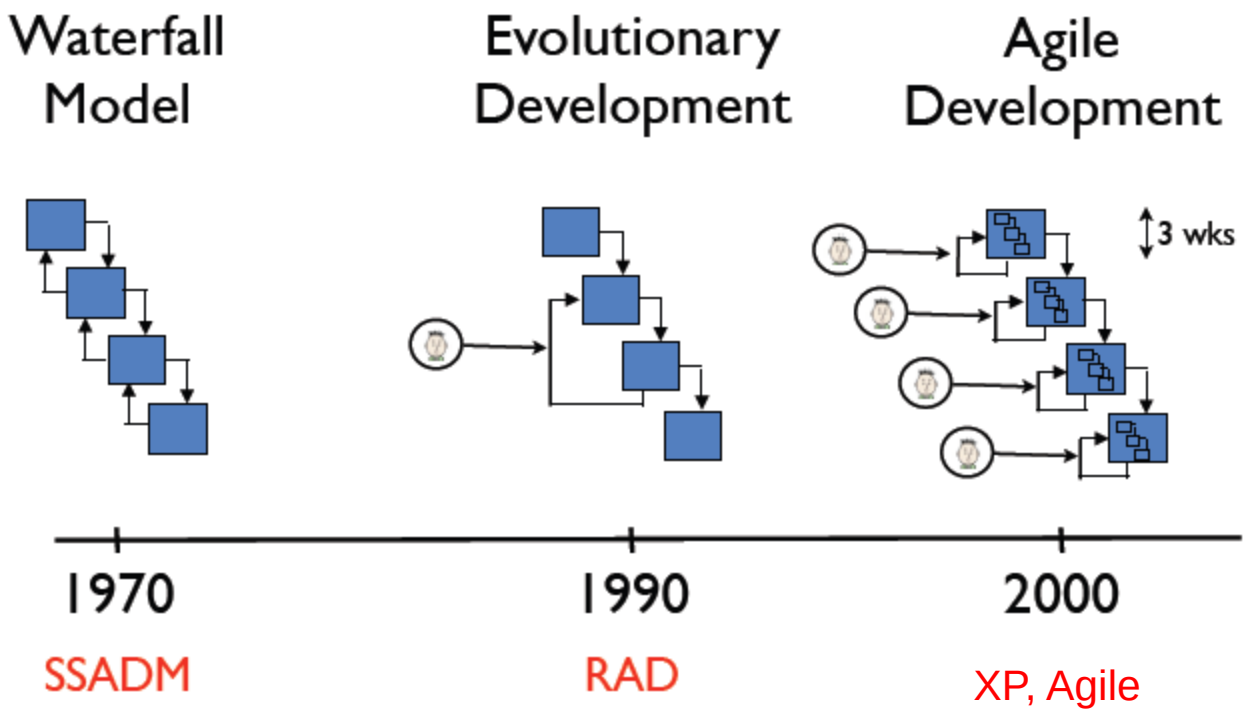


[1 lecture slides, DIT, Sarah Jane Delany
[2] http://www.waterfall-model.com/the-waterfall-model-avantages-and-disadvantages/

Other Approached

Software Engineering Life Cycle models

The current trend Is the Agile approach



[1 lecture slides, DIT, Sarah Jane Delany

Approach Vs. Method

Irrespective of the approach deployed

- The techniques and tools used to determine and document requirements / design and testing remain the same
- It is only the **level of detail and documentation** changes at each stage or iteration that changes
- The **user is more involved in an agile approach** as new functionality is typically delivered every 3 weeks with the user embedded in the development team.

User Experience

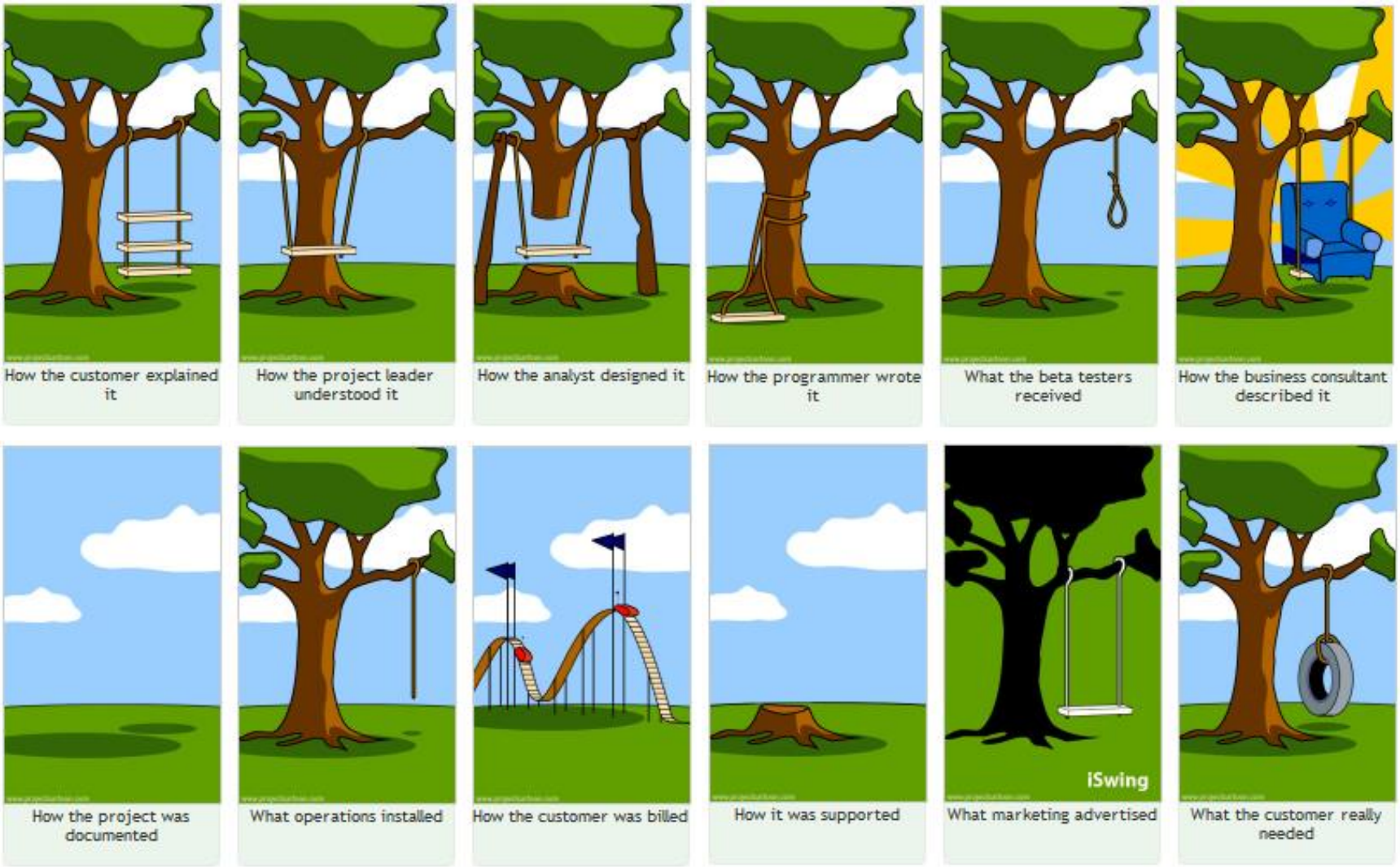
Local play

Web Play

http://www.youtube.com/watch?feature=player_embedded&v=3Sk7cOqB9Dk

THE PARABLE OF THE SWING

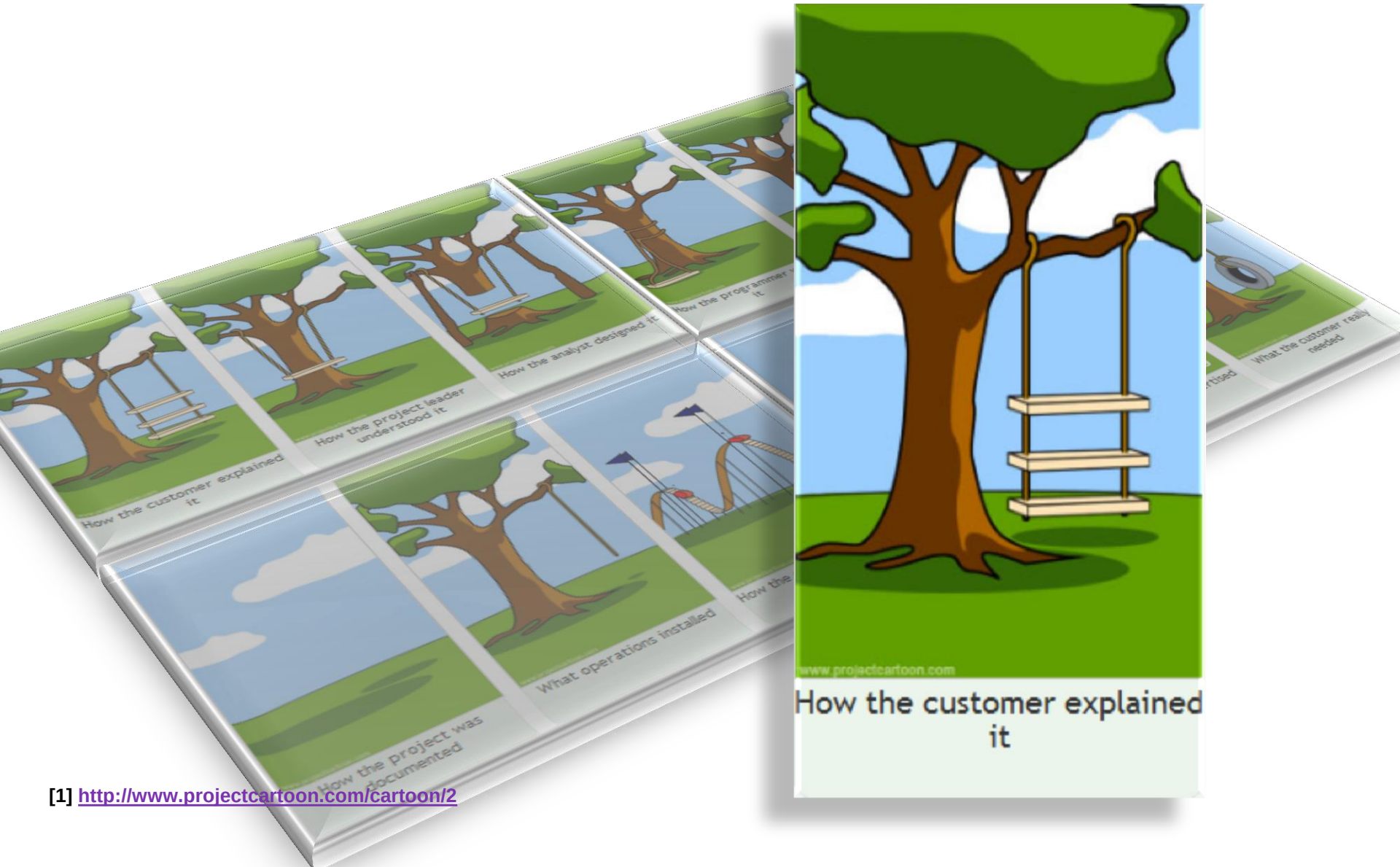
An approach you see all too often ...



The parable of the swing

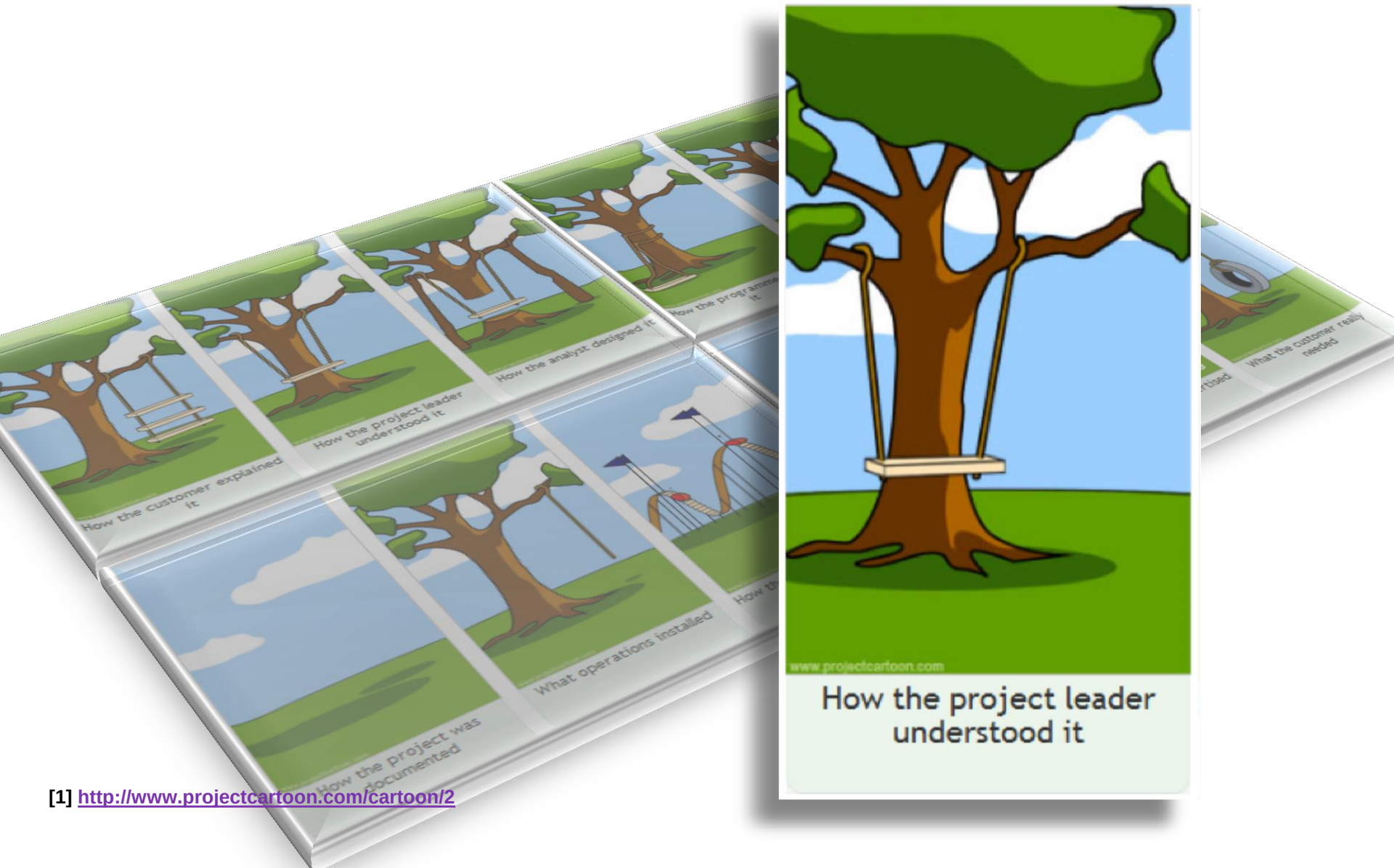
[1] http://www.projectcartoon.com/cartoon/2

How the customer explained it



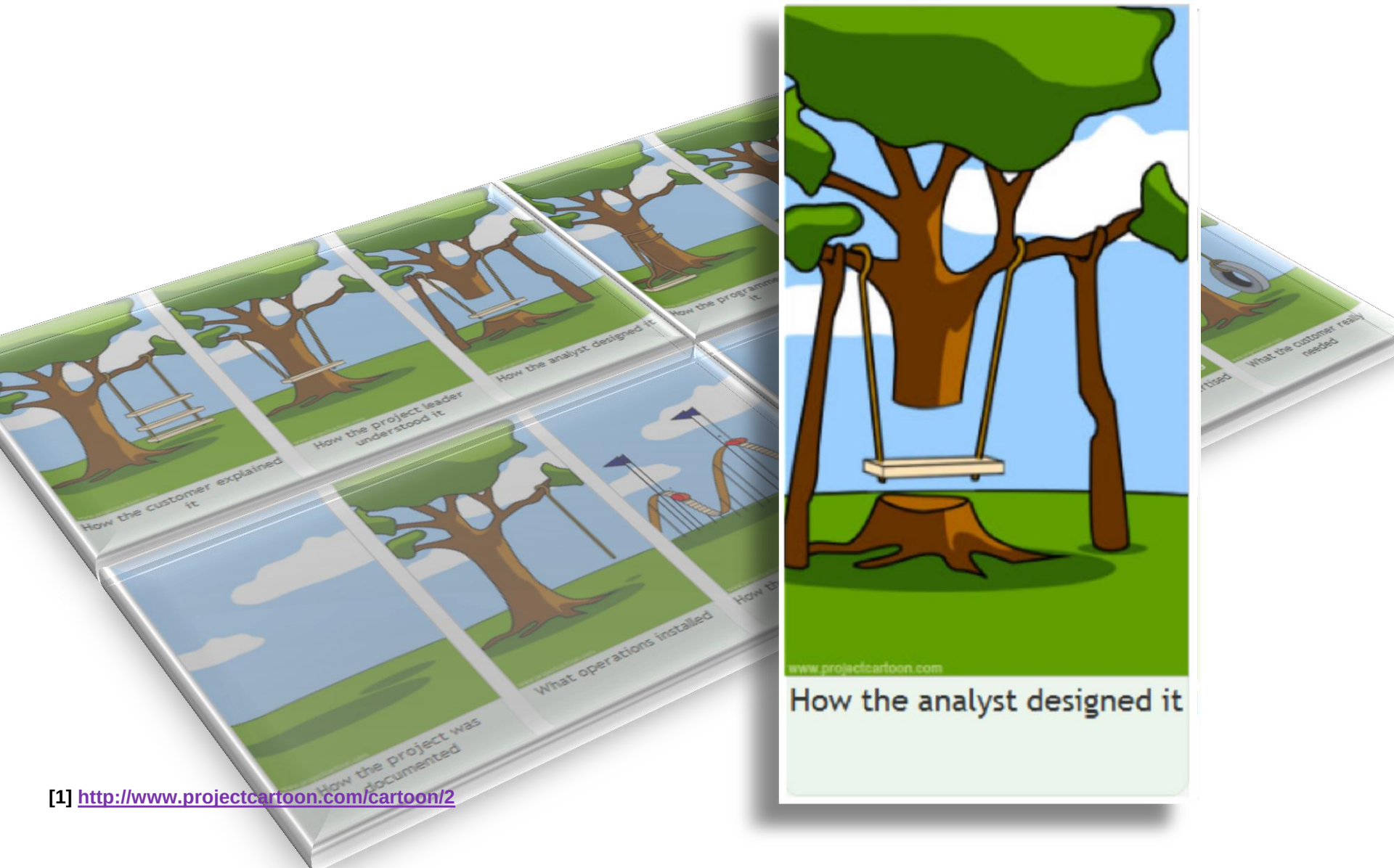
[1] http://www.projectcartoon.com/cartoon/2

How the project manager understood it



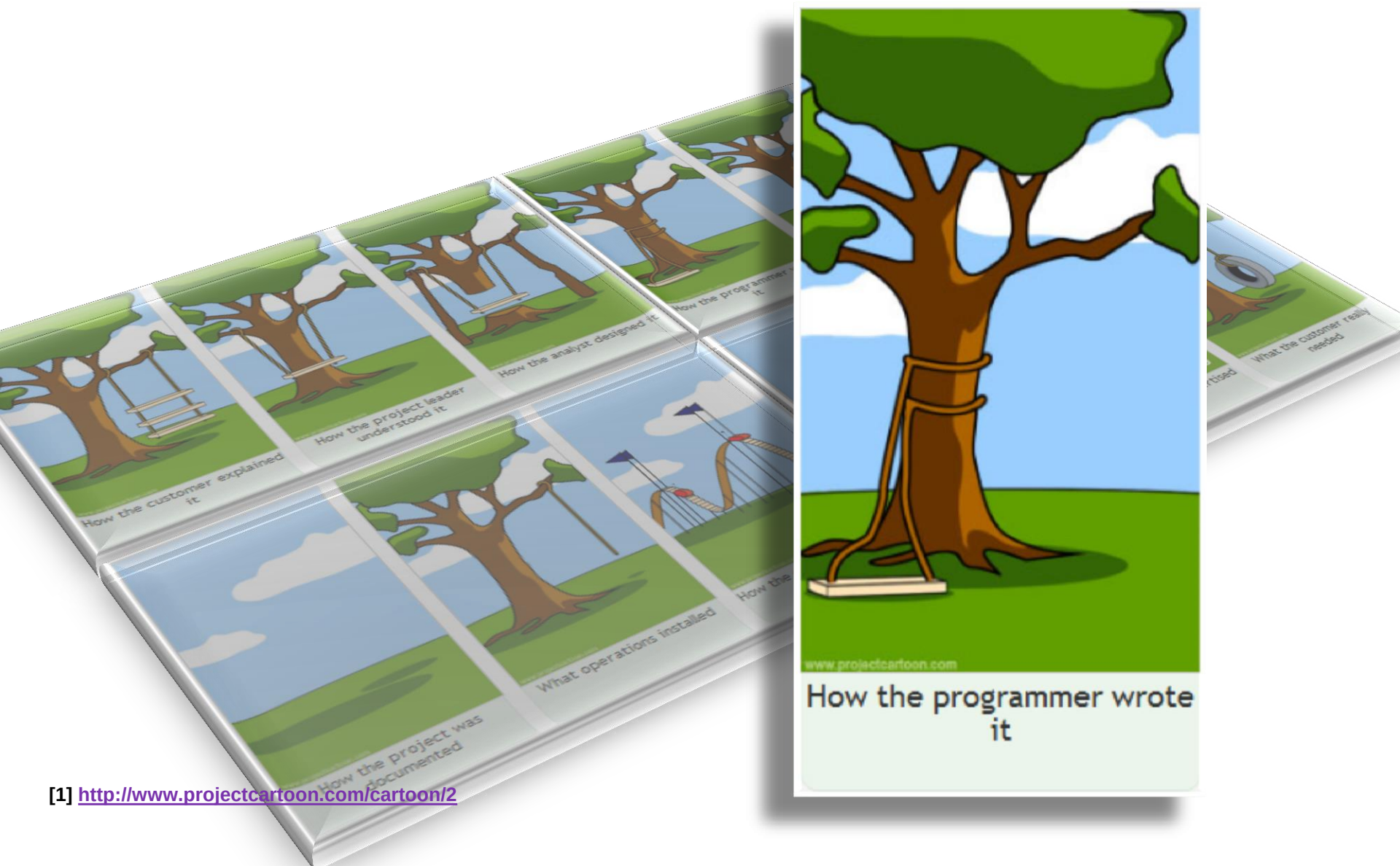
[1] http://www.projectcartoon.com/cartoon/2

How the analyst designed it



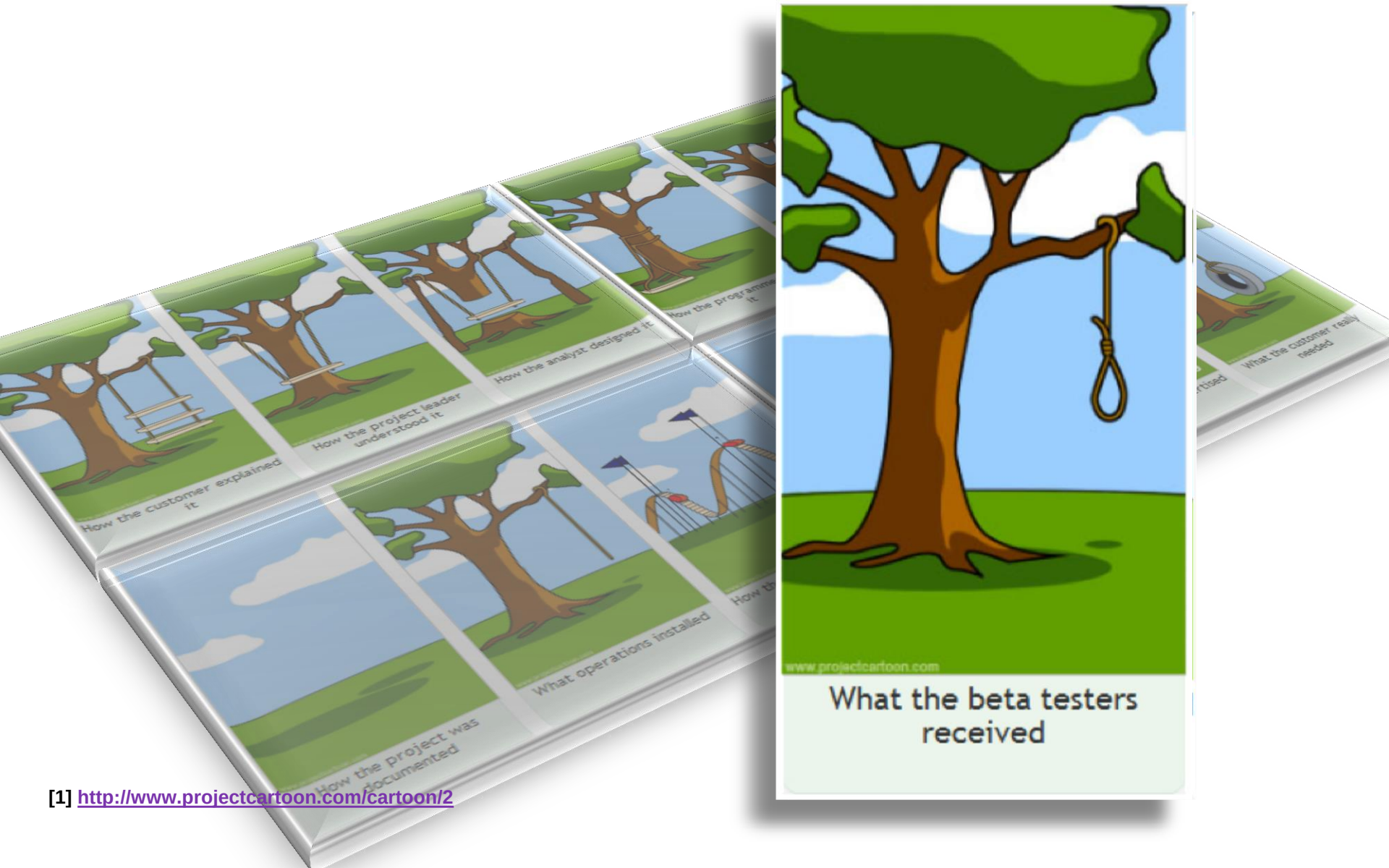
[1] http://www.projectcartoon.com/cartoon/2

How the programmer wrote it



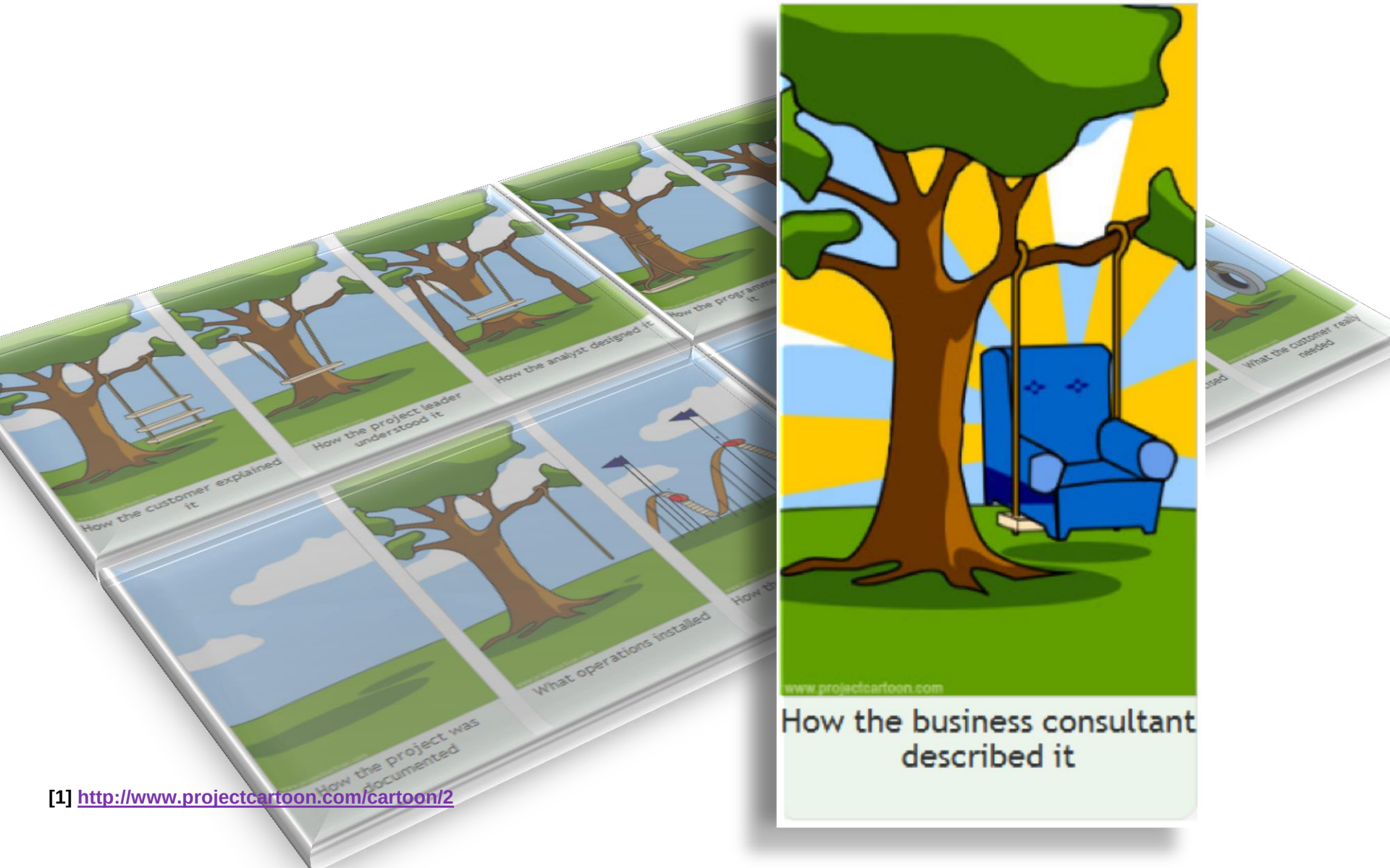
[1] http://www.projectcartoon.com/cartoon/2

What the tester received it



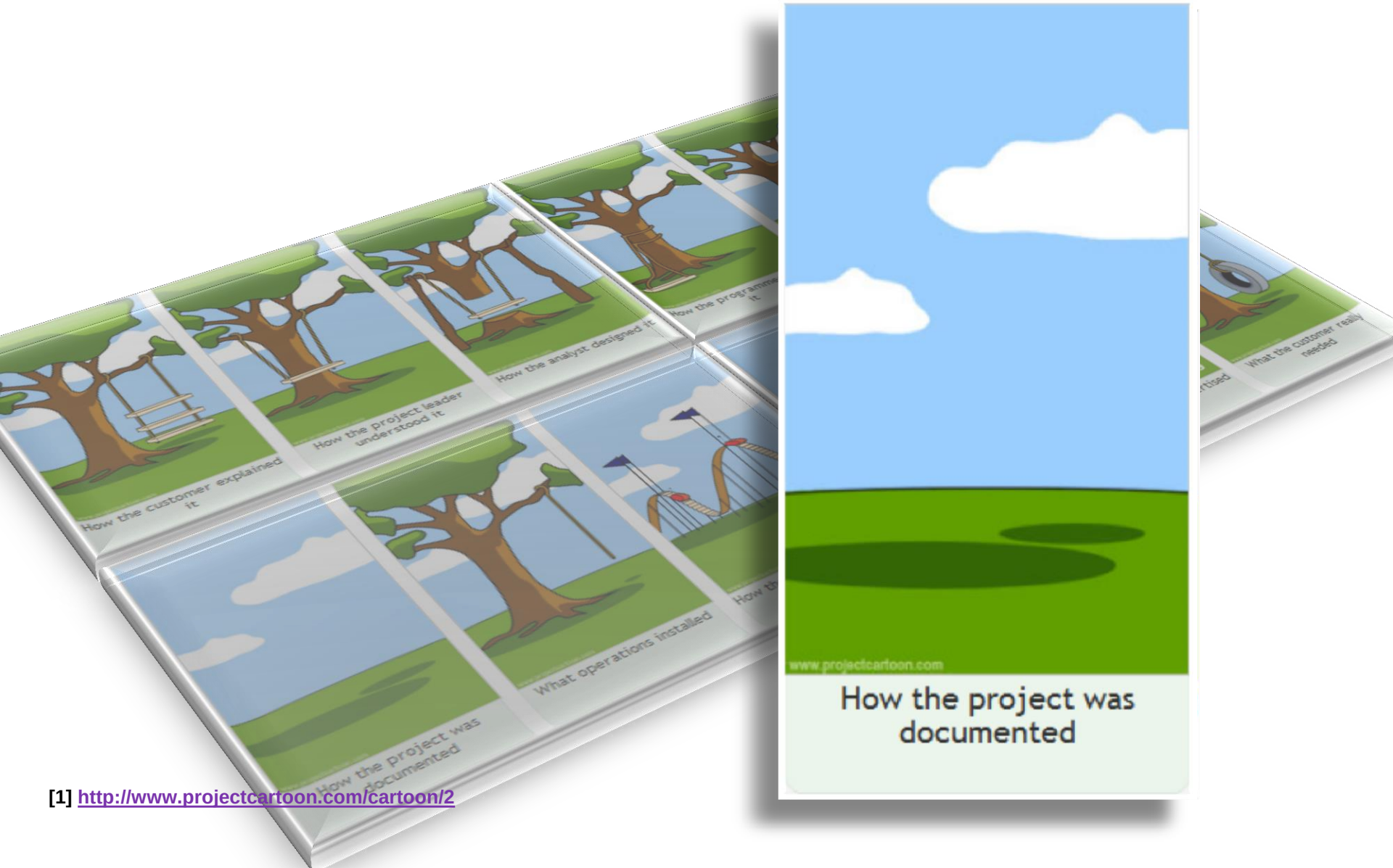
[1] http://www.projectcartoon.com/cartoon/2

How the consultant described it



[1] http://www.projectcartoon.com/cartoon/2

How the project was documented

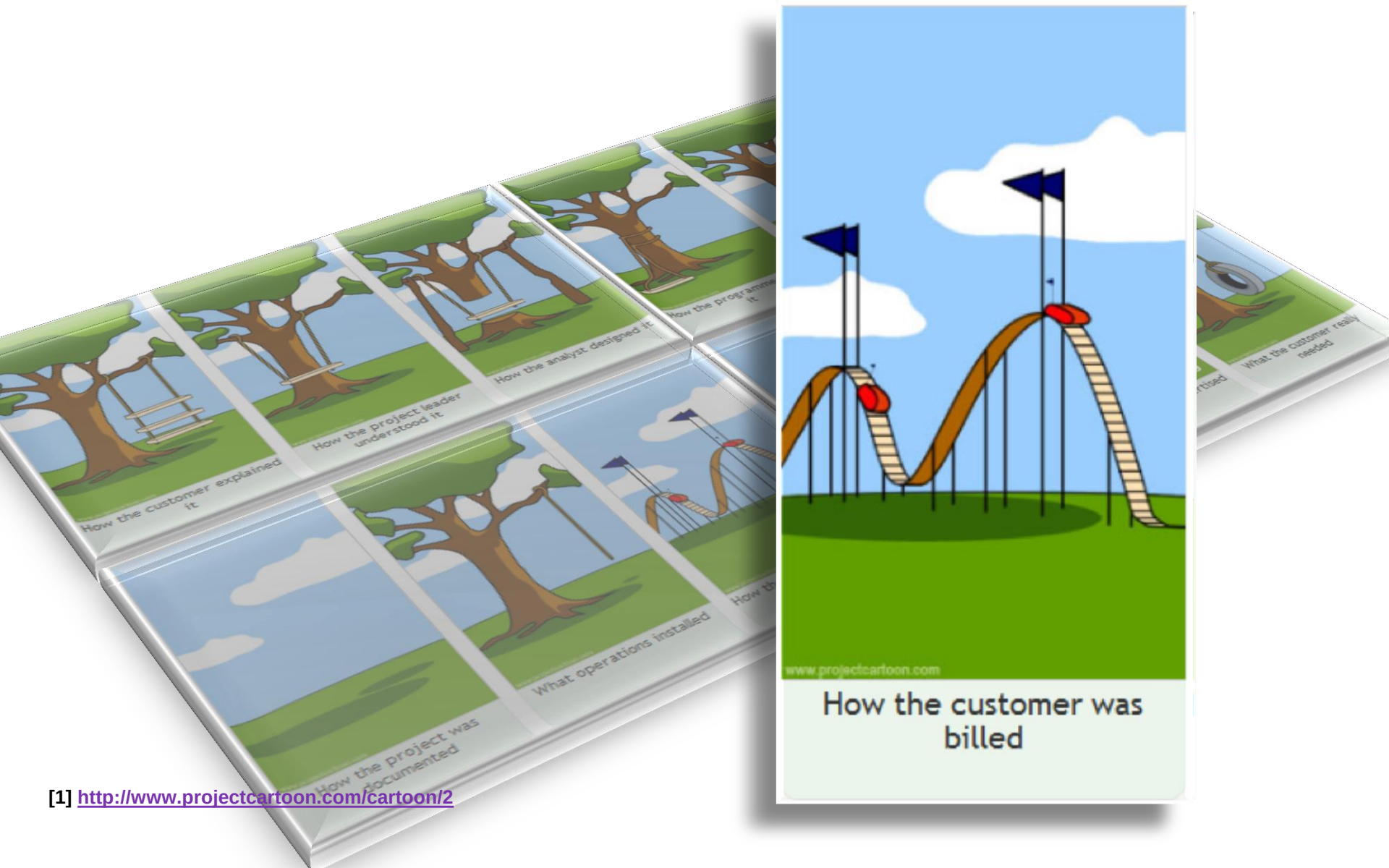


[1] http://www.projectcartoon.com/cartoon/2

What ICT operations installed

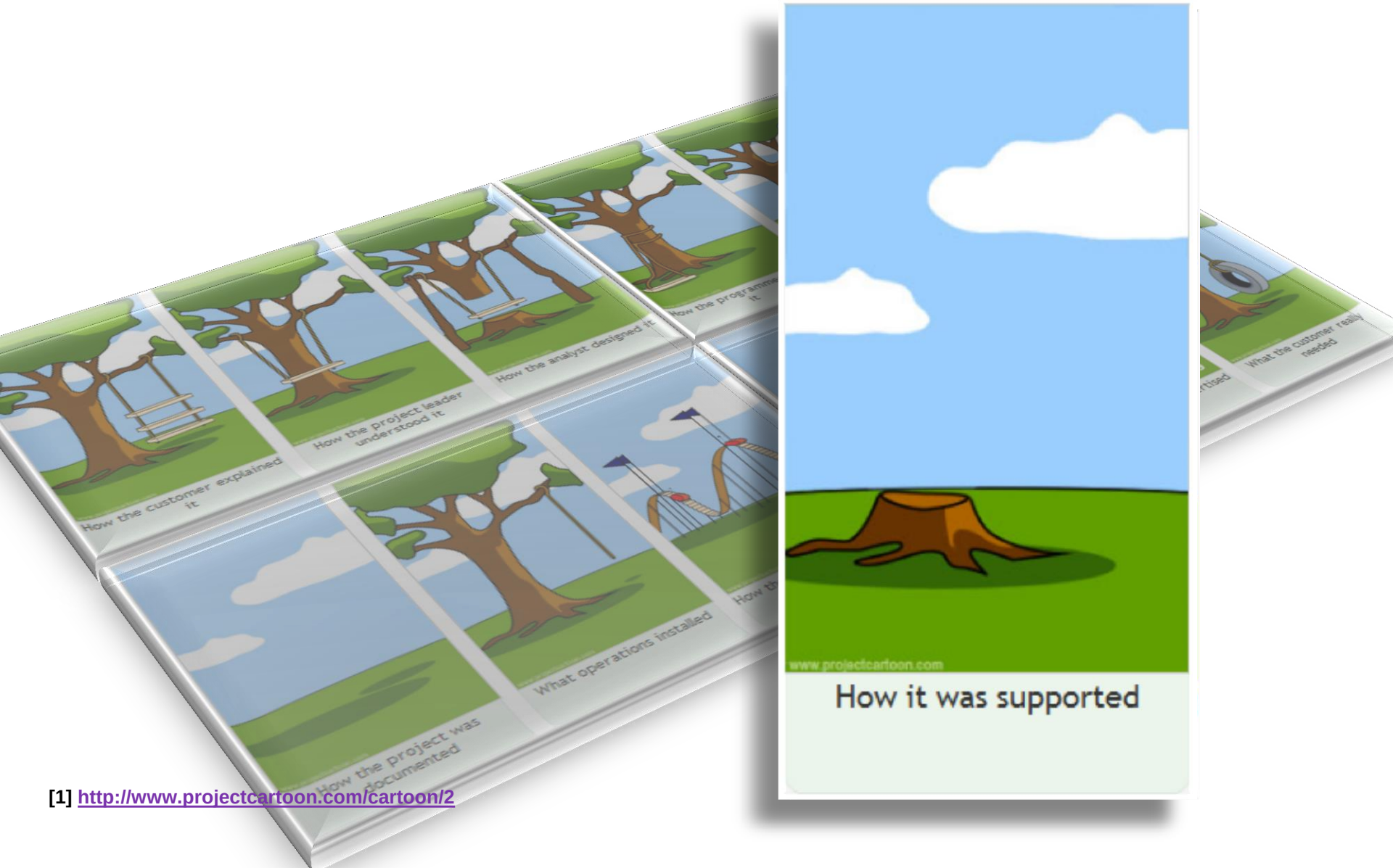
[1] http://www.projectcartoon.com/cartoon/2

How the customer was billed



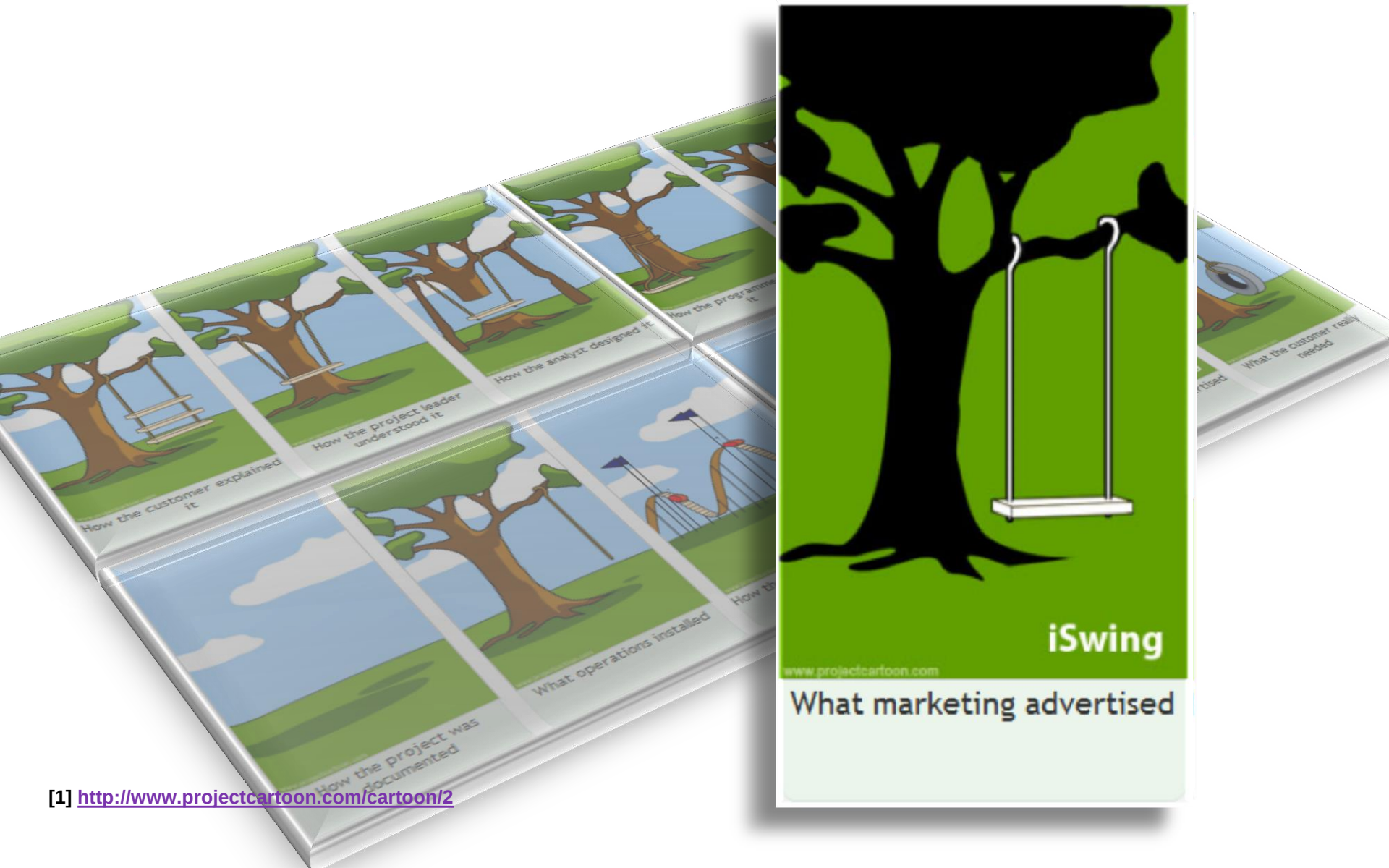
[1] http://www.projectcartoon.com/cartoon/2

How it was supported



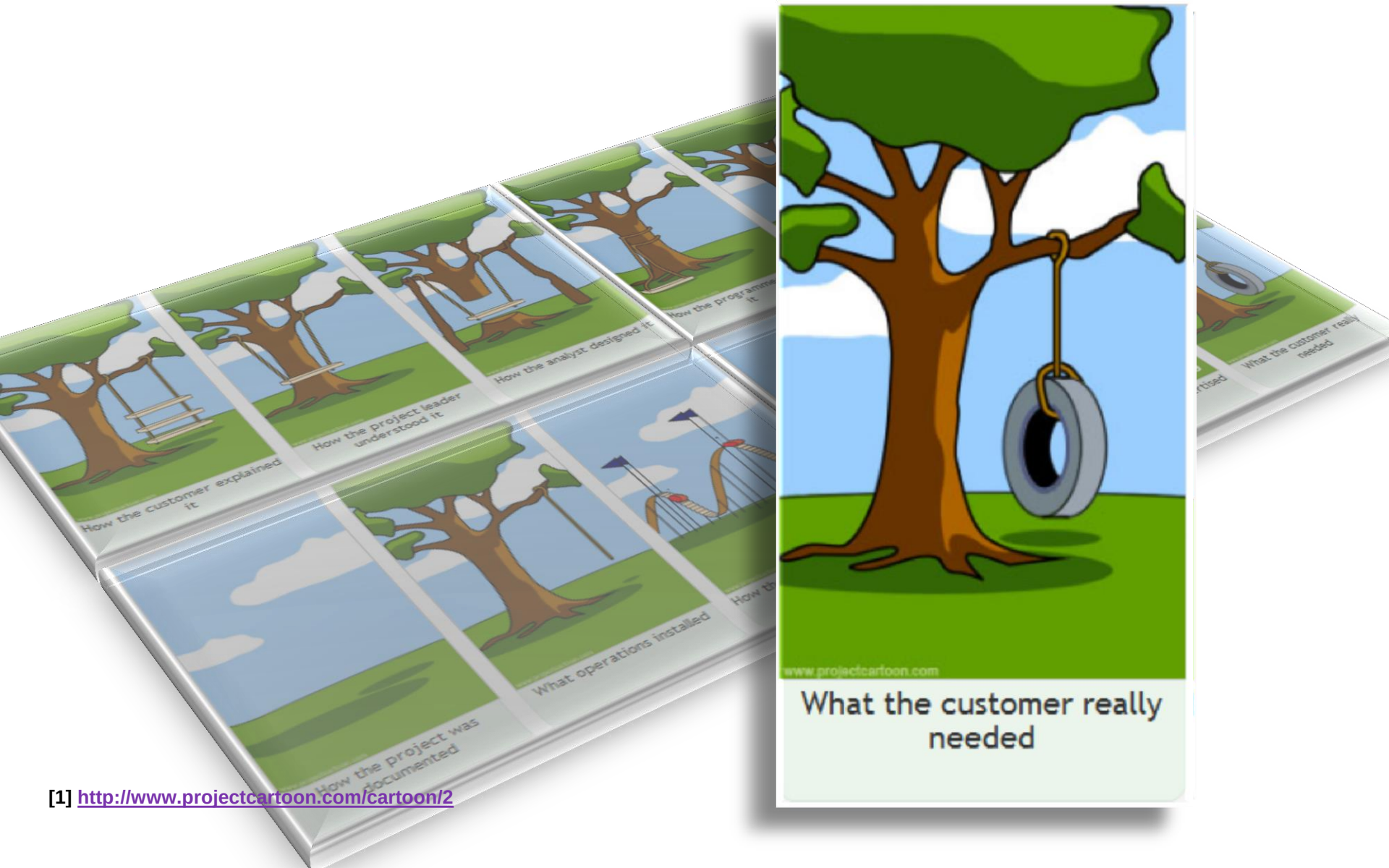
[1] http://www.projectcartoon.com/cartoon/2

What marketing advertised



[1] http://www.projectcartoon.com/cartoon/2

What the customer really needed



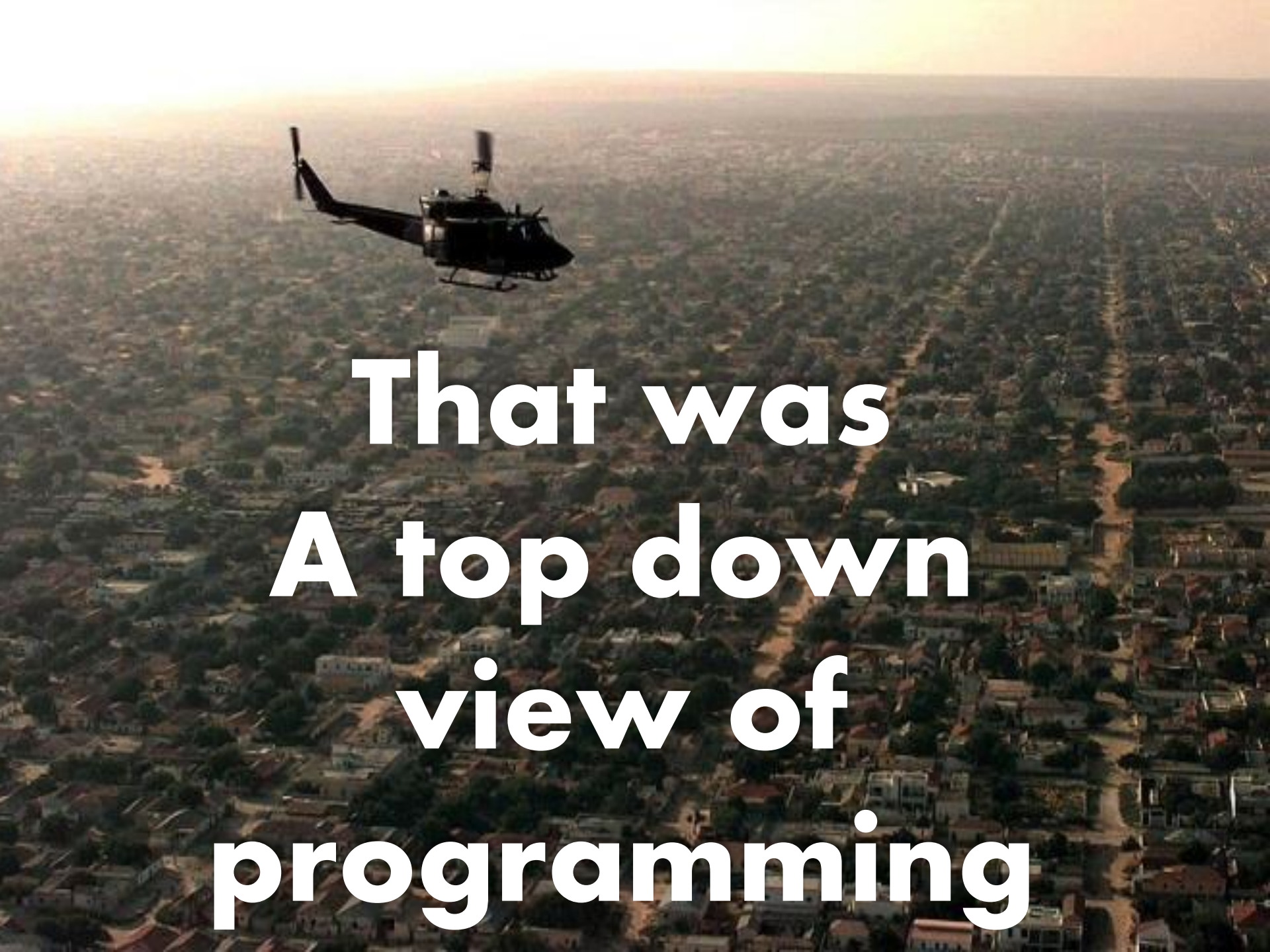
[1] http://www.projectcartoon.com/cartoon/2

Satisfy the Cat, a.k.a. User-Centred Design

Local play

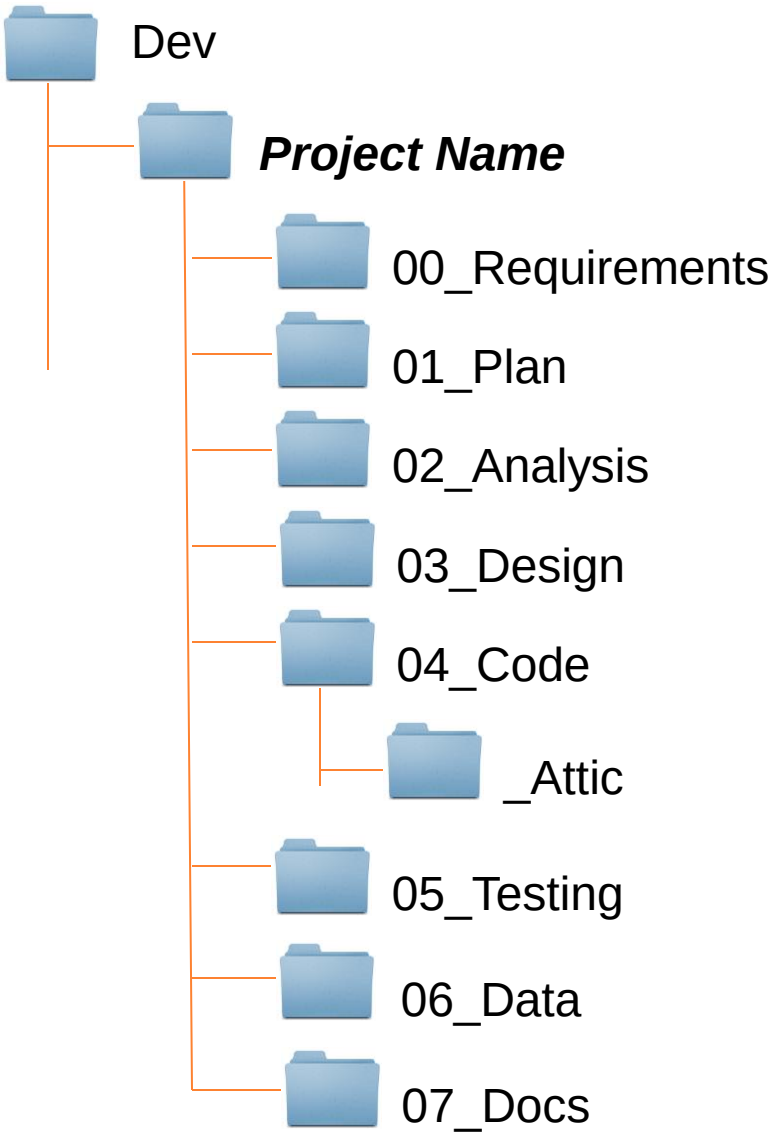
Web Play
http://www.youtube.com/watch?v=dln9xDsmCoY&feature=related

[1] http://www.youtube.com/watch?v=dln9xDsmCoY&feature=related


That was
A top down
view of
programming

We are going to start here

Programming is a Process



Programming is a ...

- Process
- And a way of thinking
- Problem solving
- Creating recipes in code
- Reusing approaches (patterns) and code (libraries) developed by others to build application, games and devices conceived by you or others

Programming is about having conversations with the computer.

Depending on how you phrase that conversation the computer does different things or combinations of things.

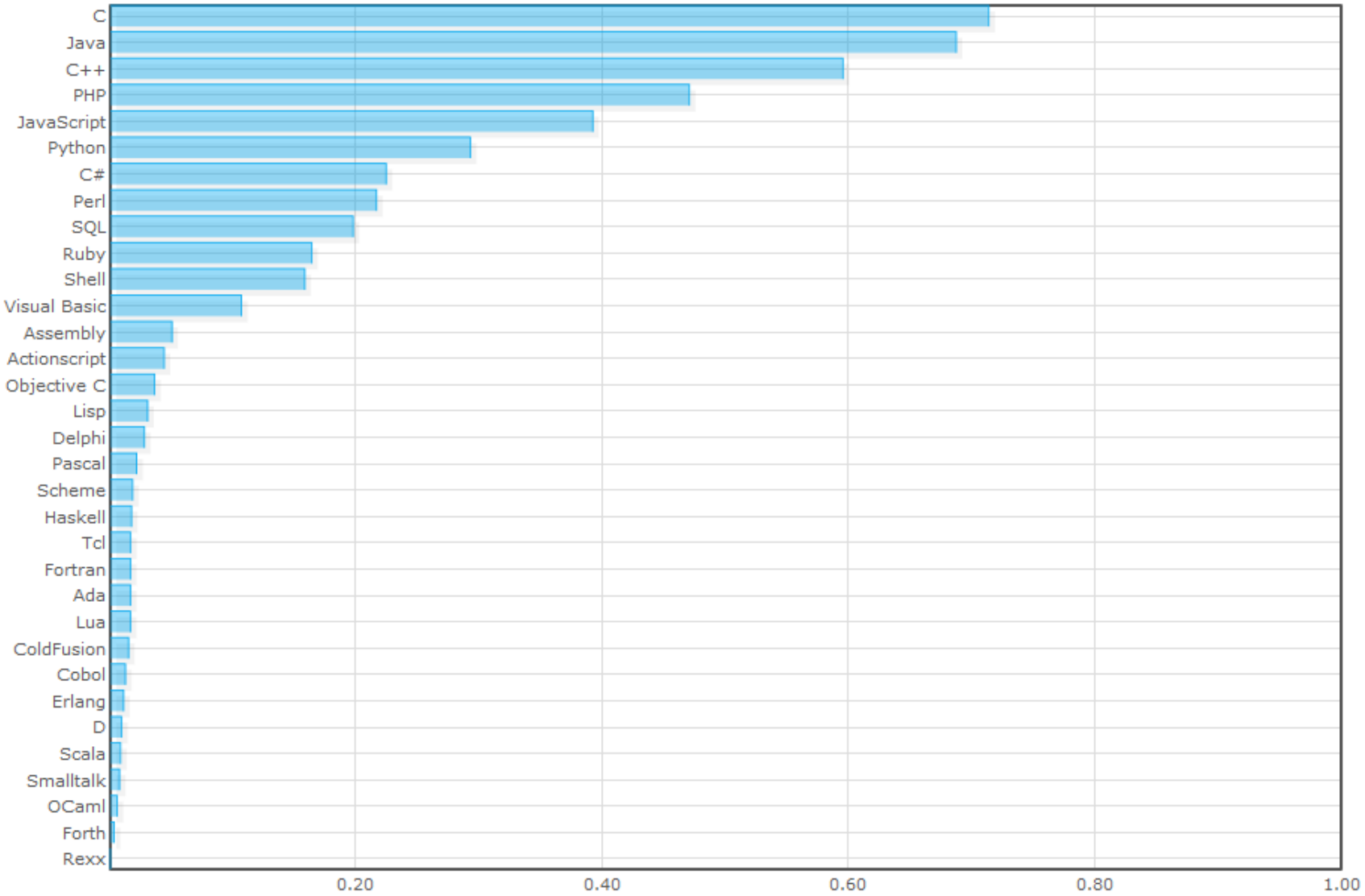
You then save these conversations so you can have them over and over again.

But to have these conversations you need to ...

Learn a new way of thinking

And a new **TYPE of WRITTEN** language for expressing to the computer what you want it to do

There are many programming languages e.g.
C, C++, Python, Visual Basic, C#, PHP, Fortran



January 2012, Normalised list based on multiple sources - http://langpop.com/

The language used depends on the problem

Certain languages are really good for certain types of problem

They allow you to easily and quickly express solutions to certain common problem within their specialist domain

But if it’s not used in the domain it was designed for, the hard can become impossible